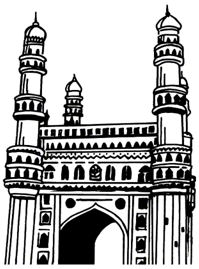
Rahul's ✔
Topper's Voice



M.B.A.

III Semester
(Osmania University)

Latest 2023 Edition

DATA BASE MANAGEMENT SYSTEMS

BUSINESS ANALYTICS
System Specialization

- ☞ Study Manual
- ☞ Important Questions
- ☞ Short Question & Answers
- ☞ Choose the Correct Answers
- ☞ Fill in the blanks
- ☞ Solved Model Papers

~~₹.699/-~~
₹.499/-

- by -

WELL EXPERIENCED LECTURER

Rahul Publications™
Hyderabad. Cell : 9391018098, 9505799122

All disputes are subjects to Hyderabad Jurisdiction only

M.B.A.

III Semester

(Osmania University)

DATA BASE MANAGEMENT SYSTEMS

BUSINESS ANALYTICS

System Specialization

Inspite of many efforts taken to present this book without errors, some errors might have crept in. Therefore we do not take any legal responsibility for such errors and omissions. However, if they are brought to our notice, they will be corrected in the next edition.

Price ₹. ~~*699/-*~~
₹. *499/-*

Sole Distributors : **Cell : 9391018098, 9505799122**

VASU BOOK CENTRE

Shop No. 2, Beside Gokul Chat, Koti, Hyderabad.
Maternity Hospital Opp. Lane, Narayan Naik Complex, Koti, Hyderabad.
Near Andhra Bank, Subway, Sultan Bazar, Koti, Hyderabad -195.

CONTENTS

DATA BASE MANAGEMENT SYSTEMS

BUSINESS ANALYTICS

DATABASE MANAGEMENT SYSTEMS

SYLLABUS

UNIT - I

Database System Architecture and Data Models

Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML), Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

UNIT - II

Relational Query Languages and Relational Database Design: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

UNIT - III

Query Processing and Optimization and Storage Strategies :

Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms, Indices, B-trees, hashing.

UNIT - IV

Transaction Processing and Database Security

Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery Authentication, Authorization and access control.

UNIT - V

SQL AND PL/SQL CONCEPTS :

Basics of SQL, DDL,DML,DCL, structure-creation, alteration, defining constraints-Primary key, foreign key, unique, not null, check, IN operator, aggregate functions, Built-in functions-numeric, date, string functions, set operations, sub-queries, correlated sub-queries, join, Exist, Any, All, view and its types., transaction control commands

Contents

Topic No. Page No.

UNIT - I

UNIT - II

Topic No. **Page No.**

UNIT - III

UNIT - IV

UNIT - V

Important Questions

UNIT - I

1. Explain about different types of DBMS architecture.

Ans :

Refer Unit-I, Q.No. 1

2. What is data abstraction? Explain.

Ans :

Refer Unit-I, Q.No. 3

3. Explain DML commands with its syntax and example.

Ans :

Refer Unit-I, Q.No. 7

4. Explain briefly about ER Model.

Ans :

Refer Unit-I, Q.No. 8

5. How To Create ER diagram in DBMS?

Ans :

Refer Unit-I, Q.No. 9

6. What is the Network Model in DBMS? Explain.

Ans :

Refer Unit-I, Q.No. 10

7. Explain values Constraints in Relational Model.

Ans :

Refer Unit-I, Q.No. 13

8. Discuss about Codd Rules in DBMS.

Ans :

Refer Unit-I, Q.No. 15

UNIT - II

1. Explain about the basic relational operations in relational algebra.

Ans :

Refer Unit-II, Q.No. 1

2. What is Relational Calculus? Explain different types of relational calculus in DBMS.

Ans :

Refer Unit-II, Q.No. 2

3. Give the brief introduction about SQL3.

Ans :

Refer Unit-II, Q.No. 3

4. Explain briefly about commands DDL constructs with examples.

Ans :

Refer Unit-II, Q.No. 4

5. Discuss about DML constructs with an examples.

Ans :

Refer Unit-II, Q.No. 5

6. What is MySQL & Explain its Features.

Ans :

Refer Unit-II, Q.No. 6

7. What is the Oracle database? Give the brief introduction about it.

Ans :

Refer Unit-II, Q.No. 9

8. Explain briefly about (SQL SERVER).

Ans :

Refer Unit-II, Q.No. 13

UNIT - III

1. What is query processing? Explain about the steps in query processing.

Ans :

Refer Unit-III, Q.No. 1

2. Explain about Query Optimization in Relational Algebra.

Ans :

Refer Unit-III, Q.No. 2

3. Write about evaluation of relational algebra expressions.

Ans :

Refer Unit-III, Q.No. 3

4. Write about equivalence rules in query optimization.

Ans :

Refer Unit-III, Q.No. 4

5. Explain about Join Strategies in Relational Algebra.

Ans :

Refer Unit-III, Q.No. 5

6. What is Indexing in DBMS? Explain.

Ans :

Refer Unit-III, Q.No. 7

7. Explain about B-Trees in DBMS.

Ans :

Refer Unit-III, Q.No. 8

8. What is hashing in DBMS? Explain about various hashing techniques.

Ans :

Refer Unit-III, Q.No. 11

UNIT - IV

1. What is concurrency control? Explain the problems of concurrency control.

Ans :

Refer Unit-IV, Q.No. 1

2. What are ACID properties? Explain.

Ans :

Refer Unit-IV, Q.No. 3

3. What is Serializabilityof scheduling in DBMS ? Explain.

Ans :

Refer Unit-IV, Q.No. 4

4. What are locks in DBMS? Explain about different types of locks.

Ans :

Refer Unit-IV, Q.No. 6

5. Explain about types of lock based protocols.

Ans :

Refer Unit-IV, Q.No. 7

6. Explain Multi-Version Schemes of Con-currency Control with example.

Ans :

Refer Unit-IV, Q.No. 9

7. What are Recovery Techniques in DBMS? Explain.

Ans :

Refer Unit-IV, Q.No. 11

8. What is user Authentication? Explain different types of user authentication techniques.

Ans :

Refer Unit-IV, Q.No. 13

UNIT - V

1. What is SQL? Explain the importance of SQL.

Ans :

Refer Unit-V, Q.No. 1

2. Explain the History of SQL.

Ans :

Refer Unit-V, Q.No. 2

3. Explain the Process of SQL.

Ans :

Refer Unit-V, Q.No. 3

4. Explain various DDL commands with an examples and syntax.

Ans :

Refer Unit-V, Q.No. 5

5. Explain insert statement with syntax and example.

Ans :

Refer Unit-V, Q.No. 7

6. Explain CREATE command in SQL.

Ans :

Refer Unit-V, Q.No. 11

7. Explain ALTER Command in SQL.

Ans :

Refer Unit-V, Q.No. 12

8. Discuss about NOT NULL constraint?

Ans :

Refer Unit-V, Q.No. 17

9. What is CHECK constraint? Explain.

Ans :

Refer Unit-V, Q.No. 18

10. Explain about IN Operator in SQL.

Ans :

Refer Unit-V, Q.No. 19

11. Explain about string functions in SQL.

Ans :

Refer Unit-V, Q.No. 24

UNIT I

Database System Architecture and Data Models

Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML), Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

1.1 Database System Architecture and Data Models

1.1.1 Data Abstraction

Q1. Explain about different types of DBMS architecture.

Ans : **(Imp.)**

Meaning

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.

Types

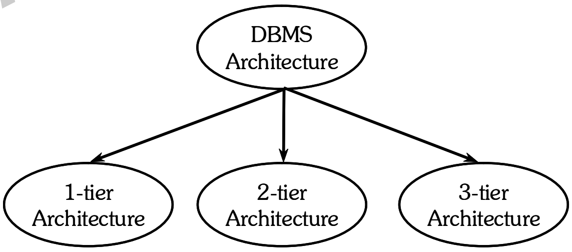


Fig.: DBMS Architecture

Database architecture can be seen as a single tier or multitier. But logically, database architecture is of two types like: 2-tier architecture and 3-tier architecture.

i) 1-Tier Architecture

- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

ii) 2-Tier Architecture

- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: **ODBC**, **JDBC** are used.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.

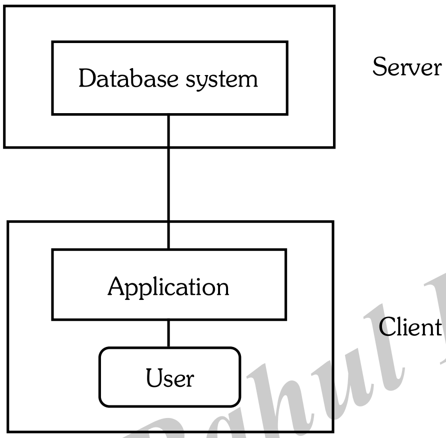


Fig: 2-tier Architecture

iii) 3-Tier Architecture

- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- The 3-Tier architecture is used in case of large web application.

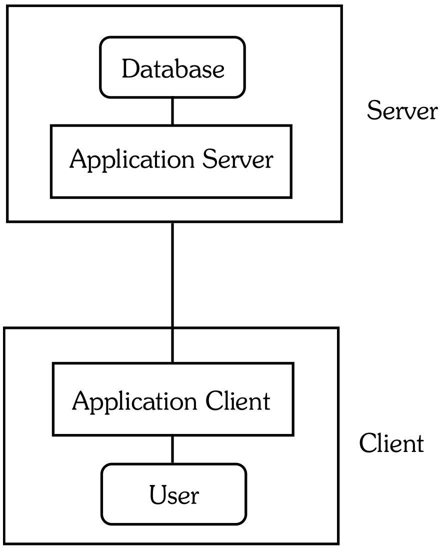


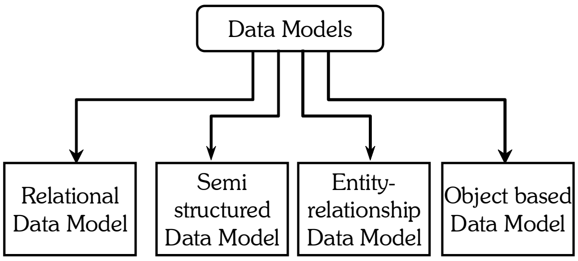
Fig.: 3-tier Architecture

Q2. Write about data models used in database.

Ans :

Meaning

Data Model is the modelling of the data description, data semantics, and consistency constraints of the data. It provides the conceptual tools for describing the design of a database at each level of data abstraction. Therefore, there are following four data models used for understanding the structure of the database:



i) **Relational Data Model:** This type of model designs the data in the form of rows and columns within a table. Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations. This model was initially described by Edgar F. Codd, in 1969. The relational data model is the widely used model which is primarily used by commercial data processing applications.

ii) **Entity-Relationship Data Model:** An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and relationship is an association among these entities. This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing. A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.

iii) **Object-based Data Model:** An extension of the ER model with notions of functions, encapsulation, and object identity, as well. This model supports a rich type system that includes structured and collection types. Thus, in 1980s, various database systems following the object-oriented approach were developed. Here, the objects are nothing but the data carrying its properties.

iv) **Semistructured Data Model:** This type of data model is different from the other three data models (explained above). The semistructured data model allows the data specifications at places where the individual data items of the same type may have different attributes sets. The Extensible Markup Language, also known as XML, is widely used for representing the semistructured data. Although XML was initially designed for including the markup information to the text document, it gains importance because of its application in the exchange of data.

Q3. What is data abstraction? Explain.

Ans : **(Imp.)**

Meaning

Data abstraction is present in our daily lives. Let us take a small example. Say, someone, asks you to switch on the fans in a room. All you will need to do is simply walk to the switchboard and turn on the switch for the fan, that's it! Do you need to know where the electricity is coming from, how the poles of the switch are connected, or exactly what the internal working of a fan is? The answer to all this is NO! That is what data abstraction is, all these background details are hidden from you inside the switchboard!

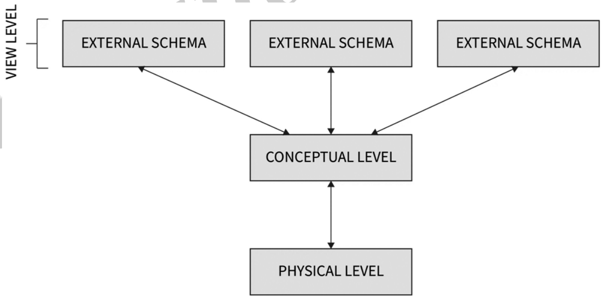
All the databases have complex data structures which are, in fact, of no use to an end user. Thus, these internal irrelevant details are hidden from the user, making the accessing of data simple and increasing the security of the data as well.

Levels

The data abstraction in **DBMS** is implemented in 3 layers:

1. Physical or Internal Level
2. Logical or Conceptual Level
3. View or External Level

The following diagram will give you a clear view of how the implementation is done.



Starting from the very bottom, we have the physical or internal layer, then we have the middle layer, the logical or conceptual layer, and finally, we have the view or external layer.

Let us discuss each of these three layers in detail.

1. Physical Level or Internal Level

This is the layer of data abstraction where the raw data is physically stored as files. This layer contains all the complex data structures and the data accessing methods defined. The physical layer is the lowest level of data abstraction in a DBMS. It is the database administrator who decides how the data is to be stored in these physical hard drives.

Example

When we access data we may get a single data or a table of data. Moreover, by the term "relational database" we visualize a table of rows and columns. But at a physical level, these tables are stored in hard drives which are located at a very secure data center.

2. Logical Level or Conceptual Level

After taking the raw data from the physical or internal level, the structure of the data is defined at the logical or conceptual level. This is like a blueprint of the raw data. This layer does not have any information about how the end user will view the data.

Example

We have data of a few products like product id, product name, and manufacturing date, and we have another set of data of customers containing customer id, customer name, and customer address. Now, we need to frame this data in proper tables of products and customers. After that, we can even frame a join to show which product has been ordered by which customer.

3. View Level or External Level

This is what an end-user gets to see. He/she does not get the entire database, but depending on the queries made from the front-end the user gets to see the data. It may be a single data from the entire database or a collection of data in tabular format. Multiple views of the same data are available to the user, the representation can be a table, a graph, or a pie chart. View Level is the highest level of data abstraction in DBMS.

Example

Concerning the example in the logical level section, let us say a customer wants to view the order history, he gets to see only the orders he had made in the past. Now, let us say a shop owner needs to see the products that are on the order list. He gets to see a table containing all the info about the products and the customers to whom they need to be delivered.

1.1.2 Data Independence

Q4. What is Data Independence in DBMS?

Ans :

Meaning

The ability to modify the schema definition of a DBMS at one level, without affecting the schema definition of the next higher level is called data independence.

In addition to the data entered by users, a database system typically holds a large amount of data. The system holds metadata about data which makes it easier to find and retrieve data. Once a set of metadata has been saved in a database, changing or updating the metadata is challenging. However, as a database management system (DBMS) grows, it must evolve to meet the needs of its users. Updating the schema or data would be a time-consuming and complicated task if all of the data were dependent.

To address the problem with updating metadata, it is organized in a tiered structure, so that changing data at one level does not affect data at another. This information is self-contained, however, all this information is linked to one another.

So, data independence aids in the separation of data from the applications that use it.

Types of Data Independence

Now that you know what data independence means, let's discuss its types. This is where your knowledge of the 3-level architecture is important! Data Independence is of two types:

1. Physical Data Independence

This is defined as the ability to modify the physical schema of the database without the modification causing any changes in the logical/conceptual or view/external level.

Importantance

- Physical data independence allows you to distinguish between conceptual and internal/physical levels. It allows you to describe the database logically without needing to identify physical structures.
- Physical data independence allows you to modify physical storage structures or devices without affecting the conceptual model of the database. Any changes made at the internal level would be absorbed by the mapping between the conceptual and internal levels, preventing any modifications to the conceptual level.

Examples

- Changing from one data structure to another.
- Making use of new storage technology, such as a hard drive or magnetic tapes
- Change the location of the database from one drive to another.
- Changing the database's file organization.

2. Logical Data Independence

Logical data independence is the ability to modify logical schema without causing any unwanted modifications to the external schema or the application programs to be rewritten.

Importantance

- Logical data is database data, which means it stores information about how data is managed within the database. Logical data independence is a method that makes sure that if we make modifications to the table format, the data should not be affected.
- The mapping between the external and conceptual levels will absorb any changes made.
- In other words, to distinguish the external level from the conceptual view, logical data independence is used. Any modifications to the conceptual representation of the data will not affect the user's view of the data.

Examples

- Without rewriting current application scripts, you can add, modify, or delete a new attribute, entity, or relationship.
- To divide an existing record into two or more records.
- Merging two records into a single one.

Q5. Differentiate between physical and logical data independence

Ans :

Difference between physical and logical data independence:-

Sl.No.	Physical Data Independence	Logical Data Independence
1.	It is concerned with the internal schema of the database.	It is concerned with the conceptual schema of the database.
2.	It is easier to achieve as compared to logical data independence.	Logical data independence is difficult to achieve as compared to physical data independence.
3.	Physical data independence is mostly concerned with how data is saved in the system.	It is mostly focused on the structure or updating data definitions.
4.	Changes at the internal level may or may not be required to increase the overall performance of the database.	When the database's logical structure needs to be modified, the changes made at the logical level are crucial.
5.	In most cases, a change at the physical level does not necessitate a change at the application program level.	If new fields are added or removed from the database, then updates are required to be made in the application software.

1.2 Data Definition Languages

Q6. What are Data Definition Language Commands? Explain.

Ans :

Data Definition Language(DDL) is a subset of SQL and a part of DBMS (Database Management System). DDL consist of Commands to commands like CREATE, ALTER, TRUNCATE and DROP. These commands are used to create or modify the tables in SQL.

DDL Commands

1. Create
2. Alter
3. truncate
4. drop

Let's discuss it one by one

1. CREATE

This command is used to create a new table in SQL. The user has to give information like table name, column names, and their datatypes.

Syntax

```
CREATE TABLE table_name
(
column_1 datatype,
column_2 datatype,
column_3 datatype,
....
);
```

Example

We need to create a table for storing Student information of a particular College. Create syntax would be as below.

```
CREATE TABLE Student_info
    (
    College_Id number(2),
    College_name varchar(30),
    Branch varchar(10)
    );
```

2. ALTER

This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.

Syntax

Syntax to add a column to an existing table.

```
ALTER TABLE table_name
ADD column_name datatype;
```

Example

In our Student_info table, we want to add a new column for CGPA. The syntax would be as below as follows.

```
ALTER TABLE Student_info
ADD CGPA number;
```

3. TRUNCATE

This command is used to remove all rows from the table, but the structure of the table still exists.

Syntax

Syntax to remove an existing table.

```
TRUNCATE TABLE table_name;
```

Example

The College Authority wants to remove the details of all students for new batches but wants to keep the table structure. The command they can use is as follows.

```
TRUNCATE TABLE Student_info;
```

4. DROP

This command is used to remove an existing table along with its structure from the Database.

Syntax

Syntax to drop an existing table.

```
DROP TABLE table_name;
```

Example

If the College Authority wants to change their Database by deleting the Student_info Table.

```
DROP TABLE Student_info;
```

1.3 Data Manipulation Language (DML)

Q7. Explain DML commands with its syntax and example.

Ans : **(Imp.)**

- DML stands for Data Manipulation Language.
- It is a language used for selecting, inserting, deleting and updating data in a database.
- It is used to retrieve and manipulate data in a relational database.

DDL commands are as follows,

1. SELECT
2. INSERT
3. UPDATE
4. DELETE

- DML performs read-only queries of data.

1. SELECT COMMAND

- **SELECT command** is used to retrieve data from the database.
- This command allows database users to retrieve the specific information they desire from an operational database.
- It returns a result set of records from one or more tables.

SELECT Command has many optional clauses are as stated below:

Clause	Description
WHERE	It specifies which rows to retrieve.
GROUP BY	It is used to arrange the data into groups.
HAVING	It selects among the groups defined by the GROUP BY clause.
ORDER BY	It specifies an order in which to return the rows.
AS	It provides an alias which can be used to temporarily rename tables or columns.

Syntax:

```
SELECT * FROM <table_name>;
```

Example : SELECT Command

```
SELECT * FROM employee;
OR
SELECT * FROM employee
where salary >=10,000;
```

2. **INSERT COMMAND**

- **INSERT** command is used for inserting a data into a table.
- Using this command, you can add one or more records to any single table in a database.
- It is also used to add records to an existing code.

Syntax: INSERT INTO <table_name> (‘column_name1‘ <datatype>, ‘column_name2‘ <datatype>, . . . , ‘column_name_n‘ <database>) VALUES (‘value1‘, ‘value2‘, . . . , ‘value n‘);

Example:

```
INSERT INTO employee (‘eid‘ int, ‘ename‘ varchar(20), ‘city‘ varchar(20))
VALUES (‘1‘, ‘ABC‘, ‘PUNE‘);
```

3. **UPDATE COMMAND**

- **UPDATE** command is used to modify the records present in existing table.
- This command updates existing data within a table.
- It changes the data of one or more records in a table.

Syntax

```
UPDATE <table_name>
SET <column_name = value>
WHERE condition;
```

Example : UPDATE Command

```
UPDATE employee
SET salary=20000
WHERE ename='ABC';
```

4. DELETE COMMAND

- **DELETE** command is used to delete some or all records from the existing table.
- It deletes all the records from a table.

Syntax

DELETE FROM <table_name> WHERE <condition>;

Example : DELETE Command

```
DELETE FROM employee
WHERE emp_id = '001';
```

If we does not write the WHERE condition, then all rows will get deleted.

1.4 Entity - Relationship Model

Q8. Explain briefly about ER Model.

Ans : **(Imp.)**

Meaning

This model defines the data elements and relationships for a specified system. It is useful in developing a conceptual design for the database & is very simple and easy to design logical view of data.

ER Diagrams

- ERD stands for Entity Relationship diagram.
- It is a graphical representation of an information system.
- ER diagram shows the relationship between objects, places, people, events etc. within that system.
- It is a data modeling technique which helps in defining the business process.
- It used for solving the design problems.

ER diagram has three main components:

1. Entity
2. Attribute
3. Relationship

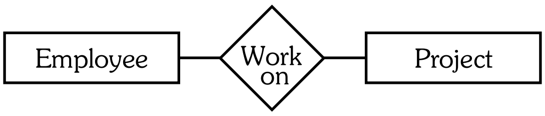
1. Entity

Any object that physically exists and is logically constructed in the real world is called as an entity. It is a real-world object that can be easily identifiable.

An entity is represented as a rectangle in an ER diagram.

Example

In an organization, employees, managers, and projects assigned can be considered entities. All these entities have some attributes or properties that give them their identity.

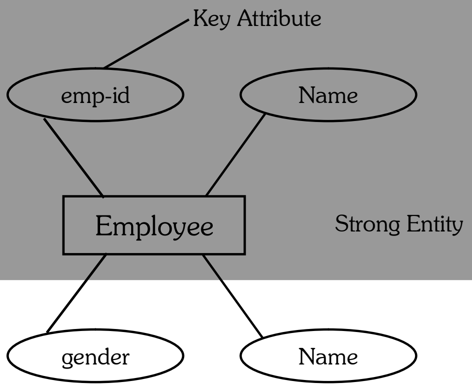


Here, in the above example, employee and project are entities.

Entities are of two types

- **Strong Entity:** Strong entities are those entity types that have a key attribute. The primary key helps in identifying each entity uniquely. this can not accept null values so it can not be a unique key. It is represented by a rectangle.

  Example: in an example of organization emp_id identifies each employee of the organization uniquely and hence, we can say that employee is a strong entity type.



- **Weak Entity:** Weak entity type doesn't have a key attribute. Weak entity types can't be identified on their own. It depends upon some other strong entity for its distinct identity. A weak entity is represented by a double outlined rectangle. The relationship between a weak entity type and strong entity type is shown with a double outlined diamond instead of a single outlined diamond. This representation can be seen in the example given below.

Here we cannot identify the address uniquely as there can be many employees from the same locality. So, for this, we need an attribute of Strong Entity Type i.e 'employee' here to uniquely identify entities of 'Address' Entity Type.

2. Attribute

An attribute is a property or characteristic of an entity. An entity may contain any number of attributes. The attributes that can uniquely define an entity are considered as the primary key. In an Entity-Relation model, attributes are represented in an elliptical shape. It also may refer to a database field. Attributes describe the instances in the database.

A database consists of tables. Each table has columns and rows. The columns in a database are called attributes.

Example: Employee has attributes like name, age, roll,emp_id, and many more. To uniquely identify the employee, we use the primary key as an emp_id(employee id) as it is not repeated. Attributes can also be subdivided into another set of attributes.

There are five such types of attributes:

- Simple attribute
- Composite attribute
- Single-valued attribute
- Multi-valued attribute
- Derived attribute.

Simple attribute

Attributes that are not further divisible into sub-attributes (atomic) are known as Simple attributes.

Example

The roll number of a student, the id number of an employee.

It is also called a key attribute. It modeled in ER diagram as a simple eclipse with underlined attribute name.

- **Composite attribute**

Composite attributes can be divided into sub-attributes which represent more basic attributes with independent meanings.

Example: the Address attribute of the EMPLOYEE entity shown can be subdivided into Street_address, City, State, and Pincode.

Composite attributes are useful to model situations in which a user sometimes refers to the composite attribute as a unit but at other times refers to its components.

If the composite attribute is referenced only as a whole, there is no need to subdivide it into component attributes. For example, if there is no need to refer to the individual components of an address (Zip Code, street, and so on), then the whole address can be designated as a simple attribute.

- **Single-valued attribute**

Attributes having single value for a particular entity instance is known as single-valued attribute.

Example, the age of a person is single-valued.

- **Multi-valued attribute**

There are many instances where an attribute has a set of values for a specific entity, known as Multivalued attributes.

Multivalued attributes are modeled in ER using a double circle.

Example: Phone number. A person may have zero, one or more phone numbers, and different employees may have different numbers of phones.

- **Derived attribute**

The value for this type of attribute can be derived from the values of other related attributes or entities instances.

Example: suppose that the employee entity set has an attribute age, which indicates the employee's age. If the employee entity set also has an attribute date-of-birth, we can calculate age from date-of-birth and the current date. Thus, age is a derived attribute.

However, the derived attribute needs to be computed every time it's required.

- **Complex attribute**

Complex attributes are formed by nesting composite and multivalued attributes arbitrarily. These attributes are rarely used in DBMS (DataBase Management System). That's why they are not so popular.

These (multi-valued and composite attributes are called 'Components' of complex attributes) components are grouped between parentheses '( )' and multi-valued attributes between curly braces '{ }', Components are separated by commas ', '.

Example

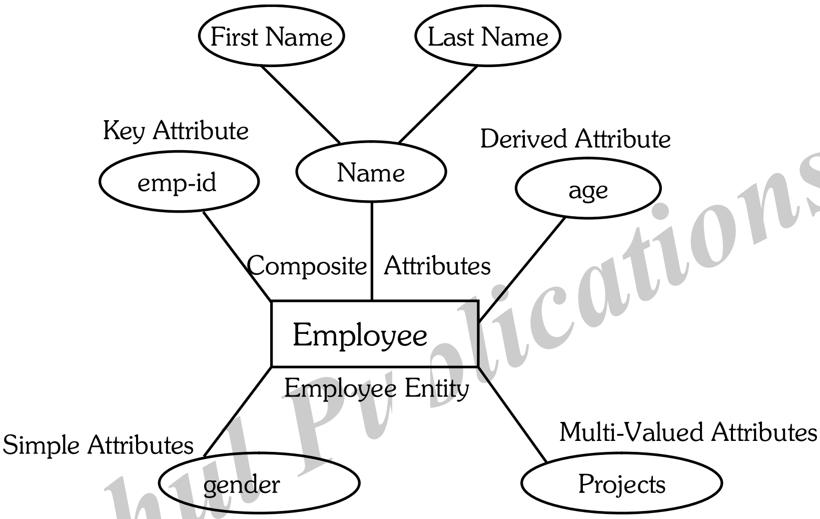
Let us consider an employee having multiple phone numbers, emails, and an address.

Here, phone number and email are examples of multi-valued attributes and address is an example of the composite attribute, because it can be divided into house number, street, city, and state.

empAdd_empPhone({email},{Phone}, Address{Housenumber,city,state})

Here,empAdd_empPhone is a complex attribute.

Here is the figure is given below represents all the attributes in the ER diagram:



3. Relationship

A relationship in a DBMS is primarily the way two or more data sets are linked. Relationships allow the datasets to share and store data in separate tables. They also help link disparate data with each other.

A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables while linking disparate data items.

Relationships are of three types and the next segment talks about the same.

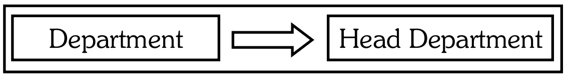
Types of relationships

- One to One
- One to Many
- Many to Many

Let us see what each one of them entails.

One to One: It is used to create a relationship between two tables in which a single row of the first table can only be related to one and only one record of a second table. This relationship tells us that a single record in Table A is related to a single record in Table B. And vice versa.

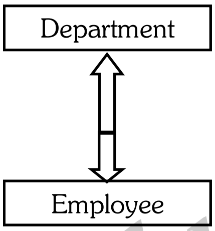
Example: In a university, each department has only one head of the department. And one HOD can take only one department. This shows a one-to-one (1:1) relationship between the department and the person as a head.



One to Many

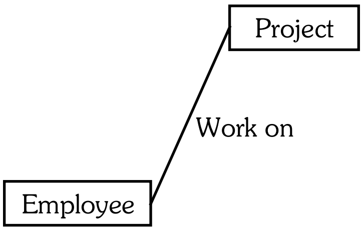
It is used to create a relationship between two tables. Any single row of the first table can be related to one or more rows of the second table, but the rows of the second table can only relate to the only row in the first table. It is also known as a many-to-one relationship.

Example: of a 1:M relationship is A department that has many employees, Each employee is assigned to one department.



Many to Many

Many to many relationships that create a relationship between two tables. Each record of the first table can relate to any records (or no records) in the second table. Similarly, each record of the second table can also relate to more than one record of the first table. It also represented an N:N relationship.



Example: there are many employees involved in each project, and every employee can involve in more than one project.

Features of ER model

The basic E-R concepts can model most database features, some aspects of a database may be more aptly expressed by certain extensions to the basic E-R model. The extended E-R features are specialization, generalization, higher- and lower-level entity sets, attribute inheritance, and aggregation.

Specialization

An entity set broken down sub-entities that are distinct in some way from other entities in the set. For instance, a subset of entities within an entity set may have attributes that are not shared by all the entities in the entity set. The E-R model provides a means for representing these distinctive entity groupings.

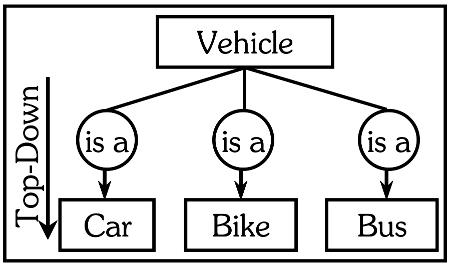
Specialization is an “aTop-down approach” where a high-level entity is specialized into two or more level entities.

Example

Consider an entity set vehicle, with attributes color and no. of tires. A vehicle may be further classified as one of the following:

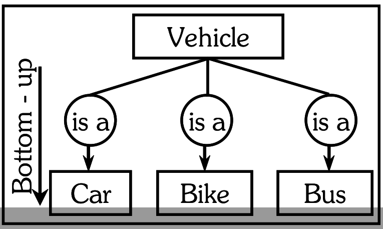
- Car
- Bike
- Bus

Each of these vehicle types is described by a set of attributes that includes all the attributes of the entity set vehicle plus possibly additional attributes. For example, car entities may be described further by the attribute gear, whereas bike entities may be described further by the attributes automatic break. The process of designating subgroupings within an entity set is called specialization. The specialization of vehicles allows us to distinguish among vehicles according to whether they are cars, buses, or bikes.



Generalization

It is a process of extracting common properties from a set of entities and creating a generalized entity from it. Generalization is a "Bottom-up approach". In which two or more entities can be combined to form a higher-level entity if they have some attributes in common.



In generalization, Subclasses are combined to make a superclass.

Example

There are three entities given, car, bus, and bike. They all have some common attributes like all cars, buses, and bikes they all have no. of tires and have some colors. So they all can be grouped and make a superclass named a vehicle.

Inheritance

An entity that is a member of a subclass inherits all the attributes of the entity as the member of the superclass, the entity also inherits all the relationships that the superclass participates in. Inheritance is an important feature of Generalization and Specialization. It allows lower-level entities to inherit the attributes of higher-level entities.

Example

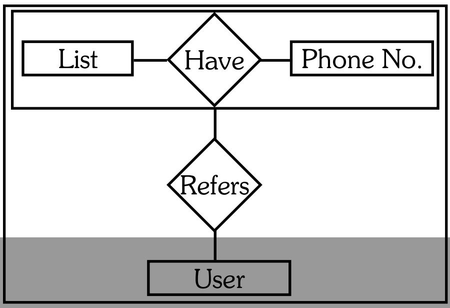
Car, bikes, and buses inherit the attributes of a vehicle. Thus, a car is described by its color and no. of tires, and additionally a gear attribute; a bike is described by its color and no. of tires attributes, and additionally automatic break attribute.

Aggregation

In aggregation, the relation between two entities is treated as a single entity. In aggregation, the relationship with its corresponding entities is aggregated into a higher-level entity.

Example

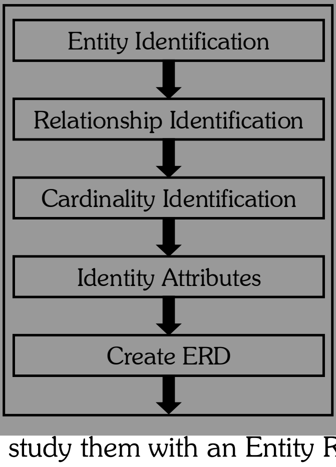
Phone numbers on your mobile phone. You can refer to them individually – your mother's number, your best friend's number, etc. But it's easier to think of them collectively, as your phone number list. It is also important to realize that each member of the aggregation still has the properties of the whole. In other words, each phone number in the list remains a phone number. The process of combining them has not altered them in any way.



Q9. How To Create ER diagram in DBMS

Ans : **(Imp.)**

Following are the steps to create an ER Diagram

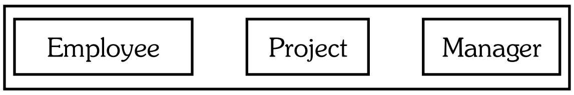


Let's study them with an Entity Relationship Diagram Example:

In an Organization, an employee is assigned to projects. An employee must be assigned to at least one or more projects. Each project is managed by a single manager. To maintain instruction quality, a manager can control only one project.

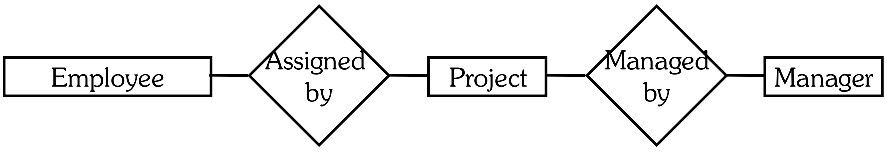
Step 1: Entity Identification: We have three entities

- Employee
- Project
- Manager



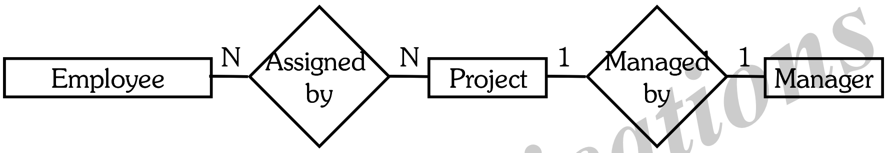
Step 2: **Relationship Identification:** We have the following two relationships.

- The employee is assigned a project
- Manager control a project



Step 3: **Cardinality Identification:** For them problem statement we know that,

- An employee can be assigned multiple projects
- A manager can manage only one course



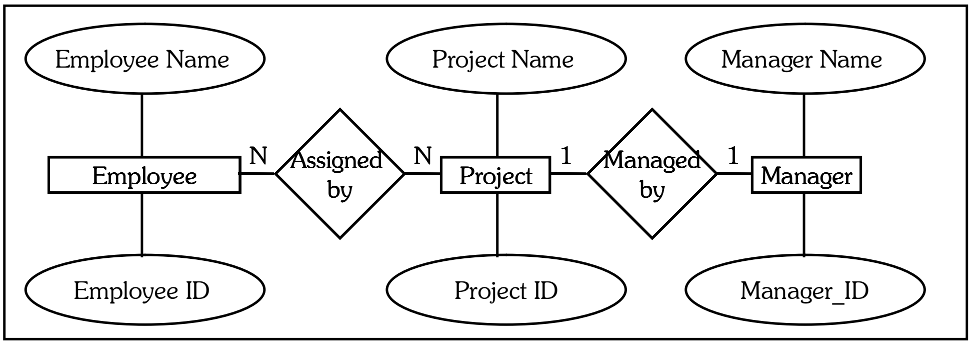
Step 4: **Identify Attributes:** Initially, it's important to identify the attributes without mapping them to a particular entity. Once you have a list of Attributes, you need to map them to the identified entities. Ensure an attribute is to be paired with exactly one entity. If you think an attribute should belong to more than one entity, use a modifier to make it unique.

Once the mapping is done, identify the primary Keys. If a unique key is not readily available, create one.

Entity	Primary Key	Attribute
Employee	Employee_ID	EmployeeName
Manager	Manager_ID	ManagerName
Project	Project_ID	ProjectName

For the sake of ease, we have considered just one attribute.

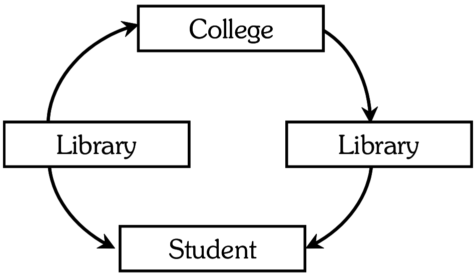
Step 5: **Create the ERD Diagram:** A more modern representation of Entity Relationship Diagram Example



Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

- It helps you to define terms related to entity relationship modeling.
- It provides a preview of how all your tables should connect, what fields are going to be on each table
- It helps to describe entities, attributes, relationships.
- ER diagrams are translatable into relational tables which allows you to build databases quickly.
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications.



1.5 Network Model

Q10. What is the Network Model in DBMS? Explain.

Ans : **(Imp.)**

Meaning

The hierarchical model is extended in the network model. Prior to the relational model, it was the most popular model. To increase database performance and standards, the network model was devised to express complicated data relationships more effectively than hierarchical models. It has entities that are grouped in a graphical format, and some of the entities can be reached by many paths.

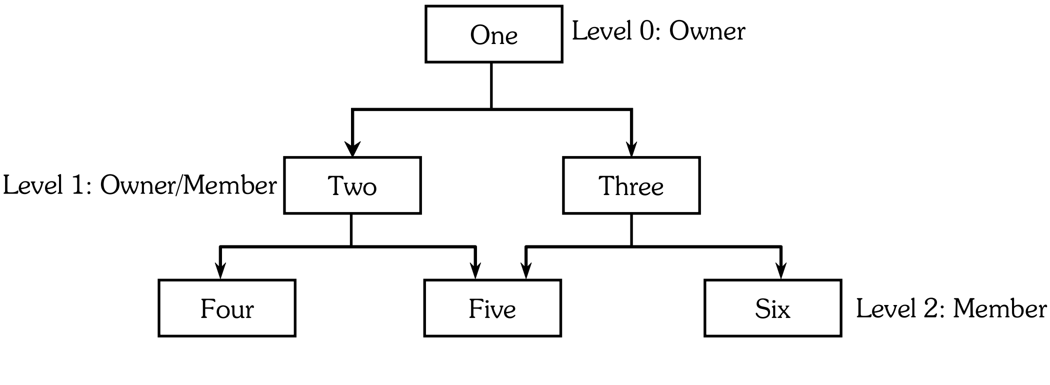
Let us take an example of a simple College database that has two departments or sections namely - the CSE Department and the Library. All the students of the college can go to both the departments. So, let us try to represent this hierarchal relationship (Refer to the diagram below for better visualization).

In the example above, the Student entity has two parents namely - CSE Department and Library. The CSE Department and Library have the same parent College

Structure of a Network Model in DBMS

Although the network model in DBMS is a hierarchal structure but is different from the hierarchal database model as there can be numerous parents of a member.

Let us take a basic hierarchal structure to visualize the structure of a network model in DBMS.



The above structure represents a network model in which ONE is the prime owner of the model (in simple terms we can say that the rest of the members are dependent on ONE). Similar to that, member FIVE has two owners namely - TWO and THREE. The network database model allows 1 : 1 (one-to-one), 1 : M (many-to-one), M : N (many-to-one) relationships among the entities or members. The modeled hierarchal structure helps in avoiding data redundancy problems as there are multiple paths to the same record.

Refer to the Examples of Network Model in DBMS section for a better understanding of a real-life example.

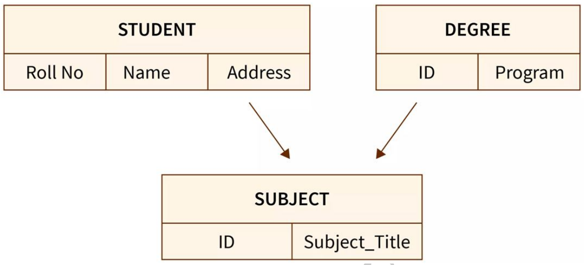
Characteristics

Let us discuss the various characteristics of the network database model.

- The network model in DBMS is better than the hierarchical model as there are more interrelations between entities.
- Supports various relationships such as one-to-one, one-to-many, and many-to-many as well.
- An entity can have various parents or owners.
- The connected structure results in high performance.
- All the entities are interconnected with each other as a connected network.
- The connected network of the database entities is represented in the form of a graph for better representation, workflow, and visualization.
- The network model in DBMS is not very flexible.
- It is also quite a complex structure to deal with.
- There can be more than one path to a certain record which makes the data retrieval faster and simple.
- The operations performed in the network database model using a circular linked list (Refer to the Operations on Network Model in DBMS section for more details).
- The Network Model in DBMS does not support the query facility.
- The 3GL programs are used by programmers to represent the relationship among the various entities of the Network Model in DBMS.

Examples of Network Model in DBMS

Let us take a basic example to visualize the structure of a network model in DBMS.



Suppose we are designing the network model for the Students database. As we can see that the Subject entity has a relationship with both the Student entity and Degree entity. So there is an edge connecting the Subject entity with both Student and Degree.

The Subject entity has two parents and the other two entities have one child entity.

Other examples of the network model in DBMS can be:-

- Store database (having relation between customers, manager, salesman, order, items, etc.).
- Finance Department database (having relation between customers, products, invoices, payments, etc.).

Examples of Network Databases

Some of the famous network databases can be:-

- TurboIMAGE
- Integrated Data Store (IDS)
- Raima Database Manager
- Univac DMS-1100
- IDMS (Integrated Database Management System), etc.

Operations on Network Model in DBMS

- **Insertion Operation:** We can insert or add a new record in the network database model but before adding any new record the database administrator or the user needs to understand the whole structure.
- **Update Operation:** We can update the data record(s). If a certain data is updated then all its children entities are also affected.
- **Deletion Operation**: We can delete the data record(s) but the deletion is a very crucial operation. Before deleting any record, we should first look out for the various connected entities so that the corresponding entities do not get affected by the deletion.
- **Retrieval Operation:** The retrieval of records in the network model in DBMS is quite complex to program but it is very fast as the entities are interconnected and various paths lead to certain records.

Q11. Explain advantages and disadvantages of Network Model.

Ans :

Advantages

- It is a simple and easy-to-construct hierarchical database model.
- The network model in DBMS allows 1 : 1 (one-to-one), 1 : M (many-to-one), M : N (many-to-one) relationships among the entities or members.
- In the network model in DBMS, there are multiple paths to the same record which helps in avoiding data redundancy problems.
- In the network model in DBMS, there is data integrity as every member entity has one or more owners. Only the prime parent has no owner but it has various inter-related children.
- The data retrieval is faster in the case of the network model in DBMS because the entities and the data are more interrelated.
- Due to the parent-child relationship, if there is a change in the parent entity, it is reflected in the children's entity as well. It also saves time as we do not need to update all the related children entities.

Disadvantages

- The network database model is very complicated due to several entities inter-related with each other. So, managing is also quite difficult.
- In the case of the addition of new entities, the database administrator or the user needs to understand the whole structure.
- Due to complex inter-related structure the addition, update, as well as deletion are very difficult.
- There is no scope for any automated query optimization.
- We need to use a pointer for navigation hence the operational anomalies exist.

1.6 Relational and Object Oriented data Models

Q12. What is Relational Model? Explain.

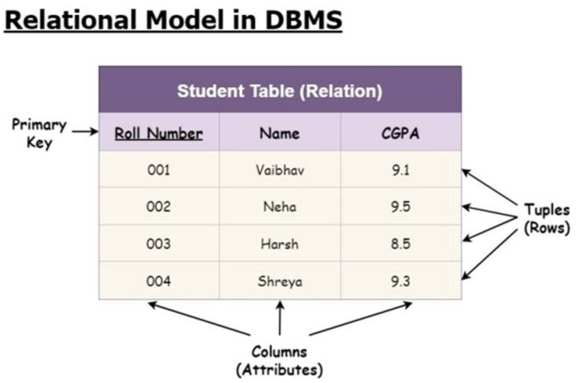
Ans :

The relational model for database management is an approach to logically represent and manage the data stored in a database. In this model, the data is organized into a collection of two-dimensional inter-related tables, also known as relations. Each relation is a collection of columns and rows, where the column represents the attributes of an entity and the rows (or tuples) represents the records.

The use of tables to store the data provided a straightforward, efficient, and flexible way to store and access structured information. Because of this simplicity, this data model provides easy data sorting and data access. Hence, it is used widely around the world for data storage and processing.

Let's look at a scenario to understand the relational model:

Consider a case where you wish to store the name, the CGPA attained, and the roll number of all the students of a particular class. This structured data can be easily stored in a table as described below:



Relational Model in DBMS

Student Table (Relation)

Roll Number	Name	CGPA
001	Vaibhav	9.1
002	Neha	9.5
003	Harsh	8.5
004	Shreya	9.3

As we can notice from the above relation

- Any given row of the relation indicates a student i.e., the row of the table describes a real-world entity.
- The columns of the table indicate the attributes related to the entity. In this case, the roll number, CGPA, and the name of the student.

Relational Model Concepts

As discussed earlier, a relational database is based on the relational model. This database consists of various components based on the relational model. These include:

- **Relation:** Two-dimensional table used to store a collection of data elements.
- **Tuple:** Row of the relation, depicting a real-world entity.
- **Attribute/Field:** Column of the relation, depicting properties that define the relation.
- **Attribute Domain:** Set of pre-defined atomic values that an attribute can take i.e., it describes the legal values that an attribute can take.
- **Degree:** It is the total number of attributes present in the relation.
- **Cardinality :** It specifies the number of entities involved in the relation i.e., it is the total number of rows present in the relation.
- **Relational Schema :** It is the logical blueprint of the relation i.e., it describes the design and the structure of the relation. It contains the table name, its attributes, and their types:

TABLE_NAME(ATTRIBUTE_1 TYPE_1, ATTRIBUTE_2 TYPE_2, ...)

For our Student relation example, the relational schema will be:

STUDENT(ROLL_NUMBER INTEGER, NAME VARCHAR(20), CGPA FLOAT)

- **Relational Instance :** It is the collection of records present in the relation at a given time.
- **Relation Key :** It is an attribute or a group of attributes that can be used to uniquely identify an entity in a table or to determine the relationship between two tables. Relation keys can be of 6 different types:
  1. Candidate Key
  2. Super Key
  3. Composite Key
  4. Primary Key
  5. Alternate Key
  6. Foreign Key

Q13. Explain values Constraints in Relational Model.

Ans : **(Imp.)**

Relational models make use of some rules to ensure the accuracy and accessibility of the data. These rules or constraints are known as Relational Integrity Constraints. These constraints are checked before performing any operation like insertion, deletion, or updation on the data present in a relational database. These constraints include:

- **Domain Constraint :** It specifies that every attribute is bound to have a value that lies inside a specific range of values. It is implemented with the help of the Attribute Domain concept.
- **Key Constraint :** It states that every relation must contain an attribute or a set of attributes (Primary Key) that can uniquely identify a tuple in that relation. This key can never be NULL or contain the same value for two different tuples.
- **Referential Integrity Constraint :** It is defined between two inter-related tables. It states that if a given relation refers to a key attribute of a different or same table, then that key must exist in the given relation.

Q14. Discuss about Anomalies in Relational Model.

Ans :

When we notice any unexpected behavior while working with the relational databases, there may be a presence of too much redundancy in the data stored in the database. This can cause anomalies in the DBMS and it can be of various types such as:

- **Insertion Anomalies:** It is the inability to insert data in the database due to the absence of other data. **For example:** Suppose we are dividing the whole class into groups for a project and the *GroupNumber* attribute is defined so that null values are not allowed. If a new student is admitted to the class but not immediately assigned to a group then this student can't be inserted into the database.
- **Deletion Anomalies:** It is the accidental loss of data in the database upon deletion of any other data element. **For example:** Suppose, we have an employee relation that contains the details of the employee along with the department they are working in. Now, if a department has one employee working in it and we remove the information of this employee from the table, there will be the loss of data related to the department also. This can lead to *data inconsistency*.
- **Modification/Update Anomalies :** It is the data inconsistency that arises from data redundancy and partial updation of data in the database. **For example:** Suppose, while updating the data into the database duplicate entries were entered. Now, if the user does not realize that the data is stored redundantly after updation, there will be data inconsistency in the database.

All these anomalies can lead to unexpected behavior and inconvenience for the user. These anomalies can be removed with the help of a process known as **normalization**.

Q15. Discuss about Codd Rules in DBMS.

Ans : **(Imp.)**

Edgar F. Codd, the creator of the relational model proposed 13 rules known as Codd Rules that states:

For a database to be considered as a perfect relational database, it must follow the following rules:

1. **Foundation Rule:** The database must be able to manage data in relational form.
2. **Information Rule:** All data stored in the database must exist as a value of some table cell.
3. **Guaranteed Access Rule:** Every unique data element should be accessible by only a combination of the table name, primary key value, and the column name.
4. **Systematic Treatment of NULL values:** Database must support NULL values.
5. **Active Online Catalog:** The organization of the database must exist in an online catalog that can be queried by authorized users.
6. **Comprehensive Data Sub-Language Rule:** Database must support at least one language that supports: data definition, view definition, data manipulation, integrity constraints, authorization, and transaction boundaries.
7. **View Updating Rule:** All views should be theoretically and practically updatable by the system.
8. **Relational Level Operation Rule:** The database must support high-level insertion, updation, and deletion operations.
9. **Physical Data Independence Rule:** Data stored in the database must be independent of the applications that can access it i.e., the data stored in the database must not depend on any other data or an application.
10. **Logical Data Independence Rule:** Any change in the logical representation of the data (structure of the tables) must not affect the user's view.
11. **Integrity independence:** Changing the integrity constraints at the database level should not reflect any change at the application level.
12. **Distribution independence:** The database must work properly even if the data is stored in multiple locations or is being used by multiple end-users.
13. **Non-subversion Rule:** Accessing the data by low-level relational language should not be able to bypass the integrity rules and constraints expressed in the high-level relational language.

Q16. State advantages and disadvantages of relational model.

Ans :

Advantages

The advantages and reasons due to which the relational model in DBMS is widely accepted as a standard are:

- **Simple and Easy To Use:** Storing data in tables is much easier to understand and implement as compared to other storage techniques.
- **Manageability:** Because of the independent nature of each relation in a relational database, it is easy to manipulate and manage. This improves the performance of the database.
- **Query capability:** With the introduction of relational algebra, relational databases provide easy access to data via high-level query language like SQL.
- **Data integrity:** With the introduction and implementation of relational constraints, the relational model can maintain data integrity in the database.

Disadvantages

The main disadvantages of relational model in DBMS occur while dealing with a huge amount of data as:

- The performance of the relational model depends upon the number of relations present in the database.
- Hence, as the number of tables increases, the requirement of physical memory increases.
- The structure becomes complex and there is a decrease in the response time for the queries.
- Because of all these factors, the cost of implementing a relational database increase.

1.7 Integrity Constraints

Q17. Explain types of integrity constraints with example.

Ans :

Integrity Constraints Over Relation

- Database integrity refers to the validity and consistency of stored data. Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate. Constraints may apply to each attribute or they may apply to relationships between tables.
- Integrity constraints ensure that changes (update deletion, insertion) made to the database by authorized users do not result in a loss of data consistency. Thus, integrity constraints guard against accidental damage to the database.

  Example: A brood group must be 'A' or 'B' or 'AB' or 'O' only (can not any other values else).

TYPES

Various types of integrity constraints are-

1. Domain Integrity
2. Entity Integrity Constraint
3. Referential Integrity Constraint
4. Key Constraints

1. Domain Integrity

Domain integrity means the definition of a valid set of values for an attribute. You define data type, length or size, is null value allowed , is the value unique or not for an attribute ,the default value, the range (values in between) and/or specific values for the attribute.

2. Entity Integrity Constraint

This rule states that in any database relation value of attribute of a primary key can't be null.

Example

Consider a relation "STUDENT" Where "Stu_id" is a primary key and it must not contain any null value whereas other attributes may contain null value e.g "Branch" in the following relation contains one null value.

Stu_id	Name	Branch
11255234	Aman	CSE
11255369	Kapil	ECE
11255324	Ajay	ME
11255237	Raman	CSE
11255678	Aastha	ECE

3. Referential Integrity Constraint

It states that if a foreign key exists in a relation then either the foreign key value must match a primary key value of some tuple in its home relation or the foreign key value must be null.

The rules are:

1. You can't delete a record from a primary table if matching records exist in a related table.
2. You can't change a primary key value in the primary table if that record has related records.
3. You can't enter a value in the foreign key field of the related table that doesn't exist in the primary key of the primary table.
4. However, you can enter a Null value in the foreign key, specifying that the records are unrelated.

Example

Consider 2 relations "stu" and "stu_1" Where "Stu_id " is the primary key in the "stu" relation and foreign key in the "stu_1" relation.

Relation "stu"

Stu_id	Name	Branch
11255234	Aman	CSE
11255369	Kapil	EcE
11255324	Ajay	ME
11255237	Raman	CSE
11255678	Aastha	ECE

Relation "stu_1"

Stu_id	Name	Branch
11255234	B TECH	4 years
11255369	B TECH	4 years
11255324	B TECH	4 years
11255237	B TECH	4 years
11255678	B TECH	4 years

Examples

Rule 1: You can't delete any of the rows in the "stu-" relation that are visible since all the "stu" are in use in the "stu_1" relation.

Rule 2: You can't change any of the "Stu_id" in the "stu" relation since all the "Stu_id" are in use in the "stu_1" relation. * *Rule 3.** The values that you can enter in the" Stu_id" field in the "stu_1" relation must be in the" Stu_id" field in the "stu" relation.

Rule 4: You can enter a null value in the "stu_1" relation if the records are unrelated.

4. Key Constraints

A Key Constraint is a statement that a certain minimal subset of the fields of a relation is a unique identifier for a tuple. The types of key constraints-

i) Primary key constraints

ii) Unique key constraints

iii) Foreign Key constraints

iv) NOT NULL constraints

v) Check constraints

i) Primary key constraints

Primary key is the term used to identify one or more columns in a table that make a row of data unique. Although the primary key typically consists of one column in a table, more than one column can comprise the primary key.

For example, either the employee's Social Security number or an assigned employee identification number is the logical primary key for an employee table. The objective is for every record to have a unique primary key or value for the employee's identification number. Because there is probably no need to have more than one record for each employee in an employee table, the employee identification number makes a logical primary key. The primary key is assigned at table creation.

The following example identifies the EMP_ID column as the PRIMARY KEY for the EMPLOYEES table:

```
CREATE TABLE EMPLOYEE_TBL
(EMP_ID          CHAR(9)         NOT NULL PRIMARY KEY,
EMP_NAME         VARCHAR (40)    NOT NULL,
EMP_ST_ADDR      VARCHAR (20)    NOT NULL,
EMP_CITY         VARCHAR (15)    NOT NULL,
EMP_ST           CHAR(2)         NOT NULL,
EMP_ZIP          INTEGER(5)      NOT NULL,
EMP_PHONE        INTEGER(10)     NULL,
EMP_PAGER        INTEGER(10)     NULL);
```

ii) Unique Key Constraints

A unique column constraint in a table is similar to a primary key in that the value in that column for every row of data in the table must have a unique value. Although a primary key constraint is placed on one column, you can place a unique constraint on another column even though it is not actually for use as the primary key.

```
CREATE TABLE EMPLOYEE_TBL
    (EMP_ID          CHAR(9)         NOT NULL     PRIMARY KEY,
    EMP_NAME         VARCHAR (40)    NOT NULL,
    EMP_ST_ADDR      VARCHAR (20)    NOT NULL,
    EMP_CITY         VARCHAR (15)    NOT NULL,
    EMP_ST           CHAR(2)         NOT NULL,
    EMP_ZIP          INTEGER(5)      NOT NULL,
    EMP_PHONE        INTEGER(10)     NULL              UNIQUE,
    EMP_PAGER        INTEGER(10)     NULL)
```

iii) Foreign Key Constraints

A foreign key is a column in a child table that references a primary key in the parent table. A foreign key constraint is the main mechanism used to enforce referential integrity between tables in a relational database. A column defined as a foreign key is used to reference a column defined as a primary key in another table.

```
CREATE TABLE EMPLOYEE_PAY_TBL
          (EMP_ID            CHAR(9)            NOT NULL,
          POSITION           VARCHAR2(15)       NOT NULL,
          DATE_HIRE          DATE               NULL,
          PAY_RATE           NUMBER(4,2)        NOT NULL,
          DATE_LAST_RAISE    DATE               NULL,
```

iv) NOT NULL Constraints

Previous examples use the keywords NULL and NOT NULL listed on the same line as each column and after the data type. NOT NULL is a constraint that you can place on a table's column. This constraint disallows the entrance of NULL values into a column; in other words, data is required in a NOT NULL column for each row of data in the table. NULL is generally the default for a column if NOT NULL is not specified, allowing NULL values in a column.

v) Check Constraints

Check (CHK) constraints can be utilized to check the validity of data entered into particular table columns. Check constraints are used to provide back-end database edits, although edits are commonly found in the front-end application as well. General edits restrict values that can be entered into columns or objects, whether within the database itself or on a front-end application. The check constraint is a way of providing another protective layer for the data.

CREATE TABLE EMPLOYEE_TBL

(EMP_ID	C	HAR(9NOT NULL,
EMP_NAME	VARCHAR2(40)	NOT NULL,
EMP_ST_ADDR	VARCHAR2(20)	NOT NULL,
EMP_CITY	VARCHAR2(15)	NOT NULL,
EMP_ST	CHAR(2)	NOT NULL,
EMP_ZIP	NUMBER(5)	NOT NULL,
EMP_PHONE	NUMBER(10)	NULL,
EMP_PAGER	NUMBER(10)	NULL),
PRIMARY KEY		(EMP_ID),

CONSTRAINT CHK_EMP_ZIP CHECK (EMP_ZIP = '46234');

1.8 DAA MANIPULAIONOPERTIONS

Q18. Explain briefly about Data Manipulation Operations.

Ans :

The majority of SQL statements are categorised as DML (Data Manipulation Language), which includes SQL commands that deal with modifying data in a database. It's the section of the SQL statement that controls who has access to the database and data. DML statements and DCL statements are grouped together. Because the DML command isn't auto-committed, it won't be able to save all database changes permanently. There's a chance they'll be rolled back.

Here are some different DML commands:

i) INSERT INTO Command

This command can be used to insert data into a row of a table. INSERT INTO would insert the values that are mentioned in the 'Student' table below.

Syntax:

```
INSERT INTO NAME_OF_TABLE (1_column, 2_column, 3_column, .... N_column)
VALUES (1_value, 2_value, 3_value, .... N_value);
```

Or

```
INSERT INTO NAME_OF_TABLE
VALUES (1_value, 2_value, 3_value, …. N_value);
```

Example

```
INSERT INTO Student(Stu_Name, DOB, Phone, Mail)
VALUES('Phoebe', '1998-05-26', 7812865845, 'user@xyz.com');
```

ii) UPDATE Command

This statement in SQL is used to update the data that is present in an existing table of a database. The UPDATE statement can be used to update single or multiple columns on the basis of our specific needs.

Syntax

UPDATE name_of_table SET 1_coumn = 1_value, 2_coumn = 2_value, 3_coumn = 3_value, …, N_coumn = N_value

WHERE condition;

And here,

name_of_table: name of the table

1_column, 2_column, 3_column, …. N_column: name of the first, second, third, …. nth column.

1_value, 2_value, 3_value, …. N_value: the new value for the first, second, third, …. nth column.

Condition: the condition used to select those rows for which the column values need to be updated.

Example

```
UPDATE Student SET Phone = 9039462901 WHERE Stu_Name = 'Phoebe';
```

The WHERE clause in the preceding query is used to select the rows for which the columns need to be adjusted, and the SET statement has been used to assign new values to a particular column. If the WHERE clause is not used at all, then all of the rows' columns will be modified. As a result, the WHERE clause is used to pick specific rows from the table.

Thus, the example query would update the phone number of the student with the name 'Phoebe'.

iii) DELETE Command

The DELETE statement can be used in SQL to delete various records from a given table. On the basis of the condition that has been set in the WHERE clause, one can delete single or multiple records.

Syntax

```
DELETE FROM name_of_table [WHERE condition];
```

Example

```
DELETE FROM Student WHERE Stu_Name = 'Phoebe';
```

The command given above would delete the record for the student with the name 'Phoebe' from the 'Student' table. Apart from this, one can also use the LOCK Table statement to explicitly acquire the shared or exclusive table lock on a specified table.

Short Question and Answers

1. What is data abstraction?

Ans :

Data abstraction is present in our daily lives. Let us take a small example. Say, someone, asks you to switch on the fans in a room. All you will need to do is simply walk to the switchboard and turn on the switch for the fan, that's it! Do you need to know where the electricity is coming from, how the poles of the switch are connected, or exactly what the internal working of a fan is? The answer to all this is NO! That is what data abstraction is, all these background details are hidden from you inside the switchboard!

All the databases have complex data structures which are, in fact, of no use to an end user. Thus, these internal irrelevant details are hidden from the user, making the accessing of data simple and increasing the security of the data as well.

2. Data Independence

Ans :

The ability to modify the schema definition of a DBMS at one level, without affecting the schema definition of the next higher level is called data independence.

In addition to the data entered by users, a database system typically holds a large amount of data. The system holds metadata about data which makes it easier to find and retrieve data. Once a set of metadata has been saved in a database, changing or updating the metadata is challenging. However, as a database management system (DBMS) grows, it must evolve to meet the needs of its users. Updating the schema or data would be a time-consuming and complicated task if all of the data were dependent.

3. What are Data Definition Language Commands? Explain.

Ans :

Data Definition Language(DDL) is a subset of SQL and a part of DBMS (Database Management System). DDL consist of Commands to commands like CREATE, ALTER, TRUNCATE and DROP. These commands are used to create or modify the tables in SQL.

DDL Commands

1. Create
2. Alter
3. truncate
4. drop

4. Explain briefly about ER Model.

Ans :

This model defines the data elements and relationships for a specified system. It is useful in developing a conceptual design for the database & is very simple and easy to design logical view of data.

ER Diagrams

- ERD stands for Entity Relationship diagram.
- It is a graphical representation of an information system.
- ER diagram shows the relationship between objects, places, people, events etc. within that system.
- It is a data modeling technique which helps in defining the business process.
- It used for solving the design problems.

5. Network Model

Ans :

The hierarchical model is extended in the network model. Prior to the relational model, it was the most popular model. To increase database performance and standards, the network model was devised to express complicated data relationships more effectively than hierarchical models. It has entities that are grouped in a graphical format, and some of the entities can be reached by many paths.

Let us take an example of a simple College database that has two departments or sections namely - the CSE Department and the Library. All the students of the college can go to both the departments. So, let us try to represent this hierarchal relationship (Refer to the diagram below for better visualization).

6. Characteristics of the network database model.

Ans :

- The network model in DBMS is better than the hierarchical model as there are more interrelations between entities.
- Supports various relationships such as one-to-one, one-to-many, and many-to-many as well.
- An entity can have various parents or owners.
- The connected structure results in high performance.
- All the entities are interconnected with each other as a connected network.
- The connected network of the database entities is represented in the form of a graph for better representation, workflow, and visualization.
- The network model in DBMS is not very flexible.
- It is also quite a complex structure to deal with.

7. Relational Model

Ans :

The relational model for database manage-ment is an approach to logically represent and manage the data stored in a database. In this model, the data is organized into a collection of two-dimensional inter-related tables, also known as relations. Each relation is a collection of columns and rows, where the column represents the attributes of an entity and the rows (or tuples) represents the records.

The use of tables to store the data provided a straightforward, efficient, and flexible way to store and access structured information. Because of this simplicity, this data model provides easy data sorting and data access. Hence, it is used widely around the world for data storage and processing.

8. Discuss about Anomalies in Relational Model.

Ans :

When we notice any unexpected behavior while working with the relational databases, there may be a presence of too much redundancy in the data stored in the database. This can cause anomalies in the DBMS and it can be of various types such as:

- **Insertion Anomalies:** It is the inability to insert data in the database due to the absence of other data. **For example:** Suppose we are dividing the whole class into groups for a project and the *GroupNumber* attribute is defined so that null values are not allowed. If a new student is admitted to the class but not immediately assigned to a group then this student can't be inserted into the database.
- **Deletion Anomalies:** It is the accidental loss of data in the database upon deletion of any other data element. **For example:** Suppose, we have an employee relation that contains the details of the employee along with the department they are working in. Now, if a department has one employee working in it and we remove the information of this employee from the table, there will be the loss of data related to the department also. This can lead to *data inconsistency*.
- **Modification/Update Anomalies :** It is the data inconsistency that arises from data redundancy and partial updation of data in the database. **For example:** Suppose, while updating the data into the database duplicate entries were entered. Now, if the user does not realize that the data is stored redundantly after updation, there will be data inconsistency in the database.

9. Integrity constraints

Ans :

- Database integrity refers to the validity and consistency of stored data. Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate. Constraints may apply to each attribute or they may apply to relationships between tables.
- Integrity constraints ensure that changes (update deletion, insertion) made to the database by authorized users do not result in a loss of data consistency. Thus, integrity constraints guard against accidental damage to the database.

Example: A brood group must be 'A' or 'B' or 'AB' or 'O' only (can not any other values else).

10. Data Manipulation Operations.

Ans :

The majority of SQL statements are categorised as DML (Data Manipulation Language), which includes SQL commands that deal with modifying data in a database. It's the section of the SQL statement that controls who has access to the database and data. DML statements and DCL statements are grouped together. Because the DML command isn't auto-committed, it won't be able to save all database changes permanently. There's a chance they'll be rolled back.

Choose the Correct Answers

1. What is the basic client/server architecture, one has to deal with? [d]

   (a) Large number of PCs (b) Web servers

   (c) Database Servers (d) All of the above

2. How many types of DBMS architectures are there? [c]

   (a) 1 (b) 2

   (c) 3 (d) 4

3. What is TRUE about 1-tier architecture? [c]

   (a) It is directly not available to the user

   (b) Changes are not done on the database

   (c) No handy tool is provided for the end user

   (d) It is not used for the development of local application

4. Basic client-server model is similar to [a]

   (a) 2-tier architecture (b) 3-tier architecture

   (c) 4-tier architecture (d) 5-tier architecture

5. Which API is used for the interaction in 2-tier architecture? [a]

   (a) ODBC (b) JDBC

   (c) Both A and B (d) None of the above

6. What is done through client-side in the 2-tier architecture? [d]

   (a) Running of user interfaces and application program

   (b) To provide query processing and transaction management functionalities

   (c) Establish a connection with the other side

   (d) Both A and C

7. In which architecture, another layer is there between client and server? [c]

   (a) 1-tier architecture

   (b) 2-tier architecture

   (c) 3-tier architecture

   (d) 4-tier architecture

8. In which architecture, client can't directly communicate with the server? [c]

   (a) 1-tier architecture (b) 2-tier architecture

   (c) 3-tier architecture (d) None of the above

9. What is present in Server but not in Client? [a]

(a) Database (b) Application Server

(c) User (d) None

10. 3-tier architecture is used in – [a]

(a) Large web application

(b) Small web application

(c) Both large and small web application

(d) Neither small nor large web application

Fill in the blanks

1. ________ is the modelling of the data description, data semantics, and consistency constraints of the data.
2. An ________ model is the logical representation of data as objects and relationships among them.
3. The ability to modify the schema definition of a DBMS at one level, without affecting the schema definition of the next higher level is called ________.
4. ________ independence is the ability to modify logical schema without causing any unwanted modifications to the external schema or the application programs to be rewritten.
5. ________ consist of Commands to commands like CREATE, ALTER, TRUNCATE and DROP.
6. DML stands for ________ .
7. ________ command is used for inserting a data into a table.
8. ERD stands for ________ .
9. An ________ is a property or characteristic of an entity.
10. ________ attributes can be divided into sub-attributes which represent more basic attributes with independent meanings.

Answers

1. Data Model
2. Entity Relationship (ER)
3. Data Independence
4. Logical data
5. DDL
6. Data Manipulation Language
7. INSERT
8. Entity Relationship diagram
9. Attribute
10. Composite

UNIT II

Relational Query Languages and Relational Database Design: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

2.1 Relational Query Languages and Relational Database Designs

2.1.1 Relational Algebra

Q1. Explain about the basic relational operations in relational algebra.

Ans : **(Imp.)**

Relational algebra in DBMS is a procedural query language. Queries in relational algebra are performed using operators. Relational Algebra is the fundamental block for modern language SQL and modern Database Management Systems such as Oracle Database, Mircosoft SQL Server, IBM Db2, etc.

Types of Relational Operations

In Relation Algebra, we have two types of Operations.

1. Basic Operations
2. Derived Operations

1. Basic Operations

Six fundamental operations are mentioned below. The majority of data retrieval operations are carried out by these. Let's know them one by one.

But, before moving in detail, let's have two tables or we can say relations STUDENT(ROLL, NAME, AGE) and EMPLOYEE (EMPLOYEE_ NO, NAME, AGE) which will be used in the below examples.

STUDENT

ROLL	NAME	AGE
1	Aman	20
2	Atul	18
3	Baljeet	19
4	Harsh	20
5	Prateek	21
6	Prateek	23

EMPLOYEE

EMPLOYEE_NO	NAME	AGE
E-1	Anant	20
E-2	Ashish	23
E-3	Baljeet	25
E-4	Harsh	20
E-5	Pranav	22

Select (σ)

Select operation is done by Selection Operator which is represented by "sigma" (σ). It is used to retrieve tuples (rows) from the table where the given condition is satisfied. It is a unary operator means it require only one operand.

Notation : σ p(R)

Where σ is used to represent SELECTION

R is used to represent RELATION

p is the logic formula

Let's understand this with an example:

Suppose we want the row(s) from STUDENT Relation where "AGE" is 20

σ AGE = 20 (STUDENT)

This will return the following output:

ROLL	NAME	AGE
1	Aman	20
4	Harsh	20

Project (π)

Project operation is done by Projection Operator which is represented by "pi"(π). It is used to retrieve certain attributes (columns) from the table. It is also known as vertical partitioning as it separates the table vertically. It is also a **unary operator**. Notation : π a(r)

Where π is used to represent PROJECTION

r is used to represent RELATION

a is the attribute list

Let's understand this with an example:

Suppose we want the names of all students from STUDENT Relation.

Π NAME(STUDENT)

This will return the following output:

NAME

Aman

Atul

Baljeet

Harsh

Prateek

As you can see from the above output it eliminates duplicates.

For multiple attributes, we can separate them using a ",".

Π ROLL,NAME(STUDENT)

Above code will return two columns, ROLL and NAME.

ROLL	NAME
1	Aman
2	Atul
3	Baljeet
4	Harsh
5	Prateek
6	Prateek

Union (∪)

Union operation is done by Union Operator which is represented by "union"(∪). It is the same as the union operator from set theory, i.e., it selects all tuples from both relations but with the exception that for the union of two relations/tables both relations must have the same set of Attributes. It is a binary operator as it requires two operands. Notation: R∪S

Where R is the first relation

S is the second relation

If relations don't have the same set of attributes, then the union of such relations will result in NULL. Let's have an example to clear the concept:

Suppose we want all the names from STUDENT and EMPLOYEE relation.

Π NAME(STUDENT) ∪ Π NAME (EMPLOYEE)

NAME

Aman

Anant

Ashish

Atul

Baljeet

Harsh

Pranav

Prateek

As we can see from above output it also eliminates duplicates.

Set Difference (–)

Set Difference as its name indicates is the difference of two relations (R-S). It is denoted by a "Hyphen"(-) and it returns all the tuples(rows) which are in relation R but not in relation S. It is also a binary operator.

Notation : R - S

Where R is the first relation

S is the second relation

Just like union, the set difference also comes with the exception of the same set of attributes in both relations.

Let's take an example where we would like to know the names of students who are in STUDENT Relation but not in EMPLOYEE Relation.

Π NAME(STUDENT) – Π NAME(EMPLOYEE)

This will give us the following output:

NAME

Aman

Atul

Prateek

Cartesian product (X)

Cartesian product is denoted by the "X" symbol. Let's say we have two relations R and S. Cartesian product will combine every tuple(row) from R with all the tuples from S. I know it sounds complicated, but once we look at an example, you'll see what I mean.

Notation: R X S

Where R is the first relation

S is the second relation

As we can see from the notation it is also a binary operator. Let's combine the two relations STUDENT and EMPLOYEE.

STUDENT X EMPLOYEE

ROLL	NAME	AGE	EMPLOYEE_NO	NAME	AGE
1	Aman	20	E-1	Anant	20
1	Aman	20	E-2	Ashish	23
1	Aman	20	E-3	Baljeet	25
1	Aman	20	E-4	Harsh	20
1	Aman	20	E-5	Pranav	22
2	Atul	18	E-1	Anant	20
2	Atul	18	E-2	Ashish	23
2	Atul	18	E-3	Baljeet	25
2	Atul	18	E-4	Harsh	20
2	Atul	18	E-5	Pranav	22

Rename (ρ)

Rename operation is denoted by "Rho" (ρ). As its name suggests it is used to rename the output relation. Rename operator too is a **binary operator**. Notation: ρ(R,S) Where R is the new relation name

S is the old relation name

Let's have an example to clear this

Suppose we are fetching the names of students from STUDENT relation. We would like to rename this relation as STUDENT_NAME.

ρ(STUDENT_NAME,” NAME(STUDENT))

STUDENT_NAME

NAME

Aman

Atul

Baljeet

Harsh

Prateek

As you can see, this output relation is named “STUDENT_NAME”.

Takeaway

- Select (σ) is used to retrieve tuples(rows) based on certain conditions.
- Project (Π) is used to retrieve attributes (columns) from the relation.
- Union (∪) is used to retrieve all the tuples from two relations.
- Set Difference (–) is used to retrieve the tuples which are present in R but not in S(R-S).
- Cartesian product (X) is used to combine each tuple from first relation with each tuple from second relation.
- Rename (ρ) is used to rename the output relation.

2. Derived Operations

Also known as extended operations, these operations can be derived from basic operations hence named Derived Operations. These include three operations: Join Operations, Intersection operation, and Division operation.

- Join Operations are binary operations that allow us to combine two or more relations.
- They are further classified into two types: Inner Join, and Outer Join.

First, let’s have two relations EMPLOYEE consisting of E_NO, E_NAME, CITY and EXPERIENCE. EMPLOYEE table contains employee’s information such as id, name, city, and experience of employee(In Years). The other relation is DEPARTMENT consisting of D_NO, D_NAME, E_NO and MIN_EXPERIENCE. DEPARTMENT table defines the mapping of an employee to its department. It contains Department Number, Department Name, Employee Id of the employee working in that department and, minimum experience required (In Years) to be in that department.

EMPLOYEE

E_NO	E_NAME	CITY	EXPERIENCE
E-1	Ram	Delhi	04
E-2	Varun	Chandigarh	09
E-3	Ravi	Noida	03
E-4	Amit	Bangalore	07

DEPARTMENT

D_NO	D_NAME	E_NO	EXPERIENCE
D-1	HR	E-1	03
D-2	IT	E-2	05
D-3	Marketing	E-3	02

Also, let's have the Cartesian Product of the above two relations. It will be much easier to understand Join Operations when we have the Cartesian Product.

E_NO	E_NAME	CITY	EXPERIENCE	D_NO	D_NAME	E_NO	MIN_EXPERIENCE
E-1	Ram	Delhi	04	D-1	HR	E-1	03
E-1	Ram	Delhi	04	D-2	IT	E-2	05
E-1	Ram	Delhi	04	D-3	Marketing	E-3	02
E-2	Varun	Chandigarh	09	D-1	HR	E-1	03
E-2	Varun	Chandigarh	09	D-2	IT	E-2	05
E-2	Varun	Chandigarh	09	D-3	Marketing	E-3	02
E-3	Ravi	Noida	03	D-1	HR	E-1	03
E-3	Ravi	Noida	03	D-2	IT	E-2	05
E-3	Ravi	Noida	03	D-3	Marketing	E-3	02
E-4	Amit	Bangalore	07	D-1	HR	E-1	03
E-4	Amit	Bangalore	07	D-2	IT	E-2	05
E-4	Amit	Bangalore	07	D-3	Marketing	E-3	02

I) Inner Join

When we perform Inner Join, only those tuples are returned which satisfies the certain condition. It is also classified into three types: Theta Join, Equi Join and Natural Join.

i) Theta Join (θ)

Theta Join combines two relations using a condition. This condition is represented by the symbol "theta"(θ). Here conditions can be inequality conditions such as >,<,>=,<=, etc.

Notation : $R \bowtie \theta S$

Where R is the first relation

S is the second relation

Let's have a simple example to understand this.

Suppose we want a relation where EXPERIENCE from EMPLOYEE >

= MIN_EXPERIENCE from DEPARTMENT.

EMPLOYEE ⋈ θEMPLOYEE.EXPERIENCE>=DEPARTMENT.MIN_EXPERIENCE DEPARTMENT

E_NO	E_NAME	CITY	EXPERIENCE	D_NO	D_NAME	E_NO	MIN_ EXPERIENCE
E-1	Ram	Delhi	04	D-1	HR	E-1	03
E-1	Ram	Delhi	04	D-3	Marketing	E-3	02
E-2	Varun	Chandigarh	09	D-1	HR	E-1	03
E-2	Varun	Chandigarh	09	D-2	IT	E-2	05
E-2	Varun	Chandigarh	09	D-3	Marketing	E-3	02
E-3	Ravi	Noida	03	D-1	HR	E-1	03
E-3	Ravi	Noida	03	D-3	Marketing	E-3	02
E-4	Amit	Bangalore	07	D-1	HR	E-1	03
E-4	Amit	Bangalore	07	D-2	IT	E-2	05
E-4	Amit	Bangalore	07	D-3	Marketing	E-3	02

Check the Cartesian Product, if in any tuple/row EXPERIENCE >= MIN_EXPERIENCE then insert this tuple/row in output relation.

ii) Equi Join

Equi Join is a special case of theta join where the condition can only contain **equality(=)** comparisons.

A non-equijoin is the inverse of an equi join, which occurs when you join on a condition other than "=".

Let's have an example where we would like to join EMPLOYEE and DEPARTMENT relation where E_NO from EMPLOYEE = E_NO from DEPARTMENT.

EMPLOYEE ⋈ EMPLOYEE.E_NO = DEPARTMENT.E_NO DEPARTMENT

E_NO	E_NAME	CITY	EXPERIENCE	D_NO	D_NAME	E_NO	MIN_ EXPERIENCE
E-1	Ram	Delhi	04	D-1	HR	E-1	03
E-2	Varun	Chandigarh	09	D-2	IT	E-2	05
E-3	Ravi	Noida	03	D-3	Marketing	E-3	02

Check Cartesian Product, if the tuple contains same E_NO, insert that tuple in output relation

iii) Natural Join (⋈)

A comparison operator is not used in a natural join. It does not concatenate like a Cartesian product. A Natural Join can be performed only if two relations share at least one common attribute. Furthermore, the attributes must share the same name and domain.

Natural join operates on matching attributes where the values of the attributes in both relations are the same and remove the duplicate ones.

Preferably Natural Join is performed on the foreign key.

Notation : R ⋈ S

Where R is the first relation

S is the second relation

Let's say we want to join EMPLOYEE and DEPARTMENT relation with E_NO as common attribute.

Notice, here E_NO has same name in both the relations and also consists of same domain, i.e., in both relations E_NO is a string.

EMPLOYEE ⋈ DEPARTMENT

E_NO	E_NAME	CITY	EXPERIENCE	D_NO	D_NAME	MIN_EXPERIENCE
E-1	Ram	Delhi	04	D-1	HR	03
E-2	Varun	Chandigarh	09	D-2	IT	05
E-3	Ravi	Noida	03	D-3	Marketing	02

But unlike above operation, where we have two columns of E_NO, here we are having only one column of E_NO. This is because **Natural Join automatically keeps single copy of common attribute**.

Outer Join

Unlike Inner Join which includes the tuple that satisfies the given condition, Outer Join also includes some/all the tuples which doesn't satisfies the given condition. It is also of three types: Left Outer Join, Right Outer Join, and Full Outer Join.

Let's say we have two relations R and S, then

Left Outer Join

As we can see from the diagram, Left Outer Join returns the matching tuples(tuples present in both relations) and the tuples which are only present in Left Relation, here R.

However, if the matching tuples are NULL, then attributes/columns of Right Relation, here S are made NULL in the output relation.

Let's understand this a bit more using an example:

EMPLOYEE ⋈ EMPLOYEE.E_NO = DEPARTMENT.E_NO DEPARTMENT

Here we are combining EMPLOYEE and DEPARTMENT relation with constraint that EMPLOYEE's E_NO must be equal to DEPARTMENT's E_NO.

E_NO	E_NAME	CITY	EXPERIENCE	D_NO	D_NAME	MIN_EXPERIENCE
E-1	Ram	Delhi	04	D-1	HR	03
E-2	Varun	Chandigarh	09	D-2	IT	05
E-3	Ravi	Noida	03	D-3	Marketing	02

As all the tuples from DEPARTMENT relation have a corresponding E_NO in EMPLOYEE relation, therefore no tuple from EMPLOYEE relation contains a NULL.

Full Outer Join

Full Outer Join returns all the tuples from both relations. However if there are no matching tuples then, their respective attributes are made NULL in output relation.

Again, combine the EMPLOYEE and DEPARTMENT relation with same constraint.

EMPLOYEE ⋈ EMPLOYEE.E_NO = DEPARTMENT.E_NO DEPARTMENT

E_NO	E_NAME	CITY	EXPERIENCE	D_NO	D_NAME	MIN_EXPERIENCE
E-1	Ram	Delhi	04	D-1	HR	03
E-2	Varun	Chandigarh	09	D-2	IT	05
E-3	Ravi	Noida	03	D-3	Marketing	02
E-4	Amit	Bangalore	07	-	-	-

Takeaway

- Theta Join (θ) combines two relations based on a condition.
- Equi Join is a type of Theta Join where only equality condition (=) is used.
- Natural Join (⋈) combines two relations based on a common attribute (preferably foreign key).
- Left Outer Join (⟕) returns the matching tuples and tuples which are only present in left relation.
- Right Outer Join (⟖) returns the matching tuples and tuples which are only present in the right relation.
- Full Outer Join (⟗) returns all the tuples present in the left and right relations.

Intersection (∩)

Intersection operation is done by Intersection Operator which is represented by "intersection"(∩).It is the same as the intersection operator from set theory, i.e., it selects all the tuples which are present in both relations. It is a binary operator as it requires two operands. Also, it eliminate duplicates. Notation : R∩S

Where R is the first relation

S is the second relation

Let's have an example to clear the concept:

Suppose we want the names which are present in STUDENT as well as in EMPLOYEE relation, Relations we used in Basic Operations.

Π NAME(STUDENT)) ∩ Π NAME(EMPLOYEE)

NAME

Baljeet

Harsh

Division (÷)

Division Operation is represented by "division" (÷ or /) operator and is used in queries which involve keyword **"every"**, **"all"**, etc.

Notation : R(X,Y)/S(Y)

Here,

R is the first relation from which data is to retrieved.

S is second relation which will help to retrieve the data.

X and Y are the attributes/columns present in relation. We can have multiple attributes in relation, but keep in mind that attributes of S must be proper subset of attributes of R.

For each corresponding value of Y, above notation will return us the value of X from tuple<X,Y> which exist **everywhere**.

It's a bit difficult to understand this in theoretical way, but you will understand this with an example.

Let's have two relations, ENROLLED and COURSE. ENROLLED consist of two attributes STUDENT_ID and COURSE_ID. It denotes the map of students who are enrolled in given courses.

COURSE contains the list of courses available.

See, here attributes/columns of COURSE relation are proper subset of attributes/columns of ENROLLED relation. Hence Division operation can be used here.

ENROLLED

STUDENT_ID	COURSE_ID
Student_1	DBMS
Student_2	DBMS
Student_1	OS
Student_3	OS

COURSE

COURSE_ID
DBMS
OS

Now the query is to return the STUDENT_ID of students who are enrolled in **every** course.

ENROLLED(STUDENT_ID, COURSE_ID)/ COURSE(COURSE_ID)

This will return the following relation as output.

STUDENT_ID
Student_1

2.1.2 Tuple and Domain Relational Calculus

Q2. What is Relational Calculus? Explain different types of relational calculus in DBMS.

Ans : **(Imp.)**

Meaning

Before understanding Relational calculus in DBMS, we need to understand Procedural Language and Declarative Language.

1. **Procedural Language:** Those Languages which clearly define how to get the required results from the Database are called Procedural Language. Relational algebra is a Procedural Language.

2. **Declarative Language:** Those Language that only cares about What to get from the database without getting into how to get the results are called Declarative Language. **Relational Calculus** is a Declarative Language.

So Relational Calculus is a Declarative Language that uses Predicate Logic or First-Order Logic to determine the results from Database.

Relational Calculus is of Two Types:

1. Tuple Relational Calculus (TRC)
2. Domain Relational Calculus (DRC)

1. Tuple Relational Calculus (TRC)

Tuple Relational Calculus in DBMS uses a tuple variable (t) that goes to each row of the table and checks if the predicate is true or false for the given row. Depending on the given predicate condition, it returns the row or part of the row.

The Tuple Relational Calculus expression Syntax

{t \| P(t)}

Where t is the tuple variable that runs over every Row, and P(t) is the predicate logic expression or condition.

Let's take an example of a Customer Database and try to see how TRC expressions work.

Customer Table

Customer_id	Name	Zip code
1	Rohit	12345
2	Rahul	13245
3	Rohit	56789

Example 1

Write a TRC query to get all the data of customers whose zip code is 12345.

TRC Query: {t \|t ∈Customer ∧t.Zipcode = 12345} or TRC Query: {t \| Customer(t) ∈ t[Zipcode] = 12345 }

Workflow of query

The tuple variable "t" will go through every tuple of the Customer table. Each row will check whether the Cust_Zipcode is 12345 or not and only return those rows that satisfies the Predicate expression condition.

The TRC expression above can be read as "Return all the tuple which belongs to the Customer Table and whose Zipcode is equal to 12345."

Result of the TRC expression above:

X to get the column values required from the database based on the predicate expression or condition.

2. Domain realtional calculus

The Domain realtional calculus expression syntax:

{<x1, x2, x3, x4...> \| P(x1, x2, x3, x4...)}

where,

<x1,x2,x3,x4...> are domain variables used to get the column values required, and **P(x1, x2, x3...)** is predicate expression or condition.

Let's take the example of Customer Database and try to understand DRC queries with some examples.

Customer Table

Customer_id	Name	Zip code
1	Rohit	12345
2	Rahul	13245
3	Rohit	56789
4	Amit	12345

Example 1

Write a DRC query to get the data of all customers with Zip code 12345.

DRC query

{< ∃ ,x2,x3> \| <x1,x2>" Customer ∈ x3 = 12345 }

Workflow of Query

In the above query x1, x2, x3 (ordered) refers to the attribute or column which we need in the result, and the predicate condition is that the first two domain variables x1 and x2 should be present while matching the condition for each row and the third domain variable x3 should be equal to 12345.

Result of the DRC query will be:

Customer_id	Name	Zip code
1	Rohit	12345
4	Amit	12345

Example 2

Write a DRC query to get the customer id of all the customer.

DRC Query: { <x1>\∃ "x2, x3(<x1, x2, x3> ∈ Customer) }

Result of the above Query will be:

Customer_id
1
2
3
4

2.1.3 SQL3

Q3. Give the brief introduction about SQL3.

Ans : **(Imp.)**

Meaning

SQL3 was accepted as the new standard for SQL in 1999, Basically, SQL3 includes data definition and management techniques from Object-Oriented dbms, OO-dbms, while maintaining the relational dbms platform. Based on this merger of concepts and techniques, DBMSs that support SQL3 are called Object-Relational or or-dbms'.

The most central data modelling notions included in SQL3 and support specification of:

- Classification hierarchies,
- Embedded structures that support composite attributes,
- Collection data-types (sets, lists/arrays, and multi-sets) that can be used for multi-valued attribute types,
- Large OBject types, LOBs, within the DB, as opposed to requiring external storage, and
- User defined data-types and functions (UDT/UDF) that can be used to define complex structures and derived attribute value calculations, among many other function extensions.

Query formulation in SQL3 remains based in the structured, relational model, though several functional additions have been made to support access to the new structures and data types

Accessing hierarchic structures

Hierarchic structures can be used at 2 levels for:

1. Distinguishing roles between entity-types and
2. Detailing attribute components.

A cascaded dot notation has been added to the SQL3 syntax to support specification of access paths within these structures. For example, the following statement selects the names and pictures of students from Bergen, Norway, using the OR DB specification given by the SQL3 declarations

```
SELECT   name, picture FROM Student
WHERE    address.city = 'Bergen'
ANDaddress.country = 'Norway';
```

The SQL3 query processor recognizes that *Student* is a sub-type of *Person* and that the attributes *name*, *picture* and *address* are inherited from *Person*, making it unnecessary for the user to:

- Specify the *Person* table in the FROM clause,
- Use the dot notation to specify the parent entity-type *Person* in the SELECT or WHERE clauses, or
- specify an explicit join between the levels in the entity-type hierarchy, here *Student* to *Perso*

Accessing multi-valued structures

SQL3 supports multi-valued (MV) attributes using a number of different implementation techniques. Basically, MV attribute structures can be defined as ordered or unordered sets and implemented as lists, arrays or tables either embedded in the parent table or 'normalized' to a linked table.

Person.address is a multi-valued complex attribute, defined as a set of addresses. In execution of the previous query the query processor must search each City and Country combination for the result. If the query intent is to locate students with a home address in Bergen, Norway and we assume that the address set ha

1st address is the home address, the query should be specified as:

```
SELECT   name, picture FROM Student
WHERE    address[1].city = 'Bergen'
AND      address[1].country = 'Norway';
```

Utilizing user defined data types (UDT)

User defined functions can be used in either the SELECT or WHERE clauses, as shown in the following example, again based on the DB specification .

```
SELECT   Avg (age) FROM Student
WHERE    Level > 4;
AND      age > 22;
```

In this query *age* is calculated by the function defined for Person.age. The SQL3 processor must calculate the relevant student.age for each graduate student (assuming that *Level* represents the number of years of higher education) and then calculate the average age of this group.

Accessing large objects

SQL3 has added data-types and storage support for unstructured binary and character large objects, BLOB and CLOB respectively, that can be used to store multimedia documents. However, ***no new*** query functionality has been added to access the content of these LOB data, though most SQL3 implementations have extended the LIKE operator so that it can also search through CLOB data. Thus, access to BLOB/CLOB data must be based on search conditions in the metadata of formatted columns or on use of the LIKE operator.

Some or-dbms implementations have extended other character string operators to operate on CLOB data, such as

- LOCATE, which returns the position of the first character or bit string within a LOB that matches the search string and
- Concatenation, substring, and length calculation.

Note that LIKE, concatenation, substring and length are original SQL operators that has been extended to function with LOBs, while LOCATE is a new SQL3 operator.

```
SELECT   Description FROM Course
WHERE    Description LIKE
         '%data management%'
OR Description LIKE '%information
management%' ;
```

An example SQL3 query

The SQL3 query refers to the University DB "Select the names and ages of students who are over 25 and have taken an advanced Data Management course within the last 3 years."

1. The clauses of the query are described below. Specifies the data to be retrieved by the query.

   In this case,

   - Student. Name and Age are inherited from the related Person record, extracted by the DBMS/query processor through a join on the primary key fields.

     Note: no join from Person to Student is specified in the query.
   - Age will be calculated by the Age_f function prior to presentation.
   - Course.Name and Course.Level form the criteria for the sorted presentation specified in line 8.
   - Course.Description will cause an output problem since each output row will have a CLOB attribute, which is large. DB2 presents the result row by row using the atomic attributes as a 'header' for the CLOB attribute.

2. Specifies the tables used in the query and gives each a short synonym for use in the query.

   Note that it is not necessary (or correct) to specify the Person table. The or-dbms 'knows' how to locate the attributes to be inherited by sub-entity types.

3. Specifies the join criteria for the tables. The result is a single table in which each row has *S.Id = T.Sid* and *T.Cid = C.Id*.

   Note that if the CLOB representing Course. Description is stored within the Course record, this query join will

   - First move a large amount of data from disk to memory prior to the join, and
   - The join result will be very large, containing a CLOB within each row of the table.

   An alternative, used by DB2, is to store only a link or pointer to the media object, called a *locator*, in the CLOB field of the Course table. This reduces the size of the table containing a CLOB and reduces the time required to manipulate it. The media object is only fetched for result presentation to the user or when it is assigned to a program variable.

4. Through 7 specify the selection criteria that must be matched in order for a row to become part of the output set. Course.Level > 1 is used to indicate an advanced course.

   The comparison value depends on knowledge of the course codes in the application domain.

5. Course Description has been defined as a CLOB data type for which the character-string comparison operator LIKE can be used. If a word or phrase is the search criteria, than it will not match the whole attribute value and must be enclosed in '%' to indicate that any preceding and following characters (text) are acceptable.

In this example, texts containing both *data* and *management* will satisfy the query no matter where these words appear. *Note:* if the phrase “data management” or “database management” were explicitly required, than line 5 of the query should be rewritten as:

AND (C.Description LIKE ‘%data management%’ OR C.Description LIKE ‘%database management%’).

6. Uses the Person: *Age* function to restrict the set of students. Note that, through inheritance, both attributes and functions defined in *Person* can be used in *Student.*

7. Specifies a date calculation to restrict the set of students. Note that the relationship has been implemented as a table and becomes searchable as such.

8. Finally, those rows from the join specified in line 3, which satisfy the criteria specified in lines 4-7, are sorted (ordered by) *course. level* and then *Course. name* before output to the user.

2.1.4 DDL AND DML Constructs

Q4. Explain briefly about commands DDL constructs with examples.

Ans : **(Imp.)**

DDL is an abbreviation of Data Definition Language.

The DDL Commands in Structured Query Language are used to create and modify the schema of the database and its objects. The syntax of DDL commands is predefined for describing the data. The commands of Data Definition Language deal with how the data should exist in the database.

Following are the five DDL commands in SQL:

1. CREATE Command
2. DROP Command
3. ALTER Command
4. TRUNCATE Command
5. RENAME Command

1. CREATE Command

CREATE is a DDL command used to create databases, tables, triggers and other database objects.

Examples of CREATE Command in SQL

Example 1

This example describes how to create a new database using the CREATE DDL command.

Syntax to Create a Database

CREATE Database Database_Name;

Suppose, you want to create a Books database in the SQL database. To do this, you have to write the following DDL Command:

Create Database Books;

Example 2

This example describes how to create a new table using the CREATE DDL command.

Syntax to create a new table:

```
CREATE TABLE table_name
(
column_Name1 data_type (size of the column ),
column_Name2 data_type (size of the column),
column_Name3 data_type (size of the column),
...
column_NameN data_type (size of the column)
);
```

Suppose, you want to create a **Student** table with five columns in the SQL database. To do this, you have to write the following DDL command:

```
CREATE TABLE Student
(
Roll_No. Int,
First_Name Varchar (20),
Last_Name Varchar (20),
Age Int,
Marks Int,
);
```

Example 3

This example describes how to create a new index using the CREATE DDL command.

Syntax to Create a new index:

CREATE INDEX Name_of_Index ON Name_of_Table (column_name_1 , column_name_2 , … . , column_name_N);

Let's take the Student table

Stu_Id	Name	Marks	City	State
100	Abhay	80	Noida	U.P
101	Sushil	75	Jaipur	Rajasthan
102	Ankit	90	Gurgaon	Haryana
103	Yogesh	93	Lucknow	U.P

Suppose, you want to create an index on the combination of the **City** and **State** field of the **Student** table. For this, we have to use the following DDL command:

CREATE INDEX index_city_State **ON** Employee (Emp_City, Emp_State);

Example 4

This example describes how to create a trigger in the SQL database using the DDL CREATE command.

Syntax to create a trigger:

```
CREATE TRIGGER [trigger_name]
    [ BEFORE | AFTER ]
    { INSERT | UPDATE | DELETE }
    ON [table_name];
```

2. DROP Command

DROP is a DDL command used to delete/remove the database objects from the SQL database. We can easily remove the entire table, view, or index from the database using this DDL command.

Examples of DROP Command in SQL

Example 1: This example describes how to remove a database from the SQL database.

Syntax to remove a database:

DROP DATABASE Database_Name;

Suppose, you want to delete the Books database from the SQL database. To do this, you have to write the following DDL command:

DROP DATABASE Books;

Example 2: This example describes how to remove the existing table from the SQL database.

Syntax to remove a table:

DROP TABLE Table_Name;

Suppose, you want to delete the Student table from the SQL database. To do this, you have to write the following DDL command:

DROP TABLE Student;

Example 3: This example describes how to remove the existing index from the SQL database.

Syntax to remove an index:

```
DROP INDEX Index_Name;
```

Suppose, you want to delete the index_city from the SQL database. To do this, you have to write the following DDL command:

```
DROP INDEX Index_city;
```

3. ALTER Command

ALTER is a DDL command which changes or modifies the existing structure of the database, and it also changes the schema of database objects.

We can also add and drop constraints of the table using the ALTER command.

Examples of ALTER Command in SQL

Example 1: This example shows how to add a new field to the existing table.

Syntax to add a newfield in the table:

```
ALTER TABLE name_of_table ADD column_name column_definition;
```

Suppose, you want to add the 'Father's_Name' column in the existing Student table. To do this, you have to write the following DDL command:

```
ALTER TABLE Student ADD Father's_Name Varchar(60);
```

Example 2: This example describes how to remove the existing column from the table.

Syntax to remove a column from the table:

```
ALTER TABLE name_of_table DROP Column_Name_1, column_Name_2, ….., column_Name_N;
```

Suppose, you want to remove the Age and Marks column from the existing Student table. To do this, you have to write the following DDL command:

```
ALTER TABLE Student DROP Age, Marks;
```

Example 3: This example describes how to modify the existing column of the existing table.

Syntax to modify the column of the table:

```
ALTER TABLE table_name MODIFY (column_name column_datatype(size));
```

Suppose, you want to change the character size of the Last_Namefield of the Student table. To do this, you have to write the following DDL command:

```
ALTER TABLE table_name MODIFY (Last_Name varchar(25));
```

4. TRUNCATE Command

TRUNCATE is another DDL command which deletes or removes all the records from the table.

This command also removes the space allocated for storing the table records.

Syntax of TRUNCATE command

```
TRUNCATE TABLE Table_Name;
```

Example

Suppose, you want to delete the record of the Student table. To do this, you have to write the following TRUNCATE DDL command:

TRUNCATE TABLE Student;

The above query successfully removed all the records from the student table. Let's verify it by using the following SELECT statement:

SELECT * **FROM** Student;

5. RENAME Command

RENAME is a DDL command which is used to change the name of the database table.

Syntax of RENAME command

RENAME **TABLE** Old_Table_Name **TO** New_Table_Name;

Example

RENAME **TABLE** Student **TO** Student_Details ;

This query changes the name of the table from Student to Student_details.

Q5. Discuss about DML constructs with an examples.

Ans : **(Imp.)**

The SQL data manipulation language (DML) is used to query and modify database data. In this chapter, we will describe how to use the SELECT, INSERT, UPDATE, and DELETE SQL DML command statements, defined below.

1. SELECT – to query data in the database
2. INSERT – to insert data into a table
3. UPDATE – to update data in a table
4. DELETE – to delete data from a table

In the SQL DML statement:

- Each clause in a statement should begin on a new line.
- The beginning of each clause should line up with the beginning of other clauses.
- If a clause has several parts, they should appear on separate lines and be indented under the start of the clause to show the relationship.
- Upper case letters are used to represent reserved words.
- Lower case letters are used to represent user-defined words.

1. SELECT Statement

The SELECT statement, or command, allows the user to extract data from tables, based on specific criteria. It is processed according to the following sequence:

SELECT DISTINCT item(s)

FROM table(s)

WHERE predicate

GROUP BY field(s)

ORDER BY fields

We can use the SELECT statement to generate an employee phone list from the Employees table as follows:

```
SELECT   FirstName, LastName, phone
FROM  Employees
ORDER BY LastName
```

This action will display employee's last name, first name, and phone number from the Employees table, seen in Table.

Last Name	First Name	Phone Number
Hagans	Ji m	604-232-3232
Wong	Bruce	604-244-2322

Table: Employees table.

In this next example, we will use a Publishers table (Table 16.2). (You will notice that Canada is mispelled in the *Publisher Country* field for Example Publishing and ABC Publishing. To correct mispelling, use the UPDATE statement to standardize the country field to Canada – see UPDATE statement later in this chapter.)

Publisher Name	Publisher City	Publisher Province	Publisher Country
Acme Publishing	Vancouver	BC	Canada
Example Publishing	Edmonton	AB	Cnada
ABC Publishing	Toronto	ON	Canda

Table: Publishers table.

If you add the publisher's name and city, you would use the SELECT statement followed by the fields name separated by a comma:

```
SELECT PubName, city
FROM Publishers
```

This action will display the publisher's name and city from the Publishers table.

If you just want the publisher's name under the display name city, you would use the SELECT statement with *no comma* separating pub_name and city:

```
SELECT  PubName city
FROM Publishers
```

Performing this action will display only the pub_name from the Publishers table with a "city" heading. If you do not include the comma, SQL Server assumes you want a new column name for pub_name.

SELECT statement with WHERE criteria

Sometimes you might want to focus on a portion of the Publishers table, such as only publishers that are in Vancouver. In this situation, you would use the SELECT statement with the WHERE criterion, i.e., WHERE city = 'Vancouver'.

These first two examples illustrate how to limit record selection with the WHERE criterion using BETWEEN. Each of these examples give the same results for store items with between 20 and 50 items in stock.

Example #1 uses the quantity, qty BETWEEN 20 and 50.

```
SELECT StorID, qty, TitleID
FROM Sales
WHERE qty BETWEEN 20 and 50   (includes the 20 and 50)
```

Example #2 on the other hand, uses *qty >=20 and qty <=50* .

```
SELECT StorID, qty, TitleID
FROM Sales
WHERE qty >= 20 and qty  <= 50
```

Example #3 illustrates how to limit record selection with the WHERE criterion using NOT BETWEEN.

```
SELECT StorID, qty, TitleID
FROM Sales
WHERE qty NOT BETWEEN 20 and 50
```

The next two examples show two different ways to limit record selection with the WHERE criterion using IN, with each yielding the same results.

Example #4 shows how to select records using *province=* as part of the WHERE statement.

```
SELECT *
FROM Publishers
WHERE province = 'BC' OR province = 'AB' OR province = 'ON'
```

Example #5 select records using *province IN* as part of the WHERE statement.

```
SELECT *
FROM Publishers
WHERE province IN ('BC', 'AB', 'ON')
```

The final two examples illustrate how NULL and NOT NULL can be used to select records. For these examples, a Books table (not shown) would be used that contains fields called Title, Quantity, and Price (of book). Each publisher has a Books table that lists all of its books.

Example #6 uses NULL.

```
SELECT price, title
FROM Books
WHERE price IS NULL
```

Example #7 uses NOT NULL.

```
SELECT price, title
FROM Books
WHERE price IS NOT NULL
```

Using wildcards in the LIKE clause

The LIKE keyword selects rows containing fields that match specified portions of character strings. LIKE is used with char, varchar, text, datetime and smalldatetime data. A *wildcard* allows the user to match fields that contain certain letters. For example, the wildcard province = 'N%' would give all provinces that start with the letter 'N'. Table 16.3 shows four ways to specify wildcards in the SELECT statement in regular express format.

%	Any string of zero or more characters
_	Any single character
[ ]	Any single character within the specified range (e.g., [a-f]) or set (e.g., [abcdef])
[^]	Any single character not within the specified range (e.g., [^a – f]) or set (e.g., [^abcdef])

Table: How to specify wildcards in the SELECT statement.

In example #1, LIKE 'Mc%' searches for all last names that begin with the letters "Mc" (e.g., McBadden).

```
SELECT LastName
FROM  Employees
WHERE LastName  LIKE 'Mc%'
```

For example #2: LIKE '%inger' searches for all last names that end with the letters "inger" (e.g., Ringer, Stringer).

```
SELECT  LastName
FROM  Employees
WHERE LastName  LIKE '%inger'
```

In, example #3: LIKE '%en%' searches for all last names that have the letters "en" (e.g., Bennett, Green, McBadden).

```
SELECT  LastName
FROM  Employees
WHERE LastName LIKE '%en%'
```

SELECT statement with ORDER BY clause

You use the ORDER BY clause to sort the records in the resulting list. Use *ASC* to sort the results in ascending order and *DESC* to sort the results in descending order.

For example, with ASC:

```
SELECT *
FROM  Employees
ORDER BY HireDate ASC
```

And with DESC:

```
SELECT *
FROM  Books
ORDER BY type, price DESC
```

SELECT statement with GROUP BY clause

The GROUP BY clause is used to create one output row per each group and produces summary values for the selected columns, as shown below.

```
SELECT type
FROM  Books
GROUP BY type
```

Here is an example using the above statement.

```
SELECT type AS 'Type', MIN(price) AS 'Minimum Price'
FROM  Books
WHERE royalty > 10
GROUP BY type
```

If the SELECT statement includes a WHERE criterion where *price is not null*,

```
SELECT type, price
FROM  Books
WHERE price is not null
```

then a statement with the GROUP BY clause would look like this:

```
SELECT type AS 'Type', MIN(price) AS 'Minimum Price'
FROM  Books
WHERE price is not null
GROUP BY type
```

Using COUNT with GROUP BY

We can use COUNT to tally how many items are in a container. However, if we want to count different items into separate groups, such as marbles of varying colours, then we would use the COUNT function with the GROUP BY command.

The below SELECT statement illustrates how to count groups of data using the COUNT function with the GROUP BY clause.

```
SELECT COUNT(*)
FROM  Books
GROUP BY type
```

Using AVG and SUM with GROUP BY

We can use the AVG function to give us the average of any group, and SUM to give the total.

Example #1 uses the AVG FUNCTION with the GROUP BY type.

```
SELECT AVG(qty)
FROM  Books
GROUP BY type
```

Example #2 uses the SUM function with the GROUP BY type.

```
SELECT SUM(qty)
FROM Books
GROUP BY type
```

Example #3 uses both the AVG and SUM functions with the GROUP BY type in the SELECT statement.

```
SELECT 'Total Sales' = SUM(qty), 'Average Sales' = AVG(qty), stor_id
FROM Sales
GROUP BY StorID ORDER BY  'Total Sales'
```

Restricting rows with HAVING

The HAVING clause can be used to restrict rows. It is similar to the WHERE condition except HAVING can include the aggregate function; the WHERE cannot do this.

The HAVING clause behaves like the WHERE clause, but is applicable to groups. In this example, we use the HAVING clause to exclude the groups with the province 'BC'.

```
SELECT au_fname AS 'Author''s First Name', province as 'Province'
FROM Authors
GROUP BY au_fname, province
HAVING province <> 'BC'
```

2. INSERT statement

The *INSERT statement* adds rows to a table. In addition,

- INSERT specifies the table or view that data will be inserted into.
- Column_list lists columns that will be affected by the INSERT.
- If a column is omitted, each value must be provided.
- If you are including columns, they can be listed in any order.
- VALUES specifies the data that you want to insert into the table. VALUES is required.
- Columns with the IDENTITY property should not be explicitly listed in the column_list or values_clause.

  The syntax for the INSERT statement is:

  ```
  INSERT [INTO] Table_name | view name [column_list]
  DEFAULT VALUES | values_list | select statement
  ```

When inserting rows with the INSERT statement, these rules apply:

- Inserting an empty string (' ') into a varchar or text column inserts a single space.
- All char columns are right-padded to the defined length.
- All trailing spaces are removed from data inserted into varchar columns, except in strings that contain only spaces. These strings are truncated to a single space.
- If an INSERT statement violates a constraint, default or rule, or if it is the wrong data type, the statement fails and SQL Server displays an error message.

When you specify values for only some of the columns in the column_list, one of three things can happen to the columns that have no values:

1. A default value is entered if the column has a DEFAULT constraint, if a default is bound to the column, or if a default is bound to the underlying user-defined data type.
2. NULL is entered if the column allows NULLs and no default value exists for the column.
3. An error message is displayed and the row is rejected if the column is defined as NOT NULL and no default exists.

This example uses INSERT to add a record to the publisher's Authors table.

INSERT INTO Authors

VALUES('555-093-467', 'Martin', 'April', '281 555-5673', '816 Market St.,' , 'Vancouver', 'BC', 'V7G3P4', 0)

This following example illustrates how to insert a partial row into the Publishers table with a column list. The country column had a default value of Canada so it does not require that you include it in your values.

```
INSERT INTO Publishers  (PubID, PubName, city, province)
VALUES ('9900', 'Acme Publishing', 'Vancouver', 'BC')
```

To insert rows into a table with an IDENTITY column, follow the below example. Do not supply the value for the IDENTITY nor the name of the column in the column list.

```
INSERT INTO jobs
VALUES ('DBA', 100, 175)
```

Inserting specific values into an IDENTITY column

By default, data cannot be inserted directly into an IDENTITY column; however, if a row is accidentally deleted, or there are gaps in the IDENTITY column values, you can insert a row and specify the IDENTITY column value.

IDENTITY_INSERT option

To allow an insert with a specific identity value, the IDENTITY_INSERT option can be used as follows.

```
SET IDENTITY_INSERT jobs ON
INSERT INTO jobs   (job_id, job_desc, min_lvl, max_lvl)
VALUES (19, 'DBA2', 100, 175)
SET IDENTITY_INSERT jobs OFF
```

Inserting rows with a SELECT statement

We can sometimes create a small temporary table from a large table. For this, we can insert rows with a SELECT statement. When using this command, there is no validation for uniqueness. Consequently, there may be many rows with the same pub_id in the example below.

This example creates a smaller temporary Publishers table using the CREATE TABLE statement. Then the INSERT with a SELECT statement is used to add records to this temporary Publishers table from the publis table.

```
        CREATE TABLE dbo.tmpPublishers (
        PubID char (4) NOT NULL ,
        PubName varchar (40) NULL ,
        city varchar (20) NULL ,
province char (2) NULL ,
country varchar (30) NULL   DEFAULT ('Canada')
)
        INSERT   tmpPublishers
        SELECT * FROM Publishers
```

In this example, we're copying a subset of data.

```
        INSERT tmpPublishers (pub_id, pub_name)
        SELECT PubID, PubName
        FROM Publishers
```

In this example, the publishers' data are copied to the tmpPublishers table and the country column is set to Canada.

```
        INSERT tmpPublishers (PubID, PubName, city, province, country)
        SELECT PubID, PubName, city, province, 'Canada'
        FROM Publishers
```

3. UPDATE statement

The *UPDATE statement* changes data in existing rows either by adding new data or modifying existing data.

This example uses the UPDATE statement to standardize the country field to be Canada for all records in the Publishers table.

UPDATE Publishers

```
SET country = 'Canada'
```

This example increases the royalty amount by 10% for those royalty amounts between 10 and 20.

```
UPDATE roysched
        SET royalty = royalty + (royalty * .10)
        WHERE royalty BETWEEN 10 and 20
```

Including subqueries in an UPDATE statement

The employees from the Employees table who were hired by the publisher in 2010 are given a promotion to the highest job level for their job type. This is what the UPDATE statement would look like.

```
        UPDATE Employees
        SET job_lvl =
        (SELECT max_lvl FROM jobs
        WHERE employee.job_id = jobs.job_id)
        WHERE DATEPART(year, employee.hire_date) = 2010
```

4. DELETE statement

The *DELETE statement* removes rows from a record set. DELETE names the table or view that holds the rows that will be deleted and only one table or row may be listed at a time. WHERE is a standard WHERE clause that limits the deletion to select records.

The DELETE syntax looks like this.

```
DELETE [FROM] {table_name | view_name }
[WHERE clause]
```

The rules for the DELETE statement are:

1. If you omit a WHERE clause, all rows in the table are removed (except for indexes, the table, constraints).
2. DELETE cannot be used with a view that has a FROM clause naming more than one table. (Delete can affect only one base table at a time.)

What follows are three different DELETE statements that can be used.

1. Deleting all rows from a table.

```
DELETE
FROM Discounts
```

2. Deleting selected rows:

```
DELETE
FROM Sales
WHERE stor_id = '6380'
```

3. Deleting rows based on a value in a subquery:

```
DELETE FROM Sales
WHERE title_id IN
(SELECT title_id FROM Books WHERE type = 'mod_cook')
```

2.2 Open Source and Commercial DBMS

2.2.1 MYSQL

Q6. What is MySQL & Explain its Features.

Ans : **(Imp.)**

Meaning

MySQL is an open-source relational database management system that works on many platforms. It provides multi-user access to support many storage engines and is backed by Oracle. So, you can buy a commercial license version from Oracle to get premium support services.

The features of MySQL are as follows:

- **Ease of Management:** The software very easily gets downloaded and also uses an event scheduler to schedule the tasks automatically.
- **Robust Transactional Support:** Holds the ACID (Atomicity, Consistency, Isolation, Durability) property, and also allows distributed multi-version support.
- **Comprehensive Application Development:** MySQL has plugin libraries to embed the database into any application. It also supports stored procedures, triggers, functions, views and many more for application development. You can refer to the RDS Tutorial, to understand Amazon's RDBMS.
- **High Performance:** Provides fast load utilities with distinct memory caches and table index partitioning.
- **Low Total Cost of Ownership:** This reduces licensing costs and hardware expenditures.
- **Open Source & 24*7 Support:** This RDBMS can be used on any platform and offers 24*7 support for open source and enterprise edition.
- **Secure Data Protection:** MySQL supports powerful mechanisms to ensure that only authorized users have access to the databases.
- **High Availability:** MySQL can run high-speed master/slave replication configurations and it offers cluster servers.
- **Scalability & Flexibility:** With MySQL you can run deeply embedded applications and create data warehouses holding a humongous amount of data.

Now, that you guys know what is MySQL, let me tell you various data types supported by MySQL.

Q7. Explain briefly about MySQL Data Types.

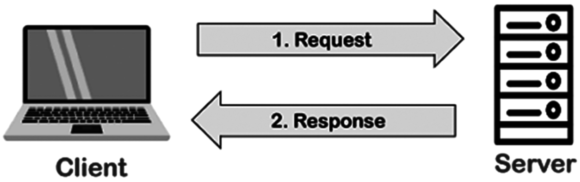
Ans :

- **Numeric:** This data type includes integers of various sizes, floating-point(real) of various precisions and formatted numbers.
- **Character-string:** These data types either have a fixed, or a varying number of characters. This data type also has a variable-length string called Character Large Object (CLOB) which is used to specify columns that have large text values.
- **Bit-string:** These data types are either of a fixed length or varying length of bits. There is also a variable-length bit string data type called *BINARY LARGE OBJECT (BLOB),* which is available to specify columns that have large binary values, such as images.
- **Boolean:** This data type has TRUE or FALSE values. Since SQL, has NULL values, a three-valued logic is used, which is UNKNOWN.
- **Date & Time:** The DATE data type has: YEAR, MONTH, and DAY in the form YYYY-MM-DD. Similarly, the TIME data type has the components HOUR, MINUTE, and SECOND in the form HH:MM: SS. These formats can change based on the requirement.
- **Timestamp & Interval:** The TIMESTAMP data type includes a minimum of six positions, for decimal fractions of seconds and an optional WITH TIME ZONE qualifier in addition to the DATE and TIME fields. The INTERVAL data type mentions a relative value that can be used to increment or decrement an absolute value of a date, time, or timestamp.

Q8. Explain working mechanism of mysqul?

Ans :

MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of MySQL environment is the same as the client-server model.



The core of the MySQL database is the MySQL Server. This server is available as a separate program and responsible for handling all the database instructions, statements, or commands. The working of MySQL database with MySQL Server are as follows:

1. MySQL creates a database that allows you to build many tables to store and manipulate data and defining the relationship between each table.
2. Clients make requests through the GUI screen or command prompt by using specific SQL expressions on MySQL.
3. Finally, the server application will respond with the requested expressions and produce the desired result on the client-side.

A client can use any MySQL GUI. But, it is making sure that your GUI should be lighter and user-friendly to make your data management activities faster and easier. Some of the most widely used MySQL GUIs are MySQL Workbench, SequelPro, DBVisualizer, and the Navicat DB Admin Tool. Some GUIs are commercial, while some are free with limited functionality, and some are only compatible with MacOS. Thus, you can choose the GUI according to your needs.

MySQL Variables

Variables are used for storing data or information during the execution of a program. It is a way of labeling data with an appropriate name that helps to understand the program more clearly by the reader. The main purpose of the variable is to store data in memory and can be used throughout the program.

User-Defined Variable

Sometimes, we want to pass values from one statement to another statement. The user-defined variable enables us to store a value in one statement and later can refer it to another statement. MySQL provides a **SET** and **SELECT** statement to declare and initialize a variable. The user-defined variable name starts with **@ symbol**.

The user-defined variables are not case-sensitive such as @name and @NAME; both are the same. A user-defined variable declares by one person cannot visible to another person. We can assign the user-defined variable into limited data types like integer, float, decimal, string, or NULL. The user-defined variable can be a maximum of **64 characters** in length.

Syntax

The following syntax is used to declare a user-defined variable.

1. By using the **SET** statement

   SET @var_name = value;

2. By using the **SELECT** statement

   SELECT @var_name := value;

Example1

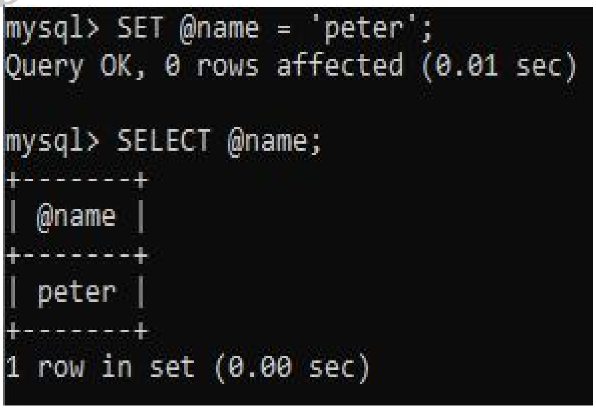
Here, we are going to assign a value to a variable by using the SET statement.

mysql> **SET** @**name**='peter';

Then, we can display the above value by using the SELECT statement.

mysql> **SELECT** @**name**;

Output

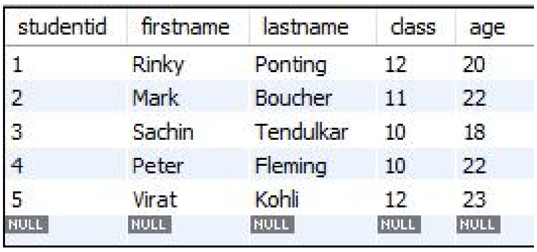


```
mysql> SET @name = 'peter';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT @name;
+-------+
| @name |
+-------+
| peter |
+-------+
1 row in set (0.00 sec)
```

Example 2

Let us create table **students** in the MySQL database, as shown below:

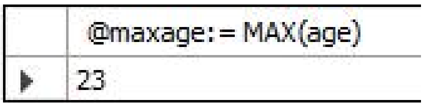


studentid	firstname	lastname	class	age
1	Rinky	Ponting	12	20
2	Mark	Boucher	11	22
3	Sachin	Tendulkar	10	18
4	Peter	Fleming	10	22
5	Virat	Kohli	12	23
NULL	NULL	NULL	NULL	NULL

Run the following statement to get the maximum age of the student in the 'students' table and assign the age to the user-defined variable **@maxage**.

mysql> **SELECT** @maxage:= **MAX** (age) **FROM** students;

It will give the following output.

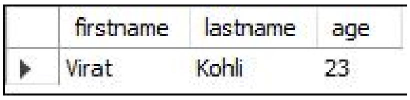


	@maxage:= MAX(age)
▶	23

Now, run the SELECT statement that uses the @maxage variable to return the maximum age of the student.

mysql> **SELECT** firstname, lastname, age **FROM** students **WHERE** age = @maxage;

After successful execution of the above statement, we will get the following result:



	firstname	lastname	age
▶	Virat	Kohli	23

2.2.2 Oracle

Q9. What is the Oracle database? Give the brief introduction about it.

Ans :

Meaning

Oracle database is a relational database management system. It is also called Oracle DB, or simply Oracle. It is produced and marketed by Oracle Corporation. It was created in 1977 by Lawrence Ellison and other engineers. It is one of the most popular relational database engines in the IT market for storing, organizing, and retrieving data.

Oracle database was the first DB that designed for **enterprise grid computing** and data warehousing. Enterprise grid computing provides the most flexible and cost-effective way to manage information and applications. It uses SQL queries as a language for interacting with the database.

Editions of Oracle database

Oracle database is compatible with a wide range of platforms such as Windows, UNIX, Linux, and macOS. It supports several operating systems like IBM AIX, HP-UX, Linux, Microsoft Windows Server, Solaris, SunOS, macOS, etc. In the late **1990s**, Oracle began supporting open platforms like GNU/Linux.

The following is a list of Oracle database editions in order of priority:

- **Enterprise Edition:** It is the most robust and secure edition. It offers all features, including superior performance and security.
- **Standard Edition:** It provides the base functionality for users that do not require Enterprise Edition's robust package.
- **Express Edition (XE):** It is the lightweight, free and limited Windows, and Linux edition.
- **Oracle Lite:** It is designed for mobile devices.
- **Personal Edition:** It's comparable to the Enterprise Edition but without the Oracle Real Application Clusters feature.

Features

Oracle database manages data with the help of an open, complete, and integrated approach. The following are features that complete the demand for powerful database management:

1. Availability

It is never offline or out of service that means supported 24*7 availability of the database. It provides high availability of databases because of the Oracle Data Guard functionality. This functionality allows using of the secondary database as a copy of the primary database during any failure. As a result, all normal processes such as backups and partial failures do not interrupt the database from being used.

2. Security

Oracle has a mechanism for controlling and accessing the database to prevent unauthorized access. It provides high security because of the Oracle Advanced Security features. It offers two solutions to protect databases that are TDE (Transparent Data Encryption) and Data Redaction. TDE supports data encryption both at the source and after export. Redaction is performed at the application level.

Oracle has some other security features like Oracle Database Vault that regulates user privileges and Oracle Label Security.

3. Scalability

It provides features like RAC (Real Application Cluster) and Portability, which makes an Oracle database scalable based on usage. In a clustered environment, it includes capabilities such as rolling instance migrations, performing upgrades, maintaining application continuity, quality of service management, etc.

4. Performance

Oracle provides performance optimization tools such as Oracle Advanced Compression, Oracle Database In-Memory, Oracle Real Application Testing, and Oracle Times Ten Application-Tier Database Cache. Their main objective is to improve system performance to the highest possible level.

5. Analytics

Oracle has the following solutions in the field of analytics:

- **OLAP (Oracle Analytic Processing):** It is an implementation of Oracle for doing complicated analytical calculations on business data.
- **Oracle Advanced Analytics:** It is a technical combination of Oracle R Enterprise and Oracle Data Mining that assists customers in determining predictive business models through data and text mining, as well as statistical data computation.

6. Management

Oracle Multitenant is a database management tool that combines a single container database with many pluggable databases in a consolidated design.

Q10. Explain the Benefits of Oracle.

Ans :

The following are the main advantages of an Oracle database:

1. Performance

Oracle has procedures and principles that help us to get high levels of database performance. We can increase query execution time and operations with the use of performance optimization techniques in its database. This technique helps to retrieve and alter data faster.

2. Portability

The Oracle database can be ported on all different platforms than any of its competitors. We can use this database on around 20 networking protocols as well as over 100 hardware platforms. This database makes it simple to write an Oracle application by making changes to the OS and hardware in a secure manner.

3. Backup and Recovery

It is always better to take a proper backup of your entire oracle online backup and recovery. The Oracle database makes it easy to accomplish recovery quickly by using the. RMAN (Recovery Manager) functionality. It can recover or restore database files during downtime or outages. It can be used for online backups, archived backups, and continuous archiving. We can also use SQL* PLUS for recovery, which is known as user-managed recovery.

4. PL/SQL

One of the greatest benefits of using the Oracle database is to support PL/SQL extension for procedural programming.

5. Multiple Database

Oracle database allows several database instances management on a single server. It provides an instance caging approach for managing CPU allocations on a server hosting database instances. The database resource management and instance caging can work together to manage services across multiple instances.

6. Flashback Technology

This advantage comes with the recent Oracle version. It allows us to recover those data that are incorrectly deleted or lost by human errors like accidental deletion of valuable data, deleting the wrong data, or dropping the table.

Q11. Explain the Disadvantages of Oracle.

Ans :

The following are the disadvantages of the Oracle database

1. Complexity

Oracle is not recommended to use when the users are not technically savvy and have limited technical skills required to deal with the Oracle Database. It is also not advised to use if the company is looking for a database with limited functionality and easy to use.

2. Cost

The price of Oracle products is very high in comparison to other databases. Therefore users are more likely to choose other less expensive options such as MS SQL Server, MySQL, etc.

3. Difficult to manage

Oracle databases are often much more complex and difficult in terms of the management of certain activities.

2.2.3 DB2

Q12. What is DB2? Explain about it.

Ans : **(Imp.)**

Meaning

DB2 is a database server developed by IBM. It is a Relational Database Management System which is designed to store, analyze and retrieve the data efficiently.

DB2 database supports Object Oriented features and non relational structure with XML.

History

The name DB2, or IBM Database 2, was first given to the Database Management System or DBMS in 1983 when IBM released DB2 on its MVS mainframe platform.

Initially DB2 was developed for specific platform of IBM.

In 1990, it was developed as a Universal Database (UDB) DB2 Server, which can run on any authoritative operating systems such as Linux, UNIX, and Windows.

Features of DB2

All of the DB2 tools have the following features

- **AI-powered functionality:** Users can employ artificial intelligence (AI) to simplify the querying process.
  - Machine learning algorithms improve performance and efficiency
  - Column store directs queries to specific columns, ultimately reducing overhead and employee workload
  - Data skipping automatically overlooks data that shouldn't be included in a query
- **Common SQL engine:** A query may be written once and used across products and platforms.
- **Can support all data types:** Structured, unstructured, and relational data can all be accessed on one platform.
- **High availability and disaster recovery:** DB2 replication functionality allows for safe storage and access.
- **Scalability:** Users can extend local storage and power levels onto cloud environments, and also scale storage and power in a managed cloud to save money.
- **Table partitioning:** In a DB2 warehouse, the database partitioning feature allows users to split data across servers to maximize computing power and allow parallel processing.

Products

These products are part of the DB2 catalog, a range that can be used on premises or in the cloud

- **DB2 Database:** A powerful local (on-premises) RDBMS that is optimized for use with online transaction processing (OLTP). It is enterprise-ready and provides high performance and resilience.
- **DB2 Warehouse:** An on-premises data warehouse that can handle machine learning, data analytics, and parallel processing.

- **DB2 on Cloud:** A cloud-based SQL database similar to Db2 Database.
- **DB2 Warehouse on Cloud:** A fully managed cloud-based and on-premises data warehouse similar to Db2 Warehouse.
- **DB2 Big SQL:** A SQL-on-Hadoop engine that provides parallel processing and querying functionality. It can be integrated with Cloudera Data Platform.
- **DB2 Event Store:** A memory-optimized database that can analyze streamed data for event-driven applications. It includes IBM Watson Studio, so users can integrate machine learning models.
- **DB2 for z/OS:** An enterprise data server for IBM Z that provides a mission-critical data solution and integration for mobile and cloud to support thousands of users.

Examples

Organizations that use an IBM server tend to use DB2. The industries that typically use DB2 include banking and financial services, insurance, manufacturing, and automotive industries Around 62.5 percent of these companies are large, with over 10,000 employees.

Many of our everyday transactions use relational databases to store and retrieve important data for banking, manufacturing, and retail, such as paying with a credit card, accessing our bank accounts, buying products or services online, and more. Relational databases spurred IBM to create the Db2 product line as well as the language used to query relational databases SQL.

Careers that use DB2

1. Database administrators install, develop, test, and maintain databases for companies. They ensure optimal performance by performing backups, data migrations, and load balancing.
2. Data engineers design and build systems for collecting and analyzing data. They typically use SQL to query relational databases like DB2 to manage the data, as well as provide troubleshooting, recovery, and security management support.
3. Data architects analyze the data infrastructure of an organization to execute database management systems that improve efficiency in workflows for specific departments.
4. Systems programmers help to install, configure, maintain, and monitor DB2 for an organization's mainframe operating system. They might be hired on a contract or as-needed basis.

2.2.4 SQL SERVER

Q13. Explain briefly about (SQL SERVER).

Ans : **(Imp.)**

Data is a collection of facts and figures and we have humungous data available to the users via the internet and other sources. To manipulate the data, Structured Query Language (SQL) in short has been introduced years ago. There are different versions of SQL available in the market provided by different organizations. In this article, we shall see the version of SQL provided by Microsoft.

1. Microsoft SQL Server or MS SQL Server for short is the query language provided for data definition and manipulation.
2. SQL Server is a Relational Database Management Systems which was developed and marketed by the Microsoft company.
3. SQL and SQL servers are built as two layers where the SQL server is on the top for interacting with the relational databases.
4. MS SQL Server also has T-SQL or Transact-SQL and the main focus of T-SQL is to handle the transactions.
5. As it is a Microsoft's developed system, it worked only on Microsoft's environment until it was made available on Linux platforms in the year 2016.

SQL Server is composed of Database engine, and Relational engine, and Storage engine. These are explained as following below.

1. Database Engine

Database is a collection of various data items on which the user can perform any kind of manipulations.

i) The database engine has a relational engine on which a user can perform queries and it also comes with a storage engine which manages the data files, indexes and procedures.

ii) The database engine also creates and executes objects like triggers, views, procedures etc.

2. Relational Engine

Relations are the connections between the two different databases or within the same database. It is stored in the form of a row and column intersection named tables.

i) It manages query processing, memory management, buffer management, threads, and much more.

ii) It has another layer named storage engine.

3. Storage Engine

i) It looks upon the storage of data.

ii) It is done using systems like disks and Storage Area Network or SAN.

Database Management Tools

SQL Server comes with a number of tools to help you with your database administration and programming tasks.

Some typical database administration and programming tasks could include:

- Create & maintain databases
- Create & maintain tables
- Create & maintain other database objects such as stored procedures, views, etc
- Create & maintain and schedule data backups
- Replication (eg, create a copy of the database)
- Create & maintain users, roles, etc
- Optimization tasks

These are some of many tasks that a database administrator (DBA) might need to perform. SQL Server provides the means for performing these tasks.

Client/Server Database Systems

SQL Server is a client/server DBMS, as opposed to a desktop system such as Access.

Client/server systems are designed to run on a central server - or servers - so that multiple users can access the same data simultaneously from across a network. Users normally access the database through an application.

For example, a web-based corporate CRM could be used by employees in various cities, or even countries, all reading and updating data via their browser.

Short Question and Answers

1. Relational algebra.

Ans :

Relational algebra in DBMS is a procedural query language. Queries in relational algebra are performed using operators. Relational Algebra is the fundamental block for modern language SQL and modern Database Management Systems such as Oracle Database, Mircosoft SQL Server, IBM Db2, etc.

2. What is Relational Calculus?

Ans :

Before understanding Relational calculus in DBMS, we need to understand Procedural Language and Declarative Language.

i) **Procedural Language:** Those Languages which clearly define how to get the required results from the Database are called Procedural Language. Relational algebra is a Procedural Language.

ii) **Declarative Language:** Those Language that only cares about What to get from the database without getting into how to get the results are called Declarative Language. **Relational Calculus** is a Declarative Language.

3. Tuple Relational Calculus (TRC)

Ans :

Tuple Relational Calculus in DBMS uses a tuple variable (t) that goes to each row of the table and checks if the predicate is true or false for the given row. Depending on the given predicate condition, it returns the row or part of the row.

4. SQL3.

Ans :

SQL3 was accepted as the new standard for SQL in 1999, Basically, SQL3 includes data definition and management techniques from Object-Oriented dbms, OO-dbms, while maintaining the relational dbms platform. Based on this merger of concepts and techniques, DBMSs that support SQL3 are called Object-Relational or or-dbms'.

The most central data modelling notions included in SQL3 and support specification of:

- Classification hierarchies,
- Embedded structures that support composite attributes,
- Collection data-types (sets, lists/arrays, and multi-sets) that can be used for multi-valued attribute types,
- Large OBject types, LOBs, within the DB, as opposed to requiring external storage, and
- User defined data-types and functions (UDT/UDF) that can be used to define complex structures and derived attribute value calculations, among many other function extensions.

5. DDL

Ans :

DDL is an abbreviation of Data Definition Language.

The DDL Commands in Structured Query Language are used to create and modify the schema of the database and its objects. The syntax of DDL commands is predefined for describing the data. The commands of Data Definition Language deal with how the data should exist in the database.

6. MySQL

Ans :

MySQL is an open-source relational database management system that works on many platforms. It provides multi-user access to support many storage engines and is backed by Oracle. So, you can buy a commercial license version from Oracle to get premium support services.

7. Features of MySQL

Ans :

- **Ease of Management:** The software very easily gets downloaded and also uses an event scheduler to schedule the tasks automatically.
- **Robust Transactional Support:** Holds the ACID (Atomicity, Consistency, Isolation, Durability) property, and also allows distributed multiversion support.
- **Comprehensive Application Development:** MySQL has plugin libraries to embed the database into any application. It also supports stored procedures, triggers, func-tions, views and many more for application development. You can refer to the RDS Tutorial, to understand Amazon's RDBMS.
- **High Performance:** Provides fast load utilities with distinct memory caches and table index partitioning.
- **Low Total Cost of Ownership:** This reduces licensing costs and hardware expenditures.
- **Open Source & 24*7 Support:** This RDBMS can be used on any platform and offers 24*7 support for open source and enterprise edition.
- **Secure Data Protection:** MySQL supports powerful mechanisms to ensure that only authorized users have access to the databases.
- **High Availability:** MySQL can run high-speed master/slave replication configurations and it offers cluster servers.

8. User-Defined Variable

Ans :

Sometimes, we want to pass values from one statement to another statement. The user-defined variable enables us to store a value in one statement and later can refer it to another statement. MySQL provides a **SET** and **SELECT** statement to declare and initialize a variable. The user-defined variable name starts with @ **symbol**.

The user-defined variables are not case-sensitive such as @name and @NAME; both are the same. A user-defined variable declares by one person cannot visible to another person. We can assign the user-defined variable into limited data types like integer, float, decimal, string, or NULL. The user-defined variable can be a maximum of **64 characters** in length.

9. Oracle

Ans :

Oracle database is a relational database management system. It is also called Oracle DB, or simply Oracle. It is produced and marketed by Oracle Corporation. It was created in 1977 by Lawrence Ellison and other engineers. It is one of the most popular relational database engines in the IT market for storing, organizing, and retrieving data.

Oracle database was the first DB that designed for **enterprise grid computing** and data warehousing. Enterprise grid computing provides the most flexible and cost-effective way to manage information and applications. It uses SQL queries as a language for interacting with the database.

10. What is DB2

Ans :

Meaning

DB2 is a database server developed by IBM. It is a Relational Database Management System which is designed to store, analyze and retrieve the data efficiently.

DB2 database supports Object Oriented features and non relational structure with XML.

History

The name DB2, or IBM Database 2, was first given to the Database Management System or DBMS in 1983 when IBM released DB2 on its MVS mainframe platform.

Initially DB2 was developed for specific platform of IBM.

11. SQL SERVER.

Ans :

Data is a collection of facts and figures and we have humungous data available to the users via the internet and other sources. To manipulate the data, Structured Query Language (SQL) in short has been introduced years ago. There are different versions of SQL available in the market provided by different organizations. In this article, we shall see the version of SQL provided by Microsoft.

1. Microsoft SQL Server or MS SQL Server for short is the query language provided for data definition and manipulation.
2. SQL Server is a Relational Database Management Systems which was developed and marketed by the Microsoft company.
3. SQL and SQL servers are built as two layers where the SQL server is on the top for interacting with the relational databases.
4. MS SQL Server also has T-SQL or Transact-SQL and the main focus of T-SQL is to handle the transactions.
5. As it is a Microsoft's developed system, it worked only on Microsoft's environment until it was made available on Linux platforms in the year 2016.

Choose the Correct Answers

1. Relational Algebra is a __________ query language that takes two relations as input and produces another relation as an output of the query. [c]

   (a) Relational (b) Structural

   (c) Procedural (d) Fundamental

2. Which of the following is a fundamental operation in relational algebra? [d]

   (a) Set intersection (b) Natural join

   (c) Assignment (d) None of the mentioned

3. Which of the following is used to denote the selection operation in relational algebra? [b]

   a) Pi (Greek) b) Sigma (Greek)

   c) Lambda (Greek) d) Omega (Greek)

4. For select operation the ________ appear in the subscript and the ___________ argument appears in the paranthesis after the sigma. [a]

   a) Predicates, relation b) Relation, Predicates

   c) Operation, Predicates d) Relation, Operation

5. The ___________ operation, denoted by, allows us to find tuples that are in one relation but are not in another. [b]

   a) Union b) Set-difference

   c) Difference d) Intersection

6. Which is a unary operation: [d]

   a) Selection operation b) Primitive operation

   c) Projection operation d) Generalized selection

7. Which is a join condition contains an equality operator: [a]

   a) Equijoins b) Cartesian

   c) Natural d) Left

8. In precedence of set operators, the expression is evaluated from [b]

   a) Left to left b) Left to right

   c) Right to left d) From user specification

9. Which of the following is not outer join? [d]

   a) Left outer join b) Right outer join

   c) Full outer join d) All of the mentioned

10. The assignment operator is denoted by [b]

    a) -> b) <-

    c) = d) ==

Fill in the blanks

1. ________ algebra in DBMS is a procedural query language.
2. Cartesian product is denoted by the ________ symbol.
3. ________ supports multi-valued (MV) attributes using a number of different implementation techniques.
4. ________ is a DDL command used to delete/remove the database objects from the SQL database.
5. ________ is another DDL command which deletes or removes all the records from the table.
6. ACID stands for ________.
7. ________ are used for storing data or information during the execution of a program.
8. ________ database supports Object Oriented features and non relational structure with XML.
9. ________ administrators install, develop, test, and maintain databases for companies.
10. ________ is a collection of various data items on which the user can perform any kind of manipulations.

ANSWERS

1. Relational
2. “X”
3. SQL3
4. DROP
5. TRUNCATE
6. Atomicity, Consistency, Isolation Durability
7. Variables
8. DB2
9. Database
10. Database

UNIT III

Query Processing and Optimization and Storage Strategies :

Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms, Indices, B-trees, hashing.

3.1 Query Processing And Optimization And Storage Strategies

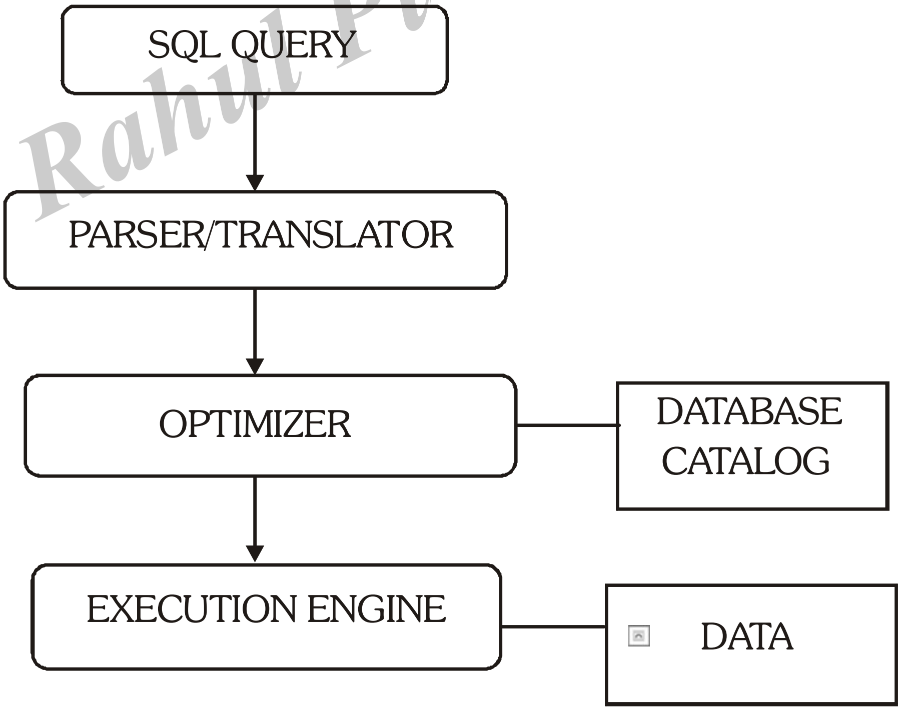
Q1. What is query processing? Explain about the steps in query processing.

Ans : **(Imp.)**

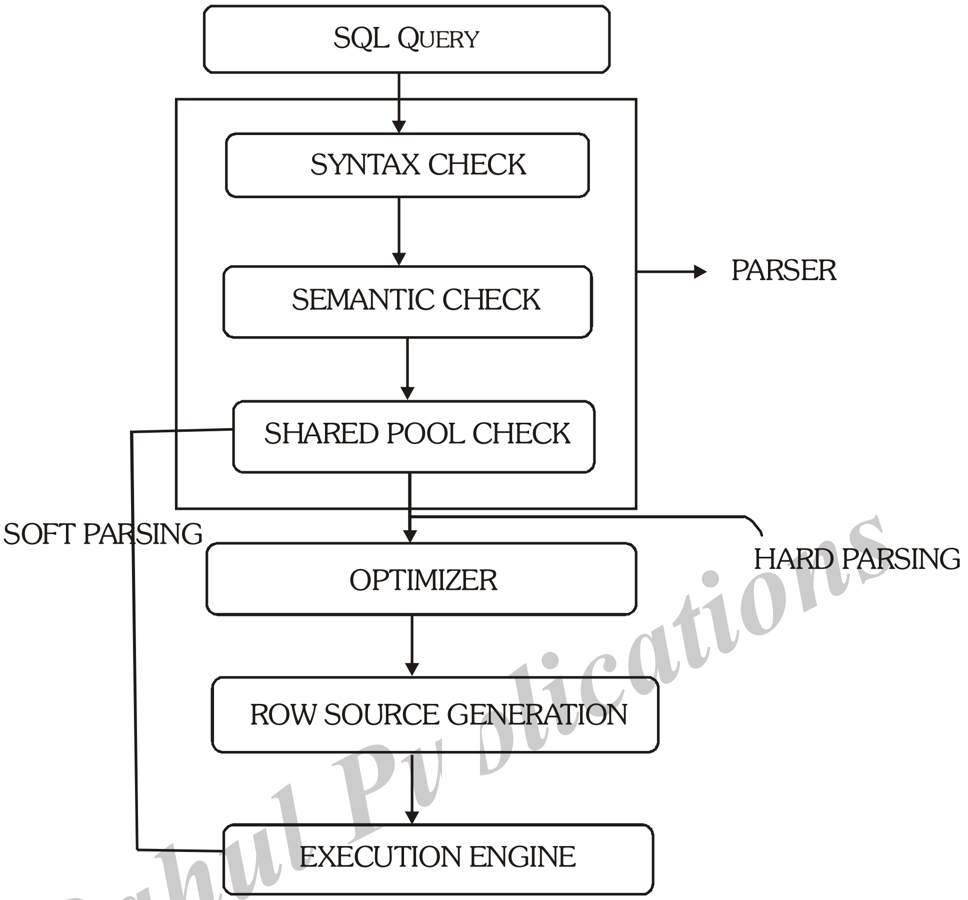
Meaning

Query Processing includes translations on high level Queries into low level expressions that can be used at physical level of file system, query optimization and actual execution of query to get the actual result.

Steps



Detailed Diagram is drawn as:



It is done in the following steps:

- **Step-I**

  Parser: During parse call, the database performs the following checks- Syntax check, Semantic check and Shared pool check, after converting the query into relational algebra.

Parser performs the following checks as (refer detailed diagram):

1. **Syntax check –** concludes SQL syntactic validity. Example:

   SELECT * FORM employee

   Here error of wrong spelling of FROM is given by this check.

2. **Semantic check –** determines whether the statement is meaningful or not. Example: query contains a tablename which does not exist is checked by this check.

3. **Shared Pool check –** Every query possess a hash code during its execution. So, this check determines existence of written hash code in shared pool if code exists in shared pool then database will not take additional steps for optimization and execution.

Hard Parse and Soft Parse

If there is a fresh query and its hash code does not exist in shared pool then that query has to pass through from the additional steps known as hard parsing otherwise if hash code exists then query does not passes through additional steps. It just passes directly to execution engine (refer detailed diagram). This is known as soft parsing.

Hard Parse includes following steps – Optimizer and Row source generation.

- **Step-II**

  Optimizer: During optimization stage, database must perform a hard parse atleast for one unique DML statement and perform optimization during this parse. This database never optimizes DDL unless it includes a DML component such as subquery that require optimization.

  It is a process in which multiple query execution plan for satisfying a query are examined and most efficient query plan is satisfied for execution.

  Database catalog stores the execution plans and then optimizer passes the lowest cost plan for execution.

  Row Source Generation

  The Row Source Generation is a software that receives a optimal execution plan from the optimizer and produces an iterative execution plan that is usable by the rest of the database. the iterative plan is the binary program that when executes by the sql engine produces the result set.

- **Step-III**

  Execution Engine: Finally runs the query and display the required result.

3.2 Evalution Of Relational Algebra Expressions

Q2. Explain about Query Optimization in Relational Algebra

Ans : **(Imp.)**

(i) Query

A query is a request for information from a database.

(ii) Query Plans

A query plan (or query execution plan) is an ordered set of steps used to access data in a SQL relational database management system.

(iii) Query Optimization

A single query can be executed through different algorithms or re-written in different forms and structures. Hence, the question of query optimization comes into the picture – Which of these forms or pathways is the most optimal? The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans.

Importance

The goal of query optimization is to reduce the system resources required to fulfill a query, and ultimately provide the user with the correct result set faster.

- First, it provides the user with faster results, which makes the application seem faster to the user.
- Secondly, it allows the system to service more queries in the same amount of time, because each request takes less time than unoptimized queries.
- Thirdly, query optimization ultimately reduces the amount of wear on the hardware (e.g. disk drives), and allows the server to run more efficiently (e.g. lower power consumption, less memory usage).

There are broadly two ways a query can be optimized:

1. **Analyze and transform equivalent relational expressions:** Try to minimize the tuple and column counts of the intermediate and final query processes (discussed here).
2. **Using different algorithms for each operation:** These underlying algorithms determine how tuples are accessed from the data structures they are stored in, indexing, hashing, data retrieval and hence influence the number of disk and block accesses (discussed in query processing).

Analyze and transform equivalent relational expressions.

Here, we shall talk about generating minimal equivalent expressions. To analyze equivalent expression, listed are a set of equivalence rules. These generate equivalent expressions for a query written in relational algebra. To optimize a query, we must convert the query into its equivalent form as long as an equivalence rule is satisfied.

1. Conjunctive selection operations can be written as a sequence of individual selections. This is called a sigma-cascade.

$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta 1}(\sigma_{\theta_2}(E))$$

Explanation:

Applying condition intersection is expensive. Instead, filter out tuples satisfying condition (inner selection) and then apply condition (outer selection) to the then resulting fewer tuples. This leaves us with less tuples to process the second time. This can be extended for two or more intersecting selections. Since we are breaking a single condition into a series of selections or cascades, it is called a “cascade”.

2. Selection is commutative.

$$\sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$$

Explanation

Condition is commutative in nature. This means, it does not matter whether we apply first or first. In practice, it is better and more optimal to apply that selection first which yields a fewer number of tuples. This saves time on our outer selection.

3. All following projections can be omitted, only the first projection is required. This is called a pi-cascade.

$$\pi L_1(\pi L_2(\ldots(\pi L_n(E))\ldots)) = \pi L_1(E)$$

Explanation

A cascade or a series of projections is meaningless. This is because in the end, we are only selecting those columns which are specified in the last, or the outermost projection. Hence, it is better to collapse all the projections into just one i.e. the outermost projection.

4. Selections on Cartesian Products can be re-written as Theta Joins.

- **Equivalence 1**

$$\sigma_\theta(E_1 \times E_2) = E_1 \infty_\theta\ E_2$$

Explanation

The cross product operation is known to be very expensive. This is because it matches each tuple of E1 (total m tuples) with each tuple of E2 (total n tuples). This yields m*n entries. If we apply a selection operation after that, we would have to scan through m*n entries to find the suitable tuples which satisfy the condition. Instead of doing all of this, it is more optimal to use the Theta Join, a join specifically designed to select only those entries in the cross product which satisfy the Theta condition, without evaluating the entire cross product first.

- **Equivalence 2**

$$\sigma_{\theta_1}(E_1 \infty_{\theta_2} E_2) = E_1 \infty_{\theta_1 \wedge \theta_2} E_2$$

Explanation

Theta Join radically decreases the number of resulting tuples, so if we apply an intersection of both the join conditions i.e. and into the Theta Join itself, we get fewer scans to do. On the other hand, a condition outside unnecessarily increases the tuples to scan.

5. Theta Joins are commutative.

$$E_1 \infty_\theta E_2 = E_2 \infty_\theta E_1$$

Explanation

Theta Joins are commutative, and the query processing time depends to some extent which table is used as the outer loop and which one is used as the inner loop during the join process (based on the indexing structures and blocks).

6. Join operations are associative.

➢ **Natural Join**

$$(E_1 \infty\, E_2) \infty\, E_3 = E_1 \infty\, (E_2 \infty\, E_3)$$

Explanation

Joins are all commutative as well as associative, so one must join those two tables first which yield less number of entries, and then apply the other join.

➢ **Theta Join**

$$(E_1 \infty_{\theta_1} E_2) \infty_{\theta_2 \Lambda \theta_3} E_3 = E_1 \infty_{\theta_1 \Lambda \theta_3} (E_2 \infty_{\theta_2} E_3)$$

Explanation

Theta Joins are associative in the above manner, where involves attributes from only E2 and E3.

7. Selection operation can be distributed.

➢ **Equivalence 1**

$$\sigma_{\theta_1 \Lambda \theta_2}(E_1 \infty_\theta E_2) = (\sigma_{\theta 1}(E_1)) \infty_\theta (\sigma_{\theta_2}(E_2))$$

Explanation

Applying a selection after doing the Theta Join causes all the tuples returned by the Theta Join to be monitored after the join. If this selection contains attributes from only E1, it is better to apply this selection to E1 (hence resulting in a fewer number of tuples) and then join it with E2.

➢ **Equivalence 2**

$$\sigma\theta_{\cap}(E_1 \infty_\theta E_2) = (\sigma\theta_{\cap}(E_1))\ \infty_\theta\ E_2$$

Explanation

This can be extended to two selection conditions, and, where Theta1 contains the attributes of only E1 and contains attributes of only E2. Hence, we can individually apply the selection criteria before joining, to drastically reduce the number of tuples joined.

8. Projection distributes over the Theta Join.

➢ **Equivalence 1**

$$\pi_{L_1 \cup L_2}(E_1 \infty_\theta E_2) = (\pi_{L_1}(E_1)) \infty_\theta (\pi_{L_2}(E_2))$$

Explanation

The idea discussed for selection can be used for projection as well. Here, if L1 is a projection that involves columns of only E1, and L2 another projection that involves the columns of only E2, then it is better to individually apply the projections on both the tables before joining. This leaves us with a fewer number of columns on either side, hence contributing to an easier join.

➢ **Equivalence 2**

$$\pi_{L_1 \cup L_2}(E_1 \infty_\theta E_2) = \pi_{L_1 \cup L_2}((\pi_{L_1 \cup L_3}(E_1)) \infty_\theta (\pi_{L_2 \cup L})$$

Explanation

Here, when applying projections L1 and L2 on the join, where L1 contains columns of only E1 and L2 contains columns of only E2, we can introduce another column E3 (which is common between both the tables). Then, we can apply projections L1 and L2 on E1 and E2 respectively, along with the added column L3. L3 enables us to do the join.

9. Union and Intersection are commutative.

$$E_1 \cup E_2 = E_2 \cup E_1$$

$$E_1 \cap E_2 = E_2 \cap E_1$$

Explanation

Union and intersection are both distributive; we can enclose any tables in parentheses according to requirement and ease of access.

10. Union and Intersection are associative.

$$(E_1 \cup E_2) \cup E_3 = E_1 \cup (E_2 \cup E_3)$$

$$(E_1 \cap E_2) \cap E_3 = E_1 \cap (E_2 \cap E_3)$$

Explanation

Union and intersection are both distributive; we can enclose any tables in parentheses according to requirement and ease of access.

11. Selection operation distributes over the union, intersection, and difference operations.

$$\sigma_P(E_1 - E_2) = \sigma_P(E_1) - \sigma_P(E_2)$$

Explanation

In set difference, we know that only those tuples are shown which belong to table E1 and do not belong to table E2. So, applying a selection condition on the entire set difference is equivalent to applying the selection condition on the individual tables and then applying set difference. This will reduce the number of comparisons in the set difference step.

12. Projection operation distributes over the union operation.

$$\pi_L(E_1 \cup E_2) = (\pi_L(E_1)) \cup (\pi_L(E_2))$$

Explanation

Applying individual projections before computing the union of E1 and E2 is more optimal than the left expression, i.e. applying projection after the union step.

Minimality

A set of equivalence rules is said to be minimal if no rule can be derived from any combination of the others. A query is said to be optimal when it is minimal.

Examples

Assume the following tables:

instructor(ID, name, dept_name, salary)

teaches(ID, course_id, sec_id, semester, year)

course(course_id, title, dept_name, credits)

$$\pi_{name,title}(\sigma_{dept.name="Music"}(\text{instructor} \infty$$

$$(\text{teaches} \infty \pi_{course.id,title}(\text{com}$$

Here, dept_name is a field of only the instructor table. Hence, we can select out the Music instructors before joining the tables, hence reducing query time.

Optimized Query

Using rule 7a, and Performing the selection as early as possible reduces the size of the relation to be joined.

$$\pi_{name,title}((\sigma_{dept\_name="Music"(instructor)}\infty(\text{teaches} \infty$$

Query 2

Find the names of all instructors in the CSE department who have taught a course in 2009, along with the titles of the courses that they taught

$$\sigma_{dept\_name="CSE"}(\sigma_{year=2009}(\text{instructor}$$

Optimized Query

We can perform an “early selection”, hence the optimized query becomes:

$$\sigma_{dept\_name="CSE"}(\text{instructor}) \infty \sigma_{year=2009}(\text{teaches})$$

Explain the evaluation of relational algebra expression (DBMS)

SQL queries are decomposed into query blocks. One query block contains a single SELECT-FROM-WHERE expression, as well as GROUP BY and HAVING clause (if any). Nested queries are split into separate query blocks.

Example

Consider an example given below-

Select lastname, firstname from employee where salary>(select max(salary) from employee where deptname =CSE ;

C=(select max(salary) from employee where deptname=CSE); // inner block

Select lastname, firstname from employee where salary>c; //outer block

Where C represents the result returned from the inner block.

- The relation algebra for the inner block is $G_{max(salary)}\ (\sigma_{dname=CSE}(\text{employee}))$
- The relation algebra for the outer blocks is $\Pi_{lastname,\ firstname}(\sigma_{salary>c}(\text{employee}))$

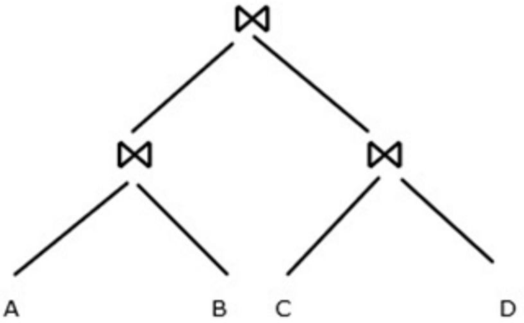
  The query optimizer would then choose an execution or evaluation plan for each block.

Q3. Write about evaluation of relational algebra expressions.

Ans : **(Imp.)**

Evaluation of relational algebra expressions

Materialized evaluation d Evaluate one operation at a time. Evaluate the expression in a bottom-up manner and stores intermediate results to temporary files.



Store the result of AÈ” B in a temporary file.

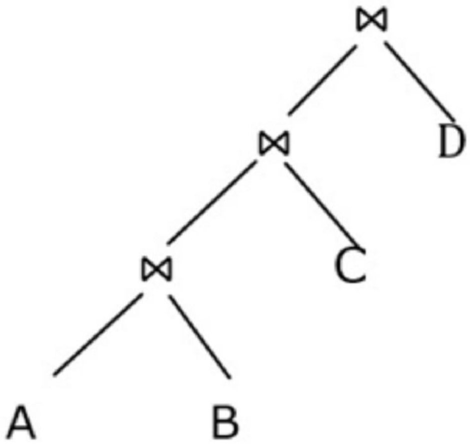
Store the result of C È” D in a temporary file.

Finally, join the results stored in temporary files.

The overall cost=sum of costs of individual operations + cost of writing intermediate results to disk, cost of writing results to results to temporary files and reading them back is quite high.

Pipelined evaluation

Evaluate several operations simultaneously. Result of one operation is passed to the next operation. Evaluate the expression in a bottom-up manner and don't store intermediate results to temporary files.



Don't store the result of AÈ” B in a temporary file. Instead the result is passed directly for projection with C and so on.

3.3 Query Equivalance

Q4. Write about equivalence rules in query optimization.

Ans : **(Imp.)**

Meaning

The equivalence rule says that expressions of two forms are the same or equivalent because both expressions produce the same outputs on any legal database instance. It means that we can possibly replace the expression of the first form with that of the second form and replace the expression of the second form with an expression of the first form. Thus, the optimizer of the query-evaluation plan uses such an equivalence rule or method for transforming expressions into the logically equivalent one.

The optimizer uses various equivalence rules on relational-algebra expressions for transforming the relational expressions. For describing each rule, we will use the following symbols:

$\theta, \theta_1, \theta_2 ...$: Used for denoting the predicates.

L_1, L_2, L_3 ... : Used for denoting the list of attributes.

E, E_1, E_2 .... : Represents the relational-algebra expressions.

Let's discuss a number of equivalence rules:

Rule 1: Cascade of σ

This rule states the deconstruction of the conjunctive selection operations into a sequence of individual selections. Such a transformation is known as a cascade of σ.

$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

Rule 2: Commutative Rule

(a) This rule states that selections operations are commutative.

$$\sigma_{\theta 1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta 1}(E))$$

(b) Theta Join (θ) is commutative.

$E_1 \infty_\theta E_2 = E_2 \infty_\theta E_1$ (θ is in subscript with the join symbol)

However, in the case of theta join, the equivalence rule does not work if the order of attributes is considered. Natural join is a special case of Theta join, and natural join is also commutative.

However, in the case of theta join, the equivalence rule does not work if the order of attributes is considered. Natural join is a special case of Theta join, and natural join is also commutative.

Rule 3: Cascade of Π

This rule states that we only need the final operations in the sequence of the projection operations, and other operations are omitted. Such a transformation is referred to as a cascade of Π.

$$\Pi\, L_1\ (\Pi\, L_2\,(...(\Pi\, L_n\,(E))...)) = \Pi\, L_1\ (E)$$

Rule 4

We can combine the selections with Cartesian products as well as theta joins

1. $\sigma_\theta (E_1 x E_2) = E_{1\theta} \infty E_2$

2. $\sigma_{\theta 1}(E_1 \infty_{\theta 2}\ E_2) = E_1 \infty_{\theta 1 \wedge \theta_2} E_2$

Rule 5: Associative Rule

(a) This rule states that natural join operations are associative.

$$(E_1 \infty E_2) \infty E_3 = E_1 \infty (E_2 \infty E_3)$$

(b) Theta joins are associative for the following expression:

$$(E_1 \infty_{\theta 1} E_2) \infty_{\theta 2 \wedge \theta_3} E_3 = E_1 \infty_{\theta 1 \wedge \theta_3} (E_2 \infty_{\theta 2} E_3)$$

In the theta associativity, θ_2 involves the attributes from E_2 and E_3 only.There may be chances of empty conditions, and thereby it concludes that Cartesian Product is also associative.

Rule 6

Distribution of the Selection operation over the Theta join.

Under two following conditions, the selection operation gets distributed over the theta-join operation:

(a) When all attributes in the selection condition θ_0 include only attributes of one of the expressions which are being joined.

$$\sigma_{\theta 0}(E_1 \infty_\theta\ E_2) = (\sigma_{\theta 0}(E_1)) \infty_\theta E_2$$

(b) When the selection condition θ_1 involves the attributes of E_1 only, and θ_2 includes the attributes of E_2 only.

$$\sigma_{\theta_1 \theta_2}(E_1 \infty_\theta E_2) = (\sigma_{\theta 1}(E_1))\infty_\theta((\sigma_{\theta_2}(E_2))$$

Rule 7

Distribution of the projection operation over the theta join.

Under two following conditions, the selection operation gets distributed over the theta-join operation:

(a) Assume that the join condition θ includes only in $L_1 \upsilon\ L_2$ attributes of E_1 and E_2 Then, we get the following expression:

$$\Pi_{L_1 \upsilon L_2}(E_1 \infty_\theta E_2) = (\Pi_{L_1}(E_1))\infty_\theta(\Pi_{L_2}(E_2))$$

(b) Assume a join as $E_1 \infty\ E_2$. Both expressions E_1 and E_2 have sets of attributes as L_1 and L_2. Assume two attributes L_3 and L_4 where L_3 be attributes of the expression E_1, involved in the θ join condition but not in $L_1 \upsilon\ L_2$ Similarly, an L_4 be attributes of the expression E_2 involved only in the θ join condition and not in $L_1 \upsilon\ L_2$ attributes. Thus, we get the following expression:

$$\Pi_{L_1 \upsilon L_2}(E_1 \infty_\theta E_2) = \Pi_{L_1 \upsilon L_2}((\Pi_{L_1 \upsilon L_3}(E_1))\infty_\theta((\Pi_{L_2 \upsilon L_4}(E_2)))$$

Rule 8

The union and intersection set operations are commutative.

$E_1 \upsilon E_2 = E_2 \upsilon E_1$

$E_1 \square E_2 = E_2 = E_1$

However, set difference operations are not commutative.

Rule 9

The union and intersection set operations are associative.

$(E_1 \upsilon E_2) \upsilon E_3 = E_1 \upsilon (E_2 \upsilon E_3)$

$(E_1 \upsilon\ E_2) \upsilon E_3 = E_1 \upsilon (E_2 \upsilon E_3)$

Rule 10

Distribution of selection operation on the intersection, union, and set difference operations.

The below expression shows the distribution performed over the set difference operation.

$\sigma_P (E_1 - E_2) = \sigma_P(E_1) - \sigma_P(E_2)$

We can similarly distribute the selection operation on õ and õ£ by replacing with -. Further, we get:

$\sigma_P (E_1 - E_2) = \sigma_P(E_1) - E_2$

Rule 11:

Distribution of the projection operation over the union operation.

This rule states that we can distribute the projection operation on the union operation for the given expressions.

$\Pi_L(E_1 \upsilon\ E_2) = (\Pi_L (E_1)) \upsilon (\Pi_L (E_2))$

Apart from these discussed equivalence rules, there are various other equivalence rules also.

3.4 Join Strategies

Q5. Explain about Join Strategies in Relational Algebra.

Ans : **(Imp.)**

Extended operators are those operators which can be derived from basic operators. There are mainly three types of extended operators in Relational Algebra:

- Join
- Intersection
- Divide

The relations used to understand extended operators are STUDENT, STUDENT_SPORTS, ALL_SPORTS and EMPLOYEE which are shown in Table 1, Table 2, Table 3 and Table 4 respectively. STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI	9156768971	18

Table 1

STUDENT_SPORTS

ROLL_NO	SPORTS
1	Badminton
2	Cricket
2	Badminton
4	Badminton

Table 2

ALL_SPORTS

SPORTS
Badminton
Cricket

Table 3

EMPLOYEE

EMP_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
5	NARESH	HISAR	9782918192	22
6	SWETA	RANCHI	9852617621	21
4	SURESH	DELHI	9156768971	18

Table 4

Intersection ($\cap$)

Intersection on two relations R1 and R2 can only be computed if R1 and R2 are union compatible (These two relation should have same number of attributes and corresponding attributes in two relations have same domain). Intersection operator when applied on two relations as R1 $\cap$ R2 will give a relation with tuples which are in R1 as well as R2. Syntax:

Relation1 $\cap$ Relation2

Example: Find a person who is student as well as employee- STUDENT $\cap$ EMPLOYEE

In terms of basic operators (union and minus) :

STUDENT $\cap$ EMPLOYEE = STUDENT + EMPLOYEE - (STUDENT U EMPLOYEE)

Result:

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18

Conditional Join (∞_c) :

Conditional Join is used when you want to join two or more relation based on some conditions. Example: Select students whose ROLL_NO is greater than EMP_NO of employees

STUDENTÈ $\infty_{c\ \text{STUDENT.ROLL\_NO>EMPLOYEE.EMP\_NOEMPLOYEE}}$

In terms of basic operators (cross product and selection) :

$\sigma_{\text{(STUDENT.ROLL\_NO>EMPLOYEE.EMP\_NO)(STUDENT}\times\text{EMPLOYEE)}}$

Result:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS	PHONE	AGE
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI	9455123451	18
3	SUJIT	ROHTAK	9156253131	20	1	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18	1	RAM	DELHI	9455123451	18

Equijoin(∞)

Equijoin is a special case of conditional join where only equality condition holds between a pair of attributes. As values of two attributes will be equal in result of equijoin, only one attribute will be appeared in result. Example: Select students whose ROLL_NO is equal to EMP_NO of employees.

STUDENT $\infty_{\text{STUDENT.ROLL\_NO=EMPLOYEE.EMP\_NOEMPLOYEE}}$

In terms Π of basic operators (cross product, selection and projection) :

$\Pi_{\text{(STUDENT.ROLL\_NO, STUDENT.NAME, STUDENT.ADDRESS, STUDENT.PHONE, STUDENT.AGE EMPLOYEE.NAME, EMPLOYEE.ADDRESS, EMPLOYEE.PHONE, EMPLOYEE>AGE)}}$(σ (STUDENT.ROLL_NO=EMPLOYEE.EMP_NO) (STUDENT × EMPLOYEE))

Result:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18	SURESH	DELHI	9156768971	18

Natural Join(∞)

It is a special case of equijoin in which equality condition hold on all attributes which have same name in relations R and S (relations on which join operation is applied). While applying natural join on two relations, there is no need to write equality condition explicitly. Natural Join will also return the similar attributes only once as their value will be same in resulting relation. Example: Select students whose ROLL_NO is equal to ROLL_NO of STUDENT_SPORTS as:

STUDENT ∞ STUDENT_SPORTS

In terms of basic operators (cross product, selection and projection) :

$\Pi_{\text{(STUDENT.ROLL\_NO, STUDENT.NAME, STUDENT.ADDRESS, STUDENT.PHONE, STUDENT.AGE STUDENT\_SPORTS.SPORTS)(ó (STUDENT.ROLL\_NO=STUDENT\_SPORTS.ROLL\_NO) (STUDENT}\times\text{STUDENT\_SPORTS))}}$

Result:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	SPORTS
1	RAM	DELHI	9455123451	18	Badminton
2	RAMESH	GURGAON	9652431543	18	Cricket
2	RAMESH	GURGAON	9652431543	18	Badminton
4	SURESH	DELHI	9156768971	18	Badminton

Natural Join is by default inner join because the tuples which does not satisfy the conditions of join does not appear in result set. e.g.; The tuple having ROLL_NO 3 in STUDENT does not match with any tuple in STUDENT_SPORTS, so it has not been a part of result set.

Left Outer Join(∞):

When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Left Outer Joins gives all tuples of R in the result set. The tuples of R which do not satisfy join condition will have values as NULL for attributes of S. Example:Select students whose ROLL_NO is greater than EMP_NO of employees and details of other students as well

STUDENT $\infty_{\text{STUDENT.ROLL\_NO>EMPLOYEE.EMP\_NOEMPLOYEE}}$

Result

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS	PHONE	AGE
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI	9455123451	18
3	SUJIT	ROHTAK	9156253131	20	1	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18	1	RAM	DELHI	9455123451	18
1	RAM	DELHI	9455123451	18	NULL	NULL	NULL	NULL	NULL

Right Outer Join(∞)

When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Right Outer Joins gives all tuples of S in the result set. The tuples of S which do not satisfy join condition will have values as NULL for attributes of R. Example: Select students whose ROLL_NO is greater than EMP_NO of employees and details of other Employees as well

$\text{STUDENT} \infty_{\text{STUDENT.ROLL\_NO>EMPLOYEE.EMP\_NOEMPLOYEE}}$

Result:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS	PHONE	AGE
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI	9455123451	18
3	SUJIT	ROHTAK	9156253131	20	1	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18	1	RAM	DELHI	9455123451	18
NULL	NULL	NULL	NULL	NULL	5	NARESH	HISAR	9782918192	22
NULL	NULL	NULL	NULL	NULL	6	SWETA	RANCHI	9852617621	21
NULL	NULL	NULL	NULL	NULL	4	SURESH	DELHI	9156768971	18

Full Outer Join(∞)

When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Full Outer Joins gives all tuples of S and all tuples of R in the result set. The tuples of S which do not satisfy join condition will have values as NULL for attributes of R and vice versa. Example:Select students whose ROLL_NO is greater than EMP_NO of employees and details of other Employees as well and other Students as well

$\text{STUDENT} \infty_{\text{STUDENT.ROLL\_NO>EMPLOYEE.EMP\_NOEMPLOYEE}}$

Result:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS	PHONE	AGE
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI	9455123451	18
3	SUJIT	ROHTAK	9156253131	20	1	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18	1	RAM	DELHI	9455123451	18
NULL	NULL	NULL	NULL	NULL	5	NARESH	HISAR	9782918192	22
NULL	NULL	NULL	NULL	NULL	6	SWETA	RANCHI	9852617621	21
NULL	NULL	NULL	NULL	NULL	4	SURESH	DELHI	9156768971	18
1	RAM	DELHI	9455123451	18	NULL	NULL	NULL	NULL	NULL

Division Operator (÷): Division operator A÷B or A/B can be applied if and only if:

- Attributes of B is proper subset of Attributes of A.
- The relation returned by division operator will have attributes = (All attributes of A – All Attributes of B)
- The relation returned by division operator will return those tuples from relation A which are associated to every B's tuple.

x	y
a	1
b	2
a	2
d	4

÷Π

B

y
1
2

The resultant of A/B is

$A \div B$

x

a

Division can be expressed in terms of Cross Product, Set Difference and Projection.

In the above example , for A/B , compute all x values that are not disqualified by some y in B.

x value is disqualified if attaching y value from B, we obtain xy tuple that is not in A.

Disqualified x values: $\Pi_x((\Pi_x(A) \times B) - A)$

So A/B = $\Pi_x(A)$ - all disqualified tuples

A/B = $\Pi_x(A)$- $\Pi x((\Pi x(A) \times B) - A)$

In the above example, disqualified tuples are

b	2
d	4

So, the resultant is

x

a

3.5 Query Optimization Algorithms

Q6. Explain various methods of query optimization Algorithms in DBMS.

Ans : **(Imp.)**

Methods of Query Optimization in DBMS

There are two methods of query optimization. They are as follows.

Cost-Based Query Optimization in DBMS

Query optimization is the process of selecting the most efficient way to execute a SQL statement. Because SQL is a nonprocedural language, the optimizer can merge, restructure, and process data in any sequence.

The Optimizer allocates a cost in numerical form for each step of a feasible plan for a given query and environment, and then discovers these values together to get a cost estimate for the plan or possible strategy. The Optimizer aims to find the plan with the lowest cost estimate after evaluating the costs of all feasible plans. As a result, the Optimizer is sometimes known as the Cost-Based Optimizer.

- **Execution Plans**

  An execution plan specifies the best way to execute a SQL statement.

  The plan describes the steps taken by Oracle Database to execute a SQL statement. Each step physically retrieves or prepares rows of data from the database for the statement's user.

An execution plan shows the total cost of the plan, which is stated on line 0, as well as the cost of each individual operation. A cost is an internal unit that appears solely in the execution plan to allow for plan comparisons. As a result, the cost value cannot be fine-tuned or adjusted.

➢ Query Blocks

The optimizer receives a parsed representation of a SQL statement as input. Each SELECT block in the original SQL statement is internally represented by a query block. A query block can be a statement at the top level, a subquery, or an unmerged view. Let's take an example where the SQL statement that follows is made up of two query sections. The inner query block is the subquery in parentheses. The remainder of the outer query block of the SQL statement obtains the names of employees in the departments whose IDs were supplied by the subquery. The query form specifies how query blocks are connected.

```
SELECT first_name, last_name
FROM   hr.employees
WHERE  department_id
IN  (SELECT department_id
FROM   hr.departments
WHERE  location_id = 1800);
```

➢ Query Sub Plans

The optimizer creates a query sub-plan for each query block.

From the bottom up, the database optimizes query blocks separately. As a result, the database optimizes the innermost query block first, generating a sub-plan for it, before generating the outer query block, which represents the full query.

The number of query block plans is proportional to the number of items in the FROM clause. As the number of objects rises, this number climbs exponentially. The possibilities for a join of five tables, for example, are far higher than those for a connection of two tables.

➢ Analogy for the Optimizer

An online trip counselor is one analogy for the optimizer.

A biker wishes to find the most efficient bicycle path from point A to point B. A query is analogous to the phrase "I need the quickest route from point A to point B" or "I need the quickest route from point A to point B via point C". To choose the most efficient route, the trip advisor employs an internal algorithm that takes into account factors such as speed and difficulty. The biker can sway the trip advisor's judgment by saying things like "I want to arrive as quickly as possible" or "I want the simplest route possible."

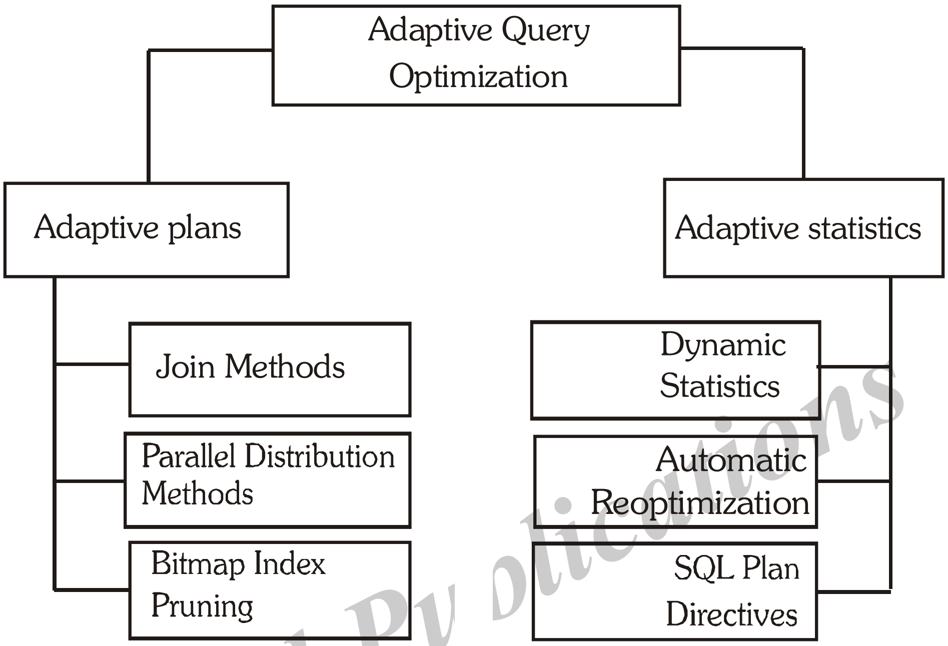
In this example, an execution plan is a possible path generated by the travel advisor. Internally, the advisor may divide the overall route into multiple subroutes (sub plans) and compute the efficiency of each subroute separately. For example, the trip advisor may estimate one subroute to take 15 minutes and be of medium difficulty, another subroute to take 22 minutes and be of low difficulty, and so on.

Based on the user-specified goals and accessible facts about roads and traffic conditions, the advisor selects the most efficient (lowest cost) overall route. The better the guidance, the more accurate the statistics. For example, if the advisor is not kept up to date on traffic delays, road closures, and poor road conditions, the proposed route may prove inefficient (high cost).

Adaptive Query Optimization in DBMS

Adaptive query optimization allows the optimizer to make run-time changes to execution plans and uncover new information that can lead to improved statistics.

When existing facts are insufficient to produce an ideal strategy, adaptive optimization comes in handy. The image below depicts the feature set for adaptive query optimization.



- **Adaptive Query Plans**

  The optimizer can defer the final plan decision for a statement with an adaptive plan until execution time.

- **Purpose of Adaptive Query Plans**

  The optimizer's ability to alter a plan based on information gained during execution can significantly increase query performance

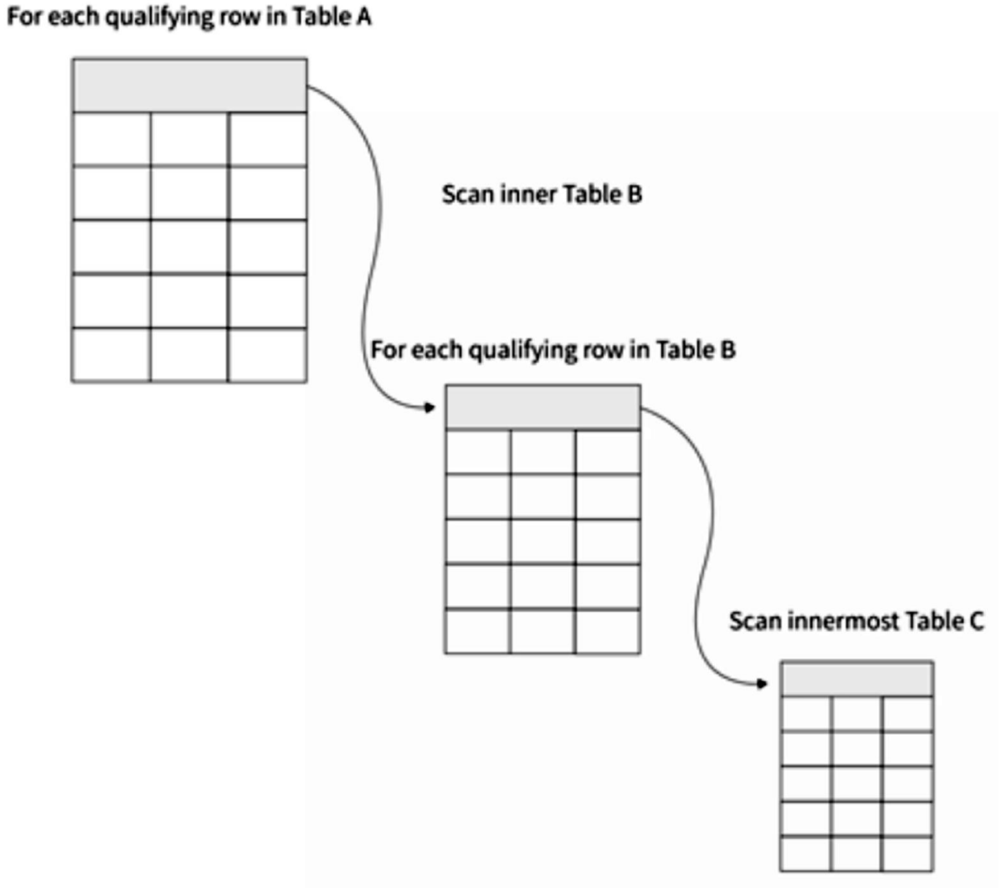
  Because the optimizer occasionally chooses an inferior default plan due to a cardinality misestimate, adaptive plans are important. The capacity to modify the plan during execution based on actual execution statistics leads to a more optimal end plan. The optimizer uses the final plan for further executions after selecting it, ensuring that the poor plan is not reused.

- **How Adaptive Query Plans Work**

  An adaptive plan is made up of several predefined sub plans and an optimizer statistics collector.

  A sub-plan is a section of a plan that the optimizer can use as an alternative during execution. A nested loops join, for example, might be converted to a hash join during execution. An optimizer statistics collector is a row source that is added at crucial points in a plan to collect run-time statistics. These statistics assist the optimizer in making a final choice amongst numerous sub plans.

During statement execution, the statistics collector collects execution information and buffers some rows received by the sub-plan. The optimizer selects a sub-plan based on the information collected by the collector. At this point, the collector stops collecting statistics and buffering rows, and permits rows to pass through instead. On subsequent executions of the child cursor, the optimizer continues to use the same plan unless the plan ages out of the cache, or a different optimizer feature (for example, adaptive cursor sharing or statistics feedback) invalidates the plan.


For each qualifying row in Table A
Scan inner Table B
For each qualifying row in Table B
Scan innermost Table C

- **Adaptive Query Plans: Parallel Distribution Methods**

  Parallel execution typically necessitates data redistribution in order to conduct operations like as parallel sorts, aggregations, and joins.

  Oracle Database supports a wide range of data dissemination mechanisms. The approach is selected by the database based on the number of rows to be distributed and the number of concurrent server processes involved in the operation. Consider the following potential scenarios:

- Few rows are distributed by many concurrent server processes. The database has the option of using the broadcast distribution method. Each row in the result set is received by each simultaneous server process in this situation.

- Few parallel server processes disseminate a large number of rows. If a data skew is found during the data redistribution, the statement's performance may suffer. To ensure that each parallel server process receives an equal number of rows, the database is more likely to use a hash distribution.

3.6 Indices

Q7. What is Indexing in DBMS? Explain.

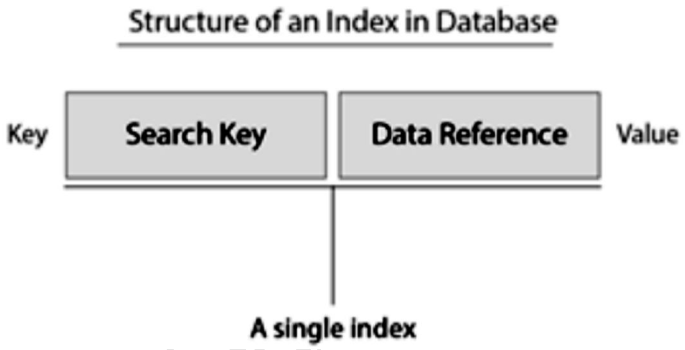
Ans : **(Imp.)**

Meaning

Indexing is used to quickly retrieve particular data from the database. Formally we can define Indexing as a technique that uses data structures to optimize the searching time of a database query. Indexing reduces the number of disks required to access a particular data by internally creating an index table.

Indexing is achieved by creating Index-table or Index.

Index usually consists of two columns which are a key-value pair. The two columns of the index table(i.e., the key-value pair) contain copies of selected columns of the tabular data of the database.



Here, Search Key contains the copy of the Primary Key or the Candidate Key of the database table. Generally, we store the selected Primary or Candidate keys in a sorted manner so that we can reduce the overall query time or search time(from linear to binary).

Data Reference contains a set of pointers that holds the address of the disk block. The pointed disk block contains the actual data referred to by the Search Key. Data Reference is also called Block Pointer because it uses block-based addressing.

Indexing Attributes

Let's discuss the various indexing attributes:

Standard (B-tree) and Bitmap

B-tree-indexing is one of the most popular and commonly used indexing techniques. B-tree is a type of tree data structure that contains 2 things namely: Index Key and its corresponding disk address. Index Key refers to a certain disk address and that disk further contains rows or tuples of data.

On the other hand, Bitmap indexing uses strings to store the address of the tuples or rows. A bitmap is a mapping from one system to the other such as integers to bits.

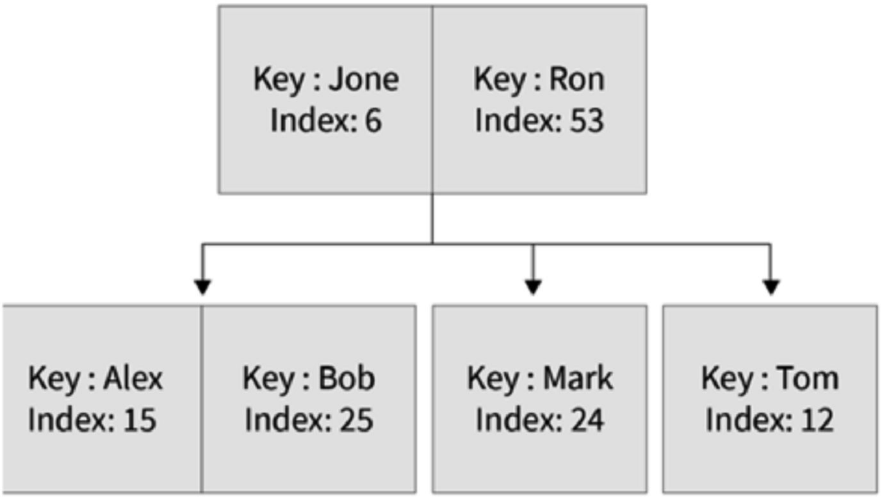
Bitmap has an advantage over B-tress as bitmap performs faster retrieval of certain data (Bitmap is made according to a certain data, hence retrieves faster). Bitmaps are also more compact than B-trees.

There is a drawback with bit mapping, bit mapping requires more overhead during tuple operations on the table. Hence, bit maps are mainly used in data warehouse environments.

Example - We want to store this three-column table in the database.

Name	Marks	Age
Jone	5	28
Alex	32	45
Tom	37	23
Ron	87	13
Mark	20	48
Bob	89	32

The B-tree representation will be like this:



Note: Oracle Database uses Bitmap and B-trees.

Ascending and Descending

As we have discussed above, columns of the index are stored in some sorted manner. Generally, we store these Search Keys in ascending order. These sorted keys allow us to search data the data fastly. We can change the sort order from ascending to descending or something different according to the most frequent queries on the database.

Syntax

Lets see the syntax to store indexing in descending order-

CREATE INDEX index_name ON table-name (column-name_1, column-name_2 DESC);

By default Sorting Order:

- **Character Data:** Sorted by ASCII values of the characters.
- **Numeric Data:** Smallest to largest numbers.
- **Date:** Earliest date to the latest date.

Column and Functional:

Generally, we prepare the index table with certain column values of the actual database but sometimes we can also use predefined SQL functions like UPPER() or LOWER() or MAX(), etc. to prepare the Search Keys.

Example - We can convert all values in a column to uppercase and stored these results in the index.

Syntax:

CREATE INDEX index-name

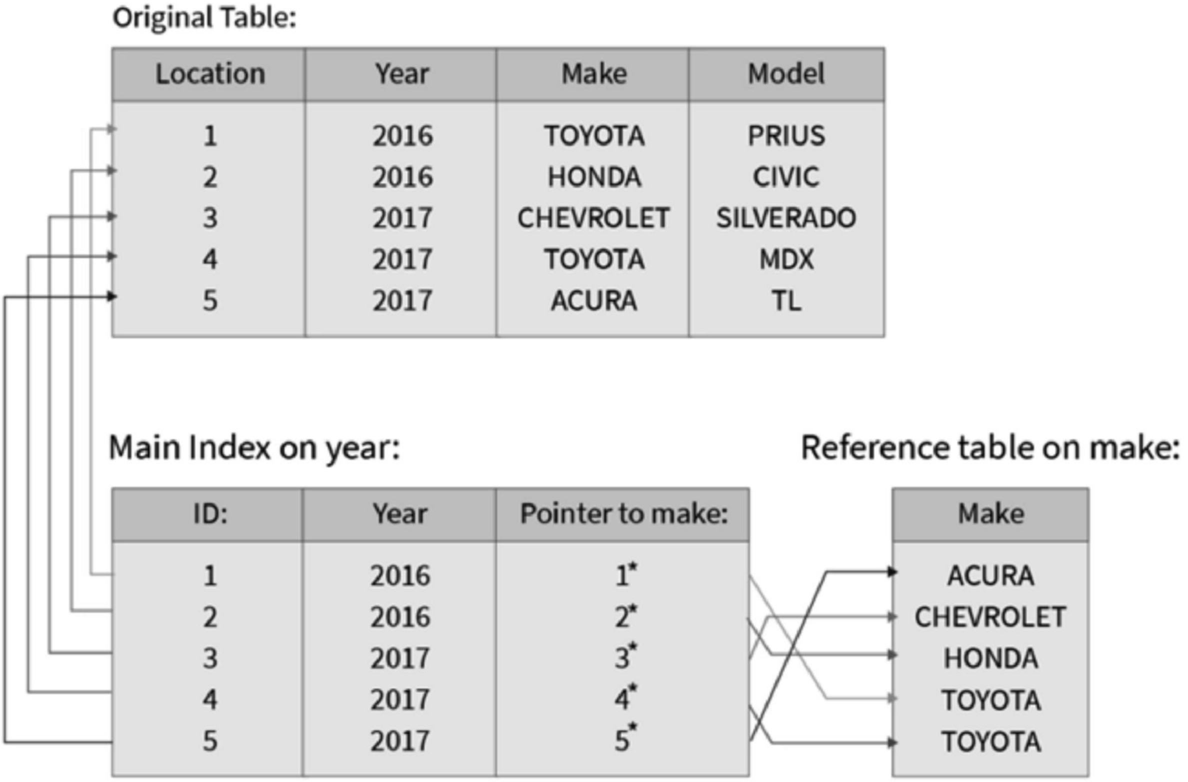
ON members(UPPER(target-column));

Note: The index table formed used columns values are also termed as Column Index or Column Index-table.

Single-Column and Concatenated

We can create a single-column index table or multi-column index table. Concatenated indexes are made according to certain WHERE clauses(WHERE clause related to the most frequent SQL Queries), hence making the searching or data retrieval faster.

Example - Let us take an example of a multi-column index table:



Original Table:

Location	Year	Make	Model
1	2016	TOYOTA	PRIUS
2	2016	HONDA	CIVIC
3	2017	CHEVROLET	SILVERADO
4	2017	TOYOTA	MDX
5	2017	ACURA	TL

Main Index on year:

ID:	Year	Pointer to make:
1	2016	1*
2	2016	2*
3	2017	3*
4	2017	4*
5	2017	5*

Reference table on make:

Make
ACURA
CHEVROLET
HONDA
TOYOTA
TOYOTA

We can use the primary key to create multiple index tables such as indexing based on year (grouping years) or indexing based on model-name etc. This multi-table indexing will help in getting specific query results faster.

Note: The multi-column is also termed a Concatenated Index.

Non-Partitioned and Partitioned

As we know index points to a certain table or block of data but sometimes the data itself is partitioned in a certain manner, so we need to partition the index table as well. Generally, we use the same table partition schema for the partition of the index table which is known as the Local Partition Index. We use the same schema so that the data retrieval speed is maintained. However, we can also create our non-partitioned index. This is known as Global Index of the partitioned table.

Example - Suppose we have a table namely a student table. If the student table is partitioned according to the roll number(primary key) then the index table of the student table should be partitioned according to roll number as well. This type of partition will help in the grouping of similar data and faster query results.

Types of Indexes

According to the attributes defined above, we divide indexing into three types:

Single Level Indexing

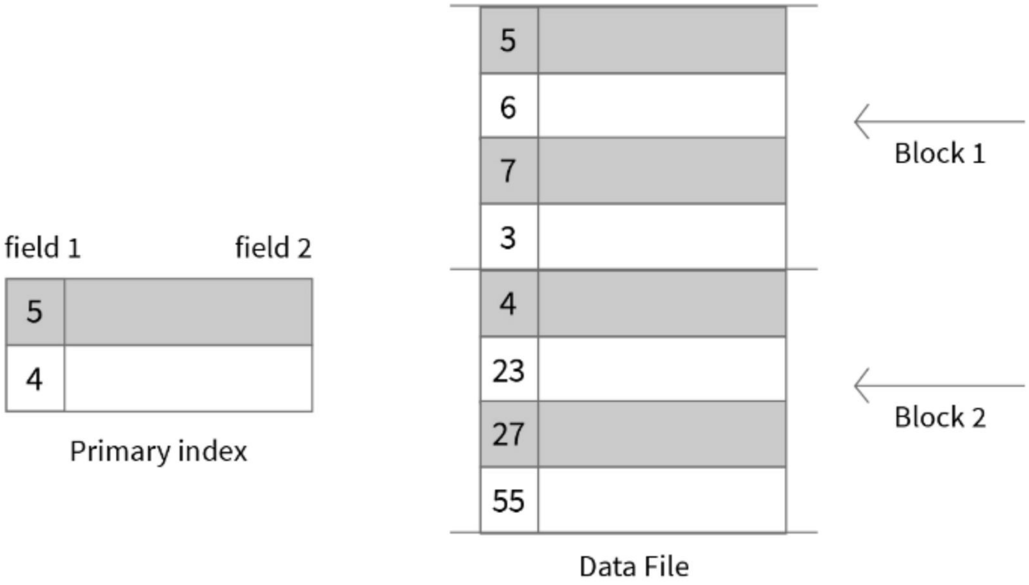
It is somewhat like the index (or the table of contents) found in a book. Index of a book contains topic names along with the page number similarly the index table of the database contains keys and their corresponding block address.

Single Level Indexing is further divided into three categories:

1. Primary Indexing

The indexing or the index table created using Primary keys is known as Primary Indexing. It is defined on ordered data. As the index is comprised of primary keys, they are unique, not null, and possess one to one relationship with the data blocks.

Example:



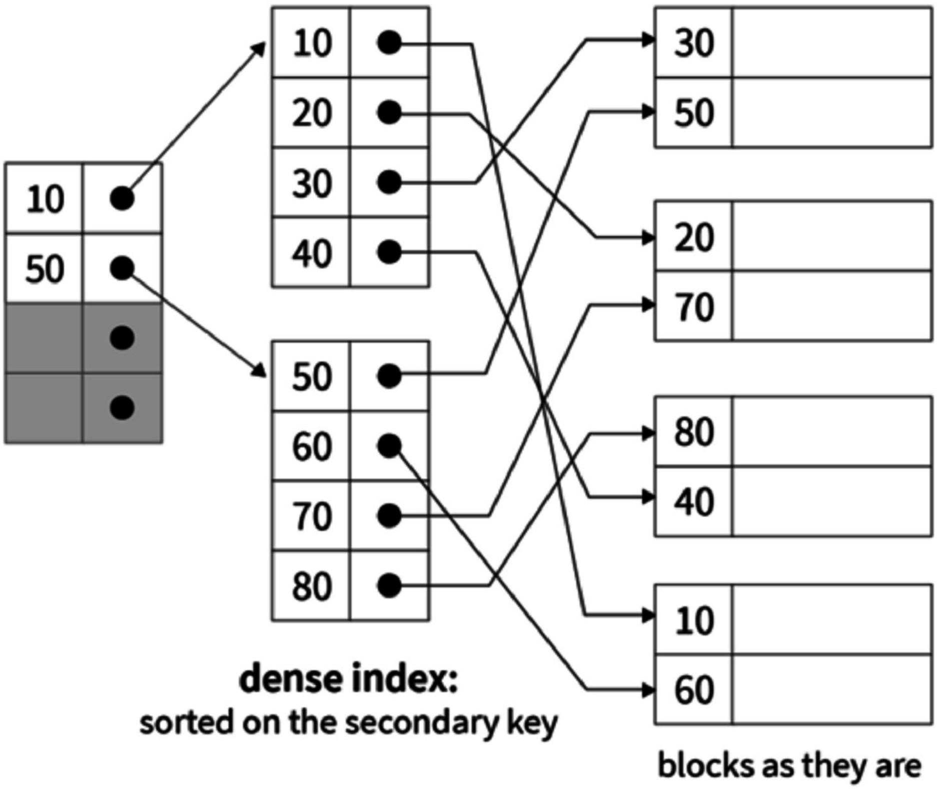
Characteristics of Primary Indexing:

- Search Keys are unique.
- Search Keys are in sorted order.
- Search Keys cannot be null as it points to a block of data.
- Fast and Efficient Searching.

2. Secondary Indexing

It is a two-level indexing technique used to reduce the mapping size of the primary index. The secondary index points to a certain location where the data is to be found but the actual data is not sorted like in the primary indexing. Secondary Indexing is also known as non-clustered Indexing.

Example:



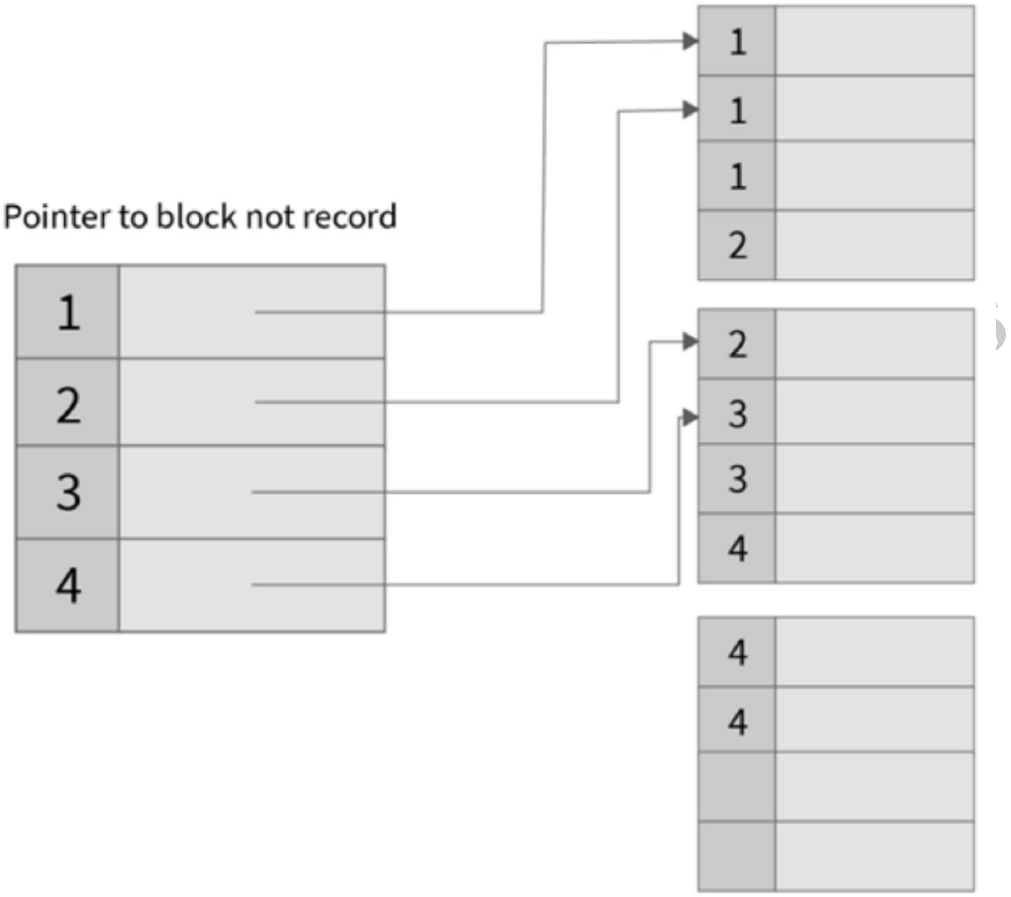
Characteristics of Secondary Indexing:

- Search Keys are Candidate Keys.
- Search Keys are sorted but actual data may or may not be sorted.
- Requires more time than primary indexing.
- Search Keys cannot be null.
- Faster than clustered indexing but slower than primary indexing.

3. Cluster Indexing

Clustered Indexing is used when there are multiple related records found at one place. It is defined on ordered data. The important thing to note here is that the index table of clustered indexing is created using non-key values which may or may not be unique. To achieve faster retrieval, we group columns having similar characteristics. The indexes are created using these groups and this process is known as Clustering Index.

Example:



Characteristics of Clustered Indexing:

- Search Keys are non-key values.
- Search Keys are sorted.
- Search Keys cannot be null.
- Search Keys may or may not be unique.
- Requires extra work to create indexing.

Ordered Indexing

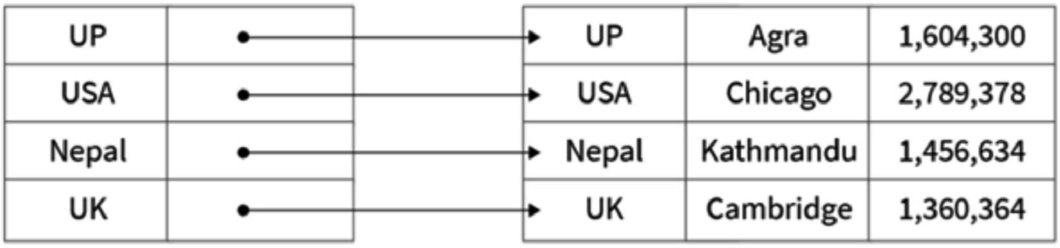
Ordered indexing is the traditional way of storing that gives fast retrieval. The indices are stored in a sorted manner hence it is also known as ordered indices.

Ordered Indexing is further divided into two categories:

1. **Dense Indexing**

   In dense indexing, the index table contains records for every search key value of the database. This makes searching faster but requires a lot more space. It is like primary indexing but contains a record for every search key.

   Example:



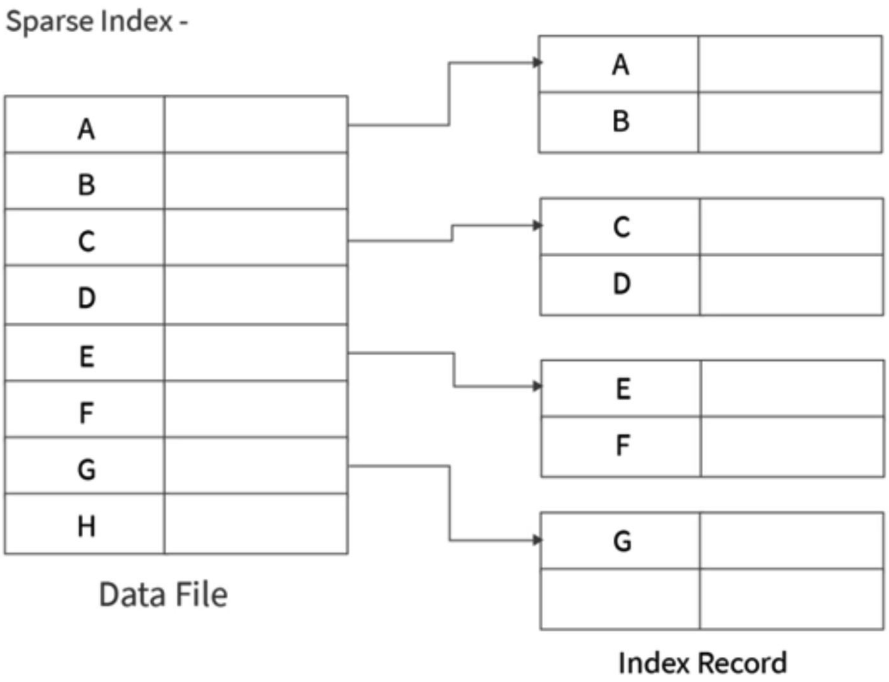
UP	●
USA	●
Nepal	●
UK	●

UP	Agra	1,604,300
USA	Chicago	2,789,378
Nepal	Kathmandu	1,456,634
UK	Cambridge	1,360,364

2. **Sparse Indexing**

   Sparse indexing consumes lesser space than dense indexing, but it is a bit slower as well. We do not include a search key for every record despite that we store a Search key that points to a block. The pointed block further contains a group of data. Sometimes we have to perform double searching this makes sparse indexing a bit slower.

   Example:

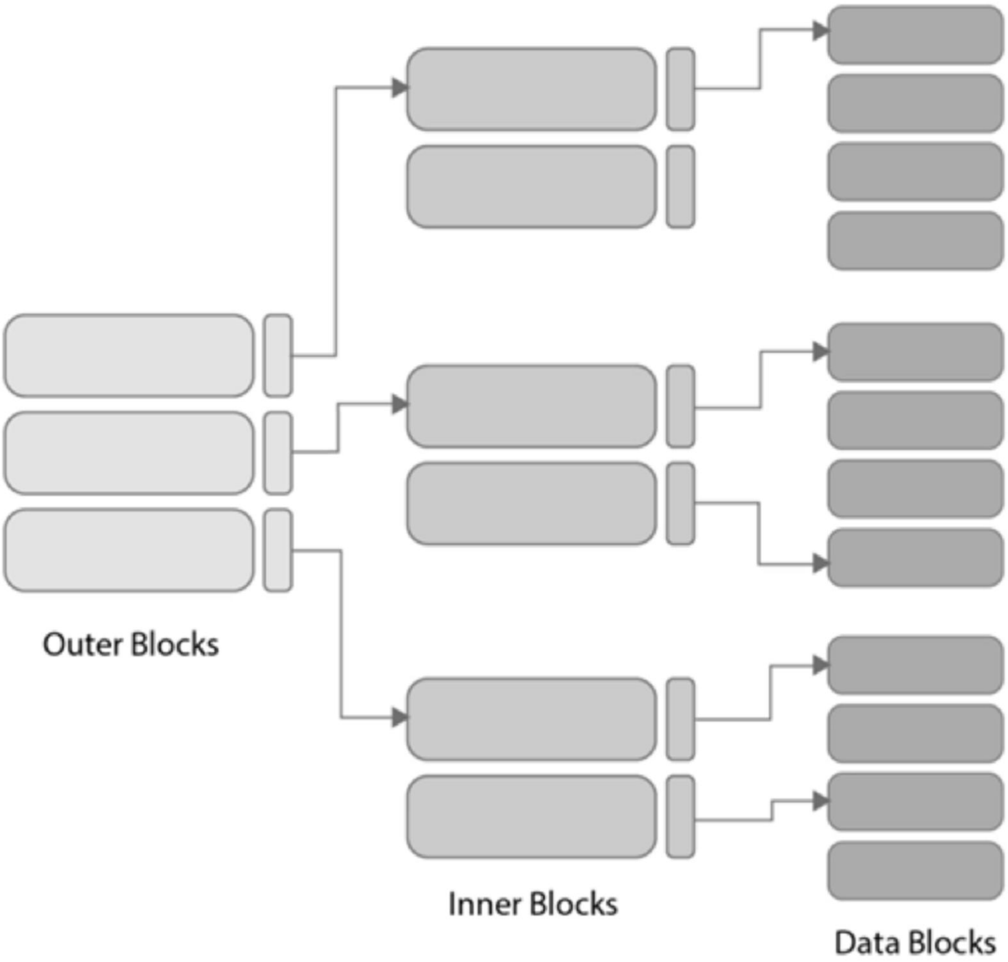


For very few search value in a Data File, there is an Index Record.
Hence the name Sparse Index.

Multi-Level Indexing

Since the index table is stored in the main memory, single-level indexing for a huge amount of data requires a lot of memory space. Hence, multilevel indexing was introduced in whichwe divide the main data block into smaller blocks. This makes the outer block of the index table small enough to be stored in the main memory.

Example:



We use the B+ Tree data structure for multilevel indexing. The leaf nodes of the B+ tree contain the actual data pointers. The leaf nodes are themselves in the form of a linked list. This linked list representation helps in both sequential and random access.

Advantages of Indexing

- Indexing helps in faster query results or quick data retrieval.
- Indexing helps in faster sorting and grouping of records
- Some Indexing uses sorted and unique keys which helps to retrieve sorted queries even faster.
- Index tables are smaller in size so require lesser memory.

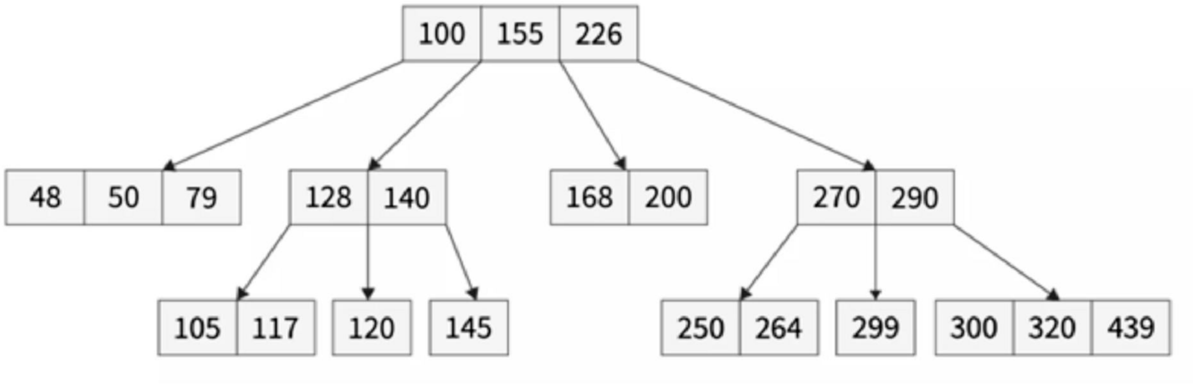
- As Index tables are smaller in size, they are stored in the main memory.
- Since CPU speed and secondary memory speed have a large difference, the CPU uses this main memory index table to bridge the gap of speeds.
- Indexing helps in better CPU utilization and better performance.

3.7 B-Trees

Q8. Explain about B-Trees in DBMS.

Ans : **(Imp.)**

B-tree in DBMS is an m-way tree which self balancesitself. Due to their balanced structure, such trees are frequently used to manage and organise enormous databases and facilitate searches. In a B-tree, each node can have a maximum of n child nodes. In DBMS, B-tree is an example of multilevel indexing. Leaf nodes and internal nodes will both have record references. B Tree is called Balanced stored trees as all the leaf nodes are at same levels.



Properties

Following are some of the properties of B-tree in DBMS:

- A non-leaf node's number of keys is one less than the number of its children.
- The number of keys in the root ranges from one to (m-1) maximum. Therefore, root has a minimum of two and a maximum of m children.
- The keys range from min([m/2]-1) to max(m-1) for all nodes (non-leaf nodes) besides the root. Thus, they can have between m and [m/2] children.
- The level of each leaf node is the same.

Need

- For having optimized searching we cannot increase a tree's height. Therefore, we want the tree to be as short as possible in height.
- Use of B-tree in DBMS, which has more branches and hence shorter height, is the solution to this problem. Access time decreases as branching and depth grow.
- Hence, use of B-tree is needed for storing data as searching and accessing time is decreased.

- The cost of accessing the disc is high when searching tables Therefore, minimising disc access is our goal.
- So to decrease time and cost, we use B-tree for storing data as it makes the Index Fast.

Q9. How Database B-Tree Indexing Works.

Ans :

- When B-tree is used for database indexing, it becomes a little more complex because it has both a key and a value. The value serves as a reference to the particular data record. A payload is the collective term for the key and value.
- For index data to particular key and value, the database first constructs a unique random index or a primary key for each of the supplied records. The keys and record byte streams are then all stored on a B+ tree. The random index that is generated is used for indexing of the data.
- So this indexing helps to decrease the searching time of data. In a B-tree, all the data is stored on the leaf nodes, now for accessing a particular data index, database can make use of binary search on the leaf nodes as the data is stored in the sorted order.
- if indexing is not used, the database reads each and every records to locate the requested record and it increases time and cost for searching the records, so B-tree indexing is very efficient.

Q10. How Searching Happens in Indexed Database?

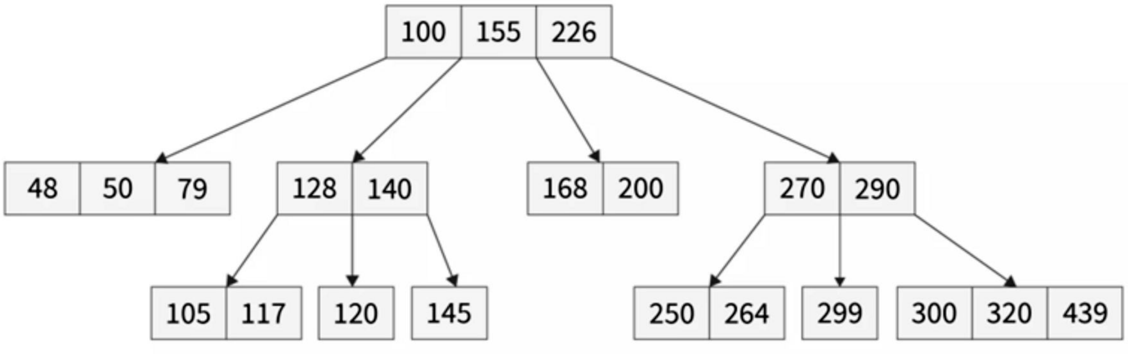
Ans :

The database does a search in the B-tree for a given key and returns the index in O(log(n)) time. The record is then obtained by running a second B+tree search in O(log(n)) time using the discovered index. So overall approx time taken for searching a record in a B-tree in DBMS Indexed databases is O(log(n)).

Examples of B-Tree

Suppose there are some numbers that need to be stored in a database, so if we store them in a B-tree in DBMS, they will be stored in a sorted order so that the searching time can be logarithmic.

Lets take a look at an example



The above data is stored in sorted order according to the values, if we wanna searching for the node containing the value 1, so the following steps will be applied:

- First, the parent node with key having data 3 is checked, as 1 is less than 3 so the left children node of 3 is checked.
- In left children, there are 2 keys, so it will check from the leftmost key as the data is stored in sorted order.
- Leftmost element is having key value as 1 which match the element to be searched, so thats how we the element we wanted to search.

3.8 Hashing

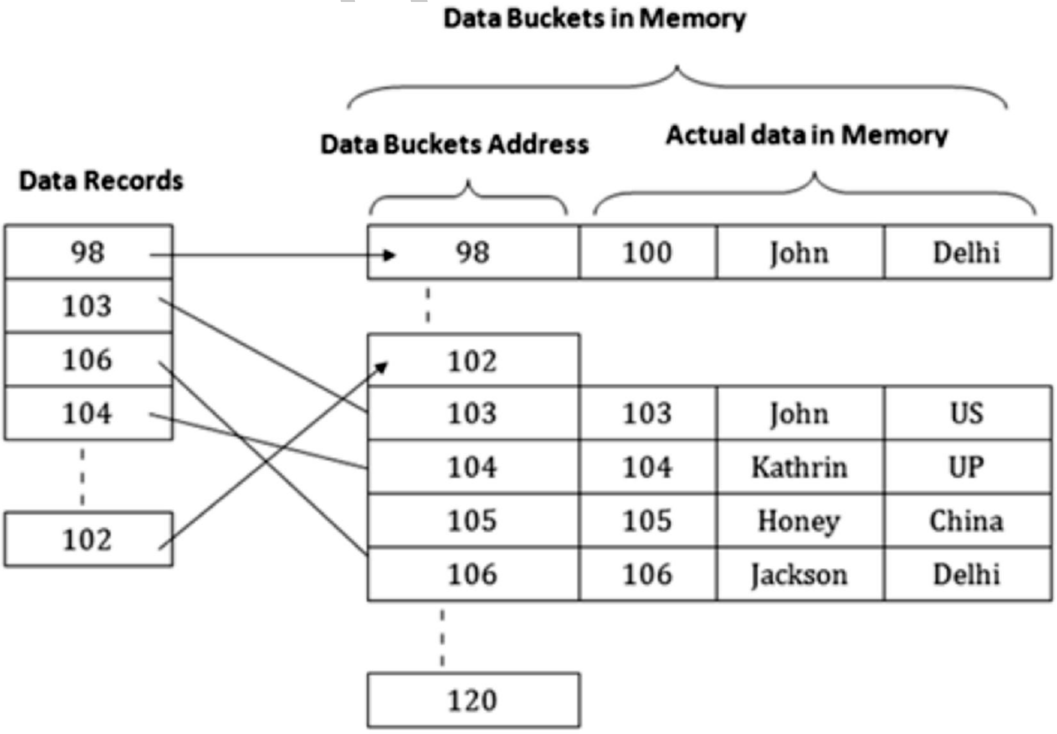
Q11. What is hashing in DBMS? Explain about various hashing techniques.

Ans : **(Imp.)**

In a huge database structure, it is very inefficient to search all the index values and reach the desired data. Hashing technique is used to calculate the direct location of a data record on the disk without using index structure.

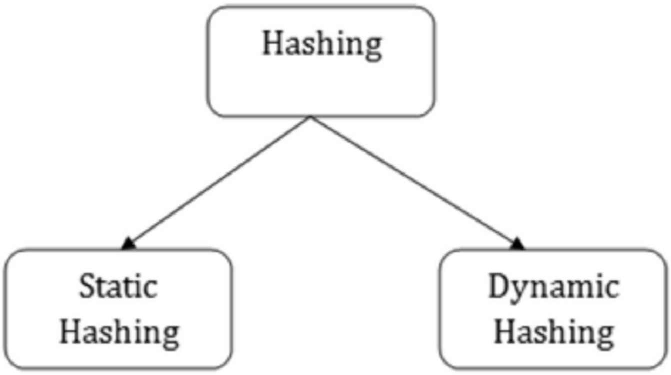
In this technique, data is stored at the data blocks whose address is generated by using the hashing function. The memory location where these records are stored is known as data bucket or data blocks.

In this, a hash function can choose any of the column value to generate the address. Most of the time, the hash function uses the primary key to generate the address of the data block. A hash function is a simple mathematical function to any complex mathematical function. We can even consider the primary key itself as the address of the data block. That means each row whose address will be the same as a primary key stored in the data block.



The above diagram shows data block addresses same as primary key value. This hash function can also be a simple mathematical function like exponential, mod, cos, sin, etc. Suppose we have mod (5) hash function to determine the address of the data block. In this case, it applies mod (5) hash function on the primary keys and generates 3, 3, 1, 4 and 2 respectively, and records are stored in those data block addresses.

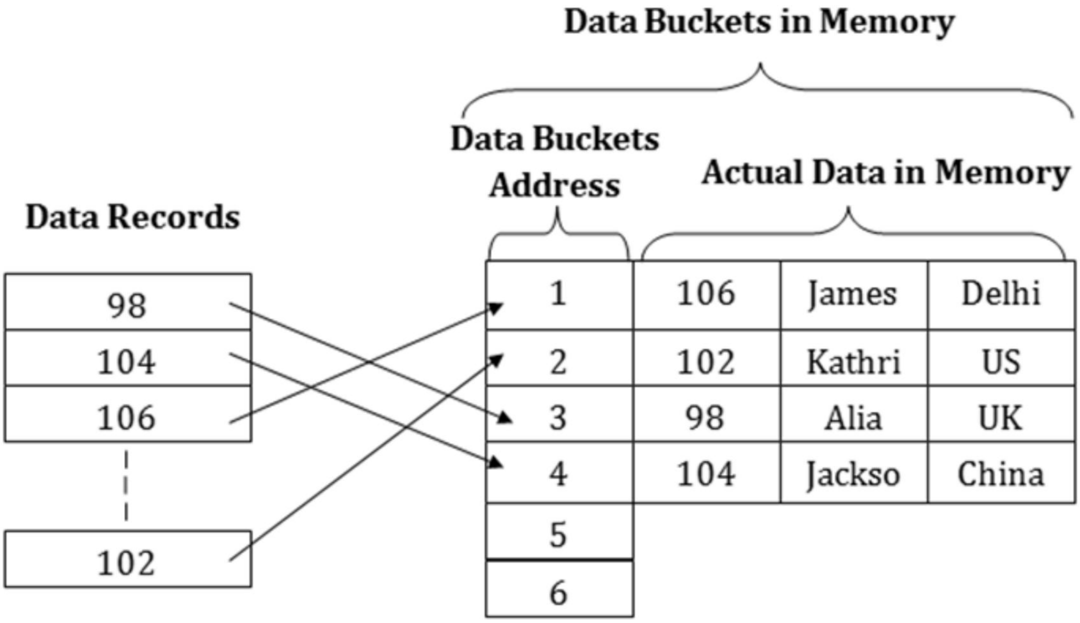
Types of Hashing:



1. Static Hashing

In static hashing, the resultant data bucket address will always be the same. That means if we generate an address for EMP_ID =103 using the hash function mod (5) then it will always result in same bucket address 3. Here, there will be no change in the bucket address.

Hence in this static hashing, the number of data buckets in memory remains constant throughout. In this example, we will have five data buckets in the memory used to store the data.



Data Buckets Address	Actual Data in Memory		
1	106	James	Delhi
2	102	Kathri	US
3	98	Alia	UK
4	104	Jackso	China
5			
6			

Operations of Static Hashing

- **Searching a record**

  When a record needs to be searched, then the same hash function retrieves the address of the bucket where the data is stored.

- **Insert a Record**

  When a new record is inserted into the table, then we will generate an address for a new record based on the hash key and record is stored in that location.

- **Delete a Record**

  To delete a record, we will first fetch the record which is supposed to be deleted. Then we will delete the records for that address in memory.

- **Update a Record**

  To update a record, we will first search it using a hash function, and then the data record is updated.

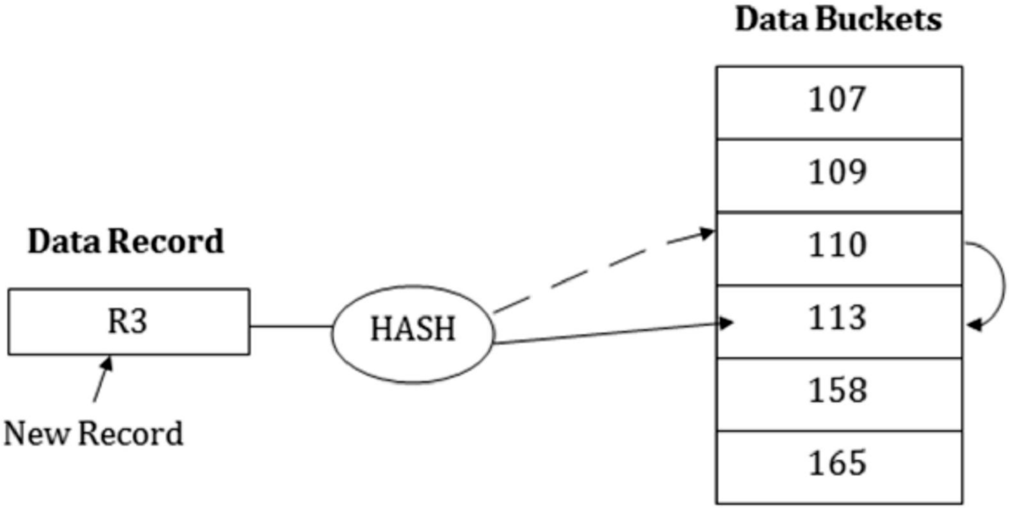
  If we want to insert some new record into the file but the address of a data bucket generated by the hash function is not empty, or data already exists in that address. This situation in the static hashing is known as bucket overflow. This is a critical situation in this method.

  To overcome this situation, there are various methods. Some commonly used methods are as follows:

2. **Open Hashing**

   When a hash function generates an address at which data is already stored, then the next bucket will be allocated to it. This mechanism is called as Linear Probing.

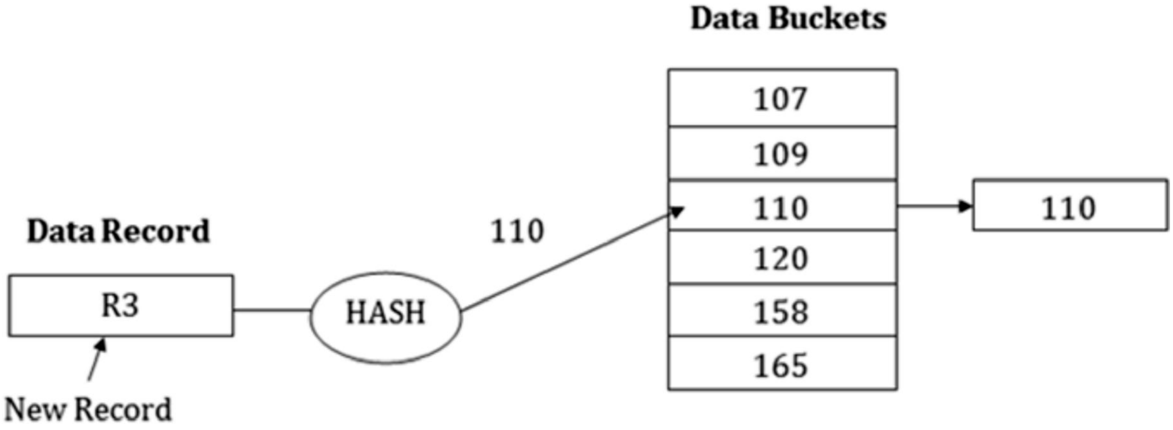
   For example: suppose R3 is a new address which needs to be inserted, the hash function generates address as 112 for R3. But the generated address is already full. So the system searches next available data bucket, 113 and assigns R3 to it.



3. Close Hashing

When buckets are full, then a new data bucket is allocated for the same hash result and is linked after the previous one. This mechanism is known as Overflow chaining.

For example: Suppose R3 is a new address which needs to be inserted into the table, the hash function generates address as 110 for it. But this bucket is full to store the new data. In this case, a new bucket is inserted at the end of 110 buckets and is linked to it.



4. Dynamic Hashing

- The dynamic hashing method is used to overcome the problems of static hashing like bucket overflow.
- In this method, data buckets grow or shrink as the records increases or decreases. This method is also known as Extendable hashing method.
- This method makes hashing dynamic, i.e., it allows insertion or deletion without resulting in poor performance.

How to search a key

- First, calculate the hash address of the key.
- Check how many bits are used in the directory, and these bits are called as i.
- Take the least significant i bits of the hash address. This gives an index of the directory.
- Now using the index, go to the directory and find bucket address where the record might be.

How to insert a new record

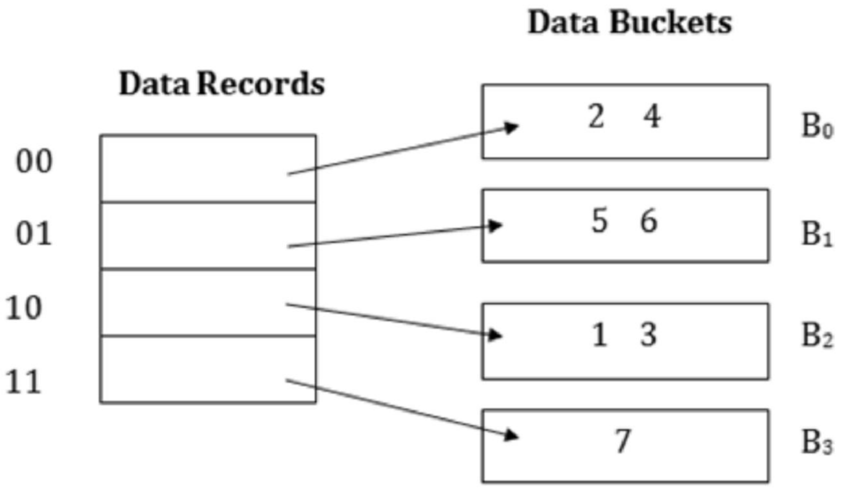
- Firstly, you have to follow the same procedure for retrieval, ending up in some bucket.
- If there is still space in that bucket, then place the record in it.
- If the bucket is full, then we will split the bucket and redistribute the records.

For example

Consider the following grouping of keys into buckets, depending on the prefix of their hash address:

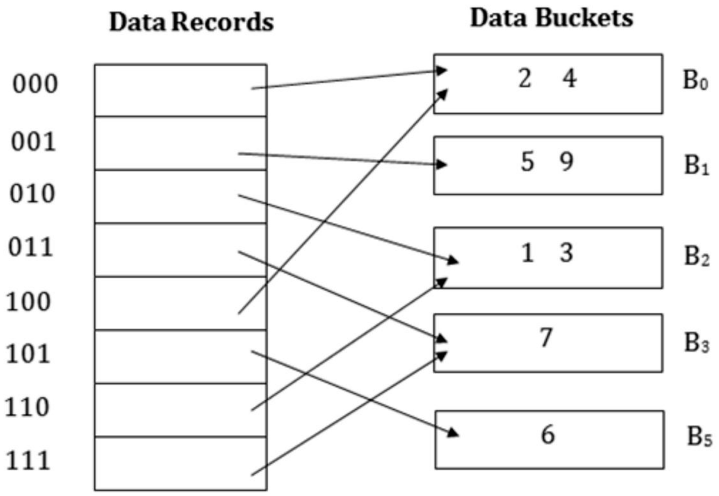
Key	Hash address
1	11010
2	00000
3	11110
4	00000
5	01001
6	10101
7	10111

The last two bits of 2 and 4 are 00. So it will go into bucket B0. The last two bits of 5 and 6 are 01, so it will go into bucket B1. The last two bits of 1 and 3 are 10, so it will go into bucket B2. The last two bits of 7 are 11, so it will go into B3.



Insert key 9 with hash address 10001 into the above structure:

- Since key 9 has hash address 10001, it must go into the first bucket. But bucket B1 is full, so it will get split.
- The splitting will separate 5, 9 from 6 since last three bits of 5, 9 are 001, so it will go into bucket B1, and the last three bits of 6 are 101, so it will go into bucket B5.
- Keys 2 and 4 are still in B0. The record in B0 pointed by the 000 and 100 entry because last two bits of both the entry are 00.
- Keys 1 and 3 are still in B2. The record in B2 pointed by the 010 and 110 entry because last two bits of both the entry are 10.
- Key 7 are still in B3. The record in B3 pointed by the 111 and 011 entry because last two bits of both the entry are 11.



Advantages

- In this method, the performance does not decrease as the data grows in the system. It simply increases the size of memory to accommodate the data.
- In this method, memory is well utilized as it grows and shrinks with the data. There will not be any unused memory lying.
- This method is good for the dynamic database where data grows and shrinks frequently.

Disadvantages

- In this method, if the data size increases then the bucket size is also increased. These addresses of data will be maintained in the bucket address table. This is because the data address will keep changing as buckets grow and shrink. If there is a huge increase in data, maintaining the bucket address table becomes tedious.

  In this case, the bucket overflow situation will also occur. But it might take little time to reach this situation than static hashing

Short Question & Answers

1. Query processing

Ans :

Query Processing includes translations on high level Queries into low level expressions that can be used at physical level of file system, query optimization and actual execution of query to get the actual result.

2. Query Optimization

Ans :

A single query can be executed through different algorithms or re-written in different forms and structures. Hence, the question of query optimization comes into the picture – Which of these forms or pathways is the most optimal? The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans.

3. Query Equivalance.

Ans :

The equivalence rule says that expressions of two forms are the same or equivalent because both expressions produce the same outputs on any legal database instance. It means that we can possibly replace the expression of the first form with that of the second form and replace the expression of the second form with an expression of the first form. Thus, the optimizer of the query-evaluation plan uses such an equivalence rule or method for transforming expressions into the logically equivalent one.

The optimizer uses various equivalence rules on relational-algebra expressions for transforming the relational expressions. For describing each rule, we will use the following symbols:

$\theta, \theta_1, \theta_2 \ldots$: Used for denoting the predicates.

$L_1, L_2, L_3 \ldots$: Used for denoting the list of attributes.

$E, E_1, E_2 \ldots.$: Represents the relational-algebra expressions.

4. Query Optimization

Ans :

Query optimization is the process of selecting the most efficient way to execute a SQL statement. Because SQL is a nonprocedural language, the optimizer can merge, restructure, and process data in any sequence.

The Optimizer allocates a cost in numerical form for each step of a feasible plan for a given query and environment, and then discovers these values together to get a cost estimate for the plan or possible strategy. The Optimizer aims to find the plan with the lowest cost estimate after evaluating the costs of all feasible plans. As a result, the Optimizer is sometimes known as the Cost-Based Optimizer.

5. Adaptive Query Optimization

Ans :

Adaptive query optimization allows the optimizer to make run-time changes to execution plans and uncover new information that can lead to improved statistics.

When existing facts are insufficient to produce an ideal strategy, adaptive optimization comes in handy. The image below depicts the feature set for adaptive query optimization.

6. Indexing

Ans :

Indexing is used to quickly retrieve particular data from the database. Formally we can define Indexing as a technique that uses data structures to optimize the searching time of a database query. Indexing reduces the number of disks required to access a particular data by internally creating an index table.

7. Primary Indexing

Ans :

The indexing or the index table created using Primary keys is known as Primary Indexing. It is defined on ordered data. As the index is comprised of primary keys, they are unique, not null, and possess one to one relationship with the data blocks.

8. Secondary Indexing

Ans :

It is a two-level indexing technique used to reduce the mapping size of the primary index. The secondary index points to a certain location where the data is to be found but the actual data is not sorted like in the primary indexing. Secondary Indexing is also known as non-clustered Indexing.

9. Sparse Indexing

Ans :

Sparse indexing consumes lesser space than dense indexing, but it is a bit slower as well. We do not include a search key for every record despite that we store a Search key that points to a block. The pointed block further contains a group of data. Sometimes we have to perform double searching this makes sparse indexing a bit slower.

10. Properties of B-tree

Ans :

- A non-leaf node's number of keys is one less than the number of its children.
- The number of keys in the root ranges from one to (m-1) maximum. Therefore, root has a minimum of two and a maximum of m children.
- The keys range from min([m/2]1) to max(m-1) for all nodes (non-leaf nodes) besides the root. Thus, they can have between m and [m/2] children.
- The level of each leaf node is the same.

11. Hashing in DBMS

Ans :

In a huge database structure, it is very inefficient to search all the index values and reach the desired data. Hashing technique is used to calculate the direct location of a data record on the disk without using index structure.

In this technique, data is stored at the data blocks whose address is generated by using the hashing function. The memory location where these records are stored is known as data bucket or data blocks.

In this, a hash function can choose any of the column value to generate the address. Most of the time, the hash function uses the primary key to generate the address of the data block. A hash function is a simple mathematical function to any complex mathematical function. We can even consider the primary key itself as the address of the data block. That means each row whose address will be the same as a primary key stored in the data block.

Choose the Correct Answer

1. Query ___ is the activity performed in extracting data from the database. [d]

   (a) Result (b) Inhibition

   (c) System (d) Processing

2. Data is ______ from the database using various steps in query processing. [c]

   (a) Extracted (b) Added

   (c) Fetched (d) Deleted

3. How many steps are involved in fetching the data from the database in query processing? [c]

   (a) 1 (b) 2

   (c) 3 (d) 4

4. What is/are the step(s) involved in fetching the data from the database in query processing? [d]

   (a) Parsing and translation (b) Optimization

   (c) Evaluation (d) All of the above

5. Initial queries from users are translated into a ______ database language such as SQL. [c]

   (a) Low-level (b) Medium-level

   (c) High-level (d) None

6. The queries are translated into ______ expressions at the level of the file system, which are used there as well. [c]

   (a) Virtual (b) Real

   (c) Physical (d) None

7. As soon as the queries are translated, they are evaluated and various ______ transformations are performed. [b]

   (a) Query-realizing (b) Query-optimizing

   (c) Query-deoptimizing (d) Query-deletion

8. Whenever a computer system processes a query, it ______ first convert it into a language humans can comprehend. [c]

   (a) Need not to (b) Must

   (c) Can (d) Maybe

9. Query languages such as SQL are best suited for humans, however, it is not best suited for the ______ of queries to a system. [c]

   (a) Temptation (b) Processing

   (c) Transmission (d) None

10. An ______ representation of a query is best suited to relational algebra. [b]

   (a) External (b) Internal

   (c) Both A and B (d) None of the above

Fill in the Blanks

1. A _______ is a request for information from a database.
2. The _______ rule says that expressions of two forms are the same or equivalent because both expressions produce the same outputs on any legal database instance.
3. _______ operators are those operators which can be derived from basic operators.
4. _______ optimization is the process of selecting the most efficient way to execute a SQL statement.
5. The _______ receives a parsed representation of a SQL statement as input.
6. The _______ can defer the final plan decision for a statement with an adaptive plan until execution time.
7. The _______ ability to alter a plan based on information gained during execution can significantly increase query performance
8. _______ is used to quickly retrieve particular data from the database.
9. _______ Indexing is used when there are multiple related records found at one place.
10. In _______ indexing, the index table contains records for every search key value of the database.

Answers

1. Query
2. Equivalence
3. Extended
4. Query
5. Optimizer
6. Optimizer
7. Optimizer's
8. Indexing
9. Clustered
10. Dense

UNIT IV

Transaction Processing and Database Security

Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery Authentication, Authorization and access control.

4.1 Transaction Processing and Database Security

4.1.1 Concurrency Control

Q1. What is concurrency control? Explain the problems of concurrency control.

Ans : **(Imp.)**

Meaning

Concurrency Control is the management procedure that is required for controlling concurrent execution of the operations that take place on a database.

But before knowing about concurrency control, we should know about concurrent execution.

Concurrent Execution in DBMS

- In a multi-user system, multiple users can access and use the same database at one time, which is known as the concurrent execution of the database. It means that the same database is executed simultaneously on a multi-user system by different users.
- While working on the database transactions, there occurs the requirement of using the database by multiple users for performing different operations, and in that case, concurrent execution of the database is performed.
- The thing is that the simultaneous execution that is performed should be done in an interleaved manner, and no operation should affect the other executing operations, thus maintaining the consistency of the database. Thus, on making the concurrent execution of the transaction operations, there occur several challenging problems that need to be solved.

Problems

In a database transaction, the two main operations are **READ** and **WRITE** operations. So, there is a need to manage these two operations in the concurrent execution of the transactions as if these operations are not performed in an interleaved manner, and the data may become inconsistent. So, the following problems occur with the Concurrent Execution of the operations:

Lost Update Problems (W - W Conflict)

The problem occurs when two different database transactions perform the read/write operations on the same database items in an interleaved manner (i.e., concurrent execution) that makes the values of the items incorrect hence making the database inconsistent.

For example

Consider the below diagram where two transactions T_X and T_Y, are performed on the same account A where the balance of account A is $300.

Time	T_x	T_y
t_1	READ (A)	–
t_2	A = A – 50	
t_3	–	READ (A)
t_4	–	A = A + 100
t_5	–	–
t_6	WRITE (A)	–
t_7		WRITE (A)

Lost Update Problem

- At time t_1, transaction T_X reads the value of account A, i.e., $300 (only read).
- At time t_2, transaction T_X deducts $50 from account A that becomes $250 (only deducted and not updated/write).
- Alternately, at time t_3, transaction T_Y reads the value of account A that will be $300 only because T_X didn't update the value yet.
- At time t_4, transaction T_Y adds $100 to account A that becomes $400 (only added but not updated/write).
- At time t_6, transaction T_X writes the value of account A that will be updated as $250 only, as T_Y didn't update the value yet.
- Similarly, at time t_7, transaction T_Y writes the values of account A, so it will write as done at time t_4 that will be $400. It means the value written by T_X is lost, i.e., $250 is lost.

Hence data becomes incorrect, and database sets to inconsistent.

Dirty Read Problems (W-R Conflict)

The dirty read problem occurs when one transaction updates an item of the database, and somehow the transaction fails, and before the data gets rollback, the updated database item is accessed by another transaction. There comes the Read-Write Conflict between both transactions.

For example:

Consider two transactions T_X and T_Y in the below diagram performing read/write operations on account A where the available balance in account A is $ 300:

Time	T_x	T_y
t_1	READ (A)	–
t_2	A = A + 50	–
t_3	WRITE (A)	–
t_4	–	READ (A)
t_5	SERVER DOWN ROLL BACK	–

Dirty Read Problem

- At time t_1, transaction T_X reads the value of account A, i.e., $300.
- At time t_2, transaction T_X adds $50 to account A that becomes $350.
- At time t_3, transaction T_X writes the updated value in account A, i.e., $350.
- Then at time t_4, transaction T_Y reads account A that will be read as $350.
- Then at time t_5, transaction T_X rollbacks due to server problem, and the value changes back to $300 (as initially).
- But the value for account A remains $350 for transaction T_Y as committed, which is the dirty read and therefore known as the Dirty Read Problem.

Unrepeatable Read Problem (W-R Conflict)

Also known as Inconsistent Retrievals Problem that occurs when in a transaction, two different values are read for the same database item.

For example:

Consider two transactions, T_X and T_Y, performing the read/write operations on account A, having an available balance = $300. The diagram is shown below:

Time	T_x	T_y
t_1	READ (A)	–
t_2	–	READ (A)
t_3		A = A + 100
t_4	–	WRITE (A)
t_5	READ (A)	–

UNREPEATABLE READ

- At time t1, transaction T_X reads the value from account A, i.e., $300.
- At time t2, transaction T_Y reads the value from account A, i.e., $300.
- At time t3, transaction T_Y updates the value of account A by adding $100 to the available balance, and then it becomes $400.
- At time t4, transaction T_Y writes the updated value, i.e., $400.

- After that, at time t5, transaction T_X reads the available value of account A, and that will be read as $400.
- It means that within the same transaction T_X, it reads two different values of account A, i.e., $ 300 initially, and after updation made by transaction T_Y, it reads $400. It is an unrepeatable read and is therefore known as the Unrepeatable read problem.

Thus, in order to maintain consistency in the database and avoid such problems that take place in concurrent execution, management is needed, and that is where the concept of Concurrency Control comes into role.

Concurrency Control

Concurrency Control is the working concept that is required for controlling and managing the concurrent execution of database operations and thus avoiding the inconsistencies in the database. Thus, for maintaining the concurrency of the database, we have the concurrency control protocols.

Concurrency Control Protocols

The concurrency control protocols ensure the atomicity, consistency, isolation, durability and serializability of the concurrent execution of the database transactions. Therefore, these protocols are categorized as:

- Lock Based Concurrency Control Protocol
- Time Stamp Concurrency Control Protocol
- Validation Based Concurrency Control Protocol.

4.1.2 ACID Property

Q2. What is known as transaction in DBMS? Explain with example.

Ans :

- The transaction is a set of logically related operation. It contains a group of tasks.
- A transaction is an action or series of actions. It is performed by a single user to perform operations for accessing the contents of the database.

Example

Suppose an employee of bank transfers Rs 800 from X's account to Y's account. This small transaction contains several low-level tasks:

X's Account

Open_Account(X)

Old_Balance = X.balance

New_Balance = Old_Balance - 800

X.balance = New_Balance

Close_Account(X)

Y's Account

Open_Account(Y)

Old_Balance = Y.balance

New_Balance = Old_Balance + 800

Y.balance = New_Balance

Close_Account(Y)

Operations of Transaction

Following are the main operations of transaction:

Read(X)

Read operation is used to read the value of X from the database and stores it in a buffer in main memory.

Write(X)

Write operation is used to write the value back to the database from the buffer.

Let's take an example to debit transaction from an account which consists of following operations:

1. R(X);
2. X = X - 500;
3. W(X);

Let's assume the value of X before starting of the transaction is 4000.

- The first operation reads X's value from database and stores it in a buffer.
- The second operation will decrease the value of X by 500. So buffer will contain 3500.
- The third operation will write the buffer's value to the database. So X's final value will be 3500.

But it may be possible that because of the failure of hardware, software or power, etc. that transaction may fail before finished all the operations in the set.

For example

If in the above transaction, the debit transaction fails after executing operation 2 then X's value will remain 4000 in the database which is not acceptable by the bank.

To solve this problem, we have two important operations:

Commit

It is used to save the work done permanently.

Rollback

It is used to undo the work done.

Q3. What are ACID properties? Explain.

Ans : **(Imp.)**

The transaction has the four properties. These are used to maintain consistency in a database, before and after the transaction.

Property of Transaction

1. Atomicity
2. Consistency
3. Isolation
4. Durability

1. Atomicity

- It states that all operations of the transaction take place at once if not, the transaction is aborted.
- There is no midway, i.e., the transaction cannot occur partially. Each transaction is treated as one unit and either run to completion or is not executed at all.

Atomicity involves the following two operations:

Abort

If a transaction aborts then all the changes made are not visible.

Commit

If a transaction commits then all the changes made are visible.

Example

Let's assume that following transaction T consisting of T1 and T2. A consists of Rs 600 and B consists of Rs 300. Transfer Rs 100 from account A to account B.

```
T1 Read(A)
   A:= A-100
   Write(A)
   Y:= Y+100
   Write(B)
```

After completion of the transaction, A consists of Rs 500 and B consists of Rs 400.

If the transaction T fails after the completion of transaction T1 but before completion of transaction T2, then the amount will be deducted from A but not added to B. This shows the inconsistent database state. In order to ensure correctness of database state, the transaction must be executed in entirety.

2. **Consistency**

- The integrity constraints are maintained so that the database is consistent before and after the transaction.
- The execution of a transaction will leave a database in either its prior stable state or a new stable state.
- The consistent property of database states that every transaction sees a consistent database instance.
- The transaction is used to transform the database from one consistent state to another consistent state.

For example

The total amount must be maintained before or after the transaction.

1. Total before T occurs = 600 + 300 = 900
2. Total after T occurs = 500 + 400 = 900

Therefore, the database is consistent. In the case when T1 is completed but T2 fails, then inconsistency will occur.

3. **Isolation**

- It shows that the data which is used at the time of execution of a transaction cannot be used by the second transaction until the first one is completed.
- In isolation, if the transaction T1 is being executed and using the data item X, then that data item can't be accessed by any other transaction T2 until the transaction T1 ends.
- The concurrency control subsystem of the DBMS enforced the isolation property.

4. **Durability**

- The durability property is used to indicate the performance of the database's consistent state. It states that the transaction made the permanent changes.
- They cannot be lost by the erroneous operation of a faulty transaction or by the system failure. When a transaction is completed, then the database reaches a state known as the consistent state. That consistent state cannot be lost, even in the event of a system's failure.
- The recovery subsystem of the DBMS has the responsibility of Durability property.

4.1.3 Serializability of Scheduling

Q4. What is Serializabilityof scheduling in DBMS ? Explain.

Ans : **(Imp.)**

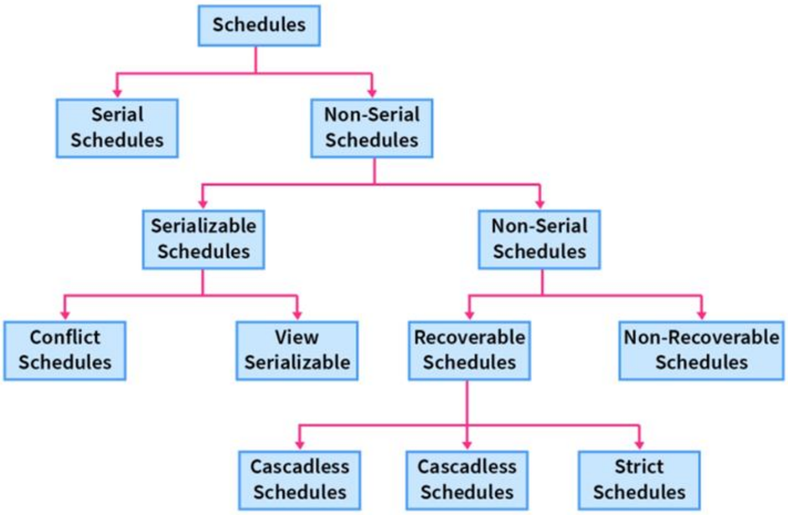
Serializability of schedules ensures that a non-serial schedule is equivalent to a serial schedule. It helps in maintaining the transactions to execute simultaneously without interleaving one another. In simple words, serializability is a way to check if the execution of two or more transactions are maintaining the database consistency or not.

Schedules and Serializable Schedules in DBMS

Schedules are a series of operations performing one transaction to the other.

R(X) means Reading the value: X; and W(X) means Writing the value: X.

Types of Schedules in DBMS



Schedules in DBMS are of two types

1. **Serial Schedule:** A schedule in which only one transaction is executed at a time, i.e., one transaction is executed completely before starting another transaction.

Example:

Transaction-1	Transaction-2
R(a)	
W(a)	
R(b)	
W(b)	
	R(b)
	W(b)
	R(a)
	R(b)

Here, we can see that Transaction-2 starts its execution after the completion of Transaction-1.

Note

Serial schedules are always serializable because the transactions only work one after the other. Also, for a transaction, there are n! serial schedules possible (where n is the number of transactions).

2. Non-serial Schedule

A schedule in which the transactions are interleaving or interchanging. There are several transactions executing simultaneously as they are being used in performing real-world database operations. These transactions may be working on the same piece of data. Hence, the serializability of non-serial schedules is a major concern so that our database is consistent before and after the execution of the transactions.

Example:

Transaction-1	Transaction-2
R(a)	
W(a)	
	R(b)
	W(b)
R(b)	
	R(a)
W(b)	
	W(a)

We can see that Transaction-2 starts its execution before the completion of Transaction-1, and they are interchangeably working on the same data, i.e., "a" and "b".

Serializable schedule

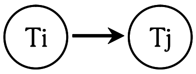
A non-serial schedule is called a serializable schedule if it can be converted to its equivalent serial schedule. In simple words, if a non-serial schedule and a serial schedule result in the same then the non-serial schedule is called a serializable schedule.

Testing of Serializability

To test the serializability of a schedule, we can use Serialization Graph or Precedence Graph. A serialization Graph is nothing but a Directed Graph of the entire transactions of a schedule.

It can be defined as a Graph G(V, E) consisting of a set of directed-edges E = {E1, E2, E3, ..., En} and a set of vertices V = {V1, V2, V3, ...,Vn}. The set of edges contains one of the two operations - READ, WRITE performed by a certain transaction.

Precedence Graph for Schedule S



Ti ->Tj, means Transaction-Ti is either performing read or write before the transaction-Tj.

Note

If there is a cycle present in the serialized graph then the schedule is non-serializable because the cycle resembles that one transaction is dependent on the other transaction and vice versa. It also means that there are one or more conflicting pairs of operations in the transactions. On the other hand, no-cycle means that the non-serial schedule is serializable.

What is a conflicting pair in transactions?

Two operations inside a schedule are called conflicting if they meet these three conditions:

1. They belong to two different transactions.
2. They are working on the same data piece.
3. One of them is performing the WRITE operation.

To conclude, let's take two operations on data: "a". The conflicting pairs are:

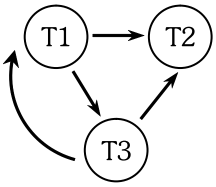
1. READ(a) - WRITE(a)
2. WRITE(a) - WRITE(a)
3. WRITE(a) - READ(a)

Note:

There can never be a read-read conflict as there is no change in the data.

Let's take an example of schedule "S" having three transactions t1, t2, and t3 working simultaneously, to get a better understanding.

t1	t2	t3
R(x)		
		R(z)
		W(z)
	R(y)	
R(y)		
	W(y)	
		W(x)
	W(z)	
W(x)		



Non-serializable schedule

R(x) of T1 conflicts with W(x) of T3, so there is a directed edge from T1 to T3. R(y) of T1 conflicts with W(y) of T2, so there is a directed edge from T1 to T2. W(y\x) of T3 conflicts with W(x) of T1, so there is a directed edge from T3 to T. Similarly, we will make edges for every conflicting pair. Now, as the cycle is formed, the transactions cannot be serializable.

Types of Serializability

Serializability of any non-serial schedule can be verified using two types mainly:

1. **Conflict Serializability and View Serializability.**

One more way to check serializability is by forming an equivalent serial schedule that results in the same as the original non-serial schedule. Since this process only focuses on the output rather than the operations taking place in between the switching of transactions, it is not practically used. Now let's discuss Conflict and View Serializability in detail.

2. **Conflict Serializability and Conflict Serializable Schedules**

A non-serial schedule is a conflict serializable if, after performing some swapping on the non-conflicting operation results in a serial schedule. It is checked using the non-serial schedule and an equivalent serial schedule. This process of checking is called Conflict Serializability.

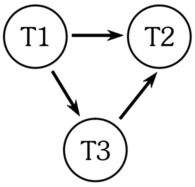
It is tedious to use if we have many operations and transactions as it requires a lot of swapping.

For checking, we will use the same Precedence Graph technique discussed above. First, we will check conflicting pairs operations(read-write, write-read, and write-write) and then form directed edges between those conflicting pair transactions. If we can find a loop in the graph, then the schedule is non-conflicting serializable otherwise it is surely a conflicting serializable schedule.

Example:

We have a schedule "S" having three transactions t1, t2, and t3 working simultaneously. Let's form is precedence graph.

t1	t2	t3
R(x)		
	R(y)	
		R(y)
	W(y)	
W(x)		
		W(x)
	R(x)	
	W(x)	



As there is no loop in the graph(the graph is DAG), it is a conflict serializable schedule as well as a serial schedule. Since it is a serial schedule, we can detect the order of transactions as well.

The order of the Transactions

t1 will execute first as there is no incoming edge on T1. t3 will execute second as it depends on T1 only. t2 will execute at last as it depends on both T1 and T3.

So, order of its equivalent serial schedule is: t1 → t3 → t2

Note :

If a schedule is conflicting serializable, then it is surely a consistent schedule. On the other hand, a non-conflicting serializable schedule may or may not be serial. To further check its serial behavior, we use the concept of View Serializability.

View Serializability and View Serializable Schedules

If a non-serial schedule is **view equivalent** to some other serial schedule then the schedule is called View Serializable Schedule. It is needed to ensure the consistency of a schedule.

The two conditions needed by schedules(S1 and S2) to be view equivalent are:

1. Initial read must be on the same piece of data.

   Example: If transaction t1 is reading "A" from database in schedule S1, then in schedule S2, t1 must read A.

2. Final write must be on the same piece of data.

   Example: If a transaction t1 updated A at last in S1, then in S2, t1 should perform final write as well.

3. The mid sequence should also be in the same order.

   Example: If t1 is reading A which is updated by t2 in S1, then in S2, t1 should read A which should be updated by t2.

This process of checking view equivalency of a schedule is called View Serializability.

Example

We have a schedule "S" having two transactions t1, and t2 working simultaneously.

S:

t1	t2
R(x)	
W(x)	
	R(x)
	W(x)
R(y)	
W(y)	

Let's form its view equivalent schedule (S') by interchanging mid-read-write operations of both the transactions. **S':**

W(x)	
R(y)	
W(y)	
	R(x)
	W(x)
	R(y)
	W(y)

Since a view equivalent schedule is possible, it is a view serializable schedule.

Note

A conflict serializable schedule is always viewed as serializable, but vice versa is not always true.

Actual process for checking view serializability

1. First, check for conflict serializability.
2. Check for a blind write. If there is a blind write, then the schedule can be view serializable. So, check its view serializability using the view equivalent schedule technique (stated above). If there is no blind write, then the schedule can never be view serializable.

Blind write is writing a value or piece of data without reading it.

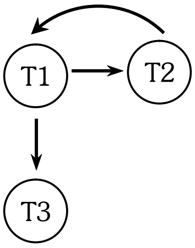
Example

We have a schedule "S" having two transactions t1, t2, and t3 working simultaneously.

S:

t1	t2	t3
R(x)		
	W(x)	
W(x)		
		W(x)

It's precedence graph:



Since there is a loop present, the schedule is non-conflicting serializable schedule. Now, there are blind writes [t2 → w(x) and t3 → w(x)] present, hence check for View Serializability.

One of its view equivalent schedules can be:

S':

t1	t2	t3
R(x)		
W(x)		
	W(x)	
		W(x)

Hence, the schedule is view serializable.

Note:

One important thing to note here is that we can form a number of sequences of the transactions such as:

t1	t2	t3
t1	t3	t2
t2	t1	t3
t2	t3	t1
t3	t1	t2
t3	t2	t1

This makes it a **N-P Hard** problem.

Benefits of Serialization

- Serialization helps in checking concurrency control between multiple transactions.
- It also helps in maintaining consistency in the database before and after any transaction.
- Serializable schedules are resource-efficient and help in improving CPU throughput (total work done in a certain time).

Q5. Check whether the given schedule S is view serialisable or not? Practice problems based on view serializability.

Ans :

Check whether the given schedule S is view serializable or not-

T1	T2	T3	T4
R(A)			
	R(A)		
		R(A)	
			R(A)
W(B)			
	W(B)		
		W(B)	
			W(B)

Sol.:

- We know, if a schedule is conflict serializable, then it is surely view serializable.
- So, let us check whether the given schedule is conflict serializable or not.

Checking Whether S is Conflict Serializable Or Not-

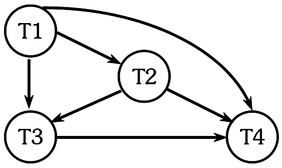
Step-01:

List all the conflicting operations and determine the dependency between the transactions-

- $W_1(B)$, $W_2(B)$ $(T_1 \rightarrow T_2)$
- $W_1(B)$, $W_3(B)$ $(T_1 \rightarrow T_3)$
- $W_1(B)$, $W_4(B)$ $(T_1 \rightarrow T_4)$
- $W_2(B)$, $W_3(B)$ $(T_2 \rightarrow T_3)$
- $W_2(B)$, $W_4(B)$ $(T_2 \rightarrow T_4)$
- $W_3(B)$, $W_4(B)$ $(T_3 \rightarrow T_4)$

Step-02

Draw the precedence graph-



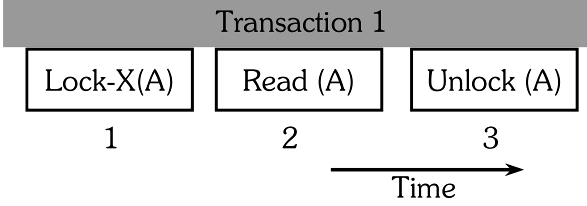
- Clearly, there exists no cycle in the precedence graph.
- Therefore, the given schedule S is conflict serializable.
- Thus, we conclude that the given schedule is also view serializable.

4.1.4 Locking and Time Stamp Based Schedulers Lock Based schedulers

Q6. What are locks in DBMS? Explain about different types of locks.

Ans : **(Imp.)**

We can define a lock-based protocol in DBMS as a mechanism that is responsible for preventing a transaction from reading or writing data until the necessary lock is obtained. The concurrency problem can be solved by securing or locking a transaction to a specific user. The lock is a variable that specifies which activities are allowed on a certain data item.



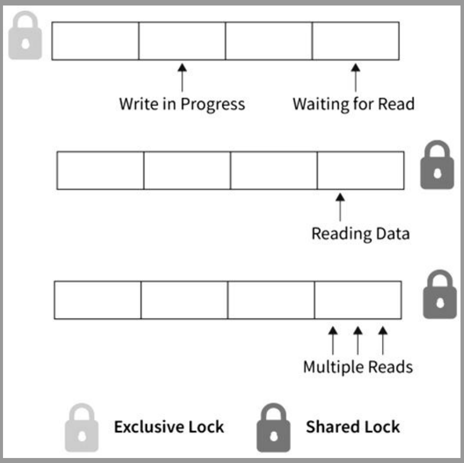
Let's go through one real-life example which will help us understand the concept of the lock-based protocol in DBMS. Every second, millions of payments take place. Consider buying a movie ticket at your preferred theatre. Now, there's a chance that two or more customers will try to reserve the very same seat without knowing it.

A scenario like this, when treated on a big scale, can damage the database consistency resulting in the corruption of data. To avoid any conflicts over user access to read and write into the database and to ensure that there is no concurrency, each transaction must be handled separately. Lock-based protocols in DBMS are used to accomplish this purpose. To know more about the concept of concurrency, read here: Concurrrency control in DBMS

Types

In DBMS Lock based Protocols, there are two modes for locking and unlocking data items Shared Lock (lock-S) and Exclusive Lock (lock-X). Let's go through the two types of locks in detail:

1. Shared Lock



- Shared Locks, which are often denoted as lock-S(), is defined as locks that provide Read-Only access to the information associated with them. Whenever a shared lock is used on a database, it can be read by several users, but these users who are reading the information or the data items will not have permission to edit it or make any changes to the data items.
- To put it another way, we can say that shared locks don't provide access to write. Because numerous users can read the data items simultaneously, multiple shared locks can be installed on them at the same time, but the data item must not have any other locks connected with it.
- A shared lock, also known as a read lock, is solely used to read data objects. Read integrity is supported via shared locks.

- Shared locks can also be used to prevent records from being updated.
- S-lock is requested via the Lock-S instruction.

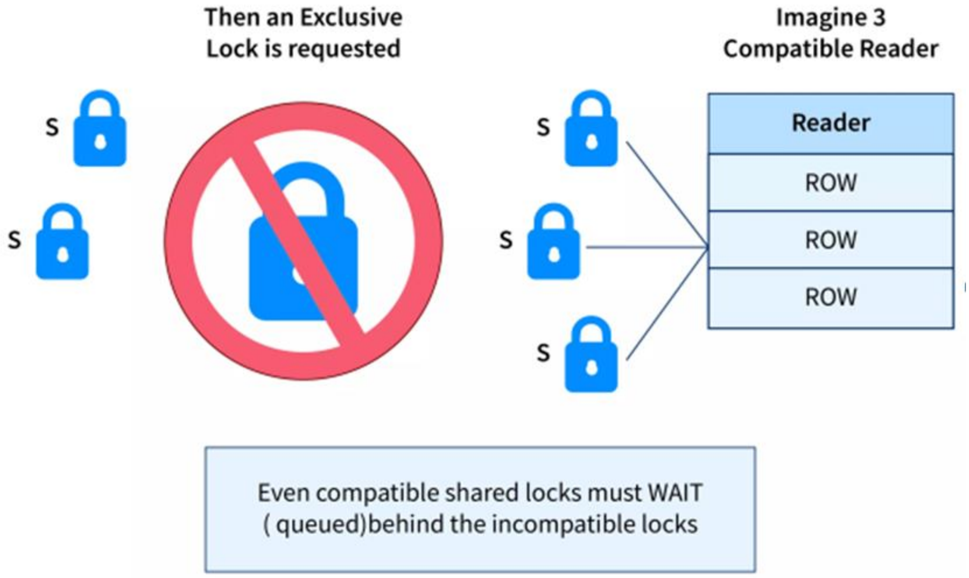
Example

Consider the situation where the value of variable X equals 50 and there are a total of 2 transactions reading X. If one transaction wants to change the value of A, another transaction that tries to read the value will read the incorrect value of the variable X. However, until it is done with reading, the Shared lock stops it from updating.

When the lock-based protocol in DBMS is applied to the transaction (let's say T1) discussed above, all the processes listed below occur.

1. T1 will gain exclusive access to the data item X.
2. Find out what the current value of data item A is.
3. The data item will be accessible once the transaction is finished.

2. Exclusive Lock



- Exclusive Lock allows the data item to be read as well as written. This is a one-time use mode that can't be utilized on the exact data item twice. To obtain X-lock, the user needs to make use of the lock-x instruction. After finishing the 'write' step, transactions can unlock the data item.
- By imposing an X lock on a transaction that needs to update a person's account balance, for example, you can allow it to proceed. As a result of the exclusive lock, the second transaction is unable to read or write.
- The other name for an exclusive lock is write lock.
- At any given time, the exclusive locks can only be owned by one transaction.

Example locks

Consider the instance where the value of a data item X is equal to 50 and a transaction requires a deduction of 20 from the data item X. By putting a Y lock on this particular transaction, we can make it possible. As a result, the exclusive lock prevents any other transaction from reading or writing.

3. Lock Compatibility Matrix

A vital point to remember when using Lock-based protocols in Database Management System is that a Shared Lock can be held by any amount of transactions. On the other hand, an Exclusive Lock can only be held by one transaction in DBMS, this is because a shared lock only reads data but does not perform any other activities, whereas an exclusive lock performs read as well as writing activities.

The figure given below demonstrates that when two transactions are involved, and both of these transactions seek to read a specific data item, the transaction is authorized, and no conflict occurs; but, in a situation when one transaction intends to write the data item and another transaction attempts to read or write simultaneously, the interaction is rejected.

	S	X
S	✓	X
X	X	X

The two methods outlined below can be used to convert between the locks:

1. Conversion from a read lock to a write lock is an upgrade.
2. Conversion from a write lock to a read lock is a downgrade.

Q7. Explain about types of lock based protocols.

Ans : **(Imp.)**

Types of Lock-Based Protocols

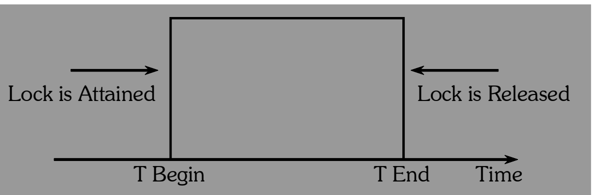
There are basically four lock based protocols in DBMS namely Simplistic Lock Protocol, Pre-claiming Lock Protocol, Two-phase Locking Protocol, and Strict Two-Phase Locking Protocol. Let's go through each of these lock-based protocols in detail.

1. Simplistic Lock Protocol

The simplistic method is defined as the most fundamental method of securing data during a transaction. Simple lock-based protocols allow all transactions to lock the data before inserting, deleting, or updating it. After the transaction is completed, the data item will be unlocked.

2. Pre-Claiming Lock Protocol

Pre-claiming Lock Protocols are known to assess transactions to determine which data elements require locks. Prior to actually starting the transaction, it asks the Database management system for all of the locks on all of the data items. The pre-claiming protocol permits the transaction to commence if all of the locks are obtained. Whenever the transaction is finished, the lock is released. This protocol permits the transaction to roll back if all of the locks are not granted and then waits until all of the locks are granted.

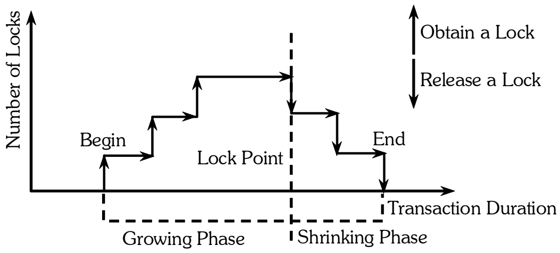


3. Two-phase Locking Protocol

If Locking as well as the Unlocking can be performed in 2 phases, a transaction is considered to follow the Two-Phase Locking protocol. The two phases are known as the growing and shrinking phases.

i) **Growing Phase:** In this phase, we can acquire new locks on data items, but none of these locks can be released.

ii) **Shrinking Phase:** In this phase, the existing locks can be released, but no new locks can be obtained.



Two-phase locking helps to reduce the amount of concurrency in a schedule but just like the two sides of a coin two-phase locking has a few cons too. The protocol raises transaction processing costs and may have unintended consequences. The likelihood of establishing deadlocks is one bad result.

4. Strict Two-Phase Locking Protocol

In DBMS, Cascaded rollbacks are avoided with the concept of a Strict Two-Phase Locking Protocol. This protocol necessitates not only two-phase locking but also the retention of all exclusive locks until the transaction commits or aborts. The two-phase is with deadlock.

It is responsible for assuring that if 1 transaction modifies data, there can be no other transaction that will be able to read it until the first transaction commits. The majority of database systems use a strict two-phase locking protocol.

Starvation and Deadlock

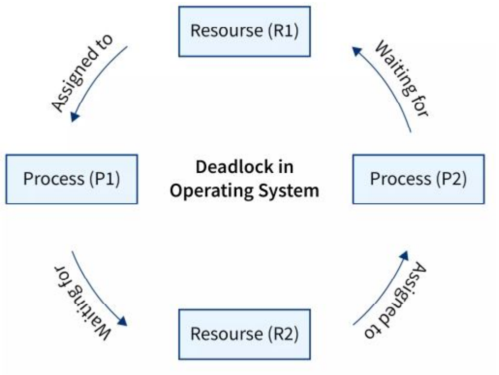
When a transaction must wait an unlimited period for a lock, it is referred to as starvation. The following are the causes of starvation :

1. When the locked item waiting scheme is not correctly controlled.
2. When a resource leak occurs.
3. The same transaction is repeatedly chosen as a victim.

Let's know how starvation can be prevented. Random process selection for resource or processor allocation should be avoided since it encourages hunger. The resource allocation priority scheme should contain ideas like aging, in which a process' priority rises as it waits longer. This prevents starvation.

DeadlockQ

In a circular chain, a deadlock situation occurs when two or more processes are expecting each other to release a resource or when more than 2 processes are waiting for the resource.



Q8. Explain about time stamp ordering protocol.

Ans :

Timestamp Ordering Protocol

- The Timestamp Ordering Protocol is used to order the transactions based on their Timestamps. The order of transaction is nothing but the ascending order of the transaction creation.
- The priority of the older transaction is higher that's why it executes first. To determine the timestamp of the transaction, this protocol uses system time or logical counter.
- The lock-based protocol is used to manage the order between conflicting pairs among transactions at the execution time. But Timestamp based protocols start working as soon as a transaction is created.
- Let's assume there are two transactions T1 and T2. Suppose the transaction T1 has entered the system at 007 times and transaction T2 has entered the system at 009 times. T1 has the higher priority, so it executes first as it is entered the system first.
- The timestamp ordering protocol also maintains the timestamp of last 'read' and 'write' operation on a data.

Basic Timestamp ordering protocol works as follows:

1. Check the following condition whenever a transaction Ti issues a **Read (X)** operation:
   - If W_TS(X) >TS(Ti) then the operation is rejected.
   - If W_TS(X) <= TS(Ti) then the operation is executed.
   - Timestamps of all the data items are updated.
2. Check the following condition whenever a transaction Ti issues a **Write(X)** operation:
   - If TS(Ti) < R_TS(X) then the operation is rejected.
   - If TS(Ti) < W_TS(X) then the operation is rejected and Ti is rolled back otherwise the operation is executed.

Where,

TS(TI) denotes the timestamp of the transaction Ti.

R_TS(X) denotes the Read time-stamp of data-item X.

W_TS(X) denotes the Write time-stamp of data-item X.

Advantages and Disadvantages of to protocol:

- TO protocol ensures serializability since the precedence graph is as follows:
- TS protocol ensures freedom from deadlock that means no transaction ever waits.
- But the schedule may not be recoverable and may not even be cascade free.

4.1.5 Multi-Version and Optimistic Concurrency Control Schemes

Q9. Explain Multi-Version Schemes of Concurrency Control with example.

Ans : **(Imp.)**

Multi-version protocol minimizes the delay for reading operation and maintains different versions of data items. For each writes operation performed, it creates a new version of transaction data so that whenever any transaction performs read operation to read that data then the appropriate created data version is selected by the control manager to make that read operation conflict-free and successful.

When the write operation and new version of data is created then that new version contains some information that is given below

1. **Content:** This field contains the data value of that version.
2. **Write_timestamp:** This field contains the timestamp of the transaction whose new version is created.
3. **Read_timestamp:** This field contains the timestamp of the transaction of that transaction that will read that newly created value.

Now let us understand this concept using an example. Let T1 and T2 be two transactions having timestamp values 15 and 10, respectively.

The transaction T2 calls for a write operation on data (let's say X) from the database. As T2 calls write operation, then a new version of data value X is created, which contains the value of X, timestamp of T2, and timestamp of that transaction which will read X, but in this case, no one is reading the newly created value so that filed remains empty.

X 10

Now let the transaction T1 (having timestamp 15) call a read operation to read the newly created value X, then the newly created variable contained will be

X 10 15

Some important cases

If timestamp of T2 is less than or equal to timestamp of T1 then

1. **Read(X):** operation performed by T1: In this case, content of X is returned to T1.
2. **Write(X):** operation performed by T1: In this case, T1 will be rolled back if timestamp of T1 is smaller than timestamp of read operation on X. And if timestamp of T1 is equal to the timestamp of write operation on X then the new version is created again with new contents.

Q10. What is an optimistic concurrency control in DBMS? Explain.

Ans :

Meaning

All data items are updated at the end of the transaction, at the end, if any data item is found inconsistent with respect to the value in, then the transaction is rolled back.

Check for conflicts at the end of the transaction. No checking while the transaction is executing. Checks are all made at once, so low transaction execution overhead. Updates are not applied until end-transaction. They are applied to local copies in a transaction space.

Phases

The optimistic concurrency control has three phases, which are explained below:

Read Phase

Various data items are read and stored in temporary variables (local copies). All operations are performed in these variables without updating the database.

Validation Phase

All concurrent data items are checked to ensure serializability will not be validated if the transaction are actually applied to the database. Any changes in the value cause the transaction rollback. The transaction timestamps are used and the write-sets and read-sets are maintained.

To check that transaction A does not interfere with transaction B the following must hold:

- TransB completes its write phase before TransA starts the read phase.
- TransA starts its write phase after TransB completes its write phase, and the read set of TransA has no items in common with the write set of TransB.
- Both the read set and write set of TransA have no items in common with the write set of TransB and TransB completes its read before TransA completes its read Phase.

Write Phase

The transaction updates applied to the database if the validation is successful. Otherwise, updates are discarded and transactions are aborted and restarted. It does not use any locks hence deadlock free, however starvation problems of data items may occur.

Problem

S: W1(X), r2(Y), r1(Y), r2(X).

T1 -3

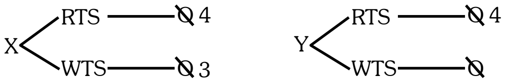
T2 – 4

Check whether timestamp ordering protocols allow schedule S.

Sol:

Initially for a data-item X, RTS(X)=0, WTS(X)=0.

Initially for a data-item Y, RTS(Y)=0, WTS(Y)=0.



For W1(X) : TS(Ti)<RTS(X) i.e.

TS(T1)<RTS(X)

TS(T1)<WTS(X)

3<0 (FALSE)

=>goto else and perform write operation w1(X) and WTS(X)=3

For r2(Y): TS(T2)<WTS(Y)

4<0 (FALSE)

=>goto else and perform read operation r2(Y) and RTS(Y)=4

For r1(Y) :TS(T1)<WTS(Y)

3<0 (FALSE)

=>goto else and perform read operation r1(Y).

For r2(X) : TS(T2)<WTS(X)

4<3 (FALSE)

=>goto else and perform read operation r2(X) and RTS(X)=4.

4.2 Database Recovery Authentication

Q11. What are Recovery Techniques in DBMS? Explain.

Ans :

A DBMS(Database Management System) is used to store, monitor, and manipulate data in a fast and efficient manner. A database has the properties of atomicity, consistency, isolation, and durability. The durability of a system is marked by the ability to preserve the data and changes made to the data. A database may fail due to any of the following reasons,

- System failures are caused due to hardware or software problems in the system.
- Transaction failures occur when a particular process that deals with the modification of data can't be completed.
- Disk crashes may be due to the inability of the system to read the disk.
- Physical damages includes problems like power failure or natural disaster.

Even though the database system fails, the data in the database must be recoverable to the last state before the failure of the system, and the database recovery techniques in DBMS are used to recover the data at such times of system failure. The recovery techniques in DBMS maintain the properties of atomicity and durability of the database. A system is not called durable if it fails during a transaction and loses all its data and a system is not called atomic, if the data is in a partial state of update during the transaction. The data recovery techniques in DBMS make sure, that the state of data is preserved to protect the atomic property and the data is always recoverable to protect the durability property. The following techniques are used to recover data in a DBMS,

- Log-based recovery.
- Recovery through Deferred Update
- Recovery through Immediate Update

The atomicity property of DBMS protects the state of data. If a manipulation is performed in a data, then the manipulation must be performed completely or the state of data must be maintained as if the manipulation never occurred. DBMS failure due to transactions may affect this property and the property is protected by the recovery techniques in DBMS.

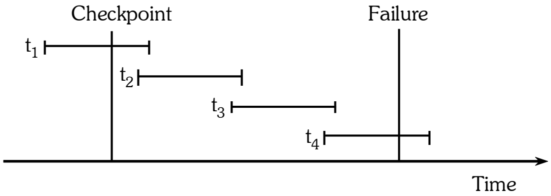
Log-Based Recovery

Any DBMS has its own **system logs** that have the records for all the activity that has occurred in the system along with timestamps on the time of the activity. Databases handle different log files for activities like errors, queries, and other changes in the database. The log is stored in the files in the following formats,

- The structure **[start_transaction, T]** denotes the start of execution of transaction T.
- **[write_item, T, X, old_value, new_value]** shows that the value of the variable, **X** is changed from **old_value** to **new_value** by the transaction **T**.
- **[read_item, T, X]** represents that the value of **X** is read by the transaction T.
- **[commit, T]** indicates the changes in the data are stored in the database through a **commit** and can't be further modified by the transaction. There will be no error after a **commit** has been made to the database.
- **[abort, T]** is used to show that the transaction, **T** is aborted.

We can use these logs to see the change in the state of the data during a transaction and can recover the data to the previous state or new state. A undo operation can be used to examine the [write_item, T, X, old_value, new_value] operation and retrieve the state of data to old_data. A redo operation can be done to convert the old state of data to the new state that was lost due to system failure and is only possible if the [commit, T] operation is performed.

Consider multiple transactions named t1,t2,t3, and t4 as shown in the image below. A checkpoint at a time during the first transaction and the system fails during the fourth transaction, but it is possible to recovery the data to the state of the checkpoint made during t1.



A checkpoint is made after all the records of a transaction are written to logs to transfer all the logs from the local storage to the permanent storage for future use.

Conceded Update Method

In the conceded update method, the updates are not made to the data until the transaction reaches the final phase or at the commit operating. After this operation is performed, the data is modified and permanently stored in the main memory. The logs are maintained throughout the operation and are used in case of failure to find the point of failure. This provides us an advantage as even if the system fails before the commit stage, the data in the database will not be modified and the status will be managed. If the system fails after the commit stage, we can redo the changes to the new stage easily compared to the process involved with the undo operation.

Logging is set up automatically in many databases, but we can also configure them manually. The following steps can be executed in the **MySQL** terminal to set up logging in to a MySQL database,

- Create a variable to store the file path of the log file(**.log**) to which the logs must be stored.

  SET GLOBAL general_log_file='/var/log/mysql/general.log';

- Set the log file format.

  SET GLOBAL log_output = 'FILE';

- The general logging feature of the database should be enabled.

  SET GLOBAL general_log = 'ON';

- The system now monitors all the activities in the database and stores them in the **general.log** file. The configuration for this is maintained in the **general_log_file** variable and can be checked with the following command,

  SHOW VARIABLES LIKE "general_log%";

Quick Update Method

In the quick update method, the update to the data is made concurrently before the transaction reaches the **commit** stage. The logs are also recorded as soon as the changes to the data are made. In the case of failure, the data may be in a partial state of the transaction, and undo operations can be performed to restore the data. We can also mark the state of the transaction and recover our data to the marked state using SQL commands. The following commands are used to achieve this,

- The **SAVEPOINT** command is used to save the current state of data in a transaction. The syntax of this command is,

  SAVEPOINT save_point_name;

- The **ROLLBACK** command is used to restore the state of the data to the save point specified by the command. The syntax of the command is,

  ROLLBACK TO save_point_name;

Q12. What are the differences between a Deferred Update and an Immediate Update?

Ans :

Deferred updates and immediate updates are database recovery techniques in DBMS that are used to maintain the transaction log files of the DBMS.

In the **Deferred** update, the state of the data in the database is not changed immediately after any transaction is executed, but as soon as the commit has been made, the changes are recorded in the log file, and the state of data is changed.

In the immediate update, at every transaction, the database is updated directly, and a log file is also maintained containing the old and new values.

Sl.No.	Deferred Update	Sl.No.	Immediate Update
1.	During a transaction, the changes are not applied immediately to the data occurs.	1.	An immediate change is made in the database as soon as the transaction
2.	The log file holds the changes that are going to be applied.	2.	The log file holds the changes along with the new and old values
3.	Buffering and Caching are used in this technique	3.	Shadow paging is used in this technique
4.	More time is required to recover the data when a system failure occurs	4.	A large number of I/O operations are performed to manage the logs during the transaction
5.	If a rollback is made, the log files are destroyed and no change is made to the database	5.	If a rollback is made, the old state of the data is restored with the records in the log file

Backup Techniques

A backup is a copy of the current state of the database that is stored in another location. This backup is useful in cases when the system is destroyed by natural disasters or physical damage. These backups can be used to recover the database to the state at which the backup was made. Different methods are being used in backup which is as follows,

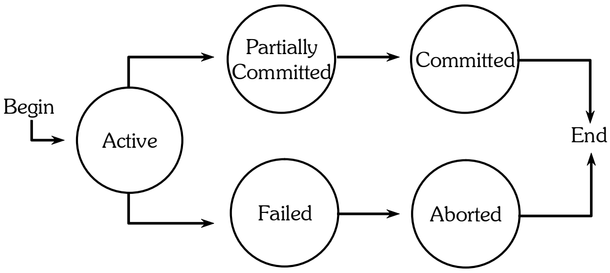
- An Immediate backup are copies that are kept in devices like hard disks or any other dives. When a disk crashes or any technical fault occurs we can use these data to recover the data.
- An Archival backup is a copy of the database kept in cloud environments or large storage systems in another region. These are used to recover the data when the system is affected by a natural disaster.

Transaction Logs

The transaction logs are used to keep track of all the transactions that have made an update to the data in the database. The following steps are followed to recover the data using transaction logs,

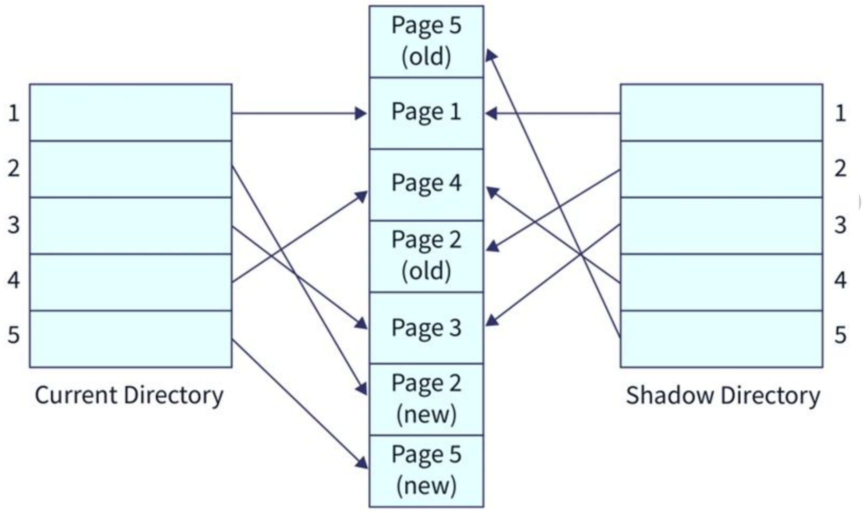
- The recovery manager searches through all the log files and finds the transactions that have a start_transaction stage and doesn't have the commit stage.
- The transactions that are of the above case are rolled back to the old state with the help of the logs using the rollback command
- The transactions that have a commit command will have made changes to the database and these changes are recorded in the logs. These changes will also be reverted using the undo operation.

Transaction State System Log



Shadow Paging

- In shadow paging, a database is split into n- multiple pages that represent a fixed-size disk memory.
- Similarly, n shadow pages are also created which are copies of the real database.
- At the beginning of a transaction, the state in the database is copied to the shadow pages.
- The shadow pages will not be modified during the transaction and only the real database will be modified.
- When the transaction reaches the commit stage, the changes are made to the shadow pages. The changes are made in a way that if the i-th part of the hard disk has modifications, then the i-th shadow page is also modified.
- If there is a failure of the system, the real pages of the database are compared with the shadow pages and recovery operations are performed.



In the Caching/Buffering method, a collection of buffers called DBMS buffers are present in the logical memory. The buffers are used to store all logs during the process and the update to the main log file is made when the transaction reaches the commit stage.

4.3 Authorisation and Access Control

Q13. What is user Authentication? Explain different types of user authentication techniques.

Ans : **(Imp.)**

Meaning

Authentication is the process of identifying users that request access to a system, network, or device. Access control often determines user identity according to credentials like username and password. Other authentication technologies like biometrics and authentication apps are also used to authenticate user identity.

5 Common Authentication Types

Cybercriminals always improve their attacks. As a result, security teams are facing plenty of authentication-related challenges. This is why companies are starting to implement more sophisticated incident response strategies, including authentication as part of the process. The list below reviews some common authentication methods used to secure modern systems.

1. Password-based authentication

Passwords are the most common methods of authentication. Passwords can be in the form of a string of letters, numbers, or special characters. To protect yourself you need to create strong passwords that include a combination of all possible options.

However, passwords are prone to phishing attacks and bad hygiene that weakens effectiveness. An average person has about 25 different online accounts, but only 54% of users use different passwords across their accounts.

The truth is that there are a lot of passwords to remember. As a result, many people choose convenience over security. Most people use simple passwords instead of creating reliable passwords because they are easier to remember.

The bottom line is that passwords have a lot of weaknesses and are not sufficient in protecting online information. Hackers can easily guess user credentials by running through all possible combinations until they find a match.

2. Multi-factor authentication

Multi-Factor Authentication (MFA) is an authentication method that requires two or more independent ways to identify a user. Examples include codes generated from the user's smartphone, Captcha tests, fingerprints, voice biometrics or facial recognition.

MFA authentication methods and technologies increase the confidence of users by adding multiple layers of security. MFA may be a good defense against most account hacks, but it has its own pitfalls. People may lose their phones or SIM cards and not be able to generate an authentication code.

3. Certificate-based authentication

Certificate-based authentication technologies identify users, machines or devices by using digital certificates. A digital certificate is an electronic document based on the idea of a driver's license or a passport.

The certificate contains the digital identity of a user including a public key, and the digital signature of a certification authority. Digital certificates prove the ownership of a public key and issued only by a certification authority.

Users provide their digital certificates when they sign in to a server. The server verifies the credibility of the digital signature and the certificate authority. The server then uses cryptography to confirm that the user has a correct private key associated with the certificate.

4. Biometric authentication

Biometrics authentication is a security process that relies on the unique biological characteristics of an individual. Here are key advantages of using biometric authentication technologies:

- Biological characteristics can be easily compared to authorized features saved in a database.
- Biometric authentication can control physical access when installed on gates and doors.
- You can add biometrics into your multi-factor authentication process.

Biometric authentication technologies are used by consumers, governments and private corporations including airports, military bases, and national borders. The technology is increasingly adopted due to the ability to achieve a high level of security without creating friction for the user. Common biometric authentication methods include:

- **Facial recognition**: Matches the different face characteristics of an individual trying to gain access to an approved face stored in a database. Face recognition can be inconsistent when comparing faces at different angles or comparing people who look similar, like close relatives. Facial liveness like ID R&D's passive facial liveness prevents spoofing.

- **Fingerprint scanners:** match the unique patterns on an individual's fingerprints. Some new versions of fingerprint scanners can even assess the vascular patterns in people's fingers. Fingerprint scanners are currently the most popular biometric technology for everyday consumers, despite their frequent inaccuracies. This popularity can be attributed to iPhones.
- **Speaker Recognition:** also known as voice biometrics, examines a speaker's speech patterns for the formation of specific shapes and sound qualities. A voice-protected device usually relies on standardized words to identify users, just like a password.
- **Eye scanners**: include technologies like iris recognition and retina scanners. Iris scanners project a bright light towards the eye and search for unique patterns in the colored ring around the pupil of the eye. The patterns are then compared to approved information stored in a database. Eye-based authentication may suffer inaccuracies if a person wears glasses or contact lenses.

5. Token-based authentication

Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange. You can then use the token to access protected systems instead of entering your credentials all over again. The digital token proves that you already have access permission. Use cases of token-based authentication include RESTful APIs that are used by multiple frameworks and clients.

Short Question and Answers

1. Concurrency control?

Ans :

Meaning

Concurrency Control is the management procedure that is required for controlling concurrent execution of the operations that take place on a database.

But before knowing about concurrency control, we should know about concurrent execution.

Concurrent Execution in DBMS

- In a multi-user system, multiple users can access and use the same database at one time, which is known as the concurrent execution of the database. It means that the same database is executed simultaneously on a multi-user system by different users.
- While working on the database transactions, there occurs the requirement of using the database by multiple users for performing different operations, and in that case, concurrent execution of the database is performed.

2. Atomicity

Ans :

- It states that all operations of the transaction take place at once if not, the transaction is aborted.
- There is no midway, i.e., the transaction cannot occur partially. Each transaction is treated as one unit and either run to completion or is not executed at all.

Atomicity involves the following two operations:

Abort

If a transaction aborts then all the changes made are not visible.

Commit

If a transaction commits then all the changes made are visible.

3. Serializability

Ans :

Serializability of schedules ensures that a non-serial schedule is equivalent to a serial schedule. It helps in maintaining the transactions to execute simultaneously without interleaving one another. In simple words, serializability is a way to check if the execution of two or more transactions are maintaining the database consistency or not.

4. Non-serial Schedule

Ans :

A schedule in which the transactions are interleaving or interchanging. There are several transactions executing simultaneously as they are being used in performing real-world database operations. These transactions may be working on the same piece of data. Hence, the serializability of non-serial schedules is a major concern so that our database is consistent before and after the execution of the transactions.

5. Types of Serializability

Ans :

Serializability of any non-serial schedule can be verified using two types mainly:

i) Conflict Serializability and View Serializability.

One more way to check serializability is by forming an equivalent serial schedule that results in the same as the original non-serial schedule. Since this process only focuses on the output rather than the operations taking place in between the switching of transactions, it is not practically used. Now let's discuss Conflict and View Serializability in detail.

ii) Conflict Serializability and Conflict Serializable Schedules

A non-serial schedule is a conflict serializable if, after performing some swapping on the non-conflicting operation results in a serial schedule. It is checked using the non-serial schedule and an equivalent serial schedule. This process of checking is called Conflict Serializability.

It is tedious to use if we have many operations and transactions as it requires a lot of swapping.

For checking, we will use the same Precedence Graph technique discussed above. First, we will check conflicting pairs operations(read-write, write-read, and write-write) and then form directed edges between those conflicting pair transactions. If we can find a loop in the graph, then the schedule is non-conflicting serializable otherwise it is surely a conflicting serializable schedule.

6. Shared Lock

Ans :

Shared Locks, which are often denoted as lock-S(), is defined as locks that provide Read-Only access to the information associated with them. Whenever a shared lock is used on a database, it can be read by several users, but these users who are reading the information or the data items will not have permission to edit it or make any changes to the data items.

- To put it another way, we can say that shared locks don't provide access to write. Because numerous users can read the data items simultaneously, multiple shared locks can be installed on them at the same time, but the data item must not have any other locks connected with it.
- A shared lock, also known as a read lock, is solely used to read data objects. Read integrity is supported via shared locks.
- Shared locks can also be used to prevent records from being updated.
- S-lock is requested via the Lock-S instruction.

7. Exclusive Lock

Ans :

- Exclusive Lock allows the data item to be read as well as written. This is a one-time use mode that can't be utilized on the exact data item twice. To obtain X-lock, the user needs to make use of the lock-x instruction. After finishing the 'write' step, transactions can unlock the data item.
- By imposing an X lock on a transaction that needs to update a person's account balance, for example, you can allow it to proceed. As a result of the exclusive lock, the second transaction is unable to read or write.
- The other name for an exclusive lock is write lock.
- At any given time, the exclusive locks can only be owned by one transaction.

8. Lock Compatibility Matrix

Ans :

A vital point to remember when using Lock-based protocols in Database Management System is that a Shared Lock can be held by any amount of transactions. On the other hand, an Exclusive Lock can only be held by one transaction in DBMS, this is because a shared lock only reads data but does not perform any other activities, whereas an exclusive lock performs read as well as writing activities.

The figure given below demonstrates that when two transactions are involved, and both of these transactions seek to read a specific data item, the transaction is authorized, and no conflict occurs; but, in a situation when one transaction intends to write the data item and another transaction attempts to read or write simultaneously, the interaction is rejected.

9. Timestamp Ordering Protocol

Ans :

- The Timestamp Ordering Protocol is used to order the transactions based on their Timestamps. The order of transaction is nothing but the ascending order of the transaction creation.
- The priority of the older transaction is higher that's why it executes first. To determine the timestamp of the transaction, this protocol uses system time or logical counter.
- The lock-based protocol is used to manage the order between conflicting pairs among transactions at the execution time. But Timestamp based protocols start working as soon as a transaction is created.
- Let's assume there are two transactions T1 and T2. Suppose the transaction T1 has entered the system at 007 times and transaction T2 has entered the system at 009 times. T1 has the higher priority, so it executes first as it is entered the system first.
- The timestamp ordering protocol also maintains the timestamp of last 'read' and 'write' operation on a data.

10. What is user Authentication?

Ans :

Authentication is the process of identifying users that request access to a system, network, or device. Access control often determines user identity according to credentials like username and password. Other authentication technologies like biometrics and authentication apps are also used to authenticate user identity.

11. Token-based authentication

Ans :

Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange. You can then use the token to access protected systems instead of entering your credentials all over again. The digital token proves that you already have access permission. Use cases of token-based authentication include RESTful APIs that are used by multiple frameworks and clients.

12. Biometric authentication

Ans :

Biometrics authentication is a security process that relies on the unique biological characteristics of an individual. Here are key advantages of using biometric authentication technologies:

- Biological characteristics can be easily compared to authorized features saved in a database.
- Biometric authentication can control physical access when installed on gates and doors.
- You can add biometrics into your multi-factor authentication process.

Biometric authentication technologies are used by consumers, governments and private corporations including airports, military bases, and national borders. The technology is increasingly adopted due to the ability to achieve a high level of security without creating friction for the user. Common biometric authentication methods include.

Choose the Correct Answers

1. A ________ consists of a sequence of query and/or update statements. [a]

   (a) Transaction (b) Commit

   (c) Rollback (d) Flashback

2. Which of the following makes the transaction permanent in the database? [b]

   (a) View (b) Commit

   (c) Rollback (d) Flashback

3. In order to undo the work of transaction after last commit which one should be used? [c]

   (a) View (b) Commit

   (c) Rollback (d) Flashback

4. Consider the following action: [d]

   What does Rollback do?

   (a) Undoes the transactions before commit

   (b) Clears all transactions

   (c) Redoes the transactions before commit

   (d) No action

5. In case of any shut down during transaction before commit which of the following statement is done automatically? [c]

   (a) View (b) Commit

   (c) Rollback (d) Flashback

6. In order to maintain the consistency during transactions, database provides [b]

   (a) Commit (b) Atomic

   (c) Flashback (d) Retain

7. Transaction processing is associated with everything below except [a]

   (a) Conforming an action or triggering a response

   (b) Producing detail summary or exception report

   (c) Recording a business activity

   (d) Maintaining a data

8. A transaction completes its execution is said to be [a]

   (a) Committed (b) Aborted

   (c) Rolled back (d) Failed

9. Which of the following is used to get back all the transactions back after rollback? [c]

   (a) Commit (b) Rollback

   (c) Flashback (d) Redo

10. ______ will undo all statements up to commit? [c]

    (a) Transaction (b) Flashback

    (c) Rollback (d) Abort

Fill in the blanks

1. ________ Control is the management procedure that is required for controlling concurrent execution of the operations that take place on a database.
2. The ________ protocols ensure the atomicity, consistency, isolation, durability and serializability of the concurrent execution of the database transactions.
3. The ________ constraints are maintained so that the database is consistent before and after the transaction.
4. The ________ property is used to indicate the performance of the database's consistent state. It states that the transaction made the permanent changes.
5. A ________ in which only one transaction is executed at a time, i.e., one transaction is executed completely before starting another transaction.
6. Shared Locks, which are often denoted as ________ .
7. ________ Lock Protocols are known to assess transactions to determine which data elements require locks.
8. ________ protocols allow all transactions to lock the data before inserting, deleting, or updating it.
9. The ________ Protocol is used to order the transactions based on their Timestamps.
10. ________ is the process of identifying users that request access to a system, network, or device.

Answer

1. Concurrency
2. Concurrency control
3. Integrity
4. Durability
5. Schedule
6. Lock-S()
7. Pre-claiming
8. Simple lock-based
9. Timestamp Ordering
10. Authentication

UNIT V

SQL AND PL/SQL CONCEPTS :

Basics of SQL, DDL,DML,DCL, structure-creation, alteration, defining constraints-Primary key, foreign key, unique, not null, check, IN operator, aggregate functions, Built-in functions-numeric, date, string functions, set operations, sub-queries, correlated sub-queries, join, Exist, Any, All, view and its types., transaction control commands

5.1 SQL And PL-SQL Concepts

5.1.1 Basics of SQF

Q1. What is SQL? Explain the importance of SQL.

Ans : **(Imp)**

Meaning

SQL is a short-form of the structured query language, and it is pronounced as S-Q-L or sometimes as See-Quell.

This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables). It is also designed for stream processing in RDSMS.

You can easily create and manipulate the database, access and modify the table rows and columns, etc. This query language became the standard of ANSI in the year of 1986 and ISO in the year of 1987.

If you want to get a job in the field of data science, then it is the most important query language to learn. Big enterprises like Facebook, Instagram, and LinkedIn, use SQL for storing the data in the back-end.

Importance

Nowadays, SQL is widely used in data science and analytics. Following are the reasons which explain why it is widely used:

- The basic use of SQL for data professionals and SQL users is to insert, update, and delete the data from the relational database.
- SQL allows the data professionals and users to retrieve the data from the relational database management systems.
- It also helps them to describe the structured data.
- It allows SQL users to create, drop, and manipulate the database and its tables.
- It also helps in creating the view, stored procedure, and functions in the relational database.
- It allows you to define the data and modify that stored data in the relational database.
- It also allows SQL users to set the permissions or constraints on table columns, views, and stored procedures.

Q2. Explain the History of SQL.

Ans :

"A Relational Model of Data for Large Shared Data Banks" was a paper which was published by the great computer scientist "E.F. Codd" in 1970.

The IBM researchers Raymond Boyce and Donald Chamberlin originally developed the SEQUEL (Structured English Query Language) after learning from the paper given by E.F. Codd. They both developed the SQL at the San Jose Research laboratory of IBM Corporation in 1970.

At the end of the 1970s, relational software Inc. developed their own first SQL using the concepts of E.F. Codd, Raymond Boyce, and Donald Chamberlin. This SQL was totally based on RDBMS. Relational Software Inc., which is now known as Oracle Corporation, introduced the Oracle V2 in June 1979, which is the first implementation of SQL language. This Oracle V2 version operates on VAX computers.

Q3. Explain the Process of SQL.

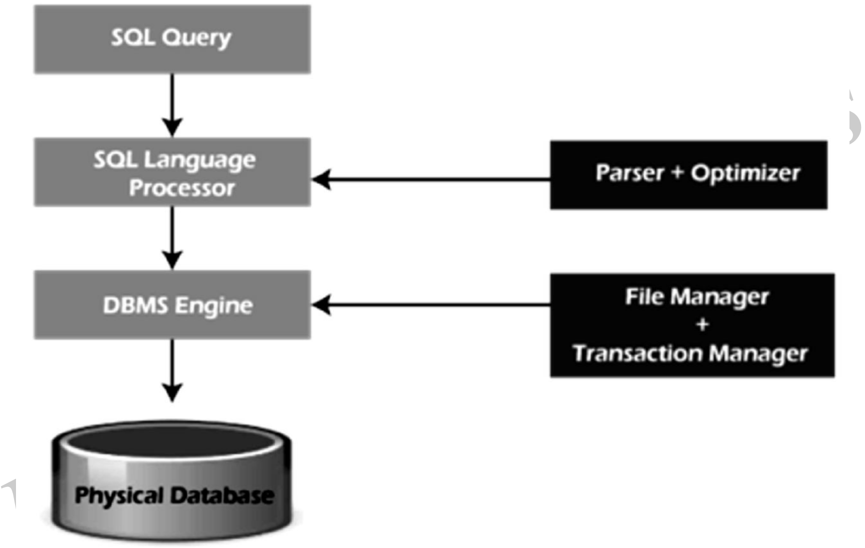
Ans :

When we are executing the command of SQL on any Relational database management system, then the system automatically finds the best routine to carry out our request, and the SQL engine determines how to interpret that particular command.

Structured Query Language contains the following four components in its process:

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine allows data professionals and users to maintain non-SQL queries. The architecture of SQL is shown in the following diagram:



Some SQL Commands

The SQL commands help in creating and managing the database. The most common SQL commands which are highly used are mentioned below:

1. **CREATE Command**

   This command helps in creating the new database, new table, table view, and other objects of the database.

2. **UPDATE Command**

   This command helps in updating or changing the stored data in the database.

3. **DELETE Command**

   This command helps in removing or erasing the saved records from the database tables. It erases single or multiple tuples from the tables of the database.

4. **SELECT Command**

   This command helps in accessing the single or multiple rows from one or multiple tables of the database. We can also use this command with the WHERE clause.

5. **DROP Command**

This command helps in deleting the entire table, table view, and other objects from the database.

6. **INSERT Command**

This command helps in inserting the data or records into the database tables. We can easily insert the records in single as well as multiple rows of the table.

Q4. Explain the advantages of SQL disadvantages of SQL.

Ans :

Advantages

SQL provides various advantages which make it more popular in the field of data science. It is a perfect query language which allows data professionals and users to communicate with the database. Following are the best advantages or benefits of Structured Query Language:

1. **No programming needed**

SQL does not require a large number of coding lines for managing the database systems. We can easily access and maintain the database by using simple SQL syntactical rules. These simple rules make the SQL user-friendly.

2. **High-Speed Query Processing**

A large amount of data is accessed quickly and efficiently from the database by using SQL queries. Insertion, deletion, and updation operations on data are also performed in less time.

3. **Standardized Language**

SQL follows the long-established standards of ISO and ANSI, which offer a uniform platform across the globe to all its users.

4. **Portability**

The structured query language can be easily used in desktop computers, laptops, tablets, and even smartphones. It can also be used with other applications according to the user's requirements.

5. **Interactive language**

We can easily learn and understand the SQL language. We can also use this language for communicating with the database because it is a simple query language. This language is also used for receiving the answers to complex queries in a few seconds.

6. **More than one Data View**

The SQL language also helps in making the multiple views of the database structure for the different database users.

Disadvantages

With the advantages of SQL, it also has some disadvantages, which are as follows:

1. **Cost**

The operation cost of some SQL versions is high. That's why some programmers cannot use the Structured Query Language.

2. **Interface is Complex**

Another big disadvantage is that the interface of Structured query language is difficult, which makes it difficult for SQL users to use and manage it.

3. **Partial Database control**

The business rules are hidden. So, the data professionals and users who are using this query language cannot have full database control.

5.1.2 DDL

Q5. Explain various DDL commands with an examples and syntax.

Ans : **(Imp.)**

The DDL Commands in Structured Query Language are used to create and modify the schema of the database and its objects. The syntax of DDL commands is predefined for describing the data. The commands of Data Definition Language deal with how the data should exist in the database.

Following are the five DDL commands in SQL:

1. CREATE Command
2. DROP Command
3. ALTER Command
4. TRUNCATE Command
5. RENAME Command

1. CREATE Command

CREATE is a DDL command used to create databases, tables, triggers and other database object Examples of CREATE Command in SQL

Example 1:

This example describes how to create a new database using the CREATE DDL command.

Syntax to Create a Database:

CREATE Database Database_Name;

Suppose, you want to create a Books database in the SQL database. To do this, you have to write the following DDL Command:

Create Database Books;

Example 2:

This example describes how to create a new table using the CREATE DDL command.

Syntax to create a new table:

```
CREATE TABLE table_name
(
column_Name1 data_type ( size of the column );
column_Name2 data_type ( size of the column);
column_Name3 data_type ( size of the column);
...
column_NameN data_type ( size of the column )
) ;
```

Suppose, you want to create a Student table with five columns in the SQL database. To do this, you have to write the following DDL command:

```
CREATE TABLE Student
(
Roll_No. Int,
First_Name Varchar (20),
Last_Name Varchar (20),
Age Int,
Marks Int,
) ;
```

Example 3:

This example describes how to create a new index using the CREATE DDL command.

Syntax to Create a new index:

```
CREATE INDEX Name_of_Index ON Name_of_Table (column_name_1, column_name_2
, … . , column_name_N);
```

Let's take the Student table:

Stu_Id	Name	Marks	City	State
100	Abhay	80	Noida	U.P
101	Sushil	75	Jaipur	Rajasthan
102	Ankit	90	Gurgaon	Haryana
103	Yogesh	93	Lucknow	U.P

Suppose, you want to create an index on the combination of the City and State field of the Student table. For this, we have to use the following DDL command:

```
CREATE INDEX index_city_State ON Employee (Emp_City, Emp_State);
```

Example 4:

This example describes how to create a trigger in the SQL database using the DDL CREATE command.

Syntax to create a trigger:

```
CREATE TRIGGER [trigger_name]
[ BEFORE | AFTER ]
{ INSERT | UPDATE | DELETE }
ON [table_name];
```

2. DROP Command

DROP is a DDL command used to delete/remove the database objects from the SQL database. We can easily remove the entire table, view, or index from the database using this DDL command.

Examples of DROP Command in SQL

Example 1:

This example describes how to remove a database from the SQL database.

Syntax to remove a database:

```
DROP DATABASE Database_Name;
```

Suppose, you want to delete the Books database from the SQL database. To do this, you have to write the following DDL command:

```
DROP DATABASE Books;
```

Example 2:

This example describes how to remove the existing table from the SQL database.

Syntax to remove a table:

DROP TABLE Table_Name;

Suppose, you want to delete the Student table from the SQL database. To do this, you have to write the following DDL command:

DROP TABLE Student;

Example 3:

This example describes how to remove the existing index from the SQL database.

Syntax to remove an index:

DROP INDEX Index_Name;

Suppose, you want to delete the index_city from the SQL database. To do this, you have to write the following DDL command:

DROP INDEX Index_city;

3. ALTER Command

ALTER is a DDL command which changes or modifies the existing structure of the database, and it also changes the schema of database objects.

We can also add and drop constraints of the table using the ALTER command.

Examples of ALTER Command in SQL

Example 1:

This example shows how to add a new field to the existing table.

Syntax to add a newfield in the table:

ALTER TABLE name_of_table ADD column_name column_definition;

Suppose, you want to add the 'Father's_Name' column in the existing Student table. To do this, you have to write the following DDL command:

ALTER TABLE Student ADD Father's_Name Varchar(60);

Example 2:

This example describes how to remove the existing column from the table.

Syntax to remove a column from the table:

ALTER TABLE name_of_table DROP Column_Name_1 , column_Name_2 , …..,

column_Name_N;

Suppose, you want to remove the Age and Marks column from the existing Student table. To do this, you have to write the following DDL command:

ALTER TABLE StudentDROP Age, Marks;

Example 3:

This example describes how to modify the existing column of the existing table.

Syntax to modify the column of the table:

```
ALTER TABLE table_name MODIFY ( column_name column_datatype(size));
```

Suppose, you want to change the character size of the Last_Namefield of the Student table. To do this, you have to write the following DDL command:

```
ALTER TABLE table_name MODIFY ( Last_Name varchar(25));
```

4. TRUNCATE Command

TRUNCATE is another DDL command which deletes or removes all the records from the table.

This command also removes the space allocated for storing the table records.

Syntax of TRUNCATE command

```
TRUNCATE TABLE Table_Name;
```

Example

Suppose, you want to delete the record of the Student table. To do this, you have to write the following TRUNCATE DDL command:

```
TRUNCATE TABLE Student;
```

The above query successfully removed all the records from the student table. Let's verify it by using the following SELECT statement:

```
SELECT * FROM Student;
```

5. RENAME Command

RENAME is a DDL command which is used to change the name of the database table.

Syntax of RENAME command

```
RENAME TABLE Old_Table_Name TO New_Table_Name;
```

Example

```
RENAME TABLE Student TO Student_Details ;
```

This query changes the name of the table from Student to Student_Details.

5.1.3 DML

Q6. What are various DML commands ? Explain select command with syntax and example.

Ans : **(Imp.)**

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

Following are the four main DML commands in SQL:

1. SELECT Command
2. INSERT Command
3. UPDATE Command
4. DELETE Command

1. SELECT Command

The SQL SELECT statement is used to select (retrieve) data from a database table.

For example,

SELECT first_name, last_name

FROM Customers;

Here, the SQL command selects the *first_name* and *last_name* of all *Customers*.

Example: SQL SELECT

SQL SELECT ALL

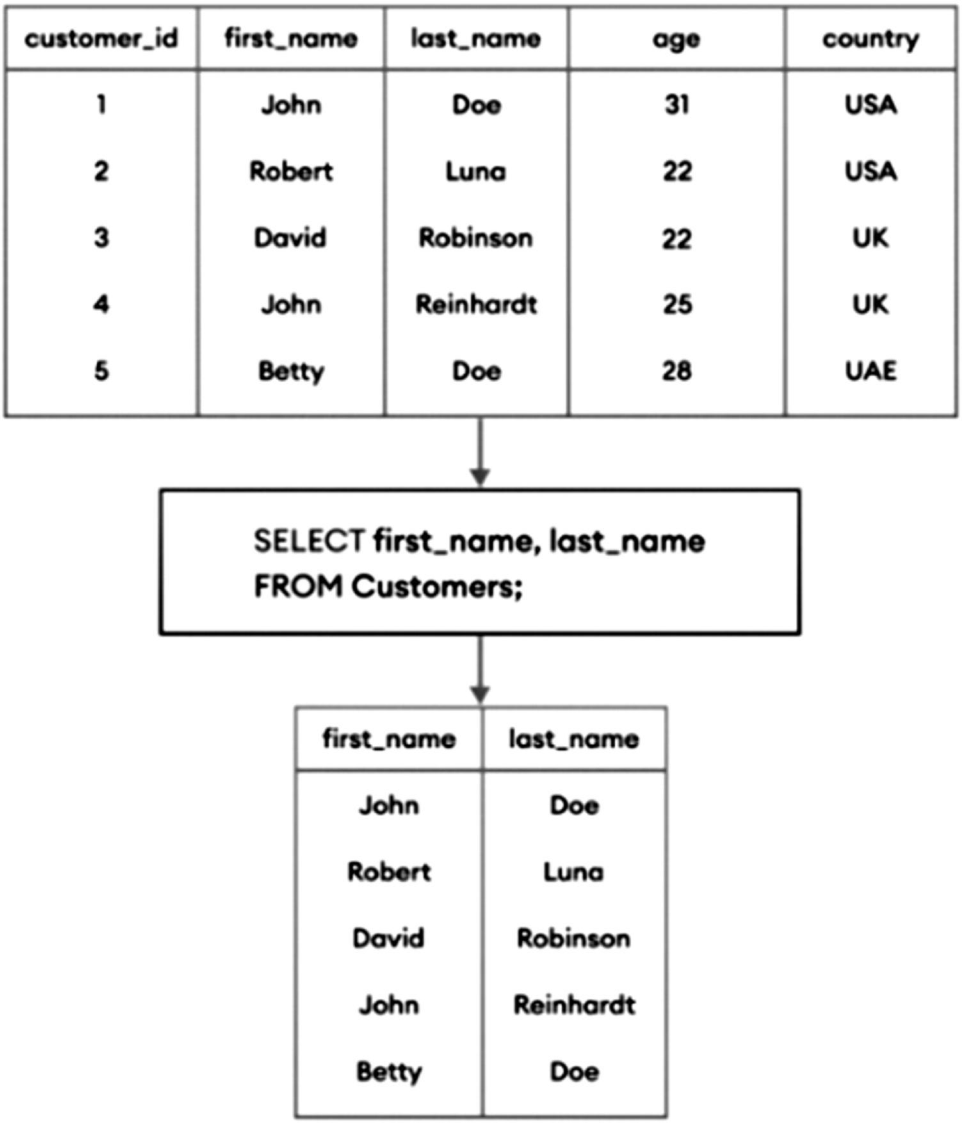
To select all columns from a database table, we use the * character. For example,

SELECT *

FROM Customers;

Here, the SQL command selects all columns of the Customers table.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, last_name
FROM Customers;
```

first_name	last_name
John	Doe
Robert	Luna
David	Robinson
John	Reinhardt
Betty	Doe

Example:

SQL SELECT All

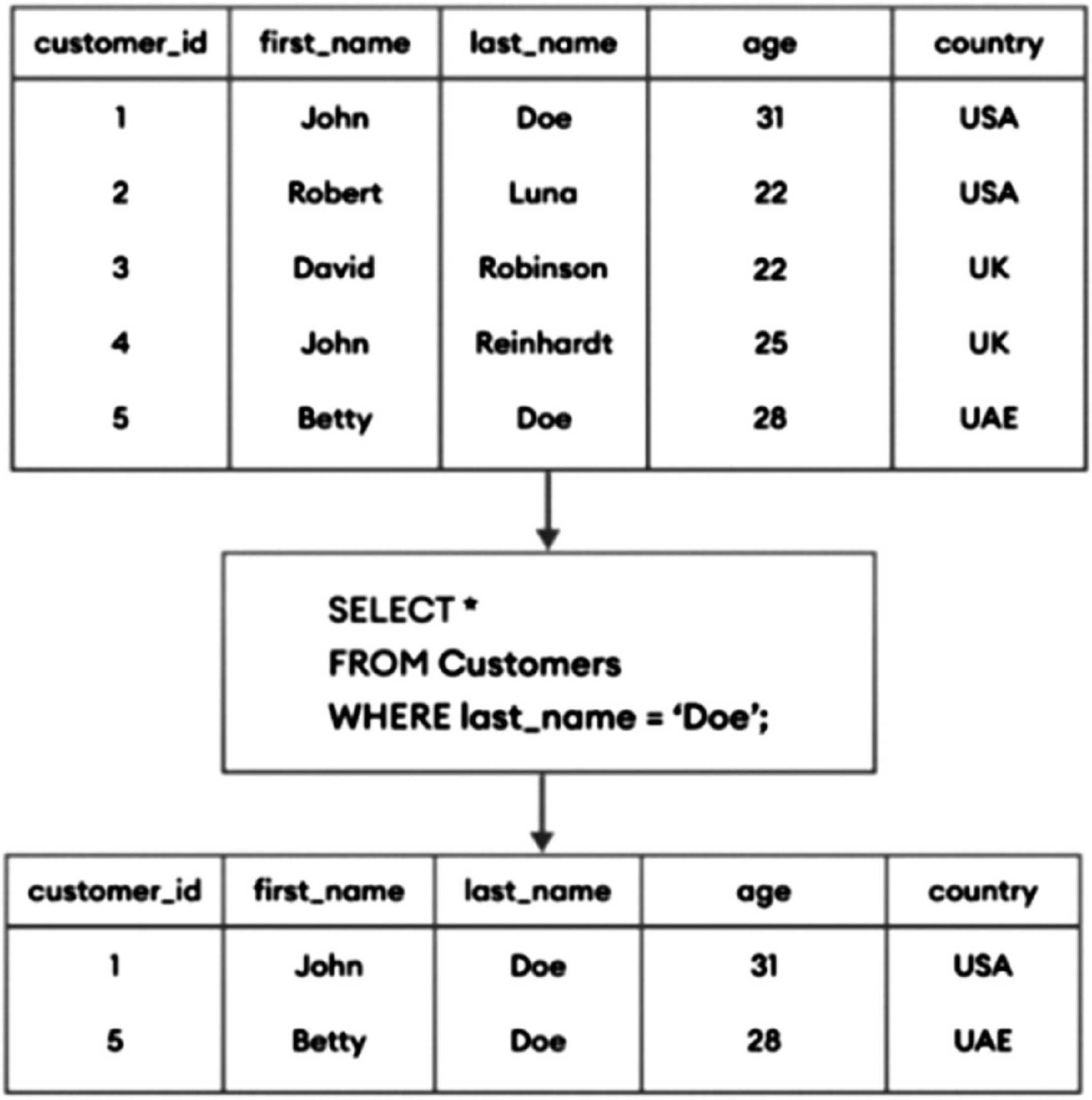
SQL SELECT WHERE Clause

A SELECT statement can have an optional WHERE clause. The WHERE clause allows us to fetch records from a database table that matches specified condition(s). For example,

```
SELECT *
FROM Customers
WHERE last_name = 'Doe';
```

Here, the SQL command selects all customers from the Customers table with last_name Doe.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers
WHERE last_name = 'Doe';
```

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
5	Betty	Doe	28	UAE

Example:

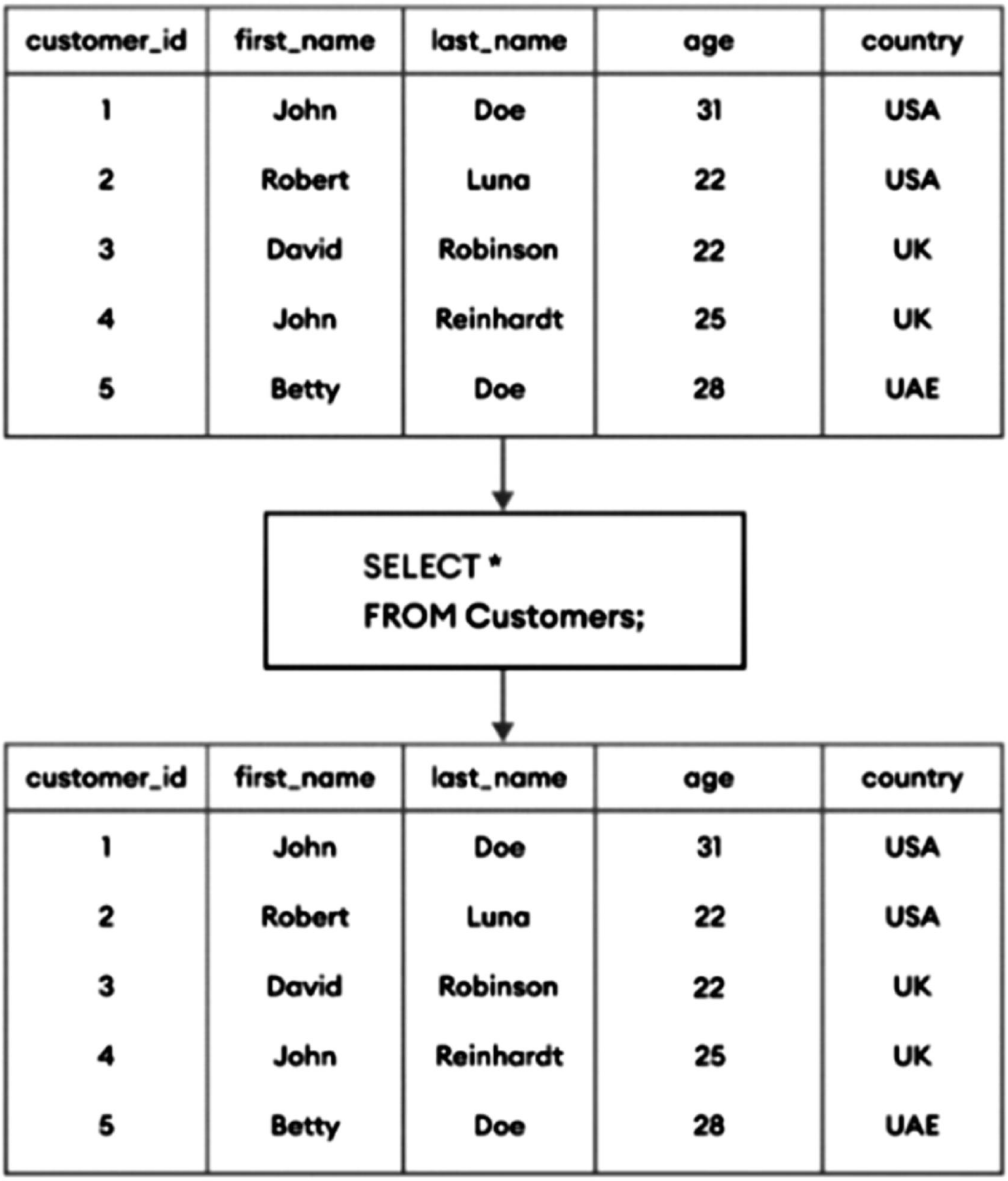
SQL SELECT with WHERE

Let’s see another example.

```
SELECT age, country
FROM Customers
WHERE country = ‘USA’;
```

Here, the SQL command fetches age and country fields of all customers whose country is USA.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers;
```

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Example:

SQL SELECT with WHERE

SQL Operators

The WHERE clause uses operators to construct conditions. Some of the commonly used operators are:

1. Equal to Operator (=)

```
SELECT *
FROM Customers
WHERE first_name = 'John';
```

RThis SQL command selects all customers from the Customers table having first_name John.

2. Greater than (>)

```
SELECT *
FROM Customers
WHERE age >25;
```

This SQL command selects all customers from the Customers table having age greater than 25.

3. AND Operator (AND)

```
SELECT *
FROM Customers
WHERE last_name = 'Doe'AND country = 'USA';
```

This SQL command selects all customers from the Customers table

having last_name Doe and country USA.

SQL AND Operator

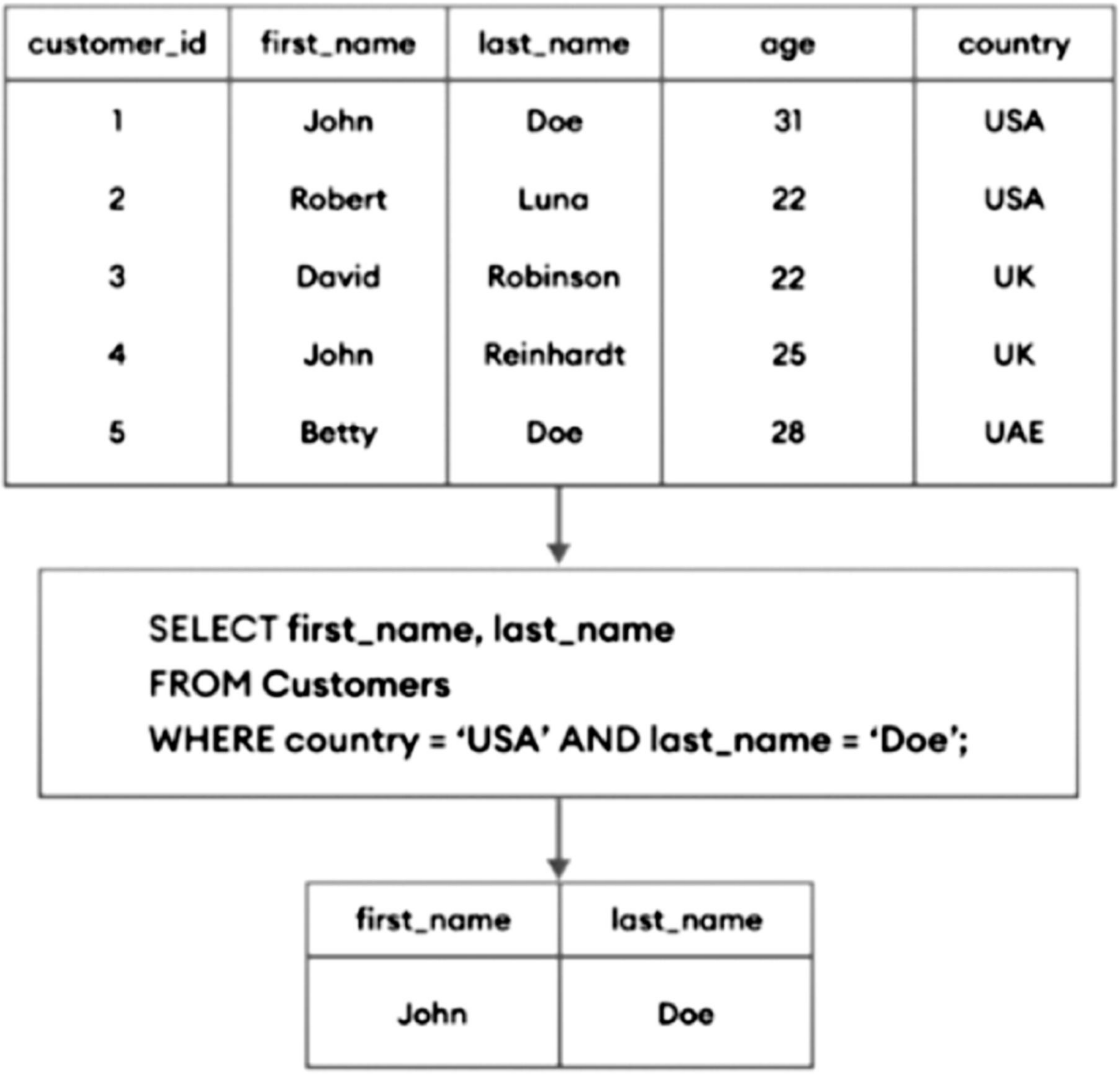
The SQL AND operator selects data if all conditions are TRUE. For example,

```
SELECT first_name, last_name
FROM Customers
WHERE country = 'USA'AND last_name = 'Doe';
```

Here, the SQL command selects first_name and last_name of all customers where the country is USA and last_name as Doe from the Customers table.

Here, the SQL command selects first_name and last_name of all customers where the country is USA and last_name as Doe from the Customers table.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, last_name
FROM Customers
WHERE country = 'USA' AND last_name = 'Doe';
```

first_name	last_name
John	Doe

Example:

SQL AND Operator

SQL OR Operator

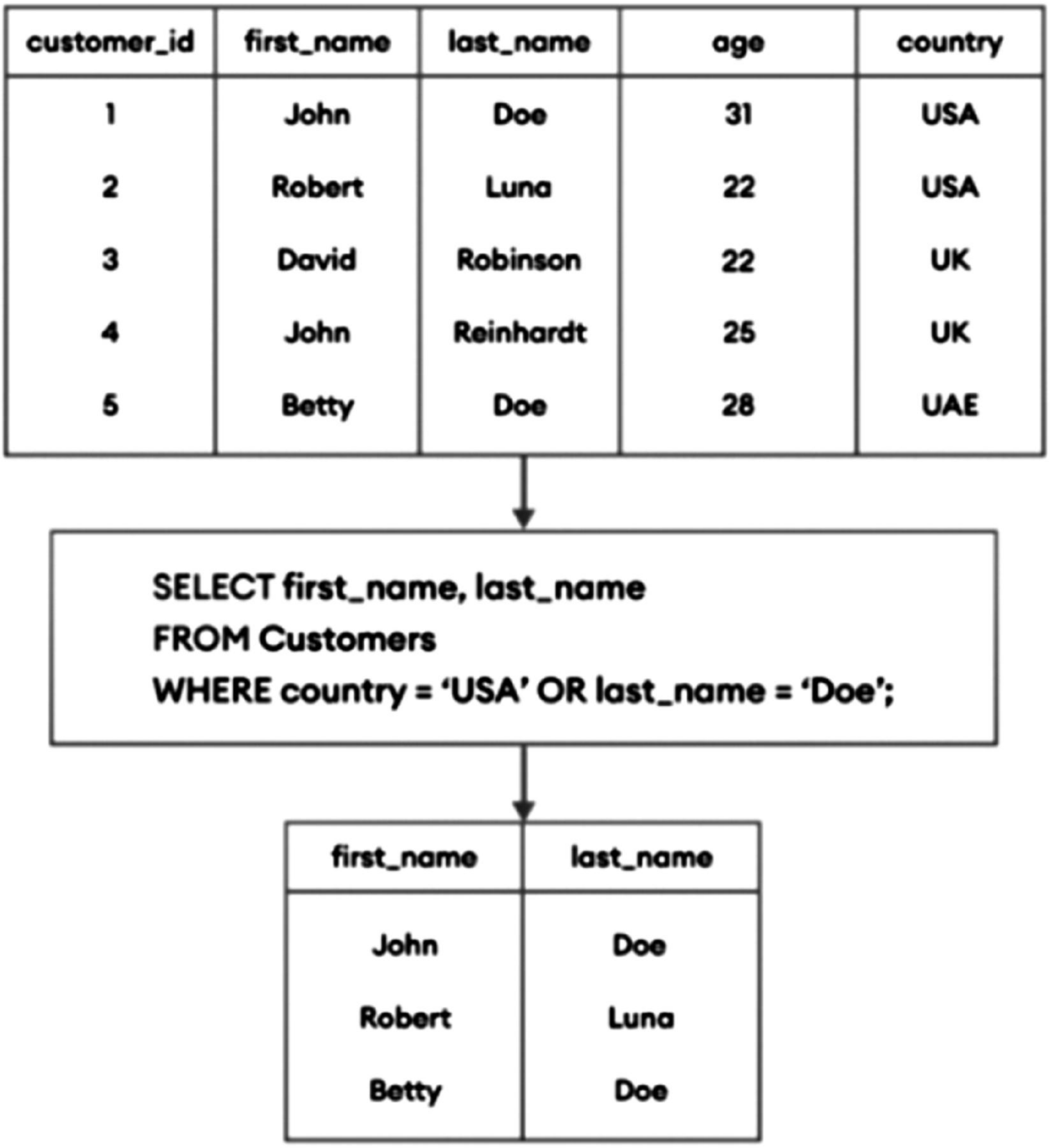
The SQL OR operator selects data if any one condition is TRUE. For example,

SELECT first_name, last_name

FROM Customers

WHERE country = 'USA'OR last_name = 'Doe';

Here, the SQL command selects first_name and last_name of all customers where the country is USA or if their last name is Doe from the Customers table.



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, last_name
FROM Customers
WHERE country = 'USA' OR last_name = 'Doe';
```

first_name	last_name
John	Doe
Robert	Luna
Betty	Doe

Example:

SQL OR Operator

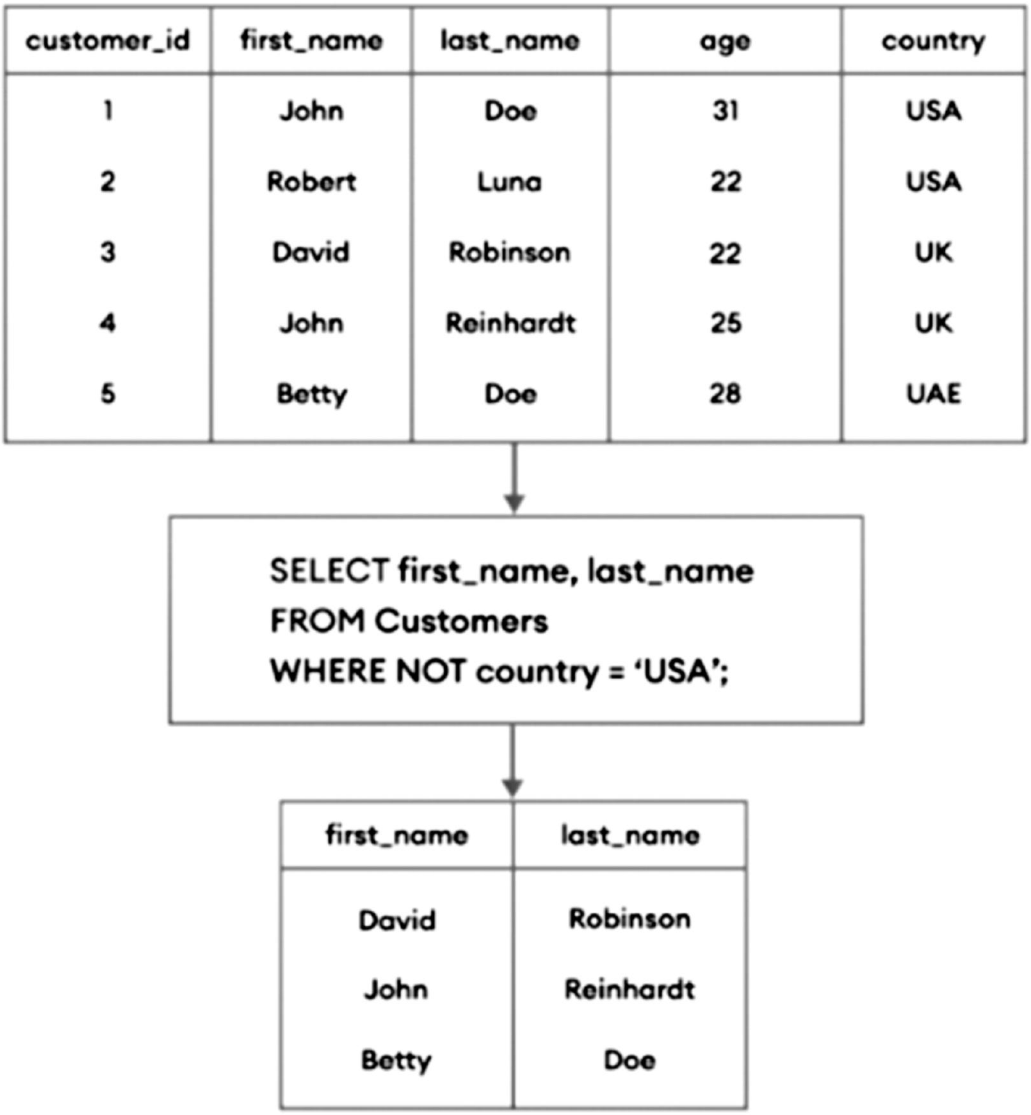
SQL NOT Operator

The SQL NOT operator selects data if the given condition is FALSE. For example

```
SELECT first_name, last_name
FROM Customers
WHERENOT country = 'USA';
```

Here, the SQL command selects first_name and last_name of all customers where the country is not USA from the Customers table.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, last_name
FROM Customers
WHERE NOT country = 'USA';
```

first_name	last_name
David	Robinson
John	Reinhardt
Betty	Doe

Example:

SQL NOT Operator

Combining Multiple Operators

It is also possible to combine multiple AND, OR and NOT operators in an SQL statement. For example,

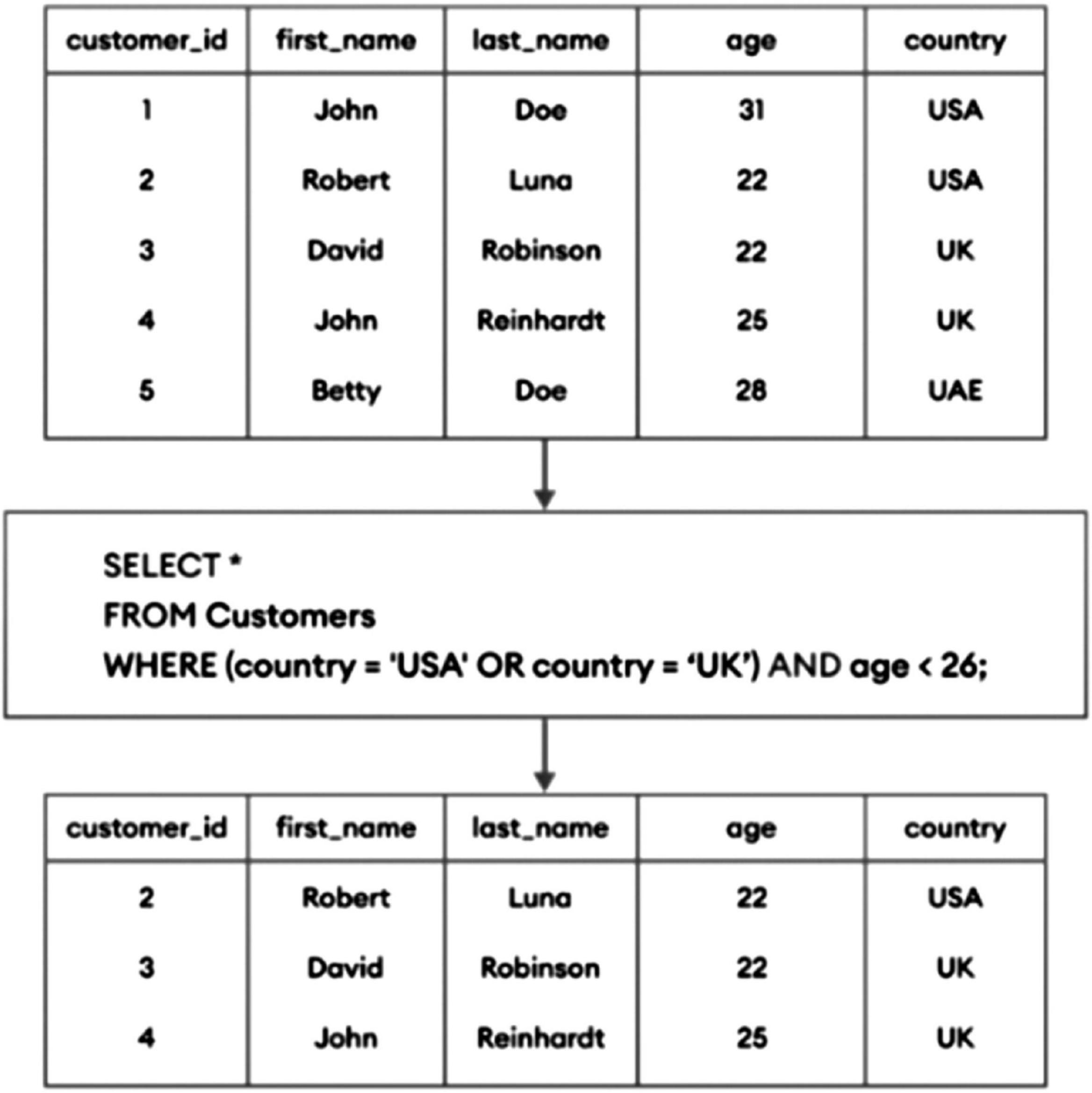
Let's suppose we want to select customers where the country is either USA or UK, and the age is less than 26.

SELECT *

FROM Customers

WHERE (country = 'USA'OR country = 'UK') AND age <26;

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers
WHERE (country = 'USA' OR country = 'UK') AND age < 26;
```

customer_id	first_name	last_name	age	country
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

Example:

SQL AND and OR Operators

Let's take a look at another example

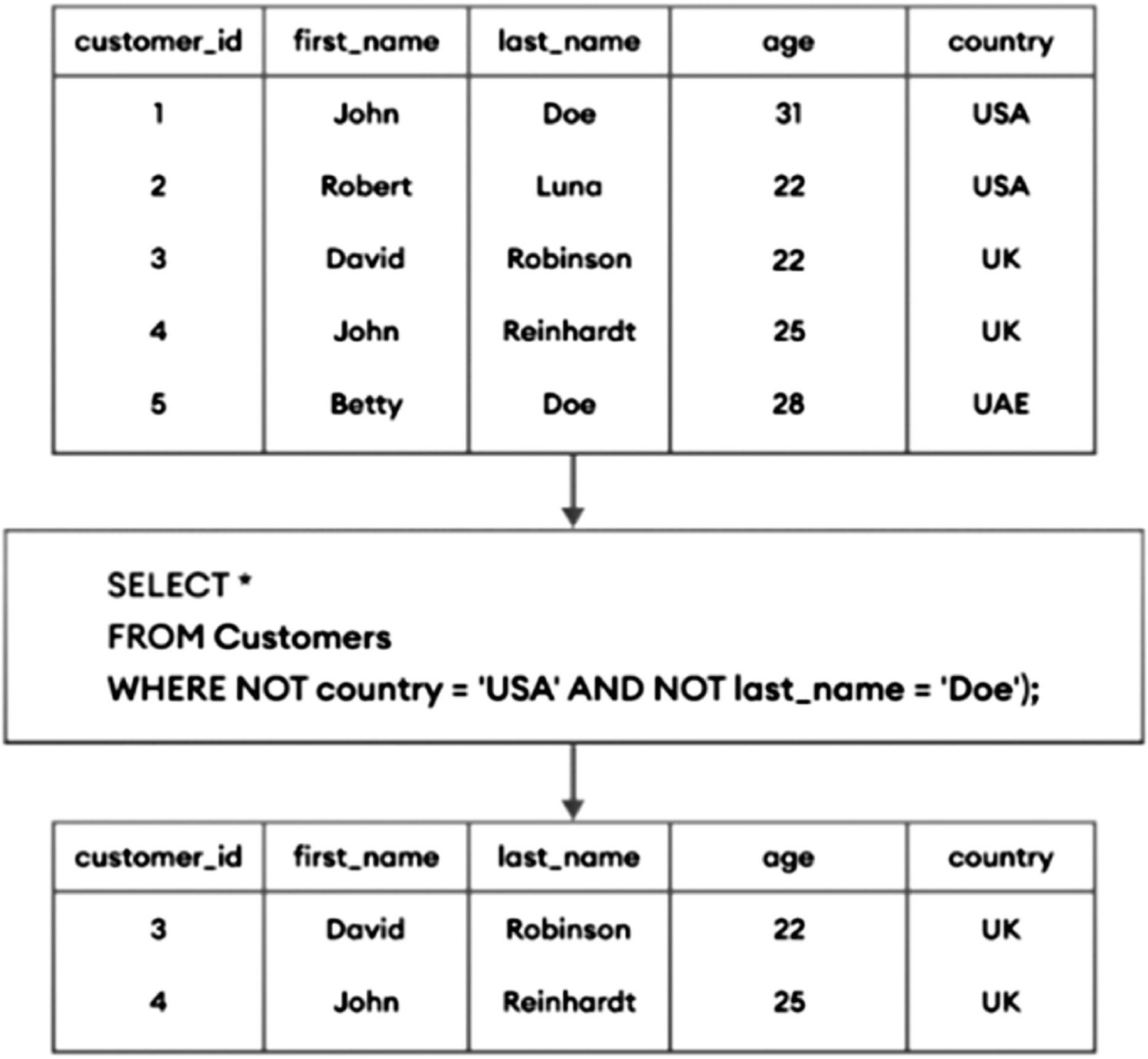
SELECT *

FROM Customers

WHERENOT country = 'USA'ANDNOT last_name = 'Doe';

Here, the SQL command selects all customers where the country is not USA and last_name is not Doe from the Customers table.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers
WHERE NOT country = 'USA' AND NOT last_name = 'Doe');
```

customer_id	first_name	last_name	age	country
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

Example:

SQL AND and NOT Operators

SQL SELECT DISTINCT Statement

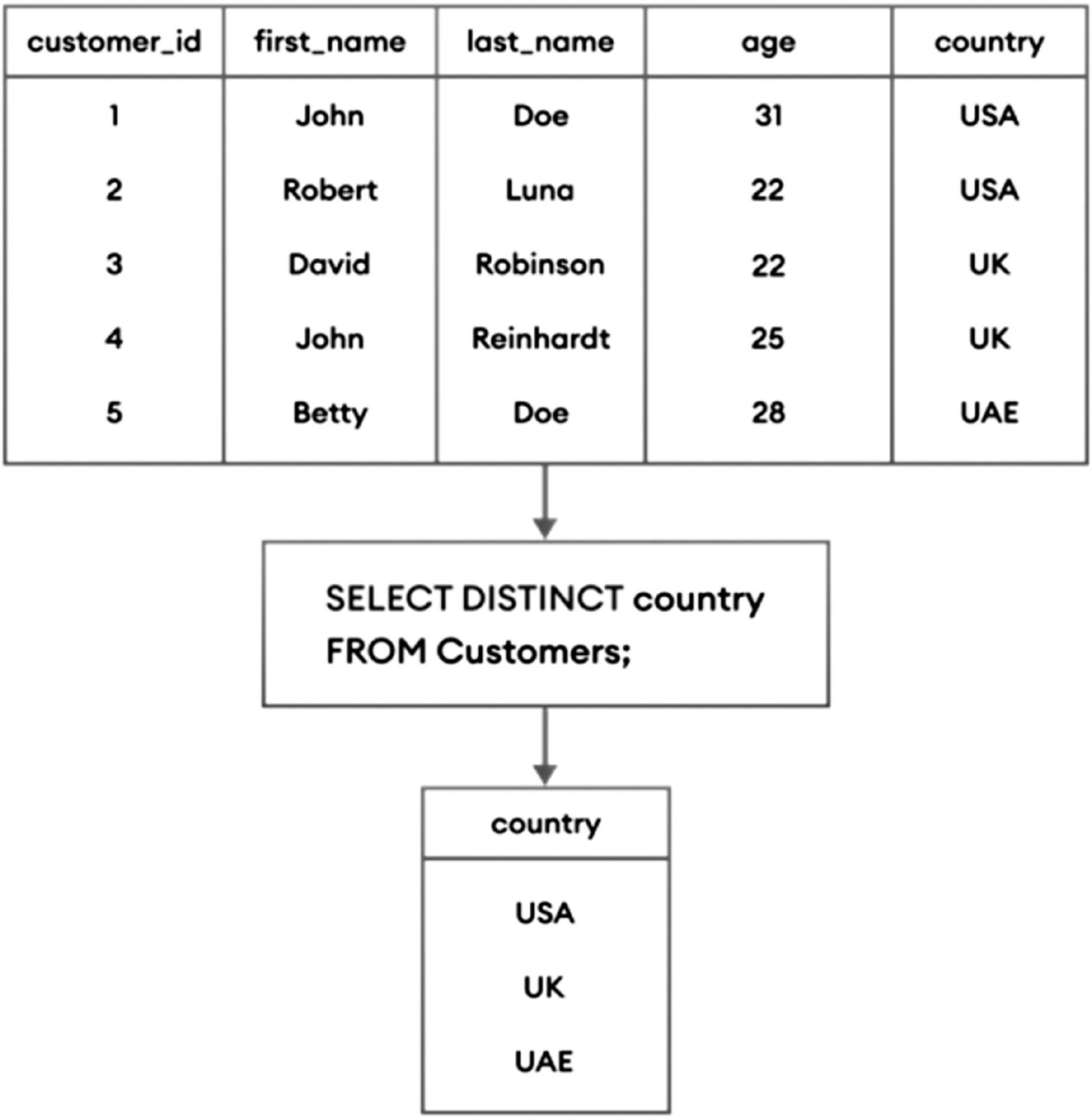
In this tutorial, you'll learn about the SQL DISTINCT clause and how to use it with the help of various examples.

The SQL SELECT DISTINCT statement selects unique rows from a database table. For example,

```
SELECTDISTINCT country
FROM Customers;
```

Here, the SQL command selects unique countries from the Customers table.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT DISTINCT country
FROM Customers;
```

country
USA
UK
UAE

Example:

Selecting unique countries

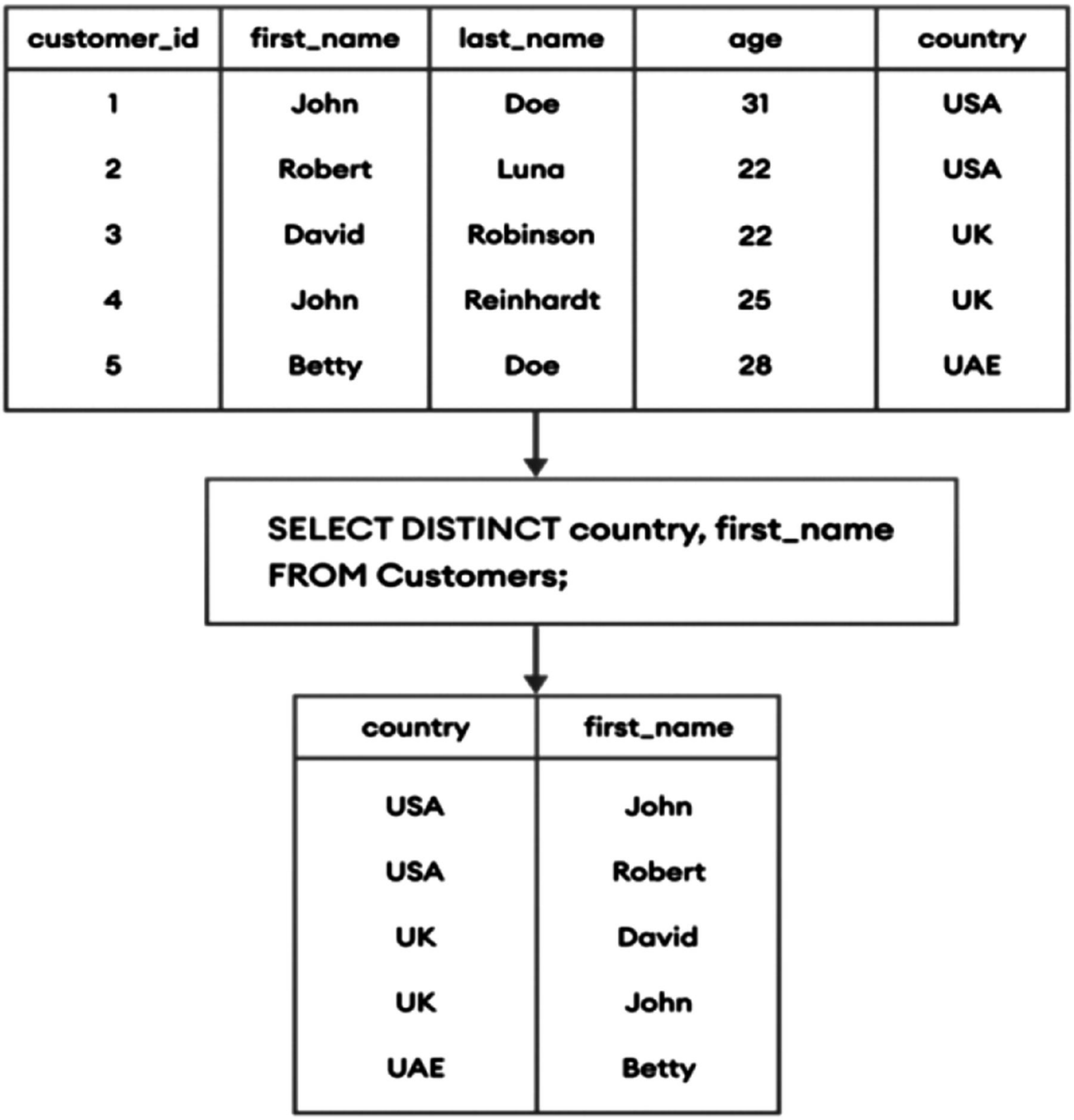
Let's see another example.

SELECTDISTINCT country, first_name

FROM Customers;

Here, the SQL command selects rows if the combination of country and first_name is unique.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT DISTINCT country, first_name
FROM Customers;
```

country	first_name
USA	John
USA	Robert
UK	David
UK	John
UAE	Betty

Example:

Selecting the unique combined fields

DISTINCT with COUNT

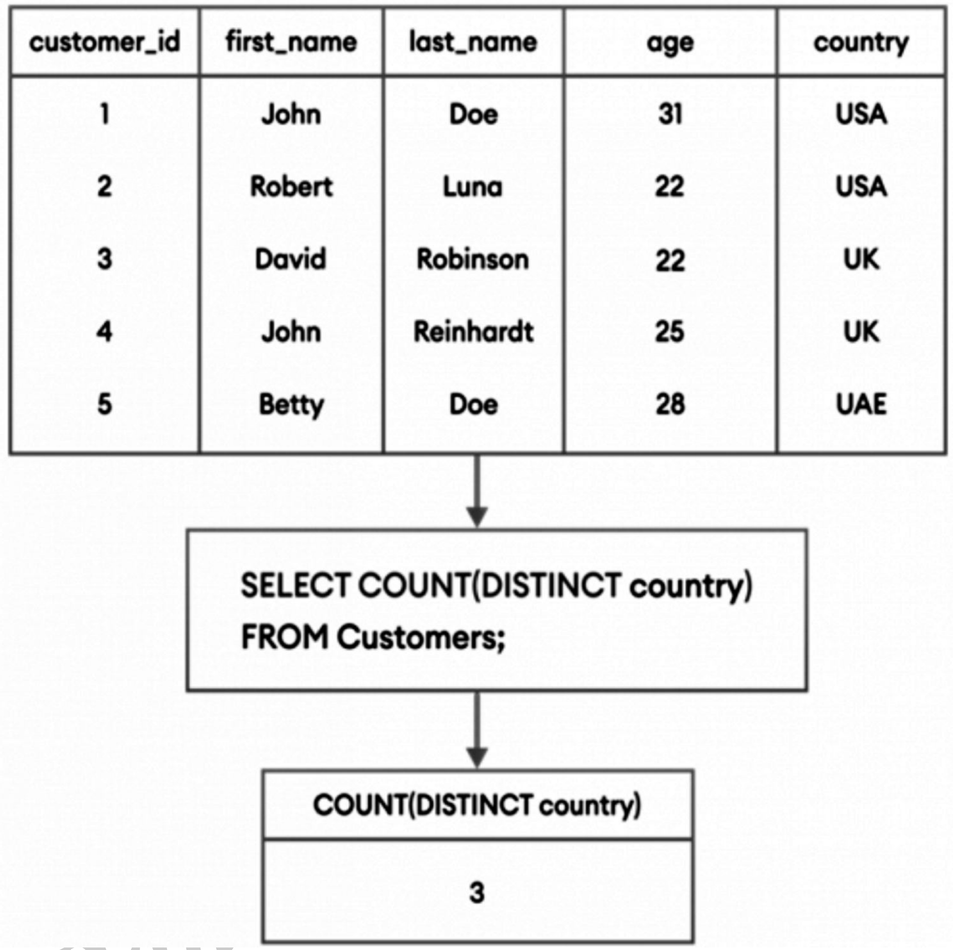
If we need to count the number of unique rows, we can use the COUNT() function with DISTINCT.

SELECTCOUNT(DISTINCT country)

FROM Customers;

Here, the SQL command returns the count of unique countries.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT COUNT(DISTINCT country)
FROM Customers;
```

COUNT(DISTINCT country)
3

Example:

Counting unique countries

Q7. Explain insert statement with syntax and example.

Ans : **(Imp.)**

SQL INSERT STATEMENT

SQL INSERT statement is a SQL query. It is used to insert a single or a multiple records in a table.

There are two ways to insert data in a table:

1. By SQL insert into statement
   1. By specifying column names
   2. Without specifying column names
2. By SQL insert into select statement

1. Inserting data directly into a table

You can insert a row in the table by using SQL INSERT INTO command.

In SQL, the INSERT INTO statement is used to insert new row(s) in a database table. For example,

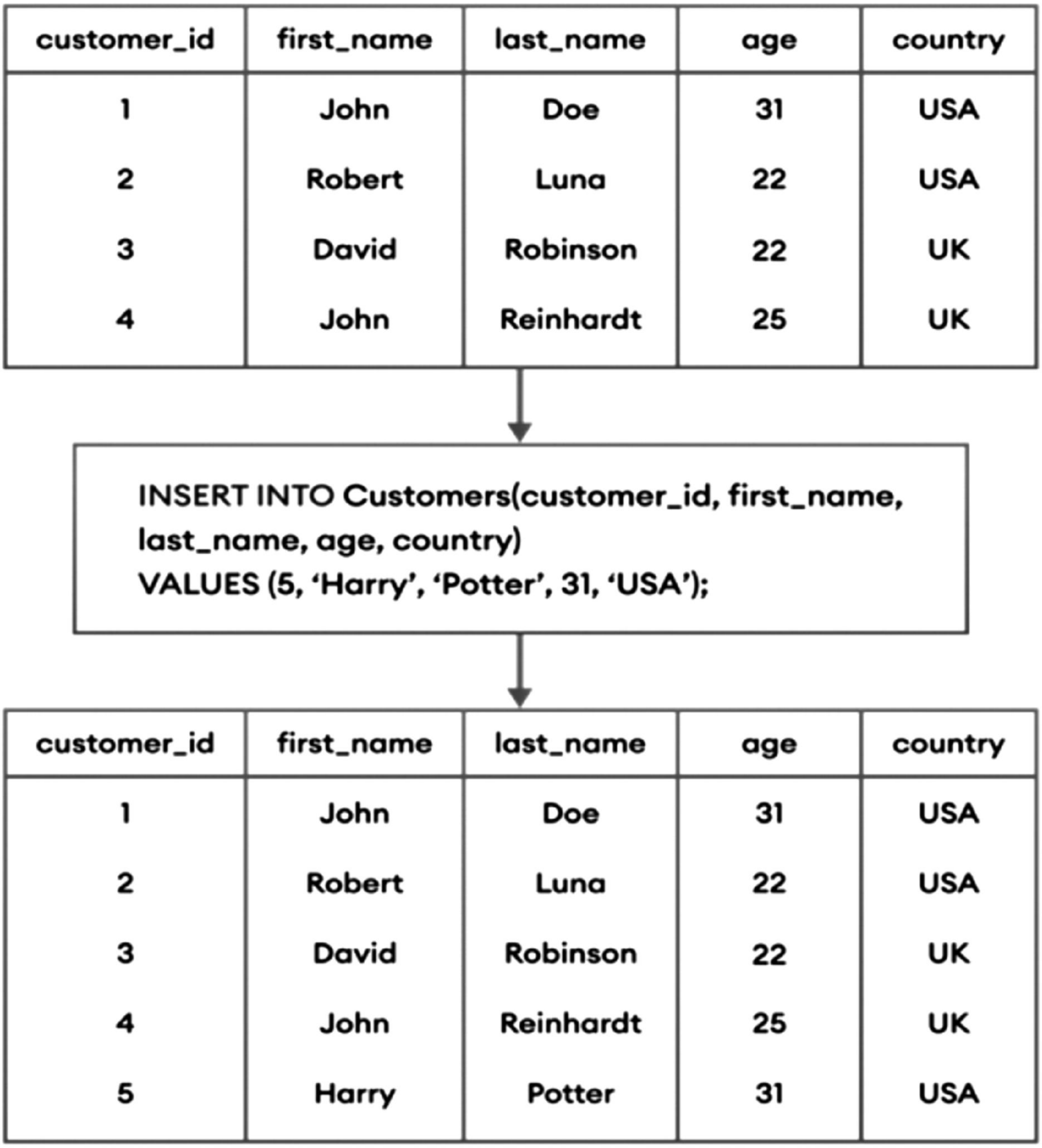
INSERTINTOCustomers(customer_id, first_name, last_name, age, country)

VALUES

(5, 'Harry', 'Potter', 31, 'USA');

Here, the SQL command inserts a new row in the Customers table with the given values.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

```
INSERT INTO Customers(customer_id, first_name,
last_name, age, country)
VALUES (5, 'Harry', 'Potter', 31, 'USA');
```

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Harry	Potter	31	USA

Example:

SQL Insert Into

Insert Row Providing Value Explicitly

It's possible to provide default values to a column (for example, auto incrementing a column). In a database table, the ID field is usually unique auto incremented.

In such cases, we can omit the value for that column during row insertion. For example,

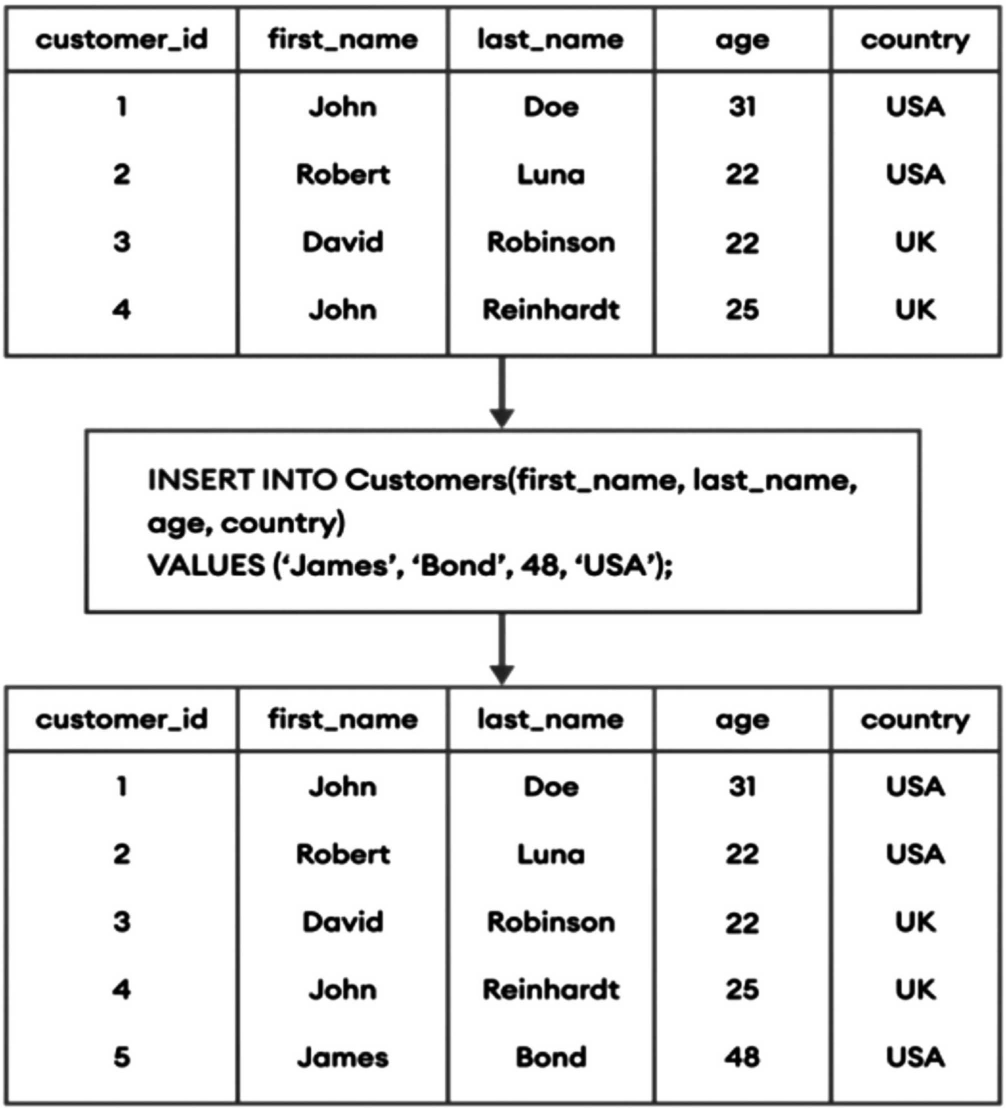
INSERTINTOCustomers(first_name, last_name, age, country)

VALUES

('James', 'Bond', 48, 'USA');

Here, the SQL command automatically sets the new customer_id for the new row and inserts it in a table.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

```
INSERT INTO Customers(first_name, last_name,
age, country)
VALUES ('James', 'Bond', 48, 'USA');
```

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	James	Bond	48	USA

Example:

SQL INSERT INTO

Insert Multiple Rows at Once in SQL

It’s also possible to insert multiple rows to a database table at once. For example,

```
INSERTINTOCustomers(first_name, last_name, age, country)

VALUES

(‘Harry’, ‘Potter’, 31, ‘USA’),

(‘Chris’, ‘Hemsworth’, 43, ‘USA’),

(‘Tom’, ‘Holland’, 26, ‘UK’);
```

Here, the SQL command inserts three rows to the Customers table.

Q8. Explain SQL UPDATE Command with syntax and example.

Ans : **(Imp.)**

SQL UPDATE

The SQL commands (UPDATE and DELETE) are used to modify the data that is already in the database. The SQL DELETE command uses a WHERE clause.

SQL UPDATE statement is used to change the data of the records held by tables. Which rows is to be update, it is decided by a condition. To specify condition, we use WHERE clause.

The UPDATE statement can be written in following form:

The SQL UPDATE statement is used to edit existing rows in a database table.

For example,

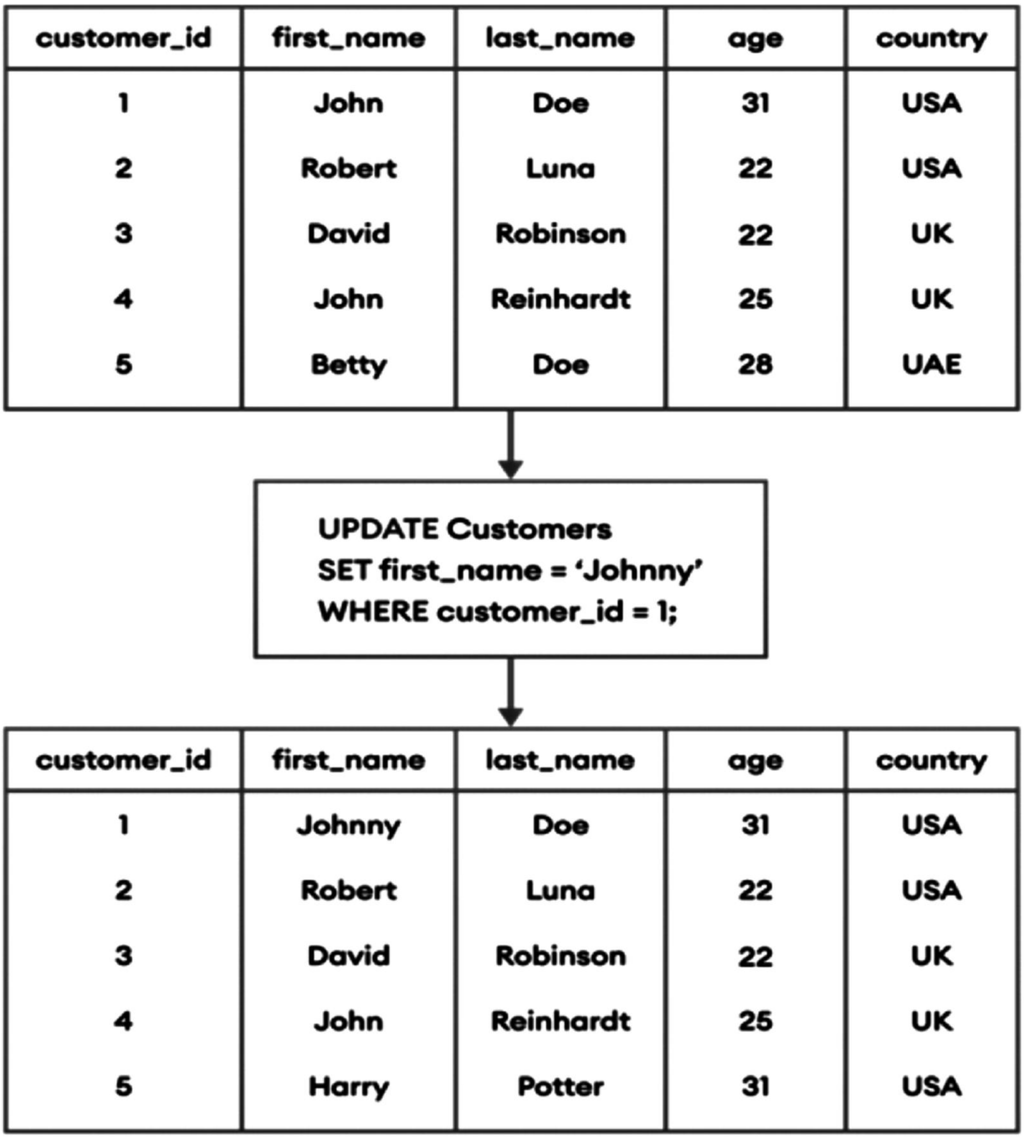
```
UPDATE Customers

SET first_name = ‘Johnny’

WHERE customer_id = 1;
```

Here, the SQL command changes the value of the first_name column will be Johnny if customer_id is equal to 1.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
UPDATE Customers
SET first_name = 'Johnny'
WHERE customer_id = 1;
```

customer_id	first_name	last_name	age	country
1	Johnny	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Harry	Potter	31	USA

Example:

SQL UPDATE Statement

Update Multiple Values in a Row

We can also update multiple values in a row at once. For example,

```
UPDATE Customers
SET first_name = 'Johnny', last_name = 'Depp'
WHERE customer_id = 1;
```

Here, the SQL command changes the value of the first_name column to Johnny and last_name to Depp if customer_id is equal to 1.

Update Multiple Rows

The UPDATE statement can update multiple rows at once. For example,

```
UPDATE Customers
SET country = 'NP'
WHERE age = 22;
```

Here, the SQL command changes the value of the country column to NP if age is 22. If there are more than one rows with age equals to 22, all the matching rows will be edited.

Update all Rows

We can update all the rows in a table at once by omitting the WHERE clause. For example,

```
UPDATE Customers
SET country = 'NP';
```

Here, the SQL command changes the value of the country column to NP for all rows.

SQL UPDATE with JOIN

SQL UPDATE JOIN means we will update one table using another table and join condition.

Let us take an example of a customer table. I have updated customer table that contains latest customer details from another source system. I want to update the customer table with latest data. In such case, I will perform join between target table and source table using join on customer ID.

Let's see the syntax of SQL UPDATE query with JOIN statement.

```
UPDATE customer_table
INNER JOIN
Customer_table
ON customer_table.rel_cust_name = customer_table.cust_id
SET customer_table.rel_cust_name = customer_table.cust_name
```

How to use multiple tables in SQL UPDATE statement with JOIN

Let's take two tables, table 1 and table 2.

Create table1

```
CREATE TABLE table1 (column1 INT, column2 INT, column3 VARCHAR (100))
INSERT INTO table1 (col1, col2, col3)
SELECT 1, 11, 'FIRST'
UNION ALL
SELECT 11,12, 'SECOND'
UNION ALL
SELECT 21, 13, 'THIRD'
UNION ALL
SELECT 31, 14, 'FOURTH'
```

Create table2

```
CREATE TABLE table2 (column1 INT, column2 INT, column3 VARCHAR (100))
INSERT INTO table2 (col1, col2, col3)
SELECT 1, 21, 'TWO-ONE'
UNION ALL
SELECT 11, 22, 'TWO-TWO'
UNION ALL
SELECT 21, 23, 'TWO-THREE'
UNION ALL
SELECT 31, 24, 'TWO-FOUR'
```

Now check the content in the table.

```
SELECT * FROM table_1
SELECT * FROM table_2
```

	Col 1	Col 2	Col 3
1	1	11	First
2	11	12	Second
3	21	13	Third
4	31	14	Fourth

	Col 1	Col 2	Col 3
1	1	21	Two-One
2	11	22	Two-Two
3	21	23	Two-Three
4	31	24	Two-Four

Our requirement is that we have table 2 which has two rows where Col 1 is 21 and 31. We want to update the value from table 2 to table 1 for the rows where Col 1 is 21 and 31.

We want to also update the values of Col 2 and Col 3 only.

The most easiest and common way is to use join clause in the update statement and use multiple tables in the update statement.

UPDATE table 1

```
SET Col 2 = t2.Col2,
Col 3 = t2.Col3
FROM table1 t1
INNER JOIN table 2 t2 ON t1.Col1 = t2.col1
WHERE t1.Col1 IN (21,31)
```

Check the content of the table

SELECT FROM table 1

SELECT FROM table 2

	Col 1	Col 2	Col 3
1	1	11	First
2	11	12	Second
3	21	23	Two-Three
4	31	24	Two-Four

	Col 1	Col 2	Col 3
1	1	21	First
2	11	22	Second
3	21	23	Two-Three
4	31	24	Two-Four

Here we can see that using join clause in update statement. We have merged two tables by the use of join clause.

Q9. Explain SQL DELETE Statement with syntax and example.

Ans :

SQL DELETE

The SQL DELETE statement is used to delete rows from a table. Generally DELETE statement removes one or more records from a table.

SQL DELETE Syntax

Let's see the Syntax for the SQL DELETE statement:

DELETE FROM table_name [WHERE condition];

Here table_name is the table which has to be deleted. The WHERE clause in SQL DELETE statement is optional here.

SQL DELETE Example

Let us take a table, named "EMPLOYEE" table.

ID	EMP_NAME	CITY	SALARY
101	Adarsh Singh	Obra	20000
102	Sanjay Singh	Meerut	21000
103	Priyanka Sharma	Raipur	25000
104	Esha Singhal	Delhi	26000

Example of delete with WHERE clause is given below:

DELETE FROM EMPLOYEE WHERE ID=101;

Resulting table after the query:

ID	EMP_NAME	CITY	SALARY
102	Sanjay Singh	Meerut	21000
103	Priyanka Sharma	Raipur	25000
104	Esha Singhal	Delhi	26000

Another example of delete statement is given below

DELETE FROM EMPLOYEE;

Resulting table after the query:

ID	EMP_NAME	CITY	SALARY

It will delete all the records of EMPLOYEE table.

It will delete the all the records of EMPLOYEE table where ID is 101.

The WHERE clause in the SQL DELETE statement is optional and it identifies the rows in the column that gets deleted.

WHERE clause is used to prevent the deletion of all the rows in the table, If you don't use the WHERE clause you might loss all the rows.

Invalid DELETE Statement for ORACLE database

You cannot use * (asterisk) symbol to delete all the records.

DELETE * FROM EMPLOYEE;

SQL DELETE TABLE

The DELETE statement is used to delete rows from a table. If you want to remove a specific row from a table you should use WHERE condition.

DELETE FROM table_name [WHERE condition];

But if you do not specify the WHERE condition it will remove all the rows from the table.

DELETE FROM table_name;

SQL DELETE ROW

Let us take an example of student.

Original table:

ID	STUDENT _NAME	ADDRESS
001	AJEET MAURYA	GHAZIABAD
002	RAJA KHAN	LUCKNOW
003	RAVI MALIK	DELHI

If you want to delete a student with id 003 from the student_name table, then the SQL DELETE query should be like this:

DELETE FROM student_name

WHERE id = 003;

Resulting table after SQL DELETE query:

ID	STUDENT_NAME	ADDRESS
001	AJEET MAURYA	GHAZIABAD
002	RAJA KHAN	LUCKNOW

SQL DELETE ALL ROWS

The statement SQL DELETE ALL ROWS is used to delete all rows from the table. If you want to delete all the rows from student table the query would be like,

DELETE FROM STUDENT_NAME;

SQL DELETE DUPLICATE ROWS

If you have got a situation that you have multiple duplicate records in a table, so at the time of fetching records from the table you should be more careful. You make sure that you are fetching unique records instead of fetching duplicate records.

To overcome with this problem we use DISTINCT keyword.

It is used along with SELECT statement to eliminate all duplicate records and fetching only unique records.

SYNTAX:

The basic syntax to eliminate duplicate records from a table is:

```
SELECT DISTINCT column1, column2,....columnN
FROM table_name
WHERE [conditions]
```

EXAMPLE:

Let us take an example of STUDENT table.

ROLL_NO	NAME	PERCENTAGE	ADDRESS
1	AJEET MAURYA	72.8	ALLAHABAD
2	CHANDAN SHARMA	63.5	MATHURA
3	DIVYA AGRAWAL	72.3	VARANASI
4	RAJAT KUMAR	72.3	DELHI
5	RAVI TYAGI	75.5	HAPUR
6	SONU JAISWAL	71.2	GHAZIABAD

Firstly we should check the SELECT query and see how it returns the duplicate percentage records.

```
SQL > SELECT PERCENTAGE FROM STUDENTS
ORDER BY PERCENTAGE;
PERCENTAGE
63.5
71.2
72.3
72.3
72.8
75.5
```

Now let us use SELECT query with DISTINCT keyword and see the result. This will eliminate the duplicate entry.

```
SQL > SELECT DISTINCT PERCENTAGE FROM STUDENTS
ORDER BY PERCENTAGE;
PERCENTAGE
63.5
71.2
72.3
72.8
75.5
```

5.1.4 DCL

Q10. Explain DCL commands with syntax and example.

Ans : **(Imp.)**

Data Control Language(DCL) is used to control privileges in Database. To perform any operation in the database, such as for creating tables, sequences or views, a user needs privileges. Privileges are of two types,

- **System:** This includes permissions for creating session, table, etc and all types of other system privileges.
- **Object:** This includes permissions for any command or query to perform any operation on the database tables.

DCL commands are as follows,

1. GRANT commands
2. REVOKE commands

- It is used to grant or revoke access permissions from any database user.

1. GRANT COMMAND

- GRANT command gives user's access privileges to the database.
- This command allows specified users to perform specific tasks.

Syntax:

```
GRANT <privilege list>
ON <relation name or view name>
TO <user/role list>;
```

Example : GRANT Command

```
GRANT ALL ON employee
TO ABC;
[WITH GRANT OPTION]
```

In the above example, user 'ABC' has been given permission to view and modify the records in the 'employee' table.

2. REVOKE COMMAND

- REVOKE command is used to cancel previously granted or denied permissions.
- This command withdraw access privileges given with the GRANT command.
- It takes back permissions from user.

Syntax:

REVOKE <privilege list>

ON <relation name or view name>

FROM <user name>;

Example : REVOKE Command

REVOKE UPDATE

ON employee

FROM ABC;

Difference between GRANT and REVOKE command.

GRANT	REVOKE
GRANT command allows a user to perform certain activities on the database.	**REVOKE command** disallows a user to perform certain activities.
It grants access privileges for database objects to other users.	It revokes access privileges for database objects previously granted to other users.

Example:
GRANT privilege_name
ON object_name
TO
{
user_name|PUBLIC|role_name
}
[WITH GRANT OPTION];	

5.2 Structure

5.2.1 Creation

Q11. Explain CREATE command in SQL.

Ans :

SQL CREATE TABLE statement is used to create table in a database.

If you want to create a table, you should name the table and define its column and each column's data type.

Let's see the simple syntax to create the table.

```
create table ”tablename”
(“column1” ”data type”,
“column2” ”data type”,
“column3” ”data type”,
...
“columnN” ”data type”);
```

The data type of the columns may vary from one database to another. For example, NUMBER is supported in Oracle database for integer value whereas INT is supported in MySQL.

Let us take an example to create a STUDENTS table with ID as primary key and NOT NULL are the constraint showing that these fields cannot be NULL while creating records in the table.

```
SQL> CREATE TABLE STUDENTS (
ID INT               NOT NULL,
NAME VARCHAR (20)    NOT NULL,
AGE INT              NOT NULL,
ADDRESS CHAR (25),
PRIMARY KEY (ID)
);
```

You can verify it, if you have created the table successfully by looking at the message displayed by the SQL Server, else you can use DESC command as follows:

SQL> DESC STUDENTS;

FIELD	TYPE	NULL	KEY	DEFAULT	EXTRA
ID	Int(11)	NO	PRI		
NAME	Varchar(20)	NO			
AGE	Int(11)	NO			
ADDRESS	Varchar(25)	YES		NULL	

4 rows in set (0.00 sec)

Now you have the STUDENTS table available in your database and you can use to store required information related to students.

SQL CREATE TABLE Example in MySQL

Let's see the command to create a table in MySQL database.

```
CREATE  TABLE  Employee
(
EmployeeID  int,
FirstName  varchar(255),
LastName  varchar(255),
Email  varchar(255),
AddressLine  varchar(255),
City  varchar(255)
);
```

CREATE TABLE IF NOT EXISTS

While creating a table that already exists, throws an error. To fix this issue, we can add the optional IF NOT EXISTS command while creating a table. For example,

```
CREATETABLEIFNOTEXISTS Companies (
idint,
namevarchar(50),
addresstext,
emailvarchar(50),
phonevarchar(10)
);
```

Here, the SQL command will only create a table if there is not one with a similar name.

CREATE TABLE AS

We can also create a table using records from any other existing table using the CREATE TABLE AS command. For example,

```
CREATETABLE USACustomers
AS (
SELECT *
FROM Customers
WHERE country = 'USA'
);
```

Here, the SQL command creates a table named USACustomers and copies the records of the nested query into the new table.

SQL Create Database

In SQL, the 'Create Database' statement is a first step for storing the structured data in the database.

The database developers and the users use this statement in SQL for creating the new database in the database systems. It creates the database with the name which has been specified in the Create Database statement.

Syntax of Create Database statement in SQL

```
CREATE  DATABASE  Database_Name;
```

In this syntax, Database_Name specifies the name of the database which we want to create in the system. We have to type the database name in query just after the 'Create Database' keyword.

Following are the most important points which are required to learn while creating a database:

- The database we want to create should be a simple and unique name, which can be easily identified.
- Database name should be no more than 128 characters.

Syntax of Create Database statement in MySQL

The same command is used in MySQL to create the new database for storing the structured data.

```
CREATE  DATABASE  Database_Name;
```

of Create Database in Oracle

There is no need to create the database in Oracle systems. In the Oracle database, we can directly create the database tables.

Examples of Create Database statement in SQL

In this article, we took the following two examples which will help how to run and perform the Create Database query in SQL:

Example 1:

This example creates the Student database. To create the Student database, you have to type the following command in Structured Query Language:

```
CREATE  DATABASE  Student  ;
```

When this query is executed successfully, then it will show the following output:

```
Database created successfully
```

You can also verify that your database is created in SQL or not by using the following query:

SHOW DATABASE

SQL does not allow developers to create the database with the existing database name. Suppose if you want to create another Student database in the same database system, then the Create Database statement will show the following error in the output:

```
Can't create database 'Student'; database exists
```

So, firstly you have to delete the existing database by using the Drop Statement. You can also replace the existing database with the help of Replace keyword.

If you want to replace the existing Student database, then you have to type the following SQL query:

```
CREATE  OR  REPLACE  DATABASE  Student  ;
```

Example 2:

Suppose, we want to create the Employee database in the system.

irstly, we have to type the following command in Structured Query Language:

```
CREATE  DATABASE  Employee  ;
```

When this query is executed successfully, then it will show the following output:

```
Database created successfully
```

5.2.2 Alteration

Q12. Explain ALTER Command in SQL.

Ans :

The ALTER TABLE statement in Structured Query Language allows you to add, modify, and delete columns of an existing table. This statement also allows database users to add and remove various SQL constraints on the existing tables.

Any user can also change the name of the table using this statement.

ALTER TABLE ADD Column statement in SQL

In many situations, you may require to add the columns in the existing table. Instead of creating a whole table or database again you can easily add single and multiple columns using the ADD keyword.

Syntax of ALTER TABLE ADD Column statement in SQL

```
ALTER  TABLE  table_name  ADD  column_name  column-definition;
```

The above syntax only allows you to add a single column to the existing table. If you want to add more than one column to the table in a single SQL statement, then use the following syntax:

```
ALTER  TABLE  table_name
ADD  (column_Name1  column-definition,
column_Name2  column-definition,
.....
column_NameN  column-definition);
```

Examples of ALTER TABLE ADD Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to add the single and multiple columns in the existing table using ALTER TABLE statement:

Example 1:

Let's take an example of a table named Cars:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

➢ Suppose, you want to add the new column Car_Model in the above table. For this, you have to type the following query in the SQL:

ALTER TABLE Cars ADD Car_Model Varchar(20);

This statement will add the Car_Model column to the Cars table.

Example 2:

Let's take an example of a table named Employee:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

➢ Suppose, you want to add two columns, Emp_ContactNo. and Emp_EmailID, in the above Employee table. For this, you have to type the following query in the SQL:

ALTER TABLE Employee ADD (Emp_ContactNo. Number(13), Emp_EmailID varchar(50);

This statement will add Emp_ContactNo. and Emp_EmailID columns to the Employee table.

ALTER TABLE MODIFY Column statement in SQL

The MODIFY keyword is used for changing the column definition of the existing table.

Syntax of ALTER TABLE MODIFY Column statement in SQL

ALTER TABLE table_name MODIFY column_name column-definition;

This syntax only allows you to modify a single column of the existing table. If you want to modify more than one column of the table in a single SQL statement, then use the following syntax:

ALTER TABLE table_name

MODIFY (column_Name1 column-definition,

column_Name2 column-definition,

.....

column_NameN column-definition);

Examples, of ALTER TABLE MODIFY Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to modify single and multiple columns of the existing table using ALTER TABLE statement:

Example 1:

Let's take an example of a table named Cars:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose, you want to modify the datatype of the Car_Color column of the above table. For this, you have to type the following query in the SQL:

  ALTER TABLE Cars ADD Car_Color Varchar(50);

Example 2:

Let's take an example of a table named Employee:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose, you want to modify the datatypes of two columns Emp_ContactNo. and Emp_EmailID of the above Employee table. For this, you have to type the following query in the SQL:

  ALTER TABLE Employee ADD (Emp_ContactNo. Int, Emp_EmailID varchar(80);

  ALTER TABLE DROP Column statement in SQL

In many situations, you may require to delete the columns from the existing table. Instead of deleting the whole table or database you can use DROP keyword for deleting the columns.

Syntax of ALTER TABLE DROP Column statement in SQL

ALTER TABLE table_name DROP Column column_name ;

Examples, of ALTER TABLE DROP Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to delete a column from the existing table using ALTER TABLE statement:

Example 1:

Let's take an example of a table named Cars:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose, you want to delete the Car_Color column from the above table. For this, you have to type the following query in the SQL:

  ALTER TABLE Cars DROP COLUMN Car_Color ;

- Let's check using the following statement that the Car_Color column is deleted from the table or not:

  SELECT * FROM Cars;

Car Name	Car Cost
Hyundai Creta	10,85,000
Hyundai Venue	9,50,000
Hyundai i20	9,00,000
Kia Sonet	10,00,000
Kia Seltos	8,00,000
Swift Dezire	7,95,000

Table: Cars

Example 2:

Let's take an example of a table named Employee:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose, you want to delete the Emp_Salary and Emp_City column from the above Employee table. For this, you have to type the following two different queries in the SQL:

```
ALTER TABLE Cars DROP COLUMN Emp_Salary ;

ALTER TABLE Cars DROP COLUMN Emp_City ;
```

ALTER TABLE RENAME Column statement in SQL

The RENAME keyword is used for changing the name of columns or fields of the existing table.

Syntax of ALTER TABLE RENAME Column statement in SQL

```
ALTER TABLE table_name RENAME COLUMN old_name to new_name;
```

Examples, of ALTER TABLE RENAME Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to change the name of a column of the existing table using ALTER TABLE statement:

Example 1:

Let's take an example of a table named Cars:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

➢ Suppose, you want to change the name of the Car_Color column of the above Cars table. For this, you have to type the following query in the SQL:

```
ALTER TABLE Cars RENAME COLUMN Car_Color to Colors;
```

This statement will change the name of a column of the Cars table. To see the changes, you have to type the following query:

```
SELECT * FROM Cars;
```

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

Example 2:

Let's take an example of a table named Employee:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose, you want to change the name of the Emp_City column of the above Employee table. For this, you have to type the following query in the SQL:

ALTER TABLE Employee RENAME COLUMN Emp_City to Emp_Address;

This statement will change the name of a column of the Employee table. To see the changes, you have to type the following query:

SELECT * FROM Employee;

Emp_Id	Emp_Name	Emp_Salary	Emp_Address
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

5.2.3 Defining Constraints

Q13. What Are Constraints in SQL? Explain.

Ans :

SQL constraints are rules that you can imply on the data in a table. It allows you to restrict only specific data that meets the regulations to go to a table. To put it simply, only if the data meets the constraint's rules, the insert operation will be successful, or else it will be aborted.

Constraints in SQL helps to maintain the accuracy, integrity, and reliability of a table's data. It can create them at a column or table level. If you declare constraints at the column level, it will apply them to a single column. On the other hand, if you declare them at the table level, it will implement them in more than one column. You can create constraints in SQL while creating a table using the CREATE TABLE command or later using the ALTER TABLE command. If you make a constraint with the ALTER TABLE command, the creation will only be successful if all the existing data meets the constraint rules.

As mentioned, you can create constraints in SQL using the CREATE TABLE command while creating a new table or ALTER TABLE command while altering an existing table. The basic syntax of creating an SQL constraint using the CREATE TABLE command is:

```
CREATE TABLE table_name(
column_name1 data_type(size) constraint_name,
column_name2 data_type(size) constraint_name,
....
);
```

In the above syntax:

- table_name: Name of the table you want to create
- column_name: Name of the column you want to create
- data_type: Data type of the value you want to add to the column
- size: Maximum size (length) of the column
- constraint_name: Name of the constraint you want to create and implement

You can also create a constraint in SQL using the ALTER TABLE command through the following syntax:

```
ALTER TABLE table_name ALTER COLUMN column_name data_type(size) constraint_name
```

The following constraints are commonly used in SQL:

- **NOT NULL -** Ensures that a column cannot have a NULL value
- **UNIQUE -** Ensures that all values in a column are different
- **PRIMARY KEY -** A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY-** Prevents actions that would destroy links between tables
- **CHECK -** Ensures that the values in a column satisfies a specific condition
- **DEFAULT -** Sets a default value for a column if no value is specified
- **CREATE INDEX -** Used to create and retrieve data from the database very quickly

5.2.3.1 PRIMAY KEY

Q14. Explain about the importance of Primary Key Constraint in SQL.

Ans : **(Imp.)**

A column or columns is called primary key (PK) that uniquely identifies each row in the table.

If you want to create a primary key, you should define a PRIMARY KEY constraint when you create or modify a table.

When multiple columns are used as a primary key, it is known as composite primary key.

In designing the composite primary key, you should use as few columns as possible. It is good for storage and performance both, the more columns you use for primary key the more storage space you require.

Points to remember for primary key:

- Primary key enforces the entity integrity of the table.
- Primary key always has unique data.
- A primary key length cannot be exceeded than 900 bytes.
- A primary key cannot have null value.
- There can be no duplicate value for a primary key.
- A table can contain only one primary key constraint.

SQL primary key for one column:

The following SQL command creates a PRIMARY KEY on the "S_Id" column when the "students" table is created.

MySQL:

```
CREATE TABLE students
(
S_Id int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
Address varchar (255),
City varchar (255),
PRIMARY KEY (S_Id)
)
```

SQL Server, Oracle, MS Access:

```
CREATE TABLE students
(
S_Id int NOT NULL PRIMARY KEY,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
Address varchar (255),
City varchar (255),
)
```

SQL primary key for multiple columns:

MySQL, SQL Server, Oracle, MS Access:

```
CREATE TABLE students
(
S_Id int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
Address varchar (255),
City varchar (255),
CONSTRAINT pk_StudentID PRIMARY KEY (S_Id, LastName)
)
```

Note: you should note that in the above example there is only one PRIMARY KEY (pk_StudentID). However it is made up of two columns (S_Id and LastName).

SQL primary key on ALTER TABLE

When table is already created and you want to create a PRIMARY KEY constraint on the "S_Id" column you should use the following SQL:

Primary key on one column:

```
ALTER TABLE students
ADD PRIMARY KEY (S_Id)
```

Primary key on multiple column:

```
ALTER TABLE students
ADD CONSTRAINT pk_StudentID PRIMARY KEY (S_Id,LastName)
```

How to DROP a PRIMARY KEY constraint?

If you want to DROP (remove) a primary key constraint, you should use following syntax:

MySQL:

```
ALTER TABLE students
DROP PRIMARY KEY
```

SQL Server / Oracle / MS Access:

```
ALTER TABLE students
DROP CONSTRAINT pk_StudentID
```

5.2.3.2 Foreign Key

Q15. Discuss about foreign key.

Ans : **(Imp.)**

In the relational databases, a foreign key is a field or a column that is used to establish a link between two tables.In simple words you can say that, a foreign key in one table used to point primary key in another table.

Let us take an example to explain it:

Here are two tables first one is students table and second is orders table.Here orders are given by students.

First table:

S_Id	Last Name	First Name	CITY
1	MAURYA	AJEET	ALLAHABAD
2	JAISWAL	RATAN	GHAZIABAD
3	ARORA	SAUMYA	MODINAGAR

First table:

O_Id	OrderNo	S_Id
1	99586465	2
2	78466588	2
3	22354846	3
4	57698656	1

- The "S_Id" column in the "Students" table is the PRIMARY KEY in the "Students" table.
- The "S_Id" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

The foreign key constraint is generally prevents action that destroy links between tables.

It also prevents invalid data to enter in foreign key column.

SQL FOREIGN KEY constraint ON CREATE TABLE:

(Defining a foreign key constraint on single column)

To create a foreign key on the "S_Id" column when the "Orders" table is created:

MySQL:

```
CREATE TABLE orders
(
O_Id int NOT NULL,
Order_No   int NOT NULL,
S_Id int,
PRIMAY KEY (O_Id),
FOREIGN KEY (S_Id) REFERENCES Persons (S_Id)
)
```

SQL FOREIGN KEY constraint for ALTER TABLE:

If the Order table is already created and you want to create a FOREIGN KEY constraint on the "S_Id" column, you should write the following syntax:

Defining a foreign key constraint on single column:

MySQL / SQL Server / Oracle / MS Access:

```
ALTER  TABLE  Orders
ADD  CONSTRAINT  fk_PerOrders
FOREIGN  KEY(S_Id)
REFERENCES  Students  (S_Id)
```

DROP SYNTAX for FOREIGN KEY COSTRAINT:

If you want to drop a FOREIGN KEY constraint, use the following syntax:

MySQL:

```
ALTER  TABLE  Orders
DROP  FOREIGN  KEY  fk_PerOrders
```

5.2.3.3 Unique

Q16. What is UNIQUE key ? Explain.

Ans :

A unique key is a set of one or more than one fields/columns of a table that uniquely identify a record in a database table.

You can say that it is little like primary key but it can accept only one null value and it cannot have duplicate values.

The unique key and primary key both provide a guarantee for uniqueness for a column or a set of columns.

There is an automatically defined unique key constraint within a primary key constraint.

There may be many unique key constraints for one table, but only one PRIMARY KEY constraint for one table.

SQL UNIQUE KEY constraint on CREATE TABLE:

If you want to create a UNIQUE constraint on the "S_Id" column when the "students" table is created, use the following SQL syntax:

(Defining a unique key constraint on single column):

```
CREATE  TABLE  students
CREATE  TABLE  students
(
S_Id  int  NOT  NULL,
LastName  varchar  (255)  NOT  NULL,
FirstName  varchar  (255),
City  varchar  (255),
UNIQUE  (S_Id)
)
```

(Defining a unique key constraint on multiple columns):

```
CREATE  TABLE  students
```

```
(
S_Id int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
City varchar (255),
CONSTRAINT uc_studentId UNIQUE (S_Id, LastName)
)
```

SQL UNIQUE KEY constraint on ALTER TABLE:

If you want to create a unique constraint on "S_Id" column when the table is already created, you should use the following SQL syntax:

(Defining a unique key constraint on single column):

```
ALTER TABLE students
ADD UNIQUE (S_Id)
```

(Defining a unique key constraint on multiple columns):

```
ALTER TABLE students
ADD CONSTRAINT uc_StudentId UNIQUE  (S_Id, LastName)
```

DROP SYNTAX FOR A FOREIGN KEY constraint:

If you want to drop a UNIQUE constraint, use the following SQL syntax:

```
ALTER TABLE students
DROP INDEX uc_studentID
```

5.2.3.4 Not Null

Q17. Discuss about NOT NULL constraint.

Ans :

The NOT NULL is a constraint in SQL which does not allow you to insert NULL values in the specified column.

If any column is defined as a NOT NULL constraint in the table, we cannot insert a new record in the table without adding the value to the column.

The following syntax adds the NOT NULL constraint to the column at the time of table creation:

```
CREATE TABLE Table_Name
(
Column_Name_1 DataType (character_size of the column_1) NOT NULL,
Column_Name_2 DataType (character_size of the column_2) NOT NULL,
Column_Name_3 DataType (character_size of the column_3) NOT NULL,
........,
Column_Name_N DataType (character_size of the column_N) NOT NULL,
)   ;
```

We can define NOT NULL constraint to one or more columns in one SQL table.

The following syntax adds the NOT NULL constraint to the column when the table already exists:

```
ALTER TABLE Table_Name ALTER COLUMN Column_Name datatype NOT NULL;
```

If you want to use NOT NULL constraint at the time of table creation, you have to follow the steps given below:

1. Create the simple new database
2. Create a new table and add NOT NULL
3. View table structure

Step 1: Create the Simple new database

Firstly, you have to make a new database in Structured Query Language.

The following query creates the Fortis_Hospital Database:

```
CREATE Database Fortis_Hospital;
```

Step 2: Create the New table and add NOT NULL

The following query creates the Doctor_Info table in the Fortis_Hospital Database and adds the NOT NULL constraint to the Doctor_ID column of the table:

If you want to remove the NOT NULL constraint from the SQL table, you can delete it by using the following syntax:

```
ALTER TABLE Table_Name MODIFY Column_Name Datatype [size] NULL;
```

The following query deletes the NOT NULL from the Doctor_Country column of the Doctor_Info table:

```
ALTER TABLE Doctor_Info MODIFY Doctor_Country Varchar[80] NULL;
```

To check the result of the above ALTER query, you have to type the following DESC command, which describes the structure of the Doctor_Info table:

```
DESC Doctor_Info;
```

Output:

Field	Type	NULL	Key	Default	Extra
Doctor_ID	INT	NO	-	NULL	-
Doctor_Name	INT	NO	-	NULL	-
Doctor_Specialist	Varchar(20)	NO	-	NULL	-
Doctor_Gender	Varchar(20)	NO	-	NULL	-
Doctor_Country	INT	Yes	-	NULL	-

As we can see in the above SQL table, the value of the NULL column is Yes for the Doctor Country field, which shows that the Doctor_Country column will accept NULL values.

Add NOT NULL constraint to Existing table.

Any database user can easily add NOT NULL constraint to the existing table by using the SQL ALTER TABLE syntax.

Syntax to Add NOT NULL constraint to Existing table:

ALTER TABLE Table_Name MODIFY Column_Name Datatype [size] NOT NULL;

The following SQL statement defines the NOT NULL constraint to the Doctor_Info table:

ALTER TABLE Doctor_Info MODIFY Doctor_Country Varchar[80] NOT NULL;

To check the result of the above ALTER query, you have to type the following DESC command to view the structure of the Doctor_Info table:

DESC Doctor_Info;

Output:

Field	Type	NULL	Key	Default	Extra
Doctor_ID	INT	NO	-	NULL	-
Doctor_Name	INT	NO	-	NULL	-
Doctor_Specialist	Varchar(20)	NO	-	NULL	-
Doctor_Gender	Varchar(20)	NO	-	NULL	-
Doctor_Country	INT	N0	-	NULL	-

5.2.3.5 Check

Q18. What is CHECK constraint? Explain.

Ans :

In SQL, the CHECK constraint is used to specify the condition that must be validated in order to insert data to a table. For example,

```
CREATETABLE Orders (
order_idINT PRIMARY KEY,
amountINTCHECK (amount >0)
);
```

Here, the amount column has a check condition: greater than 0. Now, let's try to insert records to the Orders table.

Example 1

```
— amount equal to 100
— record is inserted
INSERTINTOOrders(amount) VALUES(100);
```

Example 2

```
— amount equal to -5
— results in an error
INSERTINTOOrders(amount) VALUES(-5);
```

Create Named CHECK Constraint

It's a good practice to create named constraints so that it is easier to alter and drop constraints.

Here's an example to create named CHECK constraint:

```
— creates a named constraint named amount CK
— the constraint makes sure that amount is greater than 0
CREATETABLE Orders (
order_idINT PRIMARY KEY,
amountINT,
CONSTRAINT amountCK CHECK (amount >0)
);
```

CHECK Constraint in Existing Table

We can add the CHECK constraint to an existing table by using the ALTER TABLE clause. For example,

```
— Adding CHECK constraint without name
ALTERTABLE Orders
ADDCHECK (amount >0);
```

Here's how we can add a named CHECK constraint. For example,

```
— Adding CHECK constraint named amountCK
ALTERTABLE Orders
ADDCONSTRAINT amountCK CHECK (amount >0);
```

Remove CHECK Constraint

We can remove the CHECK constraint using the DROP clause. For example,

```
— removing CHECK constraint named amountCK
ALTERTABLE Orders
DROPCHECK amountCK;
```

5.2.3.6 IN Operator

Q19. Explain about IN Operator in SQL.

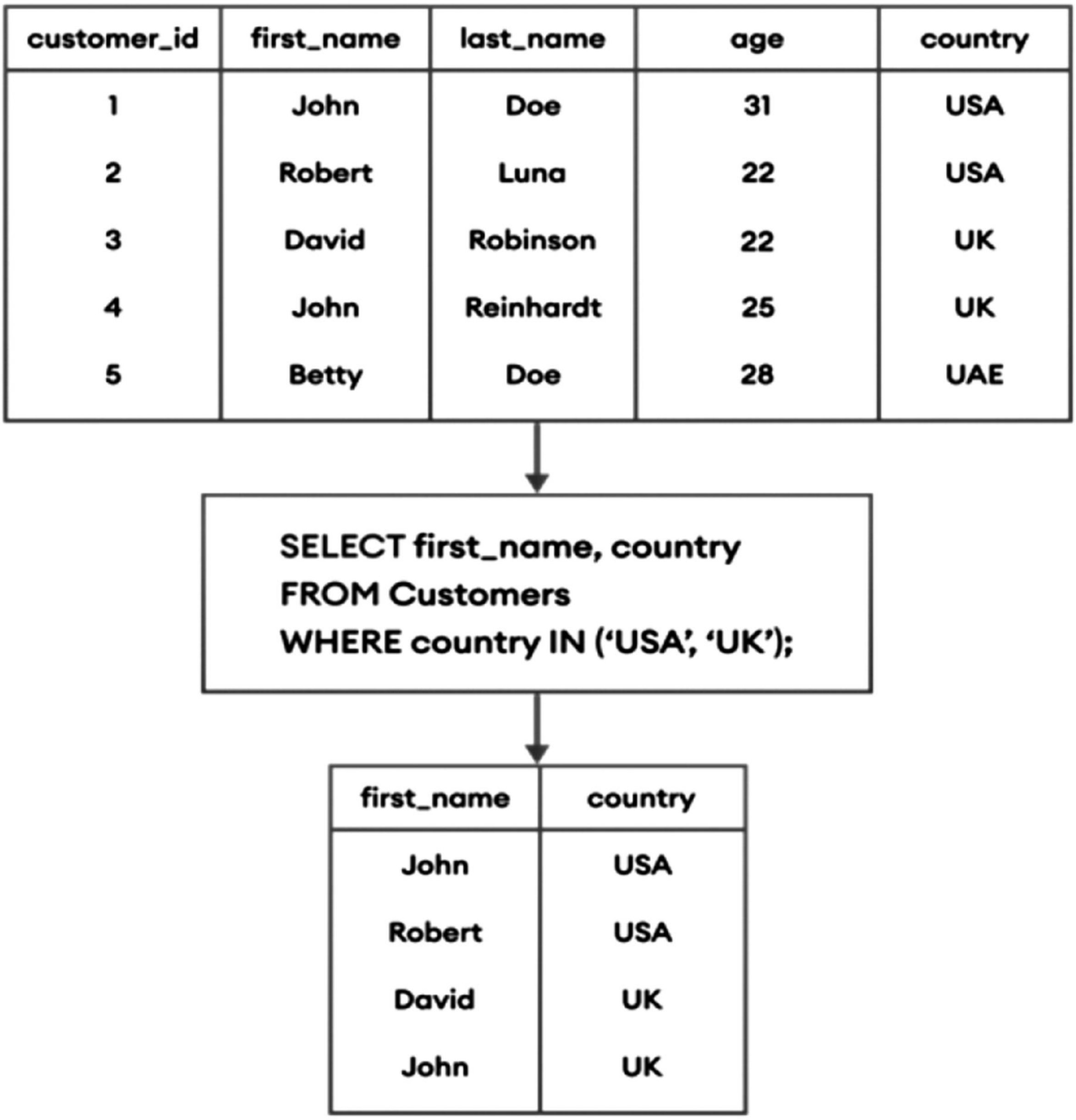
Ans :

- IN is an operator in SQL, which is generally used with a WHERE clause.
- Using the IN operator, multiple values can be specified.
- It allows us to easily test if an expression matches any value in a list of values.
- IN operator is used to replace many OR conditions.
- The IN operator is used with the WHERE clause to match values in a list. For example,

```
SELECT first_name, country
FROM Customers
WHERE country IN ('USA', 'UK');
```

- Here, the SQL command selects rows if the country is either USA or UK.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, country
FROM Customers
WHERE country IN ('USA', 'UK');
```

first_name	country
John	USA
Robert	USA
David	UK
John	UK

- **Example:** SQL IN Operator

SQL IN Operator With Columns

The IN operator can also be used to select rows in which a certain value exists in the given field. Let's see an example to clarify it.

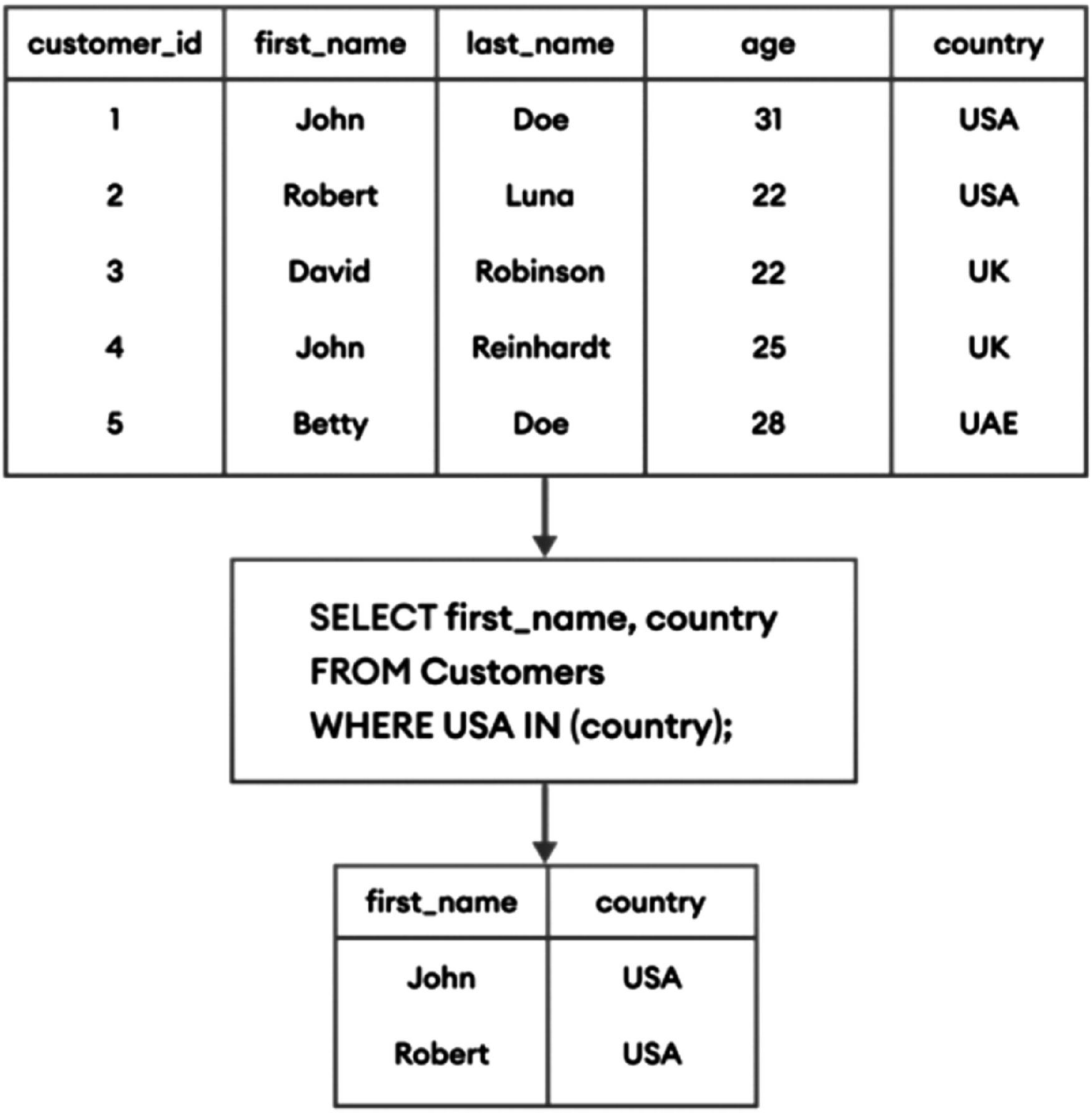
SELECT first_name, country

FROM Customers

WHERE'USA'in (country);

Here, the SQL command selects the rows if the USA value exists in the country field.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, country
FROM Customers
WHERE USA IN (country);
```

first_name	country
John	USA
Robert	USA

Example: SQL IN Operator With Value

SQL NOT IN Operator

The NOT IN operator returns is used to exclude the rows that match values in the list. It returns all the rows except the excluded rows. For example,

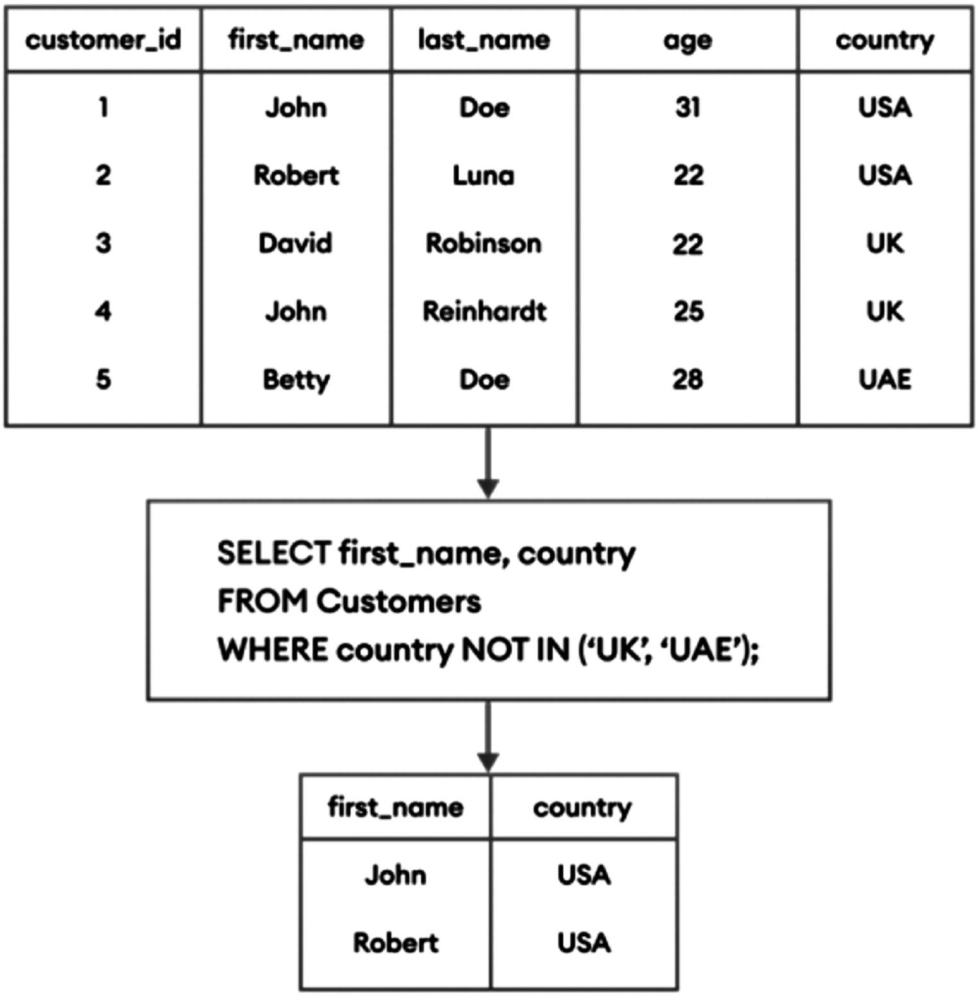
SELECT first_name, country

FROM Customers

WHERE country NOTIN ('UK', 'UAE');

Here, the SQL command selects rows if UK or UAE is not in the country column.

Table : Customers



customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, country
FROM Customers
WHERE country NOT IN ('UK', 'UAE');
```

first_name	country
John	USA
Robert	USA

Example: SQL NOT IN Operator

SQL IN Operator With Duplicate Values

By the way, the IN operator ignores duplicate values in the list. For example,

This code

```
SELECT first_name, country
FROM Customers
WHERE country IN ('USA', 'UK', 'USA');
```

is equivalent to

```
SELECT first_name, country
FROM Customers
WHERE country IN ('USA', 'UK');
```

Run Code

SQL IN Operator With Subquery

Suppose we want details of customers who have placed an order. Here's how we can do that using a subquery,

```
SELECT customer_id, first_name
Customers
WHERE customer_id IN (
SELECT customer_id
FROM Orders
);
```

5.2.3.7 Aggregate Functions

Q20. Explain about various aggregate functions in SQL.

Ans : **(Imp.)**

SQL Aggregation Function

- SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value.
- It is also used to summarize the data.

1. COUNT FUNCTION

- COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.
- COUNT function uses the COUNT(*) that returns the count of all the rows in a specified table. COUNT(*) considers duplicate and Null.

Syntax

```
COUNT(*)
or
COUNT( [ALL|DISTINCT] expression )
```

Sample table:

PRODUCT_MAST

PRODUCT	COMPANY	QTY	RATE	COST
Item1	Com1	2	10	20
Item2	Com2	3	25	75
Item3	Com1	2	30	60
Item4	Com3	5	10	50
Item5	Com2	2	20	40
Item6	Cpm1	3	25	75
Item7	Com1	5	30	150
Item8	Com1	3	10	30
Item9	Com2	2	25	50
Item10	Com3	4	30	120

Example: COUNT()

```
SELECT  COUNT(*)
FROM  PRODUCT_MAST;
```

Output:

```
10
```

Example: COUNT with WHERE

```
SELECT  COUNT(*)
FROM  PRODUCT_MAST;
WHERE  RATE>=20;
```

Output:

```
7
```

Example: COUNT() with DISTINCT

```
SELECT  COUNT(DISTINCT  COMPANY)
FROM  PRODUCT_MAST;
```

Output:

```
3
```

Example: COUNT() with GROUP BY

```
SELECT  COMPANY,  COUNT(*)
FROM  PRODUCT_MAST
GROUP  BY  COMPANY;
```

Output:

```
Com1    5
Com2    3
Com3    2
```

Example: COUNT() with HAVING

```
SELECT  COMPANY,  COUNT(*)
FROM  PRODUCT_MAST
GROUP  BY  COMPANY
HAVING  COUNT(*)>2;
```

Output:

```
Com1    5
Com2    3
```

2. SUM Function

Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

Syntax

SUM()

or

SUM([ALL|DISTINCT] expression)

Example: SUM()

```
SELECT SUM(COST)
FROM PRODUCT_MAST;
```

Output:

```
670
```

Example: SUM() with WHERE

```
SELECT SUM(COST)
FROM PRODUCT_MAST
WHERE QTY>3;
```

Output:

```
320
```

Example: SUM() with GROUP BY

```
SELECT SUM(COST)
FROM PRODUCT_MAST
WHERE QTY>3
GROUP BY COMPANY;
```

Output:

```
Com1    150
Com2    170
```

Example: SUM() with HAVING

```
SELECT COMPANY, SUM(COST)
FROM PRODUCT_MAST
GROUP BY COMPANY
HAVING SUM(COST)>=170;
```

Output:

```
Com1    335
Com3    170
```

3. AVG function

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

Syntax

```
AVG()
or
AVG( [ALL|DISTINCT] expression )
```

Example:

```
SELECT AVG(COST)
FROM PRODUCT_MAST;
```

Output:

```
67.00
```

4. MAX Function

MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

Syntax

```
MAX()
or
MAX( [ALL|DISTINCT] expression )
```

Example:

```
SELECT MAX(RATE)
FROM PRODUCT_MAST;
30
```

5. MIN Function

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.

Syntax

```
MIN()
or
MIN( [ALL|DISTINCT] expression )
```

Example:

```
SELECT MIN(RATE)
FROM PRODUCT_MAST;
```

Output:

```
10
```

5.2.4 Built in Functions

Q21. List some built in functions of SQL.

Ans :

The following is the list of built-in String functions, DateTime functions, Numeric functions and conversion functions.

I) String Functions

S.No.	Function	Description
1.	**ASCII**	Returns the ASCII code value for the leftmost character of a character expression.
2.	**CHAR**	Returns a character for an ASCII value.
3.	**CHARINDEX**	Searches for one character expression within another character expression andreturns the starting position of the first expression.
4.	**CONCAT**	Concatenates two or more string values in an end to end manner and returns a singlestring.
5.	**LEFT**	Returns a given number of characters from a character string starting from the left
6.	**LEN**	Returns a specified number of characters from a character string.
7.	**LOWER**	Converts a string to lower case.
8.	**LTRIM**	Removes all the leading blanks from a character string.
9.	**NCHAR**	Returns the Unicode character with the specified integer code, as defined by theUnicode standard.
10.	**PATINDEX**	Returns the starting position of the first occurrence of the pattern in a given string.
11.	**REPLACE**	Replaces all occurrences of a specified string with another string value.
12.	**RIGHT**	Returns the right part of a string with the specified number of characters.
13.	**RTRIM**	Returns a string after truncating all trailing spaces.
14.	**SPACE**	Returns a string of repeated spaces.
15.	**STR**	Returns character data converted from numeric data. The character data is rightjustified, with a specified length and decimal precision.
16.	**STUFF**	Inserts a string into another string. It deletes a specified length of characters from thefirst string at the start position and then inserts the second string into the first string atthe start position.
17.	**SUBSTRING**	Returns part of a character, binary, text, or image expression
18.	**UPPER**	Converts a lowercase string to uppercase.

II) DateTime Functions

S.No.	Function	Description
1.	CURRENT	Returns the current system date and time of the computer onwhich the SQL
2.	TIMESTAMP	server instance is installed. Time zone is notincluded.
3.	DATE ADD	Returns a new datetime value by adding an interval to thespecified datepart of the specified date
4.	DATEDIFF	Returns the difference in datepart between two given dates.
5.	DATENAME	Returns a datepart as a character string.
6.	DATEPART	Returns a datepart as an integer
7.	DAY	Returns the Day as an integer representing the Day part of aspecified date.
8.	GETDATE	Returns a datetime value containing the date and time of thecomputer on which the SQL Server instance is installed. It doesnot include the time zone.
9.	GETUTCDATE	Returns a datetime value inUTC format (CoordinatedUniversal Time), containing the date and time of the computeron which the SQL Server instance is installed.
10.	MONTH	Returns the Month as an integer representing the Month part ofa specified date.
11.	YEAR	Returns the Year as an integer representing the Year part of aspecified date.
12.	ISDATE	Determines whether the input is a valid date, time or datetimevalue.

III) Numeric Functions

S.No.	Function	Description
1.	ABS	Returns the absolute value of a number.
2.	AVG	Returns the average value of an expression/column values.
3.	CEILING	Returns the nearest integer value which is larger than or equal to the specifieddecimal value.
4.	COUNT	Returns the number of records in the SELECT query.
5.	FLOOR	Returns the largest integer value that is less than or equal to a number. The returnvalue is of the same data type as the input parameter.
6.	MAX	Returns the maximum value in an expression.
7.	MIN	Returns the minimum value in an expression.
8.	RAND	Returns a random floating point value using an optional seed value.
9.	ROUND	Returns a numeric expression rounded to a specified number of places right ofthe decimal point.
10.	SIGN	Returns an indicator of the sign of the input integer expression.
11.	SUM	Returns the sum of all the values or only the distinct values, in the expression.NULL values are ignored.

5.2.4.1 Numeric

Q22. Discuss about Numeric functions in SQL.

Ans : **(Imp.)**

Numeric Functions are used to perform operations on numbers and return numbers. Following are the numeric functions defined in SQL:

1. **ABS():** It returns the absolute value of a number.

   Syntax: SELECT ABS(-243.5);

Output:

```
243.5
SQL> SELECT ABS(-10);
+————————————————————————————+
| ABS(10)
+————————————————————————————+
| 10
+————————————————————————————+
```

2. **ACOS():** It returns the cosine of a number.

   Syntax: SELECT ACOS(0.25);

Output:

```
1.318116071652818
```

3. **ASIN():** It returns the arc sine of a number.

   Syntax: SELECT ASIN(0.25);

Output:

```
0.25268025514207865
```

4. **ATAN():** It returns the arc tangent of a number.

   Syntax: SELECT ATAN(2.5);

Output:

```
1.1902899496825317
```

5. **CEIL():** It returns the smallest integer value that is greater than or equal to a number.

   Syntax: SELECT CEIL(25.75);

Output:

```
26
```

6. **CEILING():** It returns the smallest integer value that is greater than or equal to a number.

   Syntax: SELECT CEILING(25.75);

Output:

```
26
```

7. **COS():** It returns the cosine of a number.

Syntax: SELECT COS(30);

Output:

0.15425144988758405

8. **COT():** It returns the cotangent of a number.

Syntax: SELECT COT(6);

Output:

-3.436353004180128

9. **DEGREES():**

It converts a radian value into degrees.

Syntax: SELECT DEGREES(1.5);

Output:

85.94366926962348

```
SQL>SELECT DEGREES(PI());
+------------------------------------------+
| DEGREES(PI())
+------------------------------------------+
| 180.000000
+------------------------------------------+
```

10. **DIV():** It is used for integer division.

Syntax: SELECT 10 DIV 5;

Output:

2

11. **EXP():** It returns e raised to the power of number.

Syntax: SELECT EXP(1);

Output:

2.718281828459045

12. **FLOOR():** It returns the largest integer value that is less than or equal to a number.

Syntax: SELECT FLOOR(25.75);

Output:

25

13. **GREATEST():** It returns the greatest value in a list of expressions.

Syntax: SELECT GREATEST(30, 2, 36, 81, 125);

Output:

125

14. LEAST(): It returns the smallest value in a list of expressions.

Syntax: SELECT LEAST(30, 2, 36, 81, 125);

Output:

2

15. LN(): It returns the natural logarithm of a number.

Syntax: SELECT LN(2);

Output:

0.6931471805599453

16. LOG10(): It returns the base-10 logarithm of a number.

Syntax: SELECT LOG(2);

Output:

0.6931471805599453

17. LOG2(): It returns the base-2 logarithm of a number.

Syntax: SELECT LOG2(6);

Output:

2.584962500721156

18. MOD(): It returns the remainder of n divided by m.

Syntax: SELECT MOD(18, 4);

Output:

2

19. PI(): It returns the value of PI displayed with 6 decimal places.

Syntax: SELECT PI();

Output:

3.141593

20. POWER(): It returns m raised to the nth power.

Syntax: SELECT POWER(4, 2);

Output:

16

21. RADIANS(): It converts a value in degrees to radians.

Syntax: SELECT RADIANS(180);

Output:

3.141592653589793

22. RAND(): It returns a random number.

Syntax: SELECT RAND();

Output:

0.33623238684258644

23. **ROUND():** It returns a number rounded to a certain number of decimal places.

Syntax: SELECT ROUND(5.553);

Output:

6

24. **SIGN():** It returns a value indicating the sign of a number.

Syntax: SELECT SIGN(255.5);

Output:

1

25. **SIN():** It returns the sine of a number.

Syntax: SELECT SIN(2);

Output:

0.9092974268256817

26. **SQRT():** It returns the square root of a number.

Syntax: SELECT SQRT(25);

Output:

5

27. **TAN():** It returns the tangent of a number.

Syntax: SELECT TAN(1.75);

Output:

-5.52037992250933

28. **ATAN2():** It returns the arctangent of the x and y coordinates, as an angle and expressed in radians.

Syntax: SELECT ATAN2(7);

Output:

1.42889927219073

29. **TRUNCATE():** This doesn't work for SQL Server. It returns 7.53635 truncated to 2 places right of the decimal point.

Syntax: SELECT TRUNCATE(7.53635, 2);

Output:

7.53

5.2.4.2 Date

Q23. Discuss about Date function in SQL.

Ans :

In SQL, dates are complicated for newbies, since while working with database, the format of the date in table must be matched with the input date in order to insert. In various scenarios instead of date, datetime (time is also involved with date) is used.

In MySql the default date functions are:

- **NOW():** Returns the current date and time. Example:

```
SELECT NOW();
```

Output:

```
2017-01-13 08:03:52
```

- **CURDATE()**: Returns the current date. Example:

```
SELECT CURDATE();
```

Output:

```
2017-01-13
```

- **CURTIME():** Returns the current time. Example:

```
SELECT CURTIME();
```

Output:

```
08:05:15
```

- **DATE()**: Extracts the date part of a date or date/time expression. Example:

For the below table named 'Test'

Id	Name	BirthTime
4120	Pratik	1996-09-26 16:44:15.581

```
SELECT Name, DATE(BirthTime) AS BirthDate FROM Test;
```

Output:

```
Name      BirthDate
Pratik    1996-09-26
```

- **EXTRACT():** Returns a single part of a date/time. Syntax:

```
EXTRACT(unit FROM date);
```

There are several units that can be considered but only some are used such as:

MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, etc.

And 'date' is a valid date expression.

Example:

For the below table named 'Test'

Id	Name	BirthTime
4120	Pratik	1996-09-26 16:44:15.581

Queries

- SELECT Name, Extract(DAY FROM BirthTime) AS BirthDay FROM Test;

Output:

```
Name      BirthDay
Pratik    26
```

- SELECT Name, Extract(YEAR FROM BirthTime) AS BirthYear FROM Test;

Output:

Name	**BirthYear**
Pratik	1996

- SELECT Name, Extract(SECOND FROM BirthTime) AS BirthSecond FROM Test;
- Output:

Name	**BirthSecond**
Pratik	581

- **DATE_ADD() :** Adds a specified time interval to a date

Syntax:

DATE_ADD(date, INTERVAL expr type);

Where, date – valid date expression and expr is the number of interval we want to add.

and type can be one of the following:

MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, etc.

Example:

For the below table named 'Test'

Id	**Name**	**BirthTime**
4120	Pratik	1996-09-26 16:44:15.581

Queries

- SELECT Name, DATE_ADD(BirthTime, INTERVAL 1 YEAR) AS BirthTimeModified FROM Test;

Output:

Name	**BirthTimeModified**
Pratik	1997-09-26 16:44:15.581

- SELECT Name, DATE_ADD(BirthTime, INTERVAL 30 DAY) AS BirthDayModified FROM Test;
- Output:

Name	**BirthDayModified**
Pratik	1996-10-26 16:44:15.581

- SELECT Name, DATE_ADD(BirthTime, INTERVAL 4 HOUR) AS BirthHourModified FROM Test;

Output:

Name	**BirthSecond**
Pratik	1996-10-26 20:44:15.581

- **DATE_SUB():** Subtracts a specified time interval from a date. Syntax for DATE_SUB is same as DATE_ADD just the difference is that DATE_SUB is used to subtract a given interval of date.
- **DATEDIFF():** Returns the number of days between two dates.Syntax:

  DATEDIFF(date1, date2);
- date1 & date2- date/time expression

Example:

SELECT DATEDIFF('2017-01-13','2017-01-03') AS DateDiff;

Output:

DateDiff

10

- **DATE_FORMAT():** Displays date/time data in different formats.Syntax:

  DATE_FORMAT(date,format);

  date is a valid date and format specifies the output format for the date/time. The formats that can be used are:

- %a-Abbreviated weekday name (Sun-Sat)
- %b-Abbreviated month name (Jan-Dec)
- %c-Month, numeric (0-12)
- %D-Day of month with English suffix (0th, 1st, 2nd, 3rd)
- %d-Day of month, numeric (00-31)
- %e-Day of month, numeric (0-31)
- %f-Microseconds (000000-999999)
- %H-Hour (00-23)
- %h-Hour (01-12)
- %I-Hour (01-12)
- %i-Minutes, numeric (00-59)
- %j-Day of year (001-366)
- %k-Hour (0-23)
- %l-Hour (1-12)
- %M-Month name (January-December)
- %m-Month, numeric (00-12)
- %p-AM or PM
- %r-Time, 12-hour (hh:mm:ss followed by AM or PM)
- %S-Seconds (00-59)
- %s-Seconds (00-59)
- %T-Time, 24-hour (hh:mm:ss)
- %U-Week (00-53) where Sunday is the first day of week
- %u-Week (00-53) where Monday is the first day of week
- %V-Week (01-53) where Sunday is the first day of week, used with %X
- %v-Week (01-53) where Monday is the first day of week, used with %x
- %W-Weekday name (Sunday-Saturday)

- %w-Day of the week (0=Sunday, 6=Saturday)
- %X-Year for the week where Sunday is the first day of week, four digits, used with %V
- %x-Year for the week where Monday is the first day of week, four digits, used with %v
- %Y-Year, numeric, four digits
- %y-Year, numeric, two digits

Example:

DATE_FORMAT(NOW(),'%d %b %y')

Result:

13 Jan 17

5.2.4.3 String Functions

Q24. Explain about string functions in SQL.

Ans :

String functions are used to perform an operation on input string and return an output string.

Following are the string functions defined in SQL:

1. **ASCII():** This function is used to find the ASCII value of a character.

   Syntax: SELECT ascii('t');

Output:

116

2. **CHAR_LENGTH():** Doesn't work for SQL Server. Use LEN() for SQL Server. This function is used to find the length of a word.

   Syntax: SELECT char_length('Hello!');

Output:

6

3. **CHARACTER_LENGTH():** Doesn't work for SQL Server. Use LEN() for SQL Server. This function is used to find the length of a line.

   Syntax: SELECT CHARACTER_LENGTH('geeks for geeks');

Output:

15

4. **CONCAT():** This function is used to add two words or strings.

   Syntax: SELECT 'Geeks' || ' ' || 'forGeeks' FROM dual;

Output:

'GeeksforGeeks'

5. **CONCAT_WS():** This function is used to add two words or strings with a symbol as concatenating symbol.

   Syntax: SELECT CONCAT_WS('_', 'geeks', 'for', 'geeks');

Output:

geeks_for_geeks

6. **FIND_IN_SET():** This function is used to find a symbol from a set of symbols.

   Syntax: SELECT FIND_IN_SET('b', 'a, b, c, d, e, f');

Output:

2

7. **FORMAT():** This function is used to display a number in the given format.

   Syntax: Format("0.981", "Percent");

Output:

'98.10%'

8. **INSERT():** This function is used to insert the data into a database.

   Syntax: INSERT INTO database (geek_id, geek_name) VALUES (5000, 'abc');

Output:

successfully updated

9. **INSTR():** This function is used to find the occurrence of an alphabet.

   Syntax:INSTR('geeks for geeks', 'e');

Output:

2 (the first occurrence of 'e')

Syntax:INSTR('geeks for geeks', 'e', 1, 2);

Output:

3 (the second occurrence of 'e')

10. **LCASE():** This function is used to convert the given string into lower case.

    Syntax: LCASE ("GeeksFor Geeks To Learn");

Output:

geeksforgeeks to learn

11. **LEFT():** This function is used to SELECT a sub string from the left of given size or characters.

    Syntax: SELECT LEFT('geeksforgeeks.org', 5);

Output:

geeks

12. **LENGTH():** This function is used to find the length of a word.

    Syntax:LENGTH('GeeksForGeeks');

Output:

13

13. **LOCATE():** This function is used to find the nth position of the given word in a string.

    Syntax: SELECT LOCATE('for', 'geeksforgeeks', 1);

Output:

6

14. LOWER(): This function is used to convert the upper case string into lower case.

Syntax: SELECT LOWER('GEEKSFORGEEKS.ORG');

Output:

geeksforgeeks.org

15. LPAD(): This function is used to make the given string of the given size by adding the given symbol.

Syntax:LPAD('geeks', 8, '0');

Output:

000geeks

16. LTRIM(): This function is used to cut the given sub string from the original string.

Syntax:LTRIM('123123geeks', '123');

Output:

geeks

17. MID(): This function is to find a word from the given position and of the given size.

Syntax: Mid ("geeksforgeeks", 6, 2);

Output:

for

18. POSITION(): This function is used to find position of the first occurrence of the given alphabet.

Syntax: SELECT POSITION('e' IN 'geeksforgeeks');

Output:

2

19. REPEAT(): This function is used to write the given string again and again till the number of times mentioned.

Syntax: SELECT REPEAT('geeks', 2);

Output:

geeksgeeks

20. REPLACE(): This function is used to cut the given string by removing the given sub string.

Syntax:REPLACE('123geeks123', '123');

Output:

geeks

21. REVERSE(): This function is used to reverse a string.

Syntax: SELECT REVERSE('geeksforgeeks.org');

Output:

'gro.skeegrofskeeg'

22. RIGHT(): This function is used to SELECT a sub string from the right end of the given size.

Syntax: SELECT RIGHT('geeksforgeeks.org', 4);

Output:

'.org'

23. RPAD(): This function is used to make the given string as long as the given size by adding the given symbol on the right.

Syntax:RPAD('geeks', 8, '0');

Output:

'geeks000'

24. RTRIM(): This function is used to cut the given sub string from the original string.

Syntax:RTRIM('geeksxyxzyyy', 'xyz');

Output:

'geeks'

25. SPACE(): This function is used to write the given number of spaces.

Syntax: SELECT SPACE(7);

Output:

'   '

26. STRCMP(): This function is used to compare 2 strings.

If string1 and string2 are the same, the STRCMP function will return 0.

If string1 is smaller than string2, the STRCMP function will return -1.

If string1 is larger than string2, the STRCMP function will return 1.

Syntax: SELECT STRCMP('google.com', 'geeksforgeeks.com');

Output:

-1

27. SUBSTR(): This function is used to find a sub string from the a string from the given position.

Syntax:SUBSTR('geeksforgeeks', 1, 5);

Output:

'geeks'

28. SUBSTRING(): This function is used to find an alphabet from the mentioned size and the given string.

Syntax: SELECT SUBSTRING('GeeksForGeeks.org', 9, 1);

Output:

'G'

29. SUBSTRING_INDEX(): This function is used to find a sub string before the given symbol.

Syntax: SELECT SUBSTRING_INDEX('www.geeksforgeeks.org', '.', 1);

Output:

'www'

30. **TRIM():** This function is used to cut the given symbol from the string.

    Syntax:TRIM(LEADING '0' FROM '000123');

Output:

123

31. **UCASE():** This function is used to make the string in upper case.

    Syntax: UCASE ("GeeksForGeeks");

Output:

GEEKSFORGEEKS

5.2.4.4 Set Operations

Q25. Explain various set operations in SQL.

Ans :

The SQL Set operation is used to combine the two or more SQL SELECT statements.

Types of Set Operation

1. Union
2. UnionAll
3. Intersect
4. Minus

1. Union

➢ The SQL Union operation is used to combine the result of two or more SQL SELECT queries.

➢ In the union operation, all the number of datatype and columns must be same in both the tables on which UNION operation is being applied.

➢ The union operation eliminates the duplicate rows from its resultset.

Syntax

```
SELECT column_name FROM table1
UNION
SELECT column_name FROM table2;
```

Example:

The First table

ID	NAME
1	Jack
2	Harry
3	Jackson

The Second table

ID	NAME
3	Jackson
4	Stephan
5	David

Union SQL query will be:

```
SELECT * FROM First
UNION
SELECT * FROM Second;
```

The resultset table will look like:

ID	NAME
1	Jack
2	Harry
3	Jackson
4	Stephan
5	David

2. Union All

Union All operation is equal to the Union operation. It returns the set without removing duplication and sorting the data.

Syntax:

```
SELECT column_name FROM table1
UNION ALL
SELECT column_name FROM table2;
```

Example:

Using the above First and Second table.

Union All query will be like:

```
SELECT * FROM First
UNION ALL
SELECT * FROM Second;
```

The resultset table will look like:

ID	NAME
1	Jack
2	Harry
3	Jackson
3	Jackson
4	Stephan
5	David

3. Intersect

- It is used to combine two SELECT statements. The Intersect operation returns the common rows from both the SELECT statements.
- In the Intersect operation, the number of datatype and columns must be the same.
- It has no duplicates and it arranges the data in ascending order by default.

Syntax

```
SELECT column_name FROM table1
INTERSECT
SELECT column_name FROM table2;
```

Example:

Using the above First and Second table.

Intersect query will be:

```
SELECT * FROM First
INTERSECT
SELECT * FROM Second;
```

The resultset table will look like:

ID	NAME
3	Jackson

4. Minus

- It combines the result of two SELECT statements. Minus operator is used to display the rows which are present in the first query but absent in the second query.
- It has no duplicates and data arranged in ascending order by default.

Syntax:

```
SELECT column_name FROM table1
MINUS
SELECT column_name FROM table2;
```

Example

Using the above First and Second table.

Minus query will be:

```
SELECT * FROM First
MINUS
SELECT * FROM Second;
```

The resultset table will look like:

ID	NAME
1	Jack
2	Harry

5.2.4.5 Sub Queries

Q26. How to write sub queries in SQL? Explain.

Ans : **(Imp.)**

In SQL a Subquery can be simply defined as a query within another query. In other words we can say that a Subquery is a query that is embedded in WHERE clause of another SQL query. Important rules for Subqueries:

- You can place the Subquery in a number of SQL clauses: WHERE clause, HAVING clause, FROM clause. Subqueries can be used with SELECT, UPDATE, INSERT, DELETE statements along with expression operator. It could be equality operator or comparison operator such as =, >, =, <= and Like operator.
- A subquery is a query within another query. The outer query is called as main query and inner query is called as subquery.
- The subquery generally executes first when the subquery doesn't have any co-relation with the main query, when there is a co-relation the parser takes the decision on the fly on which query to execute on precedence and uses the output of the subquery accordingly.
- Subquery must be enclosed in parentheses.
- Subqueries are on the right side of the comparison operator.
- ORDER BY command cannot be used in a Subquery. GROUPBY command can be used to perform same function as ORDER BY command.
- Use single-row operators with singlerow Subqueries. Use multiple-row operators with multiple-row Subqueries.

Syntax: There is not any general syntax for Subqueries. However, Subqueries are seen to be used most frequently with SELECT statement as shown below:

```
SELECT column_name
FROM table_name
WHERE column_name expression operator
( SELECT COLUMN_NAME from TABLE_NAME WHERE ... );
```

Sample Table:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
Ram	101	Chennai	9988775566
Raj	102	Coimbatore	8877665544
Sasi	103	Madurai	7766553344
Ravi	104	Salem	8989898989
Sumathi	105	Kanchipuram	8989856868

STUDENT NAME	ROLL_NO	SECTION
Ravi	104	A
Sumathi	105	B
Raj	102	A

Sample Queries

- To display NAME, LOCATION, PHONE_NUMBER of the students from DATABASE table whose section is A

  Select NAME, LOCATION, PHONE_NUMBER from DATABASE

  WHERE ROLL_NO IN

  (SELECT ROLL_NO from STUDENT where SECTION='A');

- **Explanation :** First subquery executes " SELECT ROLL_NO from STUDENT where SECTION='A' " returns ROLL_NO from STUDENT table whose SECTION is 'A'.Then outer-query executes it and return the NAME, LOCATION,

- PHONE_NUMBER from the DATABASE table of the student whose ROLL_NO is returned from inner subquery.

Output:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
Ravi	104	Salem	8989898989
Raj	102	Coimbatore	8877665544

- Insert Query Example:

  Table1: Student1

NAME	ROLL_NO	LOCATION	PHONE_NUMBER	
Ram	101	chennai	9988773344	
Raju	102	coimbatore	9090909090	
Ravi	103	salem	8989898989	

Table 2: Student 2

NAME	ROLL_NO	LOCATION	PHONE_NUMBER	
Raj	111	chennai	8787878787	
Sai	112	mumbai	6565656565	
Sri	113	coimbatore	7878787878	

To insert Student2 into Student1 table:

```
INSERT INTO Student1  SELECT * FROM Student2;
```

Output:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER	
Ram	101	chennai	9988773344	
Raju	102	coimbatore	9090909090	
Ravi	103	salem	8989898989	
Raj	111	chennai	8787878787	
Sai	112	mumbai	6565656565	
Sri	113	coimbatore	7878787878	

To delete students from Student2 table whose rollno is same as that in Student1 table and having location as chennai

```
DELETE FROM Student2
WHERE ROLL_NO IN ( SELECT ROLL_NO
          FROM Student1
          WHERE LOCATION = 'chennai');
```

Output:

1 row delete successfully.

Display Student2 table:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER	
Sai	112	mumbai	6565656565	
Sri	113	coimbatore	7878787878	

To update name of the students to geeks in Student2 table whose location is same as Raju,Ravi in Student1 table

```
UPDATE Student2
SET NAME='geeks'
WHERE LOCATION IN ( SELECT LOCATION
           FROM Student1
           WHERE NAME IN ('Raju','Ravi'));
```

Output:

1 row updated successfully.

Display Student2 table:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER	
Sai	112	mumbai	6565656565	
geeks	113	coimbatore	7878787878	

5.2.4.6 Corelated Sub Queries

Q27. What are co related sub queries in SQL? Explain.

Ans : **(Imp.)**

Let's start with an example.

See the following employees table in the sample database:

employees
employeejd
first_name
last_name
email
phone_number
hire_date
jobjd
salary
managerjd
departmentJd

The following query finds employees whose salary is greater than the average salary of all employees:

```
SELECT
        employee_id,
        first_name,
        last_name,
salary
FROM
employees
WHERE
salary> (SELECT
AVG(salary)
FROM
employees);
```

employeejd	first_name	last_name	salary
▶ 121	Adam	Fripp	8200.00
103	Alexander	Hunold	9000.00
109	Daniel	Faviet	9000.00
114	Den	Raphaely	11000.00
204	Hermann	Baer	10000.00
177	Jack	Livingston	8400.00
110	John	Chen	8200.00
145	John	Russell	14000.00
176	Jonathon	Taylor	8600.00
146	Karen	Partners	13500.00
102	Lex	De Haan	17000.00

In this example, the subquery is used in the WHERE clause. There are some points that you can see from this query:

First, you can execute the subquery that returns the average salary of all employees independently.

```
SELECT
AVG(salary)
FROM
employees;
```

Second, the database system needs to evaluate the subquery only once.

Third, the outer query makes use of the result returned from the subquery. The outer query depends on the subquery for its value. However, the subquery does not depend on the outer query. Sometimes, we call this subquery is a plain subquery.

Unlike a plain subquery, a correlated subquery is a subquery that uses the values from the outer query. Also, a correlated subquery may be evaluated once for each row selected by the outer query. Because of this, a query that uses a correlated subquery may be slow.

A correlated subquery is also known as a repeating subquery or a synchronized subquery.

SQL correlated subquery examples

Let's see few more examples of the correlated subqueries to understand them better.

SQL correlated subquery in the WHERE clause example

The following query finds all employees whose salary is higher than the average salary of the employees in their departments:

```
SELECT
    employee_id,
    first_name,
    last_name,
    salary,
    department_id
FROM
employees e
WHERE
salary> (SELECT
AVG(salary)
FROM
employees
WHERE
    department_id = e.department_id)
ORDERBY
```

```
          department_id ,
          first_name ,
          last_name;
```

Here is the output:

Here is the output:

employeejd	first_name	last_name	salary	departmentjd
▶ 201	Michael	Hartstein	13000.00	2
114	Den	Raphaely	11000.00	3
121	Adam	Fripp	8200.00	5
120	Matthew	Weiss	8000.00	5
122	Payam	Kaufling	7900.00	5
123	Shanta	Vollman	6500.00	5
103	Alexander	Hunold	9000.00	6
104	Bruce	Ernst	6000.00	6
145	John	Russell	14000.00	8
146	Karen	Partners	13500.00	8
100	Steven	King	24000.00	9
109	Daniel	Faviet	9000.00	10
108	Nancy	Greenberg	12000.00	10

In this example, the outer query is:

```
SELECT
     employee_id,
     first_name,
     last_name,
salary,
     department_id
FROM
employees e
WHERE
salary>
...
```

and the correlated subquery is:

```
SELECT
AVG( list_price )
```

```
FROM
products
WHERE
    category_id = p.category_id
```

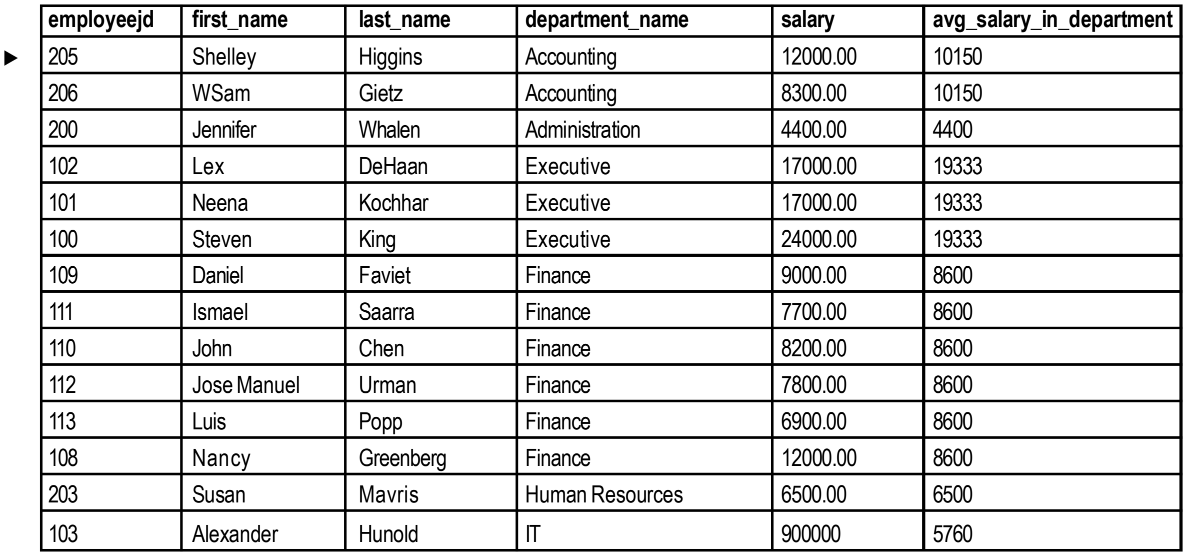
For each employee, the database system has to execute the correlated subquery once to calculate the average salary of the employees in the department of the current employee.

SQL correlated subquery in the SELECT clause example

The following query returns the employees and the average salary of all employees in their departments:

```
SELECT
    employee_id,
    first_name,
    last_name,
    department_name,
salary,
    (SELECT
ROUND(AVG(salary),0)
FROM
employees
WHERE
    department_id = e.department_id) avg_salary_in_department
FROM
employees e
INNERJOIN
departments d ON d.department_id = e.department_id
ORDERBY
    department_name,
    first_name,
     last_name;
```

The output is:



	employeejd	first_name	last_name	department_name	salary	avg_salary_in_department
▶	205	Shelley	Higgins	Accounting	12000.00	10150
	206	WSam	Gietz	Accounting	8300.00	10150
	200	Jennifer	Whalen	Administration	4400.00	4400
	102	Lex	DeHaan	Executive	17000.00	19333
	101	Neena	Kochhar	Executive	17000.00	19333
	100	Steven	King	Executive	24000.00	19333
	109	Daniel	Faviet	Finance	9000.00	8600
	111	Ismael	Saarra	Finance	7700.00	8600
	110	John	Chen	Finance	8200.00	8600
	112	Jose Manuel	Urman	Finance	7800.00	8600
	113	Luis	Popp	Finance	6900.00	8600
	108	Nancy	Greenberg	Finance	12000.00	8600
	203	Susan	Mavris	Human Resources	6500.00	6500
	103	Alexander	Hunold	IT	900000	5760

For each employee, the database system has to execute the correlated subquery once to calculate the average salary by the employee's department.

SQL correlated subquery with EXISTS operator example

We often use a correlated subquery with the EXISTS operator. For example, the following query returns all employees who have no dependents:

```
SELECT
    employee_id,
    first_name,
    last_name
FROM
employees e
WHERE
NOTEXISTS(SELECT
    *
FROM
dependents d
WHERE
d.employee_id = e.employee_id)
ORDERBY first_name ,
    last_name;
```

The following picture shows the output:

employee Jd	first_name	last_name
121	Adam	Frpp
193	Britney	Everett
179	Charles	Johnson
126	Irene	MikJolmeni
177	Jade	Livngston
178	Kimberely	Grant
120	Matthew	Weiss
122	Payam	Kauflmg
192	Sarah	Bel
123	Shanta	Vdman

5.2.5 Join

Q28. Explain about join various types in SQL.

Ans : **(Imp.)**

The join clause allows us to retrieve data from two or more related tables into a meaningful result set. We can join the table using a SELECT statement and a join condition. It indicates how SQL Server can use data from one table to select rows from another table. In general, tables are related to each other using foreign key constraints.

In a JOIN query, a condition indicates how two tables are related:

- Choose columns from each table that should be used in the join. A join condition indicates a foreign key from one table and its corresponding key in the other table.
- Specify the logical operator to compare values from the columns like =, <, or >.

Types of JOINS in SQL Server

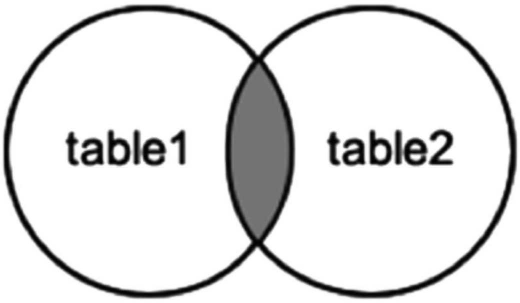
SQL Server mainly supports four types of JOINS, and each join type defines how two tables are related in a query. The following are types of join supports in SQL Server:

1. INNER JOIN
2. SELF JOIN
3. CROSS JOIN
4. OUTER JOIN

1. INNER JOIN

This JOIN returns all records from multiple tables that satisfy the specified join condition. It is the simple and most popular form of join and assumes as a default join. If we omit the INNER keyword with the JOIN query, we will get the same output.

The following visual representation explains how INNER JOIN returns the matching records from table1 and table2:



INNER JOIN Syntax

The following syntax illustrates the use of INNER JOIN in SQL Server:

```
SELECT columns
FROM table1
INNER JOIN table2 ON condition1
INNER JOIN table3 ON condition2
```

INNER JOIN Example

Let us first create two tables "Student" and "Fee" using the following statement:

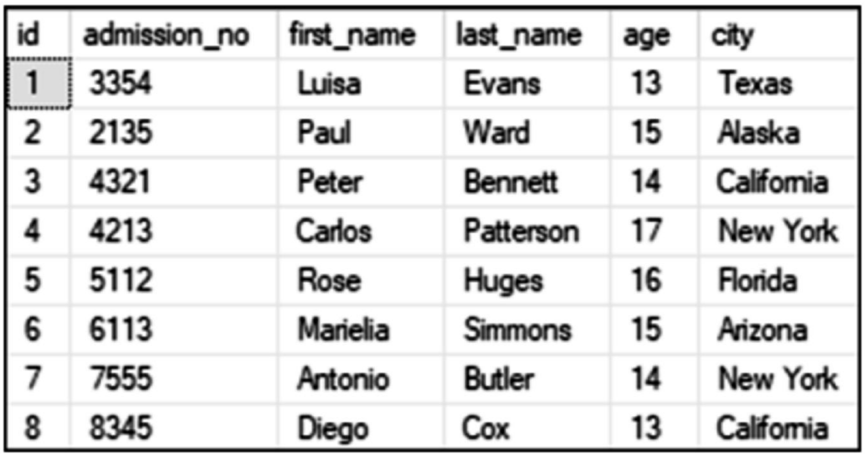
```
CREATE TABLE Student (
    id int PRIMARY KEY IDENTITY,
    admission_no varchar(45) NOT NULL,
    first_name varchar(45) NOT NULL,
    last_name varchar(45) NOT NULL,
    age int,
    city varchar(25) NOT NULL
    );
CREATE TABLE Fee (
    admission_no varchar(45) NOT NULL,
    course varchar(45) NOT NULL,
    amount_paid int,
    );
```

Next, we will insert some records into these tables using the below statements:

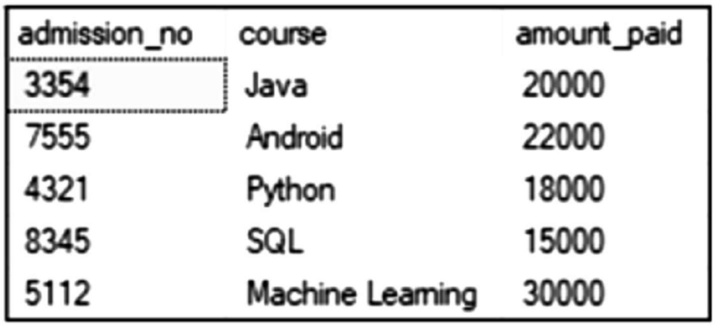
```
INSERT INTO Student (admission_no, first_name, last_name, age, city)
VALUES (3354,'Luisa', 'Evans', 13, 'Texas'),
(2135, 'Paul', 'Ward', 15, 'Alaska'),
(4321, 'Peter', 'Bennett', 14, 'California'),
(4213,'Carlos', 'Patterson', 17, 'New York'),
(5112, 'Rose', 'Huges', 16, 'Florida'),
(6113, 'Marielia', 'Simmons', 15, 'Arizona'),
(7555,'Antonio', 'Butler', 14, 'New York'),
(8345, 'Diego', 'Cox', 13, 'California');
INSERT INTO Fee (admission_no, course, amount_paid)
VALUES (3354,'Java', 20000),
(7555, 'Android', 22000),
(4321, 'Python', 18000),
(8345,'SQL', 15000),
(5112, 'Machine Learning', 30000);
```

Execute the **SELECT** statement to verify the records:

Table: Student



id	admission_no	first_name	last_name	age	city
1	3354	Luisa	Evans	13	Texas
2	2135	Paul	Ward	15	Alaska
3	4321	Peter	Bennett	14	California
4	4213	Carlos	Patterson	17	New York
5	5112	Rose	Huges	16	Florida
6	6113	Marielia	Simmons	15	Arizona
7	7555	Antonio	Butler	14	New York
8	8345	Diego	Cox	13	California



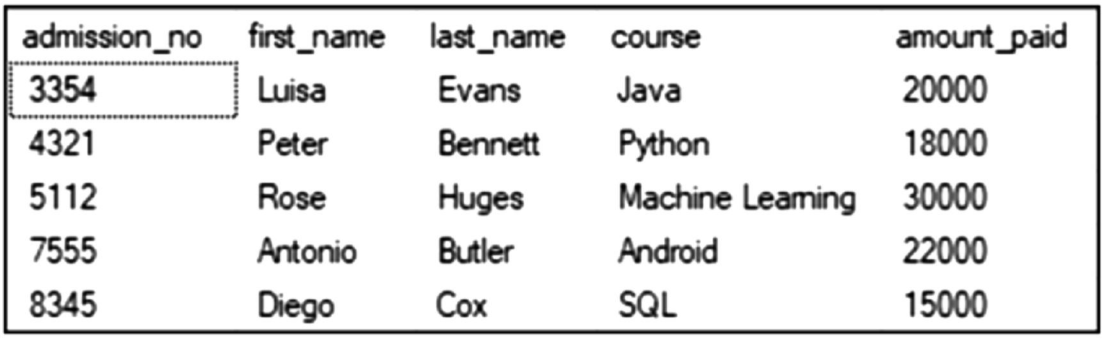
admission_no	course	amount_paid
3354	Java	20000
7555	Android	22000
4321	Python	18000
8345	SQL	15000
5112	Machine Learning	30000

We can demonstrate the INNER JOIN using the following command:

```
SELECT Student.admission_no, Student.first_name, Student.last_name, Fee.course,
Fee.amount_paid
FROM Student
INNER JOIN Fee
```

ON Student.admission_no = Fee.admission_no;

This command gives the below result:



admission_no	first_name	last_name	course	amount_paid
3354	Luisa	Evans	Java	20000
4321	Peter	Bennett	Python	18000
5112	Rose	Huges	Machine Leaming	30000
7555	Antonio	Butler	Android	22000
8345	Diego	Cox	SQL	15000

In this example, we have used the admission_no column as a join condition to get the data from both tables. Depending on this table, we can see the information of the students who have paid their fee.

2. SELF JOIN

A table is joined to itself using the SELF JOIN. It means that each table row is combined with itself and with every other table row. The SELF JOIN can be thought of as a JOIN of two copies of the same tables. We can do this with the help of table name aliases to assign a specific name to each table's instance. The table aliases enable us to use the table's temporary name that we are going to use in the query. It's a useful way to extract hierarchical data and comparing rows inside a single table.

SELF JOIN Syntax

The following expression illustrates the syntax of SELF JOIN in SQL Server. It works the same as the syntax of joining two different tables. Here, we use aliases names for tables because both the table name are the same.

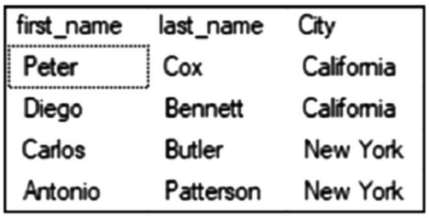
```
SELECT T1.col_name, T2.col_name...
FROM table1 T1, table1 T2
WHERE join_condition;
```

Example

We can demonstrate the SELF JOIN using the following command:

```
SELECT S1.first_name, S2.last_name, S2.city
FROM Student S1, Student S2
WHERE S1.id <> S2.iD AND S1.city = S2.city
ORDER BY S2.city;
```

This command gives the below result:



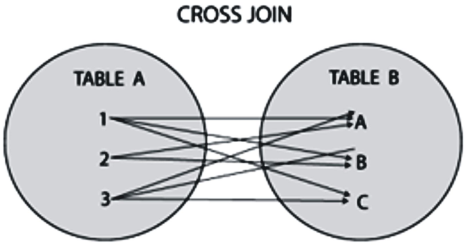
first_name	last_name	City
Peter	Cox	Califomia
Diego	Bennett	Califomia
Carlos	Butler	New York
Antonio	Patterson	New York

In this example, we have used the id and city column as a join condition to get the data from both tables.

3. CROSS JOIN

CROSS JOIN in SQL Server combines all of the possibilities of two or more tables and returns a result that includes every row from all contributing tables. It's also known as CARTESIAN JOIN because it produces the Cartesian product of all linked tables. The Cartesian product represents all rows present in the first table multiplied by all rows present in the second table.

The below visual representation illustrates the CROSS JOIN. It will give all the records from table1 and table2 where each row is the combination of rows of both tables:



CROSS JOIN Syntax

The following syntax illustrates the use of CROSS JOIN in SQL Server:

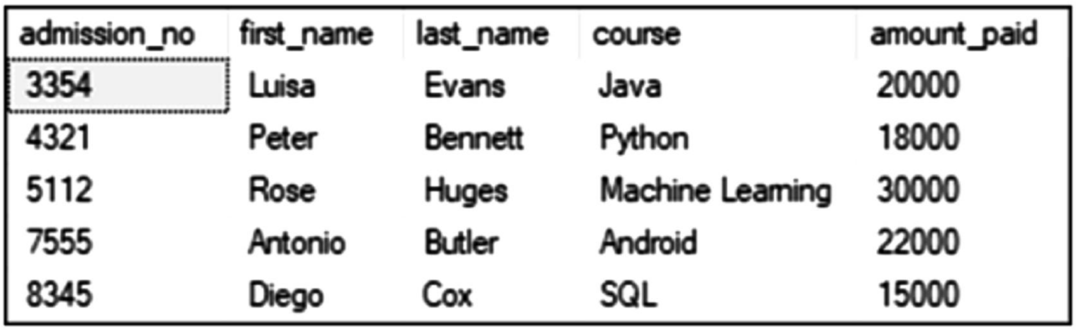
```
SELECT column_lists
FROM table1
CROSS JOIN table2;
```

Example

We can demonstrate the CROSS JOIN using the following command:

```
SELECT Student.admission_no, Student.first_name, Student.last_name, Fee.course,
Fee.amount_paid
FROM Student
CROSS JOIN Fee
WHERE Student.admission_no = Fee.admission_no;
```

This command gives the below result:



admission_no	first_name	last_name	course	amount_paid
3354	Luisa	Evans	Java	20000
4321	Peter	Bennett	Python	18000
5112	Rose	Huges	Machine Learning	30000
7555	Antonio	Butler	Android	22000
8345	Diego	Cox	SQL	15000

4. OUTER JOIN

OUTER JOIN in SQL Server returns all records from both tables that satisfy the join condition. In other words, this join will not return only the matching record but also return all unmatched rows from one or both tables.

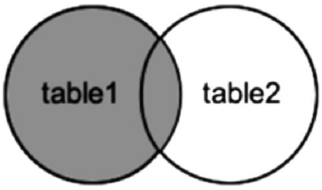
We can categories the OUTER JOIN further into three types:

- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN

LEFT OUTER JOIN

The LEFT OUTER JOIN retrieves all the records from the left table and matching rows from the right table. It will return NULL when no matching record is found in the right side table. Since OUTER is an optional keyword, it is also known as LEFT JOIN.

The below visual representation illustrates the LEFT OUTER JOIN:



LEFT OUTER JOIN Syntax

The following syntax illustrates the use of LEFT OUTER JOIN in SQL Server:

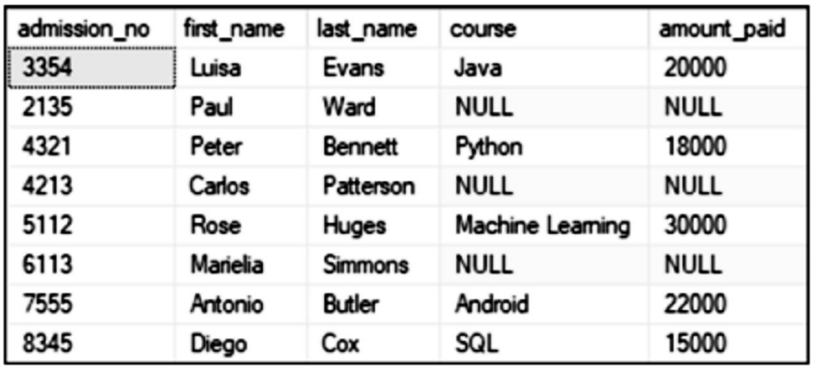
```
SELECT column_lists
FROM table1
LEFT [OUTER] JOIN table2
ON table1.column = table2.column;
```

Example

We can demonstrate the LEFT OUTER JOIN using the following command:

```
SELECT Student.admission_no, Student.first_name, Student.last_name, Fee.course,
Fee.amount_paid
FROM Student
LEFT OUTER JOIN Fee
ON Student.admission_no = Fee.admission_no;
```

This command gives the below result:



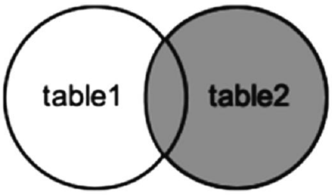
admission_no	first_name	last_name	course	amount_paid
3354	Luisa	Evans	Java	20000
2135	Paul	Ward	NULL	NULL
4321	Peter	Bennett	Python	18000
4213	Carlos	Patterson	NULL	NULL
5112	Rose	Huges	Machine Learning	30000
6113	Marielia	Simmons	NULL	NULL
7555	Antonio	Butler	Android	22000
8345	Diego	Cox	SQL	15000

This output shows that the unmatched row's values are replaced with NULLs in the respective columns.

RIGHT OUTER JOIN

The RIGHT OUTER JOIN retrieves all the records from the right-hand table and matched rows from the left-hand table. It will return NULL when no matching record is found in the left-hand table. Since OUTER is an optional keyword, it is also known as RIGHT JOIN.

The below visual representation illustrates the RIGHT OUTER JOIN:



RIGHT OUTER JOIN Syntax

The following syntax illustrates the use of RIGHT OUTER JOIN in SQL Server:

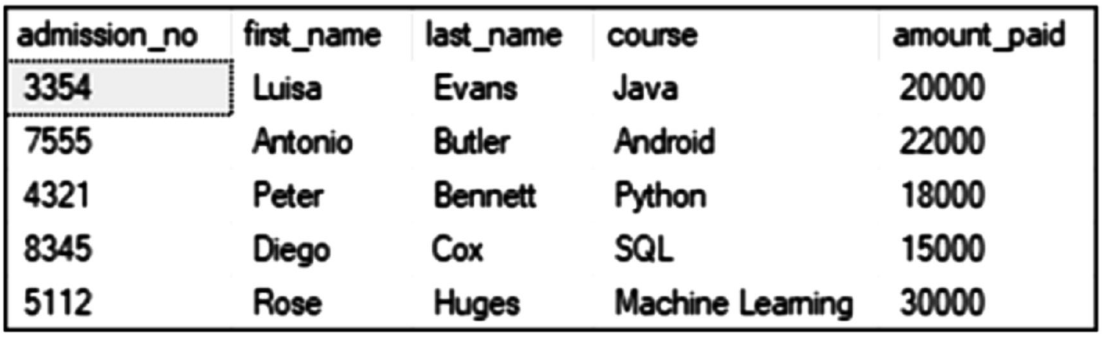
```
SELECT column_lists
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;
```

Example

The following example explains how to use the RIGHT OUTER JOIN to get records from both tables:

```
SELECT Student.admission_no, Student.first_name, Student.last_name, Fee.course,
Fee.amount_paid
FROM Student
RIGHT OUTER JOIN Fee
ON Student.admission_no = Fee.admission_no;
```

This command gives the below result:



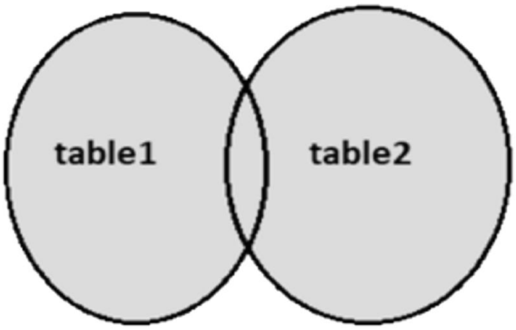
admission_no	first_name	last_name	course	amount_paid
3354	Luisa	Evans	Java	20000
7555	Antonio	Butler	Android	22000
4321	Peter	Bennett	Python	18000
8345	Diego	Cox	SQL	15000
5112	Rose	Huges	Machine Leaming	30000

In this output, we can see that no column has NULL values because all rows in the Fee table are available in the Student table based on the specified condition.

FULL OUTER JOIN

The FULL OUTER JOIN in SQL Server returns a result that includes all rows from both tables. The columns of the right-hand table return NULL when no matching records are found in the left-hand table. And if no matching records are found in the right-hand table, the left-hand table column returns NULL.

The below visual representation illustrates the FULL OUTER JOIN:



FULL OUTER JOIN Syntax

The following syntax illustrates the use of FULL OUTER JOIN in SQL Server:

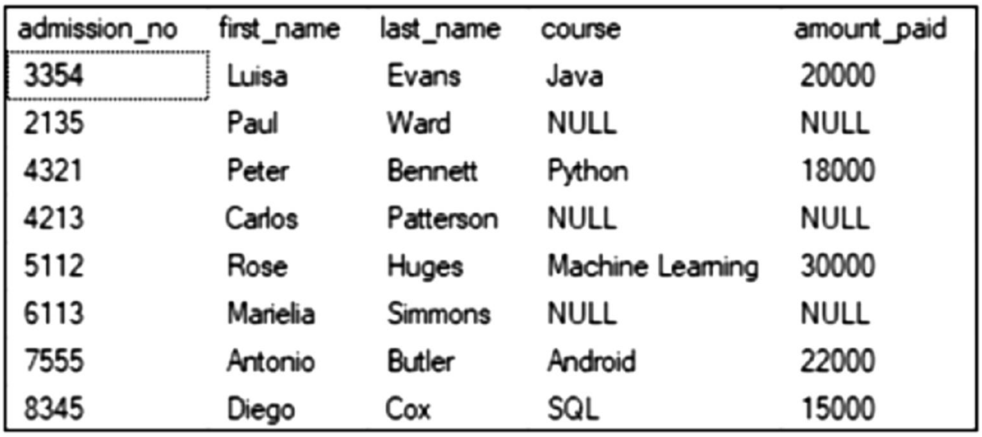
```
SELECT column_lists
FROM table1
FULL [OUTER] JOIN table2
ON table1.column = table2.column;
```

Example

The following example explains how to use the FULL OUTER JOIN to get records from both tables:

```
SELECT Student.admission_no, Student.first_name, Student.last_name, Fee.course,
Fee.amount_paid
FROM Student
FULL OUTER JOIN Fee
ON Student.admission_no = Fee.admission_no;
```

This command gives the below result:



admission_no	first_name	last_name	course	amount_paid
3354	Luisa	Evans	Java	20000
2135	Paul	Ward	NULL	NULL
4321	Peter	Bennett	Python	18000
4213	Carlos	Patterson	NULL	NULL
5112	Rose	Huges	Machine Leaming	30000
6113	Marielia	Simmons	NULL	NULL
7555	Antonio	Butler	Android	22000
8345	Diego	Cox	SQL	15000

In this output, we can see that the column has NULL values when no matching records are found in the left-hand and right-hand table based on the specified condition.

5.2.6 Exist

Q29. Explain about EXIST conditions in SQL.

Ans :

The EXISTS condition in SQL is used to check whether the result of a correlated nested query is empty (contains no tuples) or not. The result of EXISTS is a boolean value True or False. It can be used in a SELECT, UPDATE, INSERT or DELETE statement.

Syntax:

```
SELECT  column_name(s)
FROM table_name
WHERE EXISTS
    (SELECT  column_name(s)
FROM table_name
WHERE  condition);
```

Examples:

Consider the following two relation "Customers" and "Orders".

Customers

customer_id	lnanie	fname	website
401	Singh	Dolly	abc.com
402	Chauhan	Anuj	def.com
403	Kumar	Niteesh	ghi.com
404	Gupta	Shubham	jkl.com
405	Walecha	Divya	abc.com
406	Jain	Sandeep	jkl.com
407	Mehta	Rajiv	abc.com
408	Mehra	Anand	abc.com

Orders

order_id	c_id	order_date
1	407	2017-03-03
2	405	2017-03-05
3	408	2017-01-18
4	404	2017-02-05

Queries

Using EXISTS condition with SELECT statement

To fetch the first and last name of the customers who placed atleast one order.

SELECT fname, lname

```
FROM Customers
WHERE EXISTS (SELECT *
      FROM Orders
WHERE Customers.customer_id = Orders.c_id);
```

Output:

fname	Iname
Shubham	Gupta
Divya	Walecha
Rajiv	Mehta
Anand	Mehra

Using NOT with EXISTS

Fetch last and first name of the customers who has not placed any order.

```
SELECT lname, fname
FROM Customer
WHERE NOT EXISTS (SELECT * FROM Orders
      WHERE Customers.customer_id = Orders.c_id);
```

Output:

lname	fname
Singh	Dolly
Chauhan	Anuj
Kumar	Niteesh
Jain	Sandeep

Using EXISTS condition with DELETE statement

Delete the record of all the customer from Order Table whose last name is ‘Mehra’.

```
DELETE
FROM Orders
WHERE EXISTS (SELECT *
      FROM customers
      WHERE Customers.customer_id = Orders.cid
      AND Customers.lname = ‘Mehra’);
SELECT * FROM Orders;
```

Output:

order_id	c_id	order_date
1	407	2017-03-03
2	405	2017-03-05
4	404	2017-02-05

Using EXISTS condition with UPDATE statement

Update the lname as ‘Kumari’ of customer in Customer Table whose customer_id is 401.

```
UPDATE Customers
SET lname = ‘Kumari’
WHERE EXISTS (SELECT *FROM Customers
        WHERE customer_id = 401);
SELECT * FROM Customers;
```

Output:

custonier_id	lname	fname	website
401	Kumari	Dolly	abc.com
402	Chauhan	Anuj	def.com
403	Kumar	Niteesh	ghi.com
404	Gupta	Shubham	jkl.com
405	Walecha	Divya	abc.com
406	Jain	Sandeep	jkl.com
407	Mehta	Rajiv	abc.com
408	Mehra	Anand	abc.com

5.2.7 ANY

Q30. Explain about ANY statement in SQL.

Ans :

SQL ANY compares a value of the first table with all values of the second table and returns the row if there is a match with any value.

For example, if we want to find teachers whose age is similar to any of the student’s age, we can use

```
SELECT *
FROM Teachers
WHERE age = ANY (
        SELECT age
        FROM Students
    );
```

Here, the sub query

```
SELECT age
FROM Students
```

returns all the ages from the Students table. And, the condition

```
WHERE age = ANY (...)
```

compares the student ages (returned by subquery) with the teacher's age. If there is any match, the corresponding row of the Teachers table is selected.

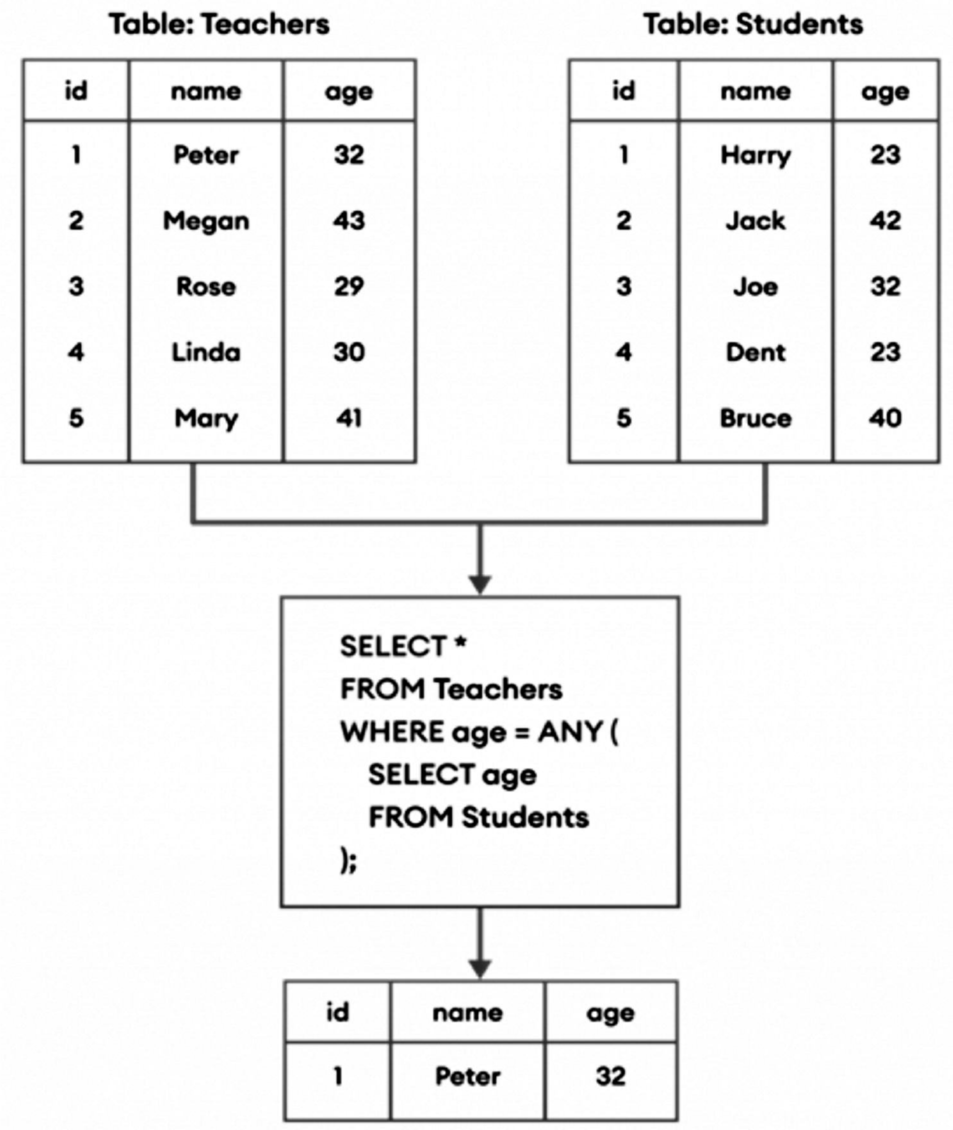


Table: Teachers

id	name	age
1	Peter	32
2	Megan	43
3	Rose	29
4	Linda	30
5	Mary	41

Table: Students

id	name	age
1	Harry	23
2	Jack	42
3	Joe	32
4	Dent	23
5	Bruce	40

```
SELECT *
FROM Teachers
WHERE age = ANY (
  SELECT age
  FROM Students
);
```

id	name	age
1	Peter	32

Example:

ANY in SQL

5.2.8 ALL

Q31. Explain ALL statement in SQL.

Ans :

SQL ALL compares a value of the first table with all values of the second table and returns the row if there is a match with all values.

For example, if we want to find teachers whose age is greater than all students, we can use

```
SELECT *
FROM Teachers
WHERE age >  ALL (
    SELECT age
FROM Students
);
```

Here, the sub query

```
SELECT age
FROM Students
```

returns all the ages from the Students table. And, the condition

```
WHERE age > ALL (...)
```

compares the student ages (returned by subquery) with the teacher's age. If the teacher's age is greater than all student's ages, the corresponding row of the Teachers table is selected.

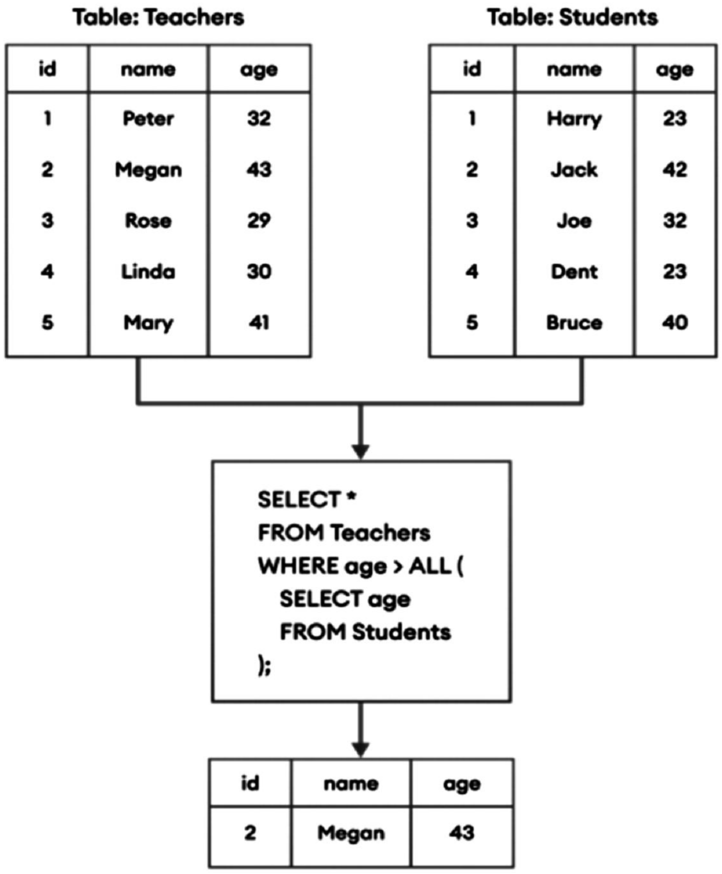


Table: Teachers

id	name	age
1	Peter	32
2	Megan	43
3	Rose	29
4	Linda	30
5	Mary	41

Table: Students

id	name	age
1	Harry	23
2	Jack	42
3	Joe	32
4	Dent	23
5	Bruce	40

```
SELECT *
FROM Teachers
WHERE age > ALL (
  SELECT age
  FROM Students
);
```

id	name	age
2	Megan	43

Example:

ALL in SQL

5.2.9 View and Its Types

Q32. What is view in SQL? Explain about it.

Ans : **(Imp.)**

A view is a database object that has no values. It is a virtual table, which is created according to the result set of an SQL query. However, it looks similar to an actual table containing rows and columns. Therefore, we can say that its contents are based on the base table. It is operated similarly to the base table but does not contain any data of its own. Its name is always unique, like tables. The views differ from tables as they are definitions that are created on top of other tables (or views). If any changes occur in the underlying table, the same changes reflected in the views also.

Uses of views

The primary use of view in SQL Server is to implement the security mechanism. It prevents users from seeing specific columns and rows from tables. It only shows the data returned by the query that was declared when the view was created. The rest of the information is completely hidden from the end-user.

Types of views

The SQL Server categories the views into two types:

1. **User-Defined Views**

Users define these views to meet their specific requirements. It can also divide into two types one is the simple view, and another is the complex view. The simple view is based on the single base table without using any complex queries. The complex view is based on more than one table along with group by clause, order by clause, and join conditions.

2. **System-Defined Views**

System-defined views are predefined and existing views stored in SQL Server, such as Tempdb, Master, and temp. Each system views has its own properties and functions. They can automatically attach to the user-defined databases. We can divide the System-defined views in SQL Server into three types: Information Schema, Catalog View, and Dynamic Management View.

SQL Server allows us to create a view in mainly two ways:

- Using T-SQL Query
- Using SQL Server Management Studio

SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

A view is created with the CREATE VIEW statement.

CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Note:

A view always shows up-to-date data! The database engine recreates the view, every time a user queries it.

SQL CREATE VIEW Examples

The following SQL creates a view that shows all customers from Brazil:

```
CREATE VIEW [Brazil Customers] AS
SELECT CustomerName, ContactName
FROM Customers
WHERE Country = 'Brazil';
```

We can query the view above as follows:

Example

```
SELECT * FROM [Brazil Customers];
```

The following SQL creates a view that selects every product in the “Products” table with a price higher than the average price:

Example

```
CREATE VIEW [Products Above Average Price] AS
SELECT ProductName, Price
FROM Products
WHERE Price > (SELECT AVG(Price) FROM Products);
```

We can query the view above as follows:

Example

```
SELECT * FROM [Products Above Average Price];
```

SQL Updating a View

A view can be updated with the CREATE OR REPLACE VIEW statement.

SQL CREATE OR REPLACE VIEW Syntax

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

The following SQL adds the “City” column to the “Brazil Customers” view:

Example

```
CREATE OR REPLACE VIEW [Brazil Customers] AS
SELECT CustomerName, ContactName, City
FROM Customers
WHERE Country = 'Brazil';
```

SQL Dropping a View

A view is deleted with the DROP VIEW statement.

SQL DROP VIEW Syntax

DROP VIEW view_name;

The following SQL drops the “Brazil Customers” view:

Example

DROP VIEW [Brazil Customers];

5.3 Transaction Control Commands

Q33. Explain about Transaction Control commands with examples.

Ans : **(Imp.)**

- In SQL, TCL stands for Transaction control language.
- A single unit of work in a database is formed after the consecutive execution of commands is known as a transaction.
- There are certain commands present in SQL known as TCL commands that help the user manage the transactions that take place in a database.
- COMMIT. ROLLBACK and SAVEPOINT are the most commonly used TCL commands in SQL.

1. COMMIT

COMMIT command in SQL is used to save all the transaction-related changes permanently to the disk. Whenever DDL commands such as INSERT, UPDATE and DELETE are used, the changes made by these commands are permanent only after closing the current session. So before closing the session, one can easily roll back the changes made by the DDL commands. Hence, if we want the changes to be saved permanently to the disk without closing the session, we will use the commit command.

Syntax:

COMMIT;

Example:

We will select an existing database, i.e., school.

mysql> USE school;

MySQL 5.5 Command Line Client

```
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.62 MySQL Community Server (GPL)
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type ‘help;* or ‘\h‘ for help. Type ‘\c’ to clear the current input statement.
mysql> USE school;
Oatabase changed
```

To create a table named t_school, we will execute the following query:

```
mysql> CREATE TABLE t_school(ID INT, School_Name VARCHAR(40), Number_Of_Students
INT,Number_Of_Teachers INT, Number_Of_Classrooms INT, EmailID VARCHAR(40));
```

```
mysql> CREATE TABLE t_school(ID INT, School_Name VARCHAR(40), Number_Of_Students
INT, Number_Of_Teachers INT, Number_Of

Classrooms INT, EmaillD VARCHAR(40));

Query OK, 0 rows affected (0.24 sec)
```

BEGIN / START TRANSACTION command is used to start the transaction.

```
mysql> START TRANSACTION;

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)
```

Now, we will execute the following query to insert multiple records at the same time in the t_school table.

```
mysql> INSERT INTO t_school(ID, School_Name, Number_Of_Students, Number_Of_
Teachers, Number_Of_Classrooms, EmailID) VALUES (1, ”Boys Town Public School”,
1000, 80, 12, ”btps15@gmail.com”),(2, ”Guru Govind Singh Public School”, 800, 35, 15, ”
ggps25@gmail.com”), (3, ”Delhi Public School”, 1200, 30, 10, ”dps101@gmail.com”),
(4, ”Ashoka Universal School”, 1110, 40, 40, ”aus17@gmail.com”),(5, ”Calibers English
Medium School”, 9000, 31, 50, ”cems@gmail.com”);
```

```
mysql> INSERT INTO t_school(ID, School_Name, Number_Of_Students, Number_Of_Teachers,
Number_Of_Classrooms, EmaillD) VALU ES(1, “Boys Town Public School”, 1000, 80, 12,
btpsl5@gmail.com”), (2, “Guru Govind Singh Public School”, 800, 35, 15, “ggps25@gmail.com”), (3,
“Delhi Public School”, 1200, 30, 10, “dpsl01@gmail.com”), (4, “Ashoka Universal School”, 1110, 40,
40, “ausl7@gmail.com”), (5, “Calibers English Medium School”, 9000, 31, 50, “cems@gmail.com”);
Query OK, 5 rows affected (0.06 sec) Records: 5 Duplicates: 0 Warnings: 0
```

We will now execute the SELECT query to verify the execution of the INSERT INTO query executed above.

```
mysql> SELECT *FROM t_school;
```

After executing the SELECT query on the t_school table, you will get the following output:

ID	School_Name	Number_Of_Students	Number_Of_Teachers	Number_Of_ Classrooms	EmailID
1	Boys Town Public School	1000	80	12	btps15@gmail.com
2	Guru Govind Singh Public School	800	35	15	ggps25@gmail.com
3	Delhi Public School	1200	30	10	dps101@gmail.com
4	Ashoka Universal School	1110	40	40	aus17@gmail.com
5	Calibers English Medium School	9000	31		

The output of the SELECT query shows that all the records are inserted successfully.

We will execute the COMMIT command to save the results of the operations carried on the t_school table.

```
mysql>  COMMIT;

mysql> COMMIT;

Query OK, 0 rows affected (0.03 sec)
```

Autocommit is by default enabled in MySQL. To turn it off, we will set the value of autocommit as 0.

```
mysql>  SET  autocommit  =  0;

mysql> SET autocommit = 0;

Query OK, 0 rows affected (0.08 sec)
```

MySQL, by default, commits every query the user executes. But if the user wishes to commit only the specific queries instead of committing every query, then turning off the autocommit is useful.

2. SAVEPOINT

We can divide the database operations into parts. For example, we can consider all the insert related queries that we will execute consecutively as one part of the transaction and the delete command as the other part of the transaction. Using the SAVEPOINT command in SQL, we can save these different parts of the same transaction using different names.

For example, we can save all the insert related queries with the savepoint named INS. To save all the insert related queries in one savepoint, we have to execute the SAVEPOINT query followed by the savepoint name after finishing the insert command execution.

Syntax:

```
SAVEPOINT  savepoint_name;
```

3. ROLLBACK

While carrying a transaction, we must create savepoints to save different parts of the transaction. According to the user's changing requirements, he/she can roll back the transaction to different savepoints. Consider a scenario: We have initiated a transaction followed by the table creation and record insertion into the table. After inserting records, we have created a savepoint INS. Then we executed a delete query, but later we thought that mistakenly we had removed the useful record. Therefore in such situations, we have an option of rolling back our transaction. In this case, we have to roll back our transaction using the ROLLBACK command to the savepoint INS, which we have created before executing the DELETE query.

Syntax:

```
ROLLBACK  TO  savepoint_name;
```

Examples to understand the SAVEPOINT and ROLLBACK commands:

Example 1:

We will select an existing database, i.e., school.

```
mysql>  USE  school;
```

MySQL 5.5 Command Line Client

```
Enter password: *****

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 4

Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h* for help. Type '\c' to clear the current input statement.

mysql> USE school;

Database changed
```

To create a table named t_school, we will execute the following query:

```
mysql> CREATE TABLE t_school(ID INT, School_Name VARCHAR(40),

Number_Of _Students INT, Number_Of_Teachers INT, Number_Of_Classrooms INT,
EmailID  VARCHAR(40));

mysql> CREATE TABLE t_school(ID INT, School_Name VARCHAR(40), Number_Of_Students
INT, Number_Of_Teachers INT, Number_Of

Classrooms INT, EmaillD VARCHAR(40));

Query OK, 0 rows affected (0.24 sec)
```

Now, we will execute the following query to insert multiple records at the same time in the t_school table.

```
mysql> INSERT INTO t_school(ID, School_Name, Number_Of_Students, Number_Of_
Teachers, Number_Of_Classrooms, EmailID) VALUES(1, "Boys Town Public School", 1000, 80, 12,
"btps15@gmail.com"), (2, "Guru Govind Singh Public School",800, 35, 15, "ggps25@gmail.com"),
(3, "Delhi Public School", 1200, 30, 10, "dps101@gmail.com"), (4, "Ashoka Universal School",
1110, 40, 40, "aus17@gmail.com"), (5, "Calibers English Medium School", 9000, 31, 50,
"cems@gmail.com");

mysql> INSERT INTO t_school(ID, School_Name, NumberOf Students, NumberOfTeachers,
NumberOfClassrooms, EmaillD) VALU

ES(1, "Boys Town Public School", 1000, 80, 12, "btpsl5@gmail.com"), (2, "Guru Govind Singh
Public School", 800, 35, 15,

"ggps25@gmail.com"), (3, "Delhi Public School", 1200, 30, 10, "dpsl01@gmail.com"), (4, "Ashoka
Universal School", 1110,

40, 40, "ausl7@gmail.com"), (5, "Calibers English Medium School", 9000, 31, 50,
"cems@gmail.com");

Query OK, 5 rows affected (0.06 sec)

Records: 5 Duplicates: 0 Warnings: 0
```

We will now execute the SELECT query to verify the execution of the INSERT INTO query executed above.

```
mysql>  SELECT  *FROM  t_school;
```

After executing the SELECT query on the t_school table, you will get the following output:

ID	School_Name	Number_Of_Students	Number_Of_Teachers	Number_Of_ Classrooms	EmailID
1	Boys Town Public School	1000	80	12	btps15@gmail.com
2	Guru Govind Singh Public School	800	35	15	ggps25@gmail.com
3	Delhi Public School	1200	30	10	dps101@gmail.com
4	Ashoka Universal School	1110	40	40	aus17@gmail.com
5	Calibers English Medium School	9000	31		

The output of the SELECT query shows that all the records are inserted successfully.

BEGIN / START TRANSACTION command is used to start the transaction.

```
mysql>  START  TRANSACTION;

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)
```

As we know, the SAVEPOINT command in SQL is used to save the different parts of the same transaction using different names. Consider till this point as one part of our transaction. We will save this part using a savepoint named Insertion.

```
mysql>  SAVEPOINT  Insertion;

mysql> SAVEPOINT Insertion;

Query OK, 0 rows affected (0.00 sec)
```

Now, we will execute the update command on the t_school table to set the Number_Of_Students as 9050 for the record with ID 5.

```
mysql>  UPDATE  t_school  SET  Number_Of_Students  =  9050  WHERE  ID  =  5;

mysql> UPDATE t_school SET Number_Of_Students = 9050 WHERE ID = 5;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0
```

To verify that the record with ID 5 now has the Number Of Students as 9050, we will

execute the SELECT query.

```
mysql> SELECT *FROM t school:
```

After executing the SELECT query on the t_school table, you will get the following output:

ID	School_Name	Number_Of_Students	Number_Of_Teachers	Number_Of_ Classrooms	EmailID
1	Boys Town Public School	1000	80	12	btps15@gmail.com
2	Guru Govind Singh Public School	800	35	15	ggps25@gmail.com
3	Delhi Public School	1200	30	10	dps101@gmail.com
4	Ashoka Universal School	1110	40	40	aus17@gmail.com
5	Calibers English Medium School	9050	31		

The output of the SELECT query shows that the record with ID 5 is updated successfully.

Consider the update operation as one part of our transaction. We will save this part using a savepoint named Updation.

mysql> SAVEPOINT Updation;

mysql> SAVEPOINT Updation;

Query OK, 0 rows affected (0.00 sec)

Suddenly, our requirement changed, and we realized that we had updated a record that was not supposed to be. In such a scenario, we need to roll back our transaction to the savepoint, which was created prior to the execution of the UPDATE command.

mysql> ROLLBACK TO Insertion;

mysql> ROLLBACK TO Insertion;

Query OK, 0 rows affected (0.09 sec)

We didn't need the updation carried on the record. Hence, we have rolled back to the savepoint named Insertion.

For confirming that we have got the same t_school table that we had before carrying out the updation operation, we will again execute the SELECT query.

mysql> SELECT *FROM t_school;

ID	School_Name	Number_Of_Students	Number_Of_Teachers	Number_Of_ Classrooms	EmailID
1	Boys Town Public School	1000	80	12	btps15@gmail.comm
2	Guru Govind Singh Public School	800	35	15	gps25@gmail.comm
3	Delhi Public School	1200	30	10	ps101@gmail.comm
4	Ashoka Universal School	1110	40	40	aus17@gmail.comm
5	Calibers English Medium School	9000	31		

The SELECT query output confirms that the transaction is now successfully rolled back to the savepoint 'Insertion'.

Example 2:

We will select an existing database, i.e., bank.

mysql> USE bank;

MySQL 5.5 Command Line Client

```
Enter password: *****

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 8

Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2060, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE bank;

Database changed
```

To create a table named customer, we will execute the following query:

```
mysql> CREATE TABLE customer(Customer_ID INT PRIMARY KEY, Name VARCHAR(20),
  Age INT, Salary INT, Salary_BankAccount VARCHAR(20));

mysql> CREATE TABLE customer(Customer_ID INT PRIMARY KEY, Name VARCHAR(20), Age
INT, Salary INT, Salary_BankAccount VARC

HAR (20));

Query OK, 0 rows affected (0.17 sec)
```

Now, we will execute the following query to insert multiple records at the same time in the customer table.|

```
mysql> INSERT INTO customer(Customer_ID, Name, Age, Salary, Salary_BankAccount) VALUES(l,
"Aryan Jain", 51, 56000, "SBI"), (2, "Arohi Dixit", 21,25000, "Axis"), (3, "Vin eetGarg", 24, 31000,
"ICICI"), (4, "Anuja Sharma", 26, 49000, "HDFC"), (5, "Deepak Kohli ", 28, 65000, "SBI");

mysql> INSERT INTO customer(Customer_ID, Name, Age, Salary, Salary_BankAccount) VALUES(1,
"Aryan lain", 51, 56000, "SBI"), (2, "Arohi Dixit", 21, 25000, "Axis"), (3, "Vineet Garg", 24, 31000,
"ICICI"), (4, "Anuja Sharma", 26, 49000, "HDFC"), (5, "Deepak Kohli", 28, 65000, "SBI");

Query OK, 5 rows affected (0.07 sec)

Records: 5 Duplicates: 0 Warnings: 0
```

We will now execute the SELECT query to verify the execution of the INSERT INTO query executed above.

```
mysql> SELECT *FROM customer;
```

After executing the SELECT query on the t_school table, you will get the following output:

Customer_ID	Name	Age	Salary	Salary_BankAccount
1	Aryan Jain	51	56000	SBI
2	Arohi Dixit	21	25000	Axis
3	Vineet Garg	24	31000	ICICI
4	Anuja Sharma	26	49000	HDFC
5	Deepak Kohli	28	65000	SBI

The output of the SELECT query shows that all the records are inserted successfully.

BEGIN / START TRANSACTION command is used to start the transaction.

```
mysql>  START  TRANSACTION;
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)
```

As we know, the SAVEPOINT command in SQL is used to save the different parts of the same transaction using different names. Consider till this point as one part of our transaction. We will save this part using a savepoint named Insertion.

```
mysql> SAVEPOINT Insertion;
mysql> SAVEPOINT Insertion;
Query OK, 0 rows affected (0.00 sec)
```

We will execute the delete command on the customer table to remove the record with ID 5.

```
mysql> DELETE FROM customer WHERE Customer ID = 5;
mysql> DELETE FROM customer WHERE Customer_ID = 5;
Query OK, 1 row affected (0.00 sec)
```

We will execute the SELECT query to verify that the record with ID 5 has been removed.

```
mysql> SELECT *FROM customer;
```

Customer_ID	Name	Age	Salary	Salary_BankAccount
1	Aryan Jain	51	56000	SBI
2	Arohi Dixit	21	25000	Axis
3	Vineet Garg	24	31000	ICICI
4	Anuja Sharma	26	49000	HDFC

The output of the SELECT query shows that the record with ID 5 is removed successfully.

Consider the delete operation as one part of our transaction. We will save this part using a savepoint named Deletion.

```
mysql>  SAVEPOINT  Deletion;
```

Short Question and Answers

1. SQL.

Ans :

SQL is a short-form of the structured query language, and it is pronounced as S-Q-L or sometimes as See-Quell.

This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables). It is also designed for stream processing in RDSMS.

You can easily create and manipulate the database, access and modify the table rows and columns, etc. This query language became the standard of ANSI in the year of 1986 and ISO in the year of 1987.

If you want to get a job in the field of data science, then it is the most important query language to learn. Big enterprises like Facebook, Instagram, and LinkedIn, use SQL for storing the data in the back-end.

2. DDL Commands.

Ans :

The DDL Commands in Structured Query Language are used to create and modify the schema of the database and its objects. The syntax of DDL commands is predefined for describing the data. The commands of Data Definition Language deal with how the data should exist in the database.

Following are the five DDL commands in SQL:

1. CREATE Command
2. DROP Command
3. ALTER Command
4. TRUNCATE Command
5. RENAME Command

3. DML Commands.

Ans :

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

Following are the four main DML commands in SQL:

1. SELECT Command
2. INSERT Command
3. UPDATE Command
4. DELETE Command

4. SQL UPDATE

Ans :

The SQL commands (UPDATE and DELETE) are used to modify the data that is already in the database. The SQL DELETE command uses a WHERE clause.

SQL UPDATE statement is used to change the data of the records held by tables. Which rows is to be update, it is decided by a condition. To specify condition, we use WHERE clause.

The UPDATE statement can be written in following form:

The SQL UPDATE statement is used to edit existing rows in a database table.

5. DCL Commands.

Ans :

Data Control Language(DCL) is used to control privileges in Database. To perform any operation in the database, such as for creating tables, sequences or views, a user needs privileges. Privileges are of two types,

- **System:** This includes permissions for creating session, table, etc and all types of other system privileges.
- **Object:** This includes permissions for any command or query to perform any operation on the database tables.

6. Explain CREATE Command in SQL.

Ans :

SQL CREATE TABLE statement is used to create table in a database.

If you want to create a table, you should name the table and define its column and each column's data type.

Let's see the simple syntax to create the table.

```
create table ”tablename”
(“column1” ”data type”,
“column2” ”data type”,
“column3” ”data type”,
...
“columnN” ”data type”);
```

7. Constraints in SQL.

Ans :

SQL constraints are rules that you can imply on the data in a table. It allows you to restrict only specific data that meets the regulations to go to a table. To put it simply, only if the data meets the constraint's rules, the insert operation will be successful, or else it will be aborted.

Constraints in SQL helps to maintain the accuracy, integrity, and reliability of a table's data. It can create them at a column or table level. If you declare constraints at the column level, it will apply them to a single column. On the other hand, if you declare them at the table level, it will implement them in more than one column. You can create constraints in SQL while creating a table using the CREATE TABLE command or later using the ALTER TABLE command. If you make a constraint with the ALTER TABLE command, the creation will only be successful if all the existing data meets the constraint rules.

8. Primary Key.

Ans :

A column or columns is called primary key (PK) that uniquely identifies each row in the table.

If you want to create a primary key, you should define a PRIMARY KEY constraint when you create or modify a table.

When multiple columns are used as a primary key, it is known as composite primary key.

In designing the composite primary key, you should use as few columns as possible. It is good for storage and performance both, the more columns you use for primary key the more storage space you require.

Points to remember for primary key:

- Primary key enforces the entity integrity of the table.
- Primary key always has unique data.
- A primary key length cannot be exceeded than 900 bytes.
- A primary key cannot have null value.
- There can be no duplicate value for a primary key.
- A table can contain only one primary key constraint.

9. Foreign Key.

Ans :

In the relational databases, a foreign key is a field or a column that is used to establish a link between two tables.In simple words you can say that, a foreign key in one table used to point primary key in another table.

Let us take an example to explain it:

Here are two tables first one is students table and second is orders table.Here orders are given by students.

First table:

S_Id	Last Name	First Name	CITY
1	MAURYA	AJEET	ALLAHABAD
2	JAISWAL	RATAN	GHAZIABAD
3	ARORA	SAUMYA	MODINAGAR

First table:

O_Id	OrderNo	S_Id
1	99586465	2
2	78466588	2
3	22354846	3
4	57698656	1

- The "S_Id" column in the "Students" table is the PRIMARY KEY in the "Students" table.
- The "S_Id" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

The foreign key constraint is generally prevents action that destroy links between tables.

It also prevents invalid data to enter in foreign key column.

10. UNIQUE.

Ans :

A unique key is a set of one or more than one fields/columns of a table that uniquely identify a record in a database table.

You can say that it is little like primary key but it can accept only one null value and it cannot have duplicate values.

The unique key and primary key both provide a guarantee for uniqueness for a column or a set of columns.

There is an automatically defined unique key constraint within a primary key constraint.

There may be many unique key constraints for one table, but only one PRIMARY KEY constraint for one table.

Choose the Correct Answers

1. PL/SQL is a ________. [b]

   (a) Brick Structured Language (b) Block Structured Language

   (c) Banner Structured Language (d) Build Structured Language

2. What does PL/SQL stand for? [a]

   (a) PL/SQL stands for Procedural Language Extension of SQL

   (b) PL/SQL stands for Primary Language Extension of SQL

   (c) PL/SQL stands for Pattern Language Extension of SQL

   (d) PL/SQL stands for Private Language Extension of SQL

3. What is TRUE about PL/SQL functionalities? [d]

   (a) Conditions and loops are fundamental elements of procedural languages like PL/SQL.

   (b) Various types and variables can be declared, as can procedures and functions, as well as types and variables of those types.

   (c) Arrays can be used with it as well as handling exceptions (runtime errors).

   (d) All of the above

4. Oracle Database's ________ are inherited in PL/SQL. [d]

   (a) Portability (b) Robustness

   (c) Security (d) All of the above

5. PL/SQL text is made up of lexical units, which are groups of characters and can be classified as ________. [d]

   (a) Delimiters (b) Identifiers

   (c) Literals (d) All of the above

6. A Variable in PL/SQL should not exceed ________. [c]

   (a) 10 (b) 20

   (c) 30 (d) 40

7. Which of the following is/are TRUE about PL/SQL Variables? [d]

   (a) Variables serve as a means for programmers to temporarily store data during code execution.

   (b) PL/SQL programs benefit from its use.

   (c) There is nothing special about it other than being the name of a storage area.

   (d) All of the above

8. PL/SQL Variables are by default ________. [d]

(a) Case Sensitive (b) Upper Case Sensitive

(c) Lower Case Sensitive (d) Not Case Sensitive

9. PL/SQL Variable needs to be declared in the ________. [b]

(a) Variable Section (b) Declaration Section

(c) Initialization Section (d) None of the above

10. The correct syntax to declare PL/SQL variable is ________. [a]

(a) variable_name [CONSTANT] datatype [NOT NULL] [:= | DEFAULT initial_value]

(b) datatype [CONSTANT] variable_name [NOT NULL] [:= | DEFAULT initial_value]

(c) variable_name [CONSTANT] datatype [NULL] [:= | DEFAULT initial_value]

(d) datatype [CONSTANT] variable_name [NULL] [:= | DEFAULT initial_value]

Fill in the blanks

1. ________ allows the data professionals and users to retrieve the data from the relational database management systems.
2. RENAME is a ________ which is used to change the name of the database table.
3. The ________ statement is used to delete rows from a table.
4. DCL stands for ________.
5. ________ are rules that you can imply on the data in a table.
6. When multiple columns are used as a primary key, it is known as ________ primary key.
7. A ________ is a set of one or more than one fields/columns of a table that uniquely identify a record in a database table.
8. The ________ is a constraint in SQL which does not allow you to insert NULL values in the specified column.
9. ________ are used to perform operations on numbers and return numbers.
10. ________ are used to perform an operation on input string and return an output string.

Answers

1. SQL
2. DDL command
3. SQL DELETE
4. Data Control Language
5. SQL containts
6. Composite
7. Unique key
8. NOT NULL
9. Numeric Functions
10. String functions

LAB PRACTICALS

Experiment 1: Problem statement

ROADWAY TRAVELS

"Roadway Travels" is in business since 1977 with several buses connecting different places in India. Its main office is located in Hyderabad. The company wants to computerize its operations in the following areas:

Reservations

Ticketing

Cancellations

Reservations :

Reservations are directly handeled by booking office.reservations can be made 60 days in advance in either cash or credit. In case the ticket is not available,a wait listed ticket is issued to the customer. This ticket is confirmed against the cancellation.

Cancellation and modification:

Cancellations are also directly handed at the booking office. Cancellation charges will be charged.Wait listed tickets that do not get confirmed are fully refunded.

AIM: Analyze the problem and come with the entities in it. Identify what Data has to be persisted in the databases.

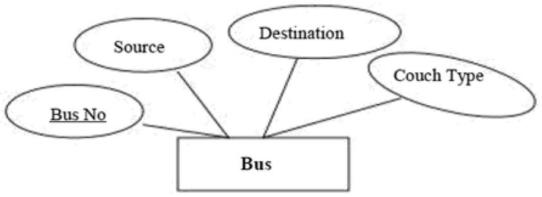
The Following are the entities:

1. Bus
2. Reservation
3. Ticket
4. Passenger
5. Cancellation

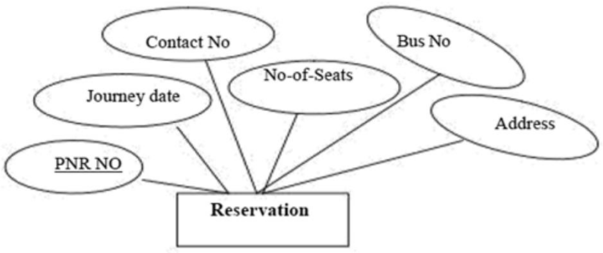
Experiment 2:

The attributes in the Entities:

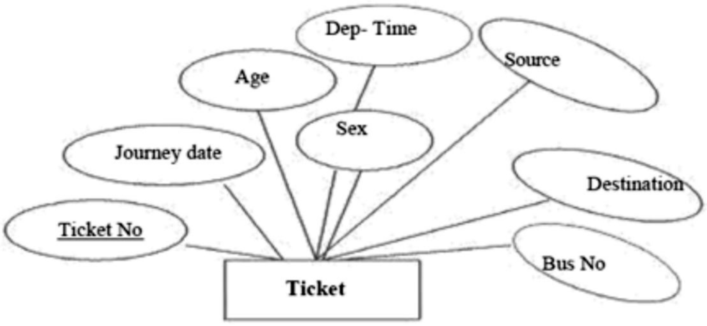
Bus:(Entity)



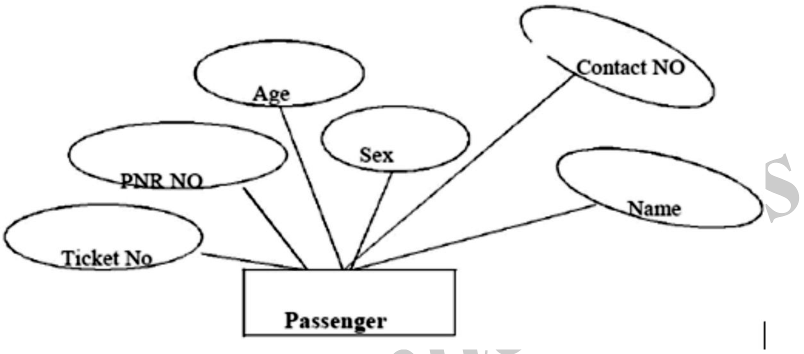
Reservation (Entity)



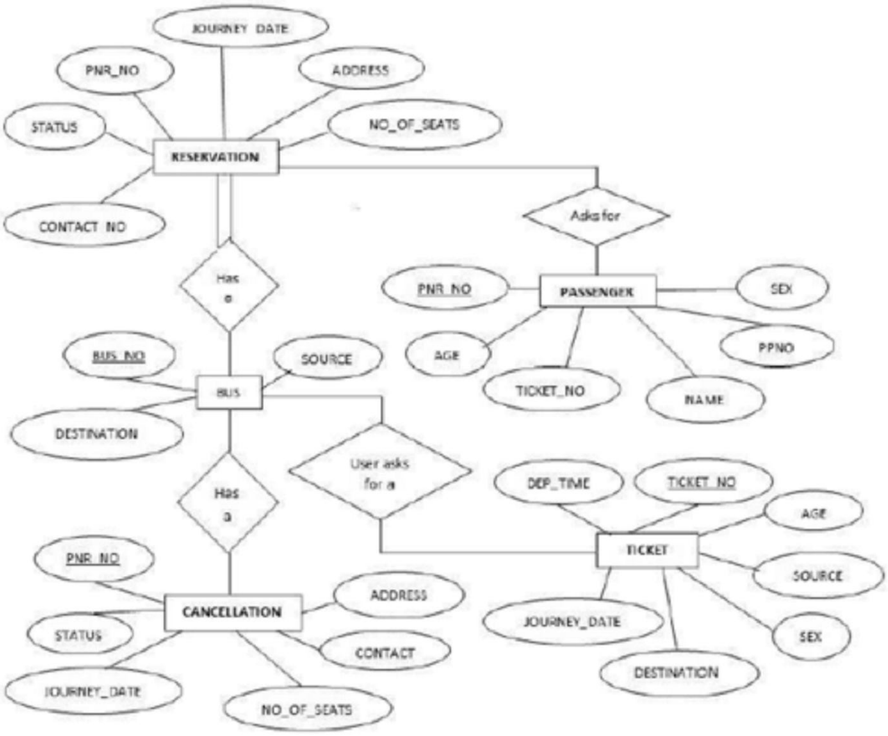
Ticket : (Entity)



Passenger :



Cancellation(Entity)



Experiment 3, 4 :

The following are tabular representation of the above entities and relationships

BUS:

COLOUME NAME	DATA TYPE	CONSTRAINT
Bus No	var char2(10)	Pr imary Key
Source	var char2(20)	
Destination	var char2(20)	
Couch Type	var char2(20)	

Reservation :

COLOUME NAME	DATA TYPE	CONSTRAINT
PNR No	number(9)	Primary Key
Journey date	Date	
No-of-seats	integer(8)	
Address	varchar2(50)	
Contact No	Number(9)	Should be equal to 10 numbers and not allow other than numeric
Bus No	varchar2(10)	Foreign key
Seat no	Number	

Ticket :

COLOUME NAME	DATA TYPE	CONSTRAINT
Ticket_No	number(9)	Primary Key
Journey date	Date	
Age	int(4)	
Sex	Char(10)	
Source	varchar2(10)	
Destination	varchar2(10)	
Dep-time	varchar2(10)	
Bus No	Number2(10)	

Passenger :

COLOUME NAME	DATA TYPE	CONSTRAINT
PNR No	Number(9)	Primary Key
Ticket No	Number(9)	Foreign key
Name	varchar2(15)	
Age	interger(4)	
Sex	char(10)	(Male / Female)
Contact no	Number(9)	Should be equal to 10 number and not allow other than numeric

Cancellation :

COLOUME NAME	DATA TYPE	CONSTRAINT
PNR No	Number(9)	Foriegn-key
Journey-date	Date	
Seat no	Integer(9)	
Contact_No	Number(9)	Should be equal to 10 numbers and not allow other than numeric

Experiment 5:

Creating of Tables on ROAD WAY TRAVELS:

Table is a primary object of database, used to store data in form of rows and columns. It is created using following command:

Create Table <table_name> (column1 datatype(size), column2 datatype(size),column(n) datatype(size));

Example:

```
SQL> create table Bus(Bus_Novarchar(5), source varchar(20), destination
varchar(20),CouchType varchar2(10),fair number);
```

Table Created.

create table for the object-relation feature we will discuss it afterwards.

Desc command

Describe command is external command of Oracle. The describe command is used to view the structure of a table as follows.

```
Desc<table name>
SQL>desc bus;
Name Null? Type
```

BUS_NO NOT NULL INTEGER2(5)

SOURCE VARCHAR2(20)

DESTINATION VARCHAR2(20)

COUCH TYPE VARCHAR2(10)

FAIR NUMBER

SQL> Describe the university database

Reservation Table:

SQL> create table Reservation(PNR_NO Numeric(9), No_of_seats Number(8), Address

varchar(50), Contact_No Numeric(9), Status char(3));

Table created.

SQL>desc Reservation

Name Null? Type

PNR_NO NUMBER(9)

NO_OF_SEATS NUMBER(8)

ADDRESS VARCHAR2(50)

CONTACT_NO NUMBER(9)

STATUS CHAR(3)

Cancellation Table:

SQL> create table Cancellation(PNR_NO Numeric(9), No_of_seats Number(8), Address

varchar(50), Contact_N o Numeric(9), Status char(3));

Table created.

SQL>desc Cancellation

Name Null? Type

PNR_NO NUMBER(9)

NO_OF_SEATS NUMBER(8)

ADDRESS VARCHAR2(50)

CONTACT_NO NUMBER(9)

STATUS CHAR(3)

Ticket Table:

SQL> create table Ticket(Ticket_No Numeric(9) primary key, age number(4), sex char(4)

Not null, source varchar(2), destination varchar(20), dep_timevarchar(4));

Table created.

SQL>desc Ticket

Name Null? Type

TICKET_NO NOT NULL NUMBER(9)

AGE NUMBER(4)

SEX NOT NULL CHAR(4)

SOURCE VARCHAR2(2)

DESTINATION VARCHAR2(20)

DEP_TIME VARCHAR2(4)

Alteration of Table

Addition of Column(s)

Addition of column in table is done using:

Alter table <table_name>add(column1 data type, column2 datatype _);

SQL> ALTER TABLE Passenger ADD FOREIGN KEY (PNR_NO) REFERENCES

Reservation(PNR_NO);

Table altered.

SQL> ALTER TABLE Cancellation ADD FOREIGN KEY (PNR_NO) REFERENCES

Reservation(PNR_NO);

Table altered.

Applying Constraints on Road Way Travels Tables.Constraints

Domain Integrit y constraints

Entity Integrity constraints

Referential Integrity constraint

Oracle allows programmers to define constraints

Column Level

Table Level

Example

SQL> create table Ticket (Ticket_No Numeric(9) , age number(4), sex char(4) Not null, sourcevarchar(2), destination varchar(20), dep_timevarchar(4));

Table created.

Check Constraint

SQL> create table Reservation(PNR_NO Numeric(9), No_of_seats Number(8), Address varchar(50), Contact_No Numeric(10) constraint ck check(length(contact_no)=10), Status char(3));

Unique Constraint

Example:

SQL> create table Ticket(Ticket_No Numeric (9) unique, age number

Primary key constraint at the column level

Example:

SQL> create table Ticket(Ticket_No Numeric(9) constraint pk primary key, age number(4), sex char(4) l, source varchar(2), destination varchar(20), dep_timevarchar(4));

Table created.

References constraint defined at column level

Example:

SQL> create table Passenger(PNR_NO Numeric(9) references r eservation , Ticket_NO

Numeric(9) references ticket, Name varchar(20), Age Number(4), Sex char(10), PPNO varchar(15)); Table created.

Foreign Key Constraint with alter command

SQL> alter table reservation add constraint fk_icode foreign key (busno) references bus(bus_no);

Experiment 6: Practicing DML commands

(a) Insert

Insert command

Insert into <table name>values(a list of data values);

Insert into <table name>(column list) values(a list of data);

SQL> insert into emp_master (empno, ename, salary) values (1122,'Smith',8000); 1 row created.

Adding values in a table using Variable method.

SQL> insert into Passenger values (& PNR_ NO, &TICKET_NO, '&Name', &Age, '&Sex', ' & PPNO');

Enter value for pnr_no: 1

Enter value for ticket_no: 1

Enter value for name: SACHIN

Enter value for age: 12

Enter value for sex: m

Enter value for ppno: sd1234

old 1: insert into Passenger values (& PNR_ NO, & TICKET_NO, '&Name', &Age, '&Sex', '& PPNO')

new 1: insert into Passenger values (1,1, 'SACHIN',12,'m','sd1234')

1 row created.

SQL> insert into Bus values ('& Bus_ No','&source','&destination');

Enter value for bus_no: 1

Enter value for source: hyd

Enter value for destination: ban

old 1: insert into Bus values ('&Bus_ No',' & source',' & destination')

new 1: insert into Bus values('1','hyd','ban')

1 row created.

(b) Select

Select Command

Select <column1>, <column2> ,_, <column (n) > from <table name>;

SQL> select * from emp_master;

Test Output:

selectempno, ename, salary from emp_master;

Test Output:

SQL> select * from Passenger;

Test Output:

SQL> select distinct deptno from emp_master;

Test Output:

(c) Update

SQL> update Passenger set age='43' where PNR_NO='2';

Test Output:

SQL>Select*from passenger;

Test Output:

(d) Delete

SQL> delete from Passenger where age< 0;

Experiment : 7 : Aim: Practice queries using ANY, ALL, IN, EXISTS, UNION, INTERSECT

Union: The union operator returns all distinct rows selected by two or more queries.

SQL> select order_no from order_master;

Test Output:

SQL> select order_no from order_detail;

Test Output:

SQL>select order_no from order_master union select order_no from

order_detail;

Test Output:

Union All :

Example:

SQL> select order_no from order_master union all select order_no from

order_detail.

Test Output

Intersect :

Example:

SQL> select order_no from order_master intersect select order_no from

order_detail;

Test Output:

Minus :

Example:

SQL> select order_no from order_master minus select order_no from order_detail;

Test Output:

Any command :

SELECT * FROM Teachers WHERE age = ANY (SELECT age FROM Students);

SQL ALL

SELECT * FROM Teachers WHERE age >ALL (SELECT age FROM Students);

SQL EXISTS

SELECTcustomer_id, first_ name FROM Customers WHEREEXISTS (SELECTorder_id

FROM Orders WHEREOrders.customer_id = Customers. customer_id);

Constraints:

UNIQUE Constraint in SQL

CREATE TABLE Student1(ID int NOT NULL, Name varchar(25) NOT NULL,

Age int, Email_IdNVARCHAR(50) UNIQUE);

INSERT INTO Student1 VALUES (1, 'Aakash', 21, 'ak@12');

INSERT INTO Student1 VALUES (2, 'George', null, 'go@45');

SELECT * FROM Student1;

— trying to insert duplicate values

INSERT INTO Student1 VALUES (3, 'Rahul', 21, 'go@45');

PRIMARY KEY Constraint in SQL

CREATE TABLE Student2(Name varchar(25) NOT NULL, ID int PRIMARY KEY,

Age int, Email_IdNVARCHAR(50) UNI QUE);

INSERT INTO Student2 VALUES ('Aakash', 1, 21, 'ak@12');

INSERT INTO Student2 VALUES ('George', 2, NULL, 'go@45');

INSERT INTO Student2 VALUES ('Rahul', 1, 21, 'rh@67');

INSERT INTO Student2 VALUES (NULL, 3, 23,'mr@89');

SELECT * FROM Student2;

FOREIGN KEY Constraint in SQL

CREATE TABLE ORDERS(O_ID int PRIMARY KEY, ORDER_NO int UNIQUE,

C_ID int, FOREIGN KEY (C_ID) REFERENCES CUSTOMERS(C_ID));

INSERT INTO ORDERS VALUES (1, 2212, 3);

INSERT INTO ORDERS VALUES (2, 2015, 1);

INSERT INTO ORDERS VALUES (3, 1983, 1);

INSERT INTO ORDERS VALUES (4, 1502, 2);

CREATE TABLE CUSTOMERS(

C_ID int PRIMARY KEY,

NAME VARCHAR(25) NOT NULL,

CITY VARCHAR(20)

);

```
INSERT INTO CUSTOMERS VALUES (1, ‘Aakash’, ‘MUMBAI’);

INSERT INTO CUSTOMERS VALUES (2, ‘George’, ‘DELHI’);

INSERT INTO CUSTOMERS VALUES (3, ‘Rahul’, ‘AHMEDABAD’);

SELECT * FROM ORDERS;

SELECT * FROM CUSTOMERS;
```

Experiment 8: Practising sub queries

(a) Subquery

Example:

```
SQL> select * from order_master where order_no = (select order_no from order_detail where

order_no = ‘0001’);
```

Test Output:

Example:

```
SQL> select * from order_master where order_no = (select order_no from order_detail);
```

Test Output

Example:

```
SQL>Select * from order_master where order_no = any(select order_no from order_detail);
```

Test Output:

```
SQL> select * from order_master where order_noin(select order_no from order_detail);
```

Test Output

Example:

```
SQL> select * from order_detail where qty_ord =all(select qty_hand from itemfile where itemrate =250);
```

Test Output:

(b) Implement Joins

Syntax for joining tables

select columns from table1, table2, ... where logical expression;

Simple Join :

Example:

```
SQL> select * from order_master ,order_detail where Order_master.order_no = order_ detail.order_no;
```

Test Output:

Example:

```
SQL> select a.*, b.* from itemfile a, order_detail b where a.max_level<b.qty_ord anda.itemcode = b.itemcode;
```

Test Output:

Self Join :

79

Example:

```
SQL> select a.ename, a.salary, b.ename, b.salary from emp a, emp b where a.mgr = b.empno;
```

Outer Join :

Example:

```
SQL> select * from order_master a, order_detail b where a.order_no = b.order_no(+);
```

Test Output:

Experiment 9: Practising COUNT, SUM , SVG, MAX, MIN, GROUPBY, HAVING, VIEW

1) **Avg (Average)**

This function will return the average of values of the column specified in the argument of the column.

Example:

```
SQL> select avg(comm) from emp_master;
```

Test Output:

2) **Min (Minimum):**

Example:

```
SQL>Select min(salary) from emp_master;
```

Test Output:

3) **Max (Maximum):**

53

Example:

```
SQL>select max(salary) from emp_master;
```

Test Output:

4) **Sum:**

Example:

```
SQL>Select sum(comm) from emp_master;
```

Test Output:

5) **Count:**

Syntax: Count(*)

Count(column name)

Count(distinct column name

Example:

```
SQL>Select count(*) from emp_master;
```

Test Output:

Example:

SQL> select count(comm) from emp_master;

Test Output:

Example:

SQL>Select count(distinct deptno) from emp_master;

Test Output:

6. **Group By Clause**

Example:

SQL>select deptno,count(*) from emp_master group by deptno;

Test Output

7. **Having Clause**

Example

SQL> select deptno,count(*) from emp_master group by deptno having Deptno is not null;

Test Output:

8. **Order By Clause**

Select<column(s)>from<Table Name> where[condition(s)][order by<column name> [asc /] desc];

Example:

SQL> select empno,ename,salary from emp_master order by salary;

Test Output:

Example

SQL> select empno,ename,salary from emp_master order by salary desc;

Test Output:

9. **Views**

Syntax: Create View <View_Name> As Select statement;

Example:

SQL>Create View EmpViewAs Select * from Employee;

View created.

Syntax: Selectcolumnname,columnname from <View_Name>;

Example:

SQL>Select Empno,Ename,Salary from EmpView where Deptno in(10,30);

Test Output:

Updatable Views:

Syntax for creating an Updatable View:

Create View Emp_vw As

Select Empno,Ename,Deptno from Employee;

View created.

SQL>Insert into Emp_ vwvalues (1126,' Brijesh', 20);

SQL>Update Emp_vw set Deptno=30 where Empno=1125;

1 row updated.

SQL>Delete from Emp_vw where Empno =1122;

View defined from Multiple tables (Which have no Referencing clause):

For insert/modify:

Test Output:

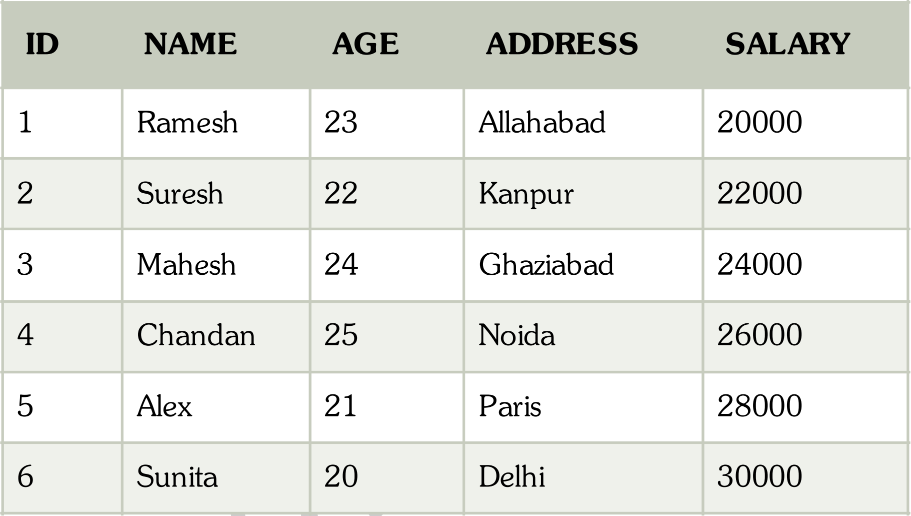
View defined from Multiple tables (Which have been created with a Referencing clause):

Syntax for creating a Master/Detail View (Join View):

```
SQL>Create View EmpDept_Vw As
Select a. Empno, a.Ename, a.Salary, a.Deptno, b.Dname From Employee a,DeptDet b
Where a.Deptno=b.Deptno;
```

View created.

Test Output:



ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	23	Allahabad	20000
2	Suresh	22	Kanpur	22000
3	Mahesh	24	Ghaziabad	24000
4	Chandan	25	Noida	26000
5	Alex	21	Paris	28000
6	Sunita	20	Delhi	30000

Create trigger:

Let’s take a program to create a row level trigger for the CUSTOMERS table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON customers
FOR EACH ROW
WHEN (NEW.ID > 0)
DECLARE
sal_diff number;
BEGIN
sal_diff := :NEW.salary  - :OLD.salary;
dbms_output.put_line(‘Old salary: ’ || :OLD.salary);
dbms_output.put_line(‘New salary: ’ || :NEW.salary);
dbms_output.put_line(‘Salary difference: ’ || sal_diff);
END;
```

/

After the execution of the above code at SQL Prompt, it produces the following result.

Trigger created.

(b) Insertion using Trigger

Let us perform some DML operations on the CUSTOMERS table. Here is one INSERT statement, which will create a new record in the table “

```
INSERT INTO CUSTOMERS (ID,NAME, AGE, ADDRESS,SALARY)
VALUES (7, 'Kriti', 22, 'HP', 7500.00 );
```

When a record is created in the CUSTOMERS table, the above create trigger, display_ salary_changes will be fired and it will display the following result -

```
Old salary:
New salary: 7500
Salary difference:
```

(c) Updating using triggers

The UPDATE statement will update an existing record in the table -

```
UPDATE customers
SET salary = salary + 500
WHERE id = 2;
```

(d) delete using triggers

```
CREATE OR REPLACE TRIGGER  "SUPPLIERS _ T2"
AFTER
delete on "SUPPLIERS"
for each row
begin
when the person performs delete operations into the table.
end;
/
ALTER TRIGGER  "SUPPLIERS_T2" ENABLE
/
```

Experiment 11: Procedures

(a) creation of procedures

Syntax to create a stored procedure

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
— Comments —
CREATE PROCEDURE procedure_name
= ,
= ,
=
AS
BEGIN
— Query —
END
GO
```

Example:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE Get Student Details
@StudentIDint = 0
AS
BEGIN
SET NOCOUNT ON;
SELECT FirstName, Last Name, Birth Date, City, Country
FROM Students WHERE Student ID=@ Student ID
END
GO
```

(b) Execution of procedure

Let’s see the code to call above created procedure.

```
BEGIN
Get Student Name('Sup','Rahul','12-12-90','hyd','india');
dbms_output.put_line ('record  inserted success fully');
```

```
END;
/
```

(c) Modification of Procedure

Syntax to modify an existing stored procedure

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
— Comments —
ALTER PROCEDURE procedure_name
= ,
= ,
=
AS
BEGIN
— Query —
END
GO
```

Example:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE Get Student Details
      @ StudentI Dint = 0
AS
BEGIN
SET NOCOUNT ON;
SELECT FirstName, LastName, City
FROM Students WHERE StudentID = @ StudentI D
END
GO
```

Syntax to drop a Procedure:

```
DROP PROCEDURE procedure_name
```

Example:

```
DROP PROCEDURE Get Student Details
```

Experiment12: CURSORS

Create procedure:

```
DECLARE
total_rows  number(2);
BEGIN
```

```
UPDATE customers
SET salary = salary + 5000;
IF sql% not found THEN
 dbms_output.put_line('no customers updated');
ELSIF sql%found THEN
 total_rows := sql% row count;
dbms_output.put _line (total_ rows ||' customers updated');
END IF;
END;
/
```

Output:

```
6 customers updated
PL/SQL procedure successfully completed.
```

(a) Declaring Cursor

t defines the cursor with a name and the associated SELECT statement.

Syntax for explicit cursor decleration

```
CURSOR name IS
SELECT statement;
```

(b) Opening Cursor

Syntax for cursor open:

```
OPEN cursor_name;
```

(c) Fetching the data

Syntax for cursor fetch:

```
FETCH cursor_name INTO variable_list;
```

(d) Closing cursor

```
Close cursor_name;
```

Example : Create customers table and have records:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	23	Allahabad	20000
2	Suresh	22	Kanpur	22000
3	Mahesh	24	Ghaziabad	24000
4	Chandan	25	Noida	26000
5	Alex	21	Paris	28000
6	Sunita	20	Delhi	30000

Create procedure:

Execute the following program to retrieve the customer name and address.

```
DECLARE
   c_id  customers.id%type;
   c_name  customers.name%type;
   c_addr  customers.address%type;
   CURSOR  c_customers  is
   SELECT  id,  name,  address  FROM  customers;
BEGIN
   OPEN  c_customers;
   LOOP
   FETCH  c_customers  into  c_id,  c_name,  c_addr;
   EXIT  WHEN  c_customers%notfound;
   dbms_output. put_ line (c_id ||” || c_ name ||” ||  c_addr);
   END  LOOP;
   CLOSE  c_customers;
     END;
     /
```

Experiment 13: Creating Forms

Running form using PL/SQL

Now, we are going to build and a form and running it that will list one record of the the following tables:

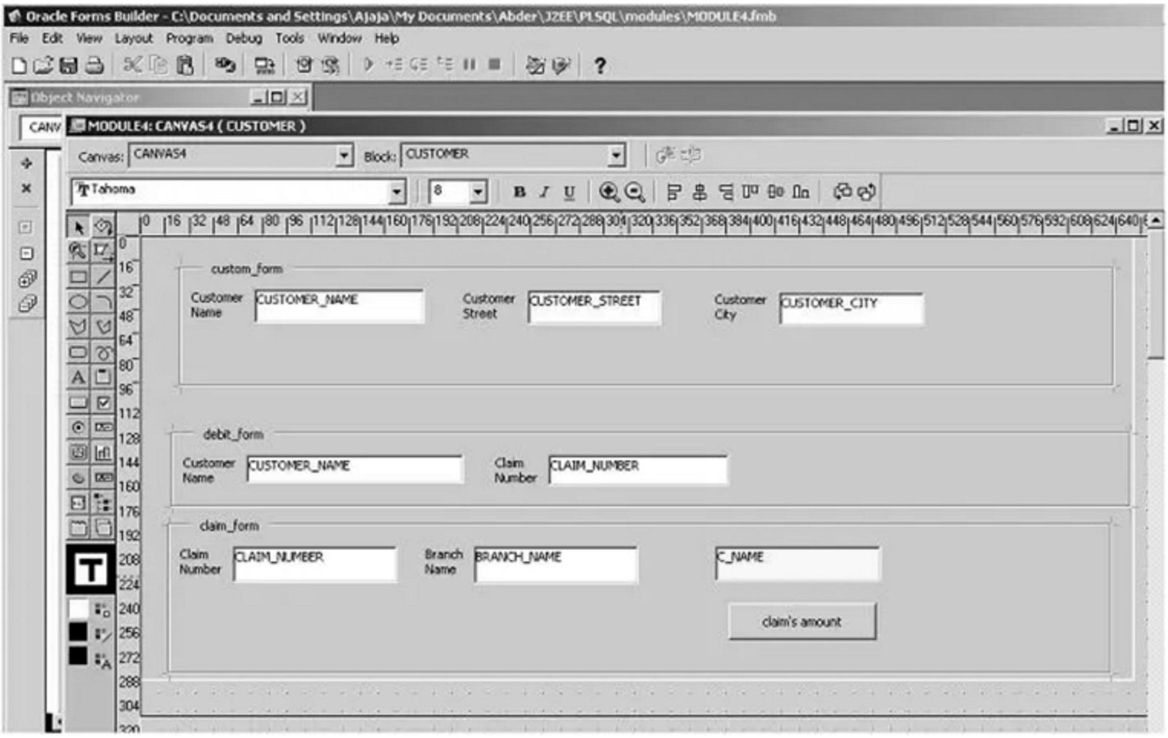
table customer (customer_name, customer_ street, customer_city)

table claim (claim_number, branch_name, amount)

table debit (customer_name, claim_number)

Plus, the form will contain a button once pressed will pop up window that allow to check the claim's table attributes and fill in related box. The code corresponding to the button will be written in PL/SQL.

1. Create the form as in chapter 4 with just one row,
   make a join "customer. customer_name = debit.customer_name" and
   the join "debit.claim_number = claim.claim_number",
   Do not include the "amount" attribute to be displayed (it can be
   deleted from the from if selected and deleted)
2. In the claim_form part, Create a display item for the branch_name box,
3. Right-click, with Property Palette, set
   In Database Section: Database Item: No
   In General Section: Name: C_NAME



Oracle Forms Builder - C:\Documents and Settings\Ajaja\My Documents\Abder\J2EE\PLSQL\modules\MODULE4.fmb

MODULE4: CANVAS4 (CUSTOMER)

Canvas: CANVAS4 Block: CUSTOMER

Tahoma 8

4. Choose LOVs on Object Navigator, and start LOV Wizard: Tools >LOV Wizard,
5. Choose "New Record Group based on a query",
6. The future pop up will list all the attributes of the "claim" relation:

   Write this query:

   SELECT ALL CLAIM.CLAIM_NUMBER, CLAIM.BRANCH_NAME, CLAIM.AMOUNT
   FROM CLAIM
   ORDER BY CLAIM.AMOUNT
7. In the wizard LOC screen, choose all the claim's table attributes,
8. In the screen of "Look up return item...",
   click on return value of branche_name and assign for it branch.branch_name,
   click on return value of claim_number and assign for it debit.claim_number, and
   click on return value of amount and assign for it C_NAME,
9. Add a title claim_form, retreive 20 items , Press next,
10. Assign amount (the one attribute to be displaced to right),
    This will finish the wizard window.
10. With Property Palette, rename the LOVs:
    On the Object Navigator window: claim_lov,
    On Record Groups: claim_rec

11. Active the “branch_name” box. With Property Palette, assign for

    “Validate from List”: Yes

11. To create a button, from the palette. With Paqlette Property

    set for it the name “claim_but”

12. Right click and choose PL/SQL Editor,

13. Select the trigger: “WHEN-BUTTON-PRESSED”,

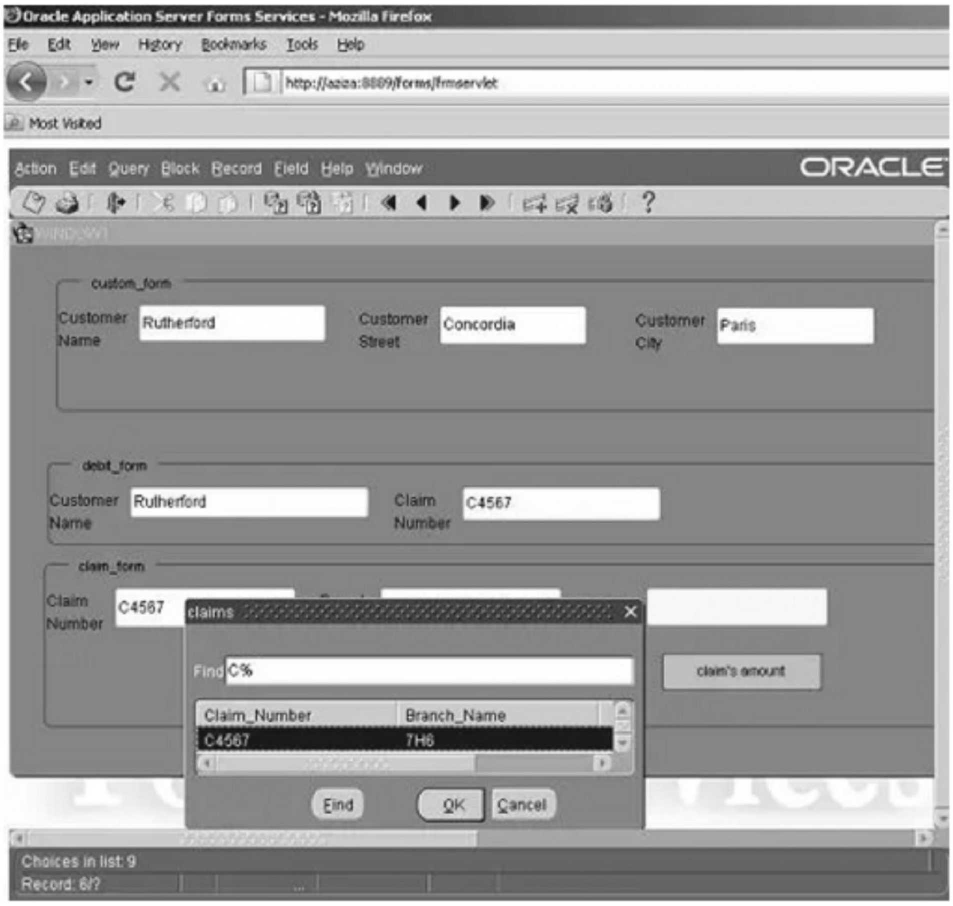
14. Within the PL/SQL editor, to display the claim_lov (i.e.: all th attributes of the claim_relation); write this code:

```
DECLARE a_value BOOLEAN;
BEGIN
a_value:=Show_lov(‘claim_lov’);
END;
```

    1 Press the Compile button to check the code,

16. Close the Window.

    Run the form created:



Oracle Application Server Forms Services - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://aziza:8889/forms/frmservlet
Most Visited
Action Edit Query Block Record Field Help Window
ORACLE
WINDOW1
claims
Find
OK
Cancel
claim's amount

custom_form

Customer Name: Rutherford

Customer Street: Concordia

Customer City: Paris

debit_form

Customer Name: Rutherford

Claim Number: C4567

claim_form

Claim Number: C4567

Find C%

Claim_Number	Branch_Name
C4567	7H6

Choices in list: 9

Record: 6/?

Experiment 14: Create, Alter and Drop Table

CREATE TABLE

A table is basic unit of storage. It is composed of rows and columns. To create a table we will name the table and the columns of the table. We follow the rules to name tables and columns:

- It must begin with a letter and can be up to 30 characters long.
- It must not be duplicate and not any reserved word.

SYNTAX to create a table:

CREATE TABLE tablename (column_name1 datatype (size), column_name2 datatype (size) …);

Example:

CREATE TABLE studen(rollno number (4), name varchar2 (15));

ALTER TABLE:

After creating a table one may have need to change the table either by add new columns or by modify existing columns. One can do so by using alter table command.

SYNTAX to add a column:

ALTER TABLE tablename ADD(col1 varchar(10),col2 number(10));

SYNTAX to modify a column:

ALTER TABLE tablename MODIFY(col1 datatype,col2 datatype);

DROP TABLE:

To remove the definition of oracle table, the drop table statement is used.Drop command is used to remove the structure of table so no rollback possible in this case.

SYNTAX to drop table:

DROP TABLE tablename;

Experiment 15

To perform Select, update, insert and delete operations on table

SELECT STATEMENT:

SELECTING ALL COLUMNS OF THE TABLE:

A 'SELECT' statement is used as a DATA RETRIEVAL statement i.e. It retrieves information from the database.

SYNTAX:

SQL> SELECT * FROM TABLE NAME;

- SELECT identifies WHAT COLUMNS.
- FROM identifies WHICH TABLE.

Simply, SELECT clause specify which column is to be displayed & FROM clause specify the table containing the columns listed in the SELECT clause.

Here, '*' is used to select all columns.

SELECTING SPECIFIC COLUMNS OF THE TABLE:

SYNTAX:

SQL> SELECT ENAME,JOB FROM EMP

SELECTING DISTINCT ELEMENTS FROM THE TABLE:

SYNTAX:

SQL> SELECT DISTINCT ENAME,JOB FROM EMP

The SELECT DISTINCT statement is used to return only distinct (different) values.It is used to remove duplicate values.

SQL> select * from empxyz;

```
NAME        AGE
anju         23
jkg          34
anju         23
```

SQL> select distinct name,age from empxyz;

```
NAME          AGE
anju          23
jkg           34
```

INSERT STATEMENT:

SYNTAX:

SQL> INSERT into CSE(student,rollno) VALUES ('MONIKA',651);

INSERT statement is used to ADD NEW ROW TO A TABLE.

Using INSERT We can only insert on row at a time. As shown in above example,

In above example CSE is the name of the TABLE & STUDENT, ROLLNO are its two ATTRIBUTES.

Enclose CHARACTER & DATE values within a SINGLE QUOTATION MARKS.

DELETE STATEMENT:

SYNTAX:

SQL> DELETE from CSE where rollno BETWEEN 605 AND 630; i.e. DELETE FROM table [WHERE condition];

If we OMIT WHERE CLAUSE then ALL ROWS OF THE COLUMN ARE DELETED.

UPDATE STATEMENT:

SYNTAX:

SQL> UPDATE cse SET rollno=21 WHERE student='ITIKA';

Here, If we do not use WHERE clause then ALL ROWS OF THE TABLE ARE UPDATED.

SPCIFIED ROW or ROWS are modified if we specify the WHERE clause

Experiment 16: To make use of different clauses where, group by, having, order by union, intersection, set difference

(a) where

Syntax

WHERE conditions;

example

SELECT *

FROM (**SELECT** ....

FROM students

WHERE age >= 40

ORDERBYlastname, firstname)

UNION

SELECT *

FROM (**SELECT** ....

FROM students

WHERE age <40

ORDERBY first name, last name)

(b) having

Oracle HAVING Clause

Syntax:

SELECT expression1, expression2, ... expression_n,

aggregate_function (aggregate_expression)

FROM tables

WHERE conditions

GROUP BY expression1, expression2, ... expression_n

HAVING having_condition;

HAVING Example: (with GROUP BY SUM function)

Let's take a table "salesdepartment"

Salesdepartment table:

```
CREATE TABLE  "SALESDEPARTMENT"
    ("ITEM"  VARCHAR2(4000),
     "SALE"  NUMBER,
     "BILLING_ADDRESS"  VARCHAR2(4000)
    )
      /
```

HAVING Example: (with GROUP BY COUNT function)

```
SELECT state, COUNT(*) AS "Number of customers"
FROM customers
WHERE salary > 10000
GROUP BY state
HAVING COUNT(*) >= 2;
```

(c) Order by

Syntax:

```
SELECT expressions
FROM tables
WHERE conditions
ORDER BY expression [ASC|DESC];
```

Example

```
CREATE TABLE  "SUPPLIER"
    (  "SUPPLIER_ID"  NUMBER,
     "FIRST_NAME"  VARCHAR2(4000),
     "LAST_NAME"  VARCHAR2(4000)
    )
/
```

ORDER BY Example:(sorting in descending order)

If you want to sort your result in descending order, you should use the DESC attribute in your ORDER BY clause:

Execute this Query:

```
SELECT *
FROM supplier
ORDER BY last_name DESC;
```

(d) group by

Syntax:

```
SELECT expression1, expression2, ... expression_n,
    aggregate_function (aggregate_expression)
FROM tables
WHERE conditions
    GROUP BY expression1, expression2, ... expression_n;
```

Example

```
CREATE TABLE "SALESDEPARTMENT"
    (  "ITEM"  VARCHAR2(4000),
     "SALE"  NUMBER,
     "BILLING_ADDRESS"  VARCHAR2(4000)
    )
/
```

GROUP BY Example: (with COUNT function)

```
CREATE TABLE  "CUSTOMERS"
    (  "NAME"  VARCHAR2(4000),
     "AGE"  NUMBER,
     "SALARY"  NUMBER,
     "STATE"  VARCHAR2(4000)
    )
/
```

GROUP BY Example: (with MIN function)

```
CREATE TABLE "EMPLOYEES"
    (  "EMP_ID"  NUMBER,
     "NAME"  VARCHAR2(4000),
     "AGE"  NUMBER,
     "DEPARTMENT"  VARCHAR2(4000),
     "SALARY"  NUMBER
    )

      /
```

(e) Union

Syntax

```
SELECT expression1, expression2, ... expression_n
FROM table1
WHERE conditions
UNION
SELECT expression1, expression2, ... expression_n
FROM table2
WHERE conditions;
```

Example: Fetch single field

```
SELECT supplier_id
FROM suppliers
UNION
SELECT supplier_id
FROM order_details
```

UNION Example: (Using ORDER BY)

```
SELECT supplier_id, supplier_name
FROM suppliers
WHERE supplier_id <= 20
UNION
SELECT s_id, s_name
FROM shopkeepers
WHERE s_name = 'dhirubhai'
ORDER BY 1;
```

f) UNION ALL

```
SELECT expression1, expression2, … expression_n
FROM table1
WHERE conditions
UNION ALL
SELECT expression1, expression2, … expression_n
FROM table2
WHERE conditions;
```

Example

```
SELECT supplier_id
FROM suppliers
UNION ALL
SELECT supplier_id
FROM order_details;
```

(g) Intersect

Syntax

```
SELECT expression1, expression2, … expression_n
FROM table1
WHERE conditions
INTERSECT
SELECT expression1, expression2, … expression_n
FROM table2
WHERE conditions;
```

Example:

```
SELECT supplier_id
FROM suppliers
INTERSECT
SELECT supplier_id
FROM order_details;
```

INTERSECT Example: (with multiple expressions)

```
SELECT supplier _ id, last _ name, first _ name
FROM supplier
WHERE first_name <> 'dhirubhai'
INTERSECT
SELECT customer _ id, last _ name, first _ name
FROM customer
WHERE customer_id < 5;
```

(h) Minus

Syntax :

```
SELECT expression1, expression2, … expression_n
FROM table1
WHERE conditions
```

(h) MINUS

```
SELECT expression1, expression2, … expression_n
FROM table2
WHERE conditions;
```

Example

```
SELECT supplier_id
FROM suppliers
MINUS
SELECT supplier_id
FROM order_details;
```

Experiment 17: To study different constraints

(a) Check Constraints

Syntax:

CREATE TABLE table_name

(

column1 datatype null/not null,

column2 datatype null/not null,

?

CONSTRAINT constraint_ name CHECK (column_name condition) [DISABLE]

);

Example 1

CREATE TABLE student (id numeric(4), name varchar2 (50), CONSTRAINT check _ id CHECK(id BETWEEN)

(b) Primary Key - Using CREATE TABLE statement

Syntax:

CREATE TABLE table_name

(

column1 datatype null/not null,

column2 datatype null/not null,

...

CONSTRAINT constraint_ name PRIMARY KEY (column1, column2, ... column_n)

);

Example

CREATE TABLE Test2(ID Number, NAME Varchar2 CONSTRAINT test2_ pk PRIMARY KEY (ID));

Primary Key - Using ALTER TABLE statement

Syntax:

ALTER TABLE table_name

ADD CONSTRAINT constraint _name PRIMARY KEY (column1, column2,... column _ n);

Example

ALTER TABLE student ADD CONSTRAINT student _ pk PRIMARY KEY(id);

(c) NOT NULL

Example:

CREATE TABLE STUDENT (

student_idint NOT NULL,

FirstNamevarchar(255) NOT NULL,

LastNamevarchar(255) NOT NULL,

Age int

);

(d) UNIQUE

Example:

CREATE TABLE student (student_idint NOT NULL UNIQUE,

FirstNamevarchar(25) NOT NULL, Last Name varchar(25),Age int);

FOREIGN KEY Constraints

A foreign key is a field which refers to the PRIMARY KEY of another table and the table which actually has the foreign key is called child table. Let us now create a table order which we has student_id column as a foreign key refrencingstudent_id column of student table using CREATE TABLE statement.

Example:

CREATE TABLE Orders(OrderIDint PRIMARY KEY,

OrderNumberint NOT NULL, student_idint REFERENCES student(student_id));

ALTER TABLE Orders ADD FOREIGN KEY (student_id) REFERENCES student(student_id);

Experiment 18: To use aggregate, numeric, string fucntions

(a) aggregate functions

Syntax of Oracle Aggregate Functions

Given below is the syntax :

GroupFunctionName (DISTINCT / ALL ColumnName)

Example #1

AVG (DISTINCT / ALL ColumnName) Function.

- Output will be the Average value of column.
- It doesn't consider NULL values.

Code:

SELECT AVG(Salary), AVG(DISTINCT Salary) FROM Employee;

Example #2

SUM (DISTINCT / ALL ColumnName) Function.

- Output will be the SUM value of column.
- It doesn't consider NULL values.

Code:

SELECT SUM (Salary), SUM (DISTINCT Salary) FROM Employee;

Example #3

MAX (DISTINCT / ALL ColumnName) Function.

- Output will be the maximum value of column.
- It doesn't consider NULL values.

Code:

SELECT MAX(Salary), MAX(DISTINCT Salary) FROM Employee;

Example #4

MIN (DISTINCT / ALL ColumnName) Function.

- Output will be the minimum value of column.
- It doesn't consider NULL values.

Code:

SELECT MIN (Salary), MIN (DISTINCT Salary) FROM Employee;

Example #5

STDDEV (DISTINCT / ALL ColumnName) Function.

- Output will be the Standard Deviation of column.
- It doesn't consider NULL values.

Code:

SELECT STDDEV(Salary), STDDEV (DISTINCT Salary) FROM Employee;

Example #6

VARIANCE (DISTINCT / ALL ColumnName) Function.

- Output will be the Variance of N.
- It doesn't consider NULL values.

Code:

SELECT VARIANCE(Salary), VARIANCE (DISTINCT Salary) FROM Employee;

Example #7

COUNT (*/ DISTINCT / ALL ColumnName) Function.

- Output will be the number of rows.
- With COUNT function, * is used to return all rows including duplicates and NULLs.
- It is used to get the count of all rows or distinct values of column.

Code:

SELECT Deptnumber, MAX (Salary) FROM Employee;

(b) numericFucntions

1. ABS :

Syntax:

ABS(n)

ABS returns the absolute value of n.

Example:

SELECT ABS(-13) "Absolute" FROM DUAL;

2. ACOS :

Syntax:

ACOS(n)

ACOS returns the arc cosine of n. The argument n must be in the range of -1 to 1, and the function returns a value in the range of 0 to pi, expressed in radians.

Example:

SELECT ACOS(.6)"Arc_Cos" FROM DUAL;

3. ASIN :

Syntax:

ASIN(n)

ASIN returns the arc sine of n. The argument n must be in the range of -1 to 1, and the function returns a value in the range of -pi/2 to pi/2, expressed in radians.

Example:

SELECT ASIN(.6) "Arc_Sin" FROM DUAL;

4. ATAN :

Syntax:

ATAN(n)

ATAN returns the arc tangent of n. The argument n can be in an unbounded range and returns a value in the range of -pi/2 to pi/2, expressed in radians.

Example:

SELECT ATAN(.4) "Arc_Tan" FROM DUAL;

5. **ATAN2 :**

Syntax:

ATAN2(n1;n2)

ATAN2 returns the arc tangent of n1 and n2. The argument n1 can be in an unbounded range and returns a value in the range of -pi to pi, depending on the signs of n1 and n2, expressed in radians. ATAN2(n1,n2) is the same as ATAN2(n1/n2).

Example:

SELECT ATAN2(.6, .4) "Arc_Tan2" FROM DUAL;

6. **BITAND :**

Syntax:

BITAND(expr1,expr2)

BITAND computes an AND operation on the bits of expr1 and expr2, both of which must resolve to nonnegative integers, and returns an integer. This function is commonly used with the DECODE function.

7. **CEIL :**

Syntax:

CEIL (n)

CEIL returns smallest integer greater than or equal to n.

8. **COS :**

Syntax:

COS (n)

COS returns the cosine of n (an angle expressed in radians).

Example:

SELECT COS(180 * 3.14159265359/180)

"Cosine of 180 degrees" FROM DUAL;

9. **COSH :**

Syntax:

COS (n)

COSH returns the hyperbolic cosine of n.

Example:

SELECT COSH(1) "Hyperbolic cosine of 1" FROM DUAL;

10. **EXP :**

Syntax:

EXP(n)

EXP returns e raised to the nth power, where e = 2.71828183 ... The function returns a value of the same type as the argument.

Example:

SELECT EXP(3) "e to the 3rd power" FROM DUAL;

11. **FLOOR :**

Syntax:

FLOOR(n)

FLOOR returns largest integer equal to or less than n.

Example:

SELECT FLOOR(14.7) "Floor" FROM DUAL;

12. **LN :**

Syntax:

LN(n)

LN returns the natural logarithm of n, where n is greater than 0.

Example:

SELECT LN(75) "Natural log of 75" FROM DUAL;

13. **LOG:**

Syntax:

LOG(n2,n1)

LOG returns the logarithm, base n2, of n1. The base n1 can be any positive value other than 0 or 1 and n2 can be any positive value.

Example:

SELECT LOG(20,100) "Log base 20 of 100" FROM DUAL;

14. **MOD :**

Syntax:

MOD (n2,n1)

MOD returns the remainder of n2 divided by n1. Returns n2 if n1 is 0.

Example:

SELECT MOD(12,3) "Modulus" FROM DUAL;

15. NANVL :

Syntax:

NANVL(n2,n1)

The NANVL function is useful only for floating-point numbers of type BINARY_FLOAT or BINARY_DOUBLE. It instructs Oracle Database to return an alternative value n1 if the input value n2 is NaN (not a number). If n2 is not NaN, then Oracle returns n2. This function is useful for mapping NaN values to NULL.

Example:

Insert INTO float_point_demo

VALUES (0,'NaN','NaN');

SELECT * FROM float_point_demo;

16. POWER :

Syntax:

POWER(n2,n1)

POWER returns n2 raised to the n1 power. The base n2 and the exponent n1 can be any numbers, but if n2 is negative, then n1 must be an integer.

Example:

SELECT POWER(4,3) "Raised" FROM DUAL;

17. REMAINDER :

Syntax:

REMAINDER(n2,n1)

REMAINDER returns the remainder of n2 divided by n1.

Example:

SELECT bin_float, bin_double, REMAINDER (bin_float, bin_double)

FROM float_point_demo;

18. ROUND (number) :

Syntax:

ROUND(n,integer)

ROUND returns n rounded to integer places to the right of the decimal point. If you omit integer, then n is rounded to 0 places. The argument integer can be negative to round off digits left of the decimal point.

Example:

SELECT ROUND(12.196,2) "Round" FROM DUAL;

19. SIGN :

Syntax:

SIGN(n)

SIGN returns the sign of n. This function takes as an argument any numeric datatype, or any nonnumeric datatype that can be implicitly converted to NUMBER, and returns NUMBER.

Example:

SELECT SIGN(-12) "Sign" FROM DUAL;

20. SIN :

Syntax:

SIN(n)

SIN returns the sine of n (an angle expressed in radians).

Example:

SELECT SIN(20 * 3.14159265359/180) "Sine of 20 degrees" FROM DUAL;

21. SINH :

Syntax:

SINH(n)

SINH returns the hyperbolic sine of n.

Example:

SELECT SINH(3) "Hyperbolic sine of 3" FROM DUAL;

22. SQRT :

Syntax:

SQRT(n)

SQRT returns the square root of n.

Example:

SELECT SQRT(24) "Square root" FROM DUAL;

23. TAN :

Syntax:

TAN(n)

TAN returns the tangent of n (an angle expressed in radians).

Example:

SELECT TAN(90 * 3.14159265359/180) "Tangent of 90 degrees" FROM DUAL;

24. TANH :

Syntax:

TANH(n)

TANH returns the hyperbolic tangent of n.

Example:

SELECT TANH(.7)"Hyperbolic tangent of .7" FROM DUAL;

25. TRUNC (number):

Syntax:

TRUNC(n1,n2)

The TRUNC (number) function returns n1 truncated to n2 decimal places. If n2 is omitted, then n1 is truncated to 0 places. n2 can be negative to truncate (make zero) n2 digits left of the decimal point.

Example:

SELECT TRUNC(13.9,2) "Truncate" FROM DUAL;

(c) string functions

1. Contains

Contain String function followed by a pattern like >0; this means that for the particular row which was selected, the calculated score value greater than Zero.

Example: contains (text, 'function')>0

2. Equals

The equal string function comes into use to examine equality and to get an exact match which returns to a true value.

Example: Equals (text = 'function')

3. Ends with

This method finds the new value, which contains a string from starting.

Example: Ends with (<suffix>)

4. Starts with

This method gets the new value contains a starting string.

Example: Starts with(<prefix>)

5. Equalsignore Case

The equal sign or case function is for comparing a particular string to others by ignoring the considerations of the case.

Example: EqualsIgnoreCase (String other string)

6. Is Empty

The isEmpty function is come into use for the string to verify that the length () is zero.

Example: IsEmpty ()

7. Matches

The matches function is all about particular string matches with the regex(regular expression).

Example: Text.matches(regex,string)

8. Replace

Replace function is all about the string search-replace with string replacement in order to get char.

Example: Replace (char,<search string> ,<replacement string>)

9. Replace All

This Function is used to replaces all substring of the string which matches the given regex with the given replacement.

Example: Replace All (<old valregex> ,<newal>

10. Split

Splits the string around matches that given regular expression.

Example: Text.Split(<regexpattern>

Common Oracle String Functions

Below are the most common Oracle string functions which help in manipulating the string character effectively.

1. ASCII

The ASCII code comparable to the one character in the expression return.

Example: ASCII ('a')

2. Bit_Length

Return length in bits of a particular string; each Unicode value of the character is 2bytes in length (equal to 16bits)

Example: Bit_Length ('abcdef')

3. Char

It converts a numeric value to the analogous ASCII Character code.

Example: Char (35)

4. Char_Length

Blanks are not counted in string length. Return length in the number of characters of a particular string.

Example: Char_Length

5. Concat

Concat string function allows a particular string at one end and back to the same string.

Example: Concat ('text a'). concat ('text b')

6. Insert

A specified string character into a particular location in other string characters.

7. Char_Length

Blanks are not counted in string length. Return length in the number of characters of a particular string.

Example: Char_Length

5. Concat

Concat string function allows a particular string at one end and back to the same string.

Example: Concat ('text a'). concat ('text b')

6. Insert

A specified string character into a particular location in other string characters.

Example: select insert ('123456'), 2, 3, 'abcd'

7. Left

Specified number of character from the left of a string

Example: select left ('123456', 3)

8. Length

Return the length, number of character of a particular string. The length is returned to exclude any blank characters.

Example: Length (Customer_Name)

9. Locate

This function comes into use to search string in other string, but it is not found the string it returns to its original index that is 0.

Example: Locate ('d' 'abcdef')

10. LocateN

Return the numeric position of a string character in other character string. This includes an integer that enables to specify an initial position to start the search.

Example: Locate ('d', 'abcdef',3)

11. Lower

This converts a string character to lowercase

Example: Lower(Customer_Name)

12. Ortet_Length

Return a number of bytes of a particular string.

Example: Octet_Length ('abcdef')

13. Position

This function comes into use to find a substring from a string and search the location of the string in the substring. The function return to the position of starting character when the substring is equal to the found substring.

Example: Position ('d', 'abcdef')

14. Repeat

Repeat a particular expression pie times.

Example: Repeat ('abc',4)

15. Replace

Replace one or more character from a particular character expression with one or more characters.

Example: Replace ('abc1234', '123' ,'zz')

16. Right

Return a particular number of characters from the right of the string.

Example: SELECT Right ('123456',3)

17. Space: Insert blank space

Example: Space (2)

18. Substring

This function permits you to excerpt substring from the original string.

Example: Substring ('abcdef')

19. SubstringN

SubstringN help you to get the length of the string, which includes an integer in the character number.

Example: Substring ('abcdef')

20. TrimBoth

Particular strips leading & trailing character from a character string.

Example: Trim (Both '_' From 'abcdef')

21. TrimLeading

Particular strips leading characters from a character string.

Example: Trim (LEADING '_' From '_ abcdef_')

22. Trim Trailing

Particular trailing characters from a character string.

Example: Trim (TRAILING '_' From 'abcdef _')

23. Upper

It converts a string character to uppercase

Example: Upper (Cutomer_Name)

Experiment: 19 : using Joins

INNER JOIN (simple join)

Chances are, you've already written a statement that uses an Oracle INNER JOIN. It is the most common type of join. Oracle INNER JOINS return all rows from multiple tables where the join condition is met.

Syntax

The syntax for the INNER JOIN in Oracle/ PLSQL is:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.column = table2.column;
```

Example

Here is an example of an Oracle INNER JOIN:

```
SELECT suppliers.supplier_id, suppliers. supplier _ name, orders.order_date
    FROM suppliers
    INNER JOIN orders
  ON suppliers.supplier_id = orders. supplier _ id;
```

INNER JOIN (simple join)

Chances are, you've already written a statement that uses an Oracle INNER JOIN. It is the most common type of join. Oracle INNER JOINS return all rows from multiple tables where the join condition is met.

Syntax

The syntax for the INNER JOIN in Oracle/ PLSQL is:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.column = table2.column;
```

Example

Here is an example of an Oracle INNER JOIN:

```
SELECT suppliers.supplier_id, suppliers. supplier _ name, orders.order_date
    FROM suppliers
    INNER JOIN orders
  ON suppliers.supplier_id = orders. supplier _id;
```

RIGHT OUTER JOIN

Another type of join is called an Oracle RIGHT OUTER JOIN. This type of join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where the joined fields are equal (join condition is met).

Syntax

The syntax for the Oracle RIGHT OUTER JOIN is:

SELECT columns

FROM table1

RIGHT [OUTER] JOIN table2

ON table1.column = table2.column;

In some databases, the RIGHT OUTER JOIN keywords are replaced with RIGHT JOIN.

Example

Here is an example of an Oracle RIGHT OUTER JOIN:

SELECT orders.order_id, orders.order_date, suppliers.supplier_name

FROM suppliers

RIGHT OUTER JOIN orders

ON suppliers.supplier_id = orders.supplier_id;

FULL OUTER JOIN

Another type of join is called an Oracle FULL OUTER JOIN. This type of join returns all rows from the LEFT-hand table and RIGHT-hand table with nulls in place where the join condition is not met.

Syntax

The syntax for the Oracle FULL OUTER JOIN is:

SELECT columns

FROM table1

FULL [OUTER] JOIN table2

ON table1.column = table2.column;

In some databases, the FULL OUTER JOIN keywords are replaced with FULL JOIN.

Example

Here is an example of an Oracle FULL OUTER JOIN:

SELECTsuppliers.supplier_id, suppliers. supplier_name, orders.order_date

FROM suppliers

FULL OUTER JOIN orders

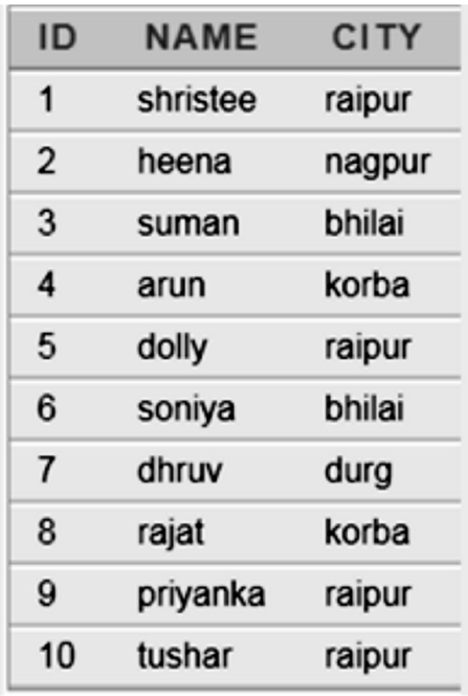
ON suppliers.supplier_id = orders. supplier _ id;

Experiment 20: working of sub – queries

ORACLE Subqueries

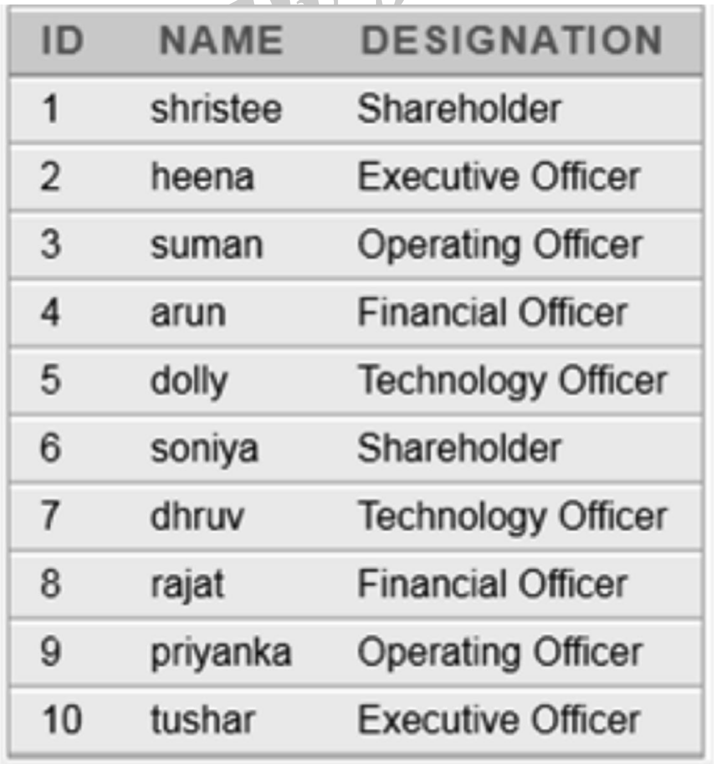
In Oracle, subqueries are the queries inside a query. Subqueries can be made using WHERE, FROM or SELECT clause.

Table 1: employee1



ID	NAME	CITY
1	shristee	raipur
2	heena	nagpur
3	suman	bhilai
4	arun	korba
5	dolly	raipur
6	soniya	bhilai
7	dhruv	durg
8	rajat	korba
9	priyanka	raipur
10	tushar	raipur

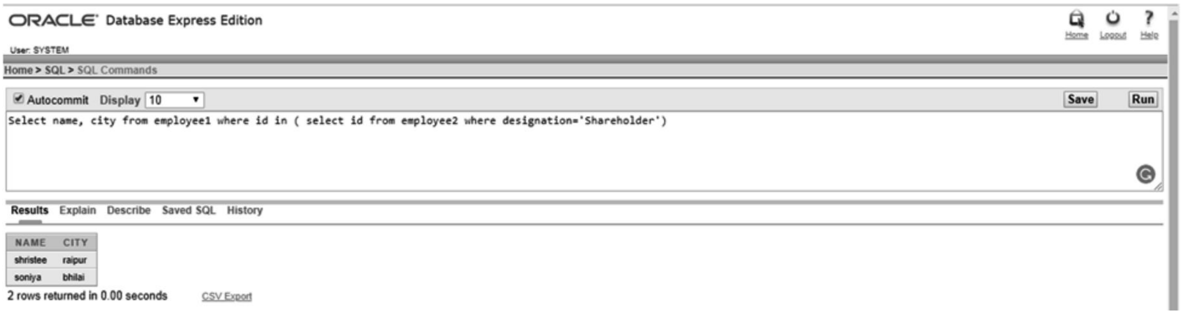
Table 2: employee2



ID	NAME	DESIGNATION
1	shristee	Shareholder
2	heena	Executive Officer
3	suman	Operating Officer
4	arun	Financial Officer
5	dolly	Technology Officer
6	soniya	Shareholder
7	dhruv	Technology Officer
8	rajat	Financial Officer
9	priyanka	Operating Officer
10	tushar	Executive Officer

Example 1

Query: Select name, city from employee1 where id in (select id from employee2 where designation='Shareholder')



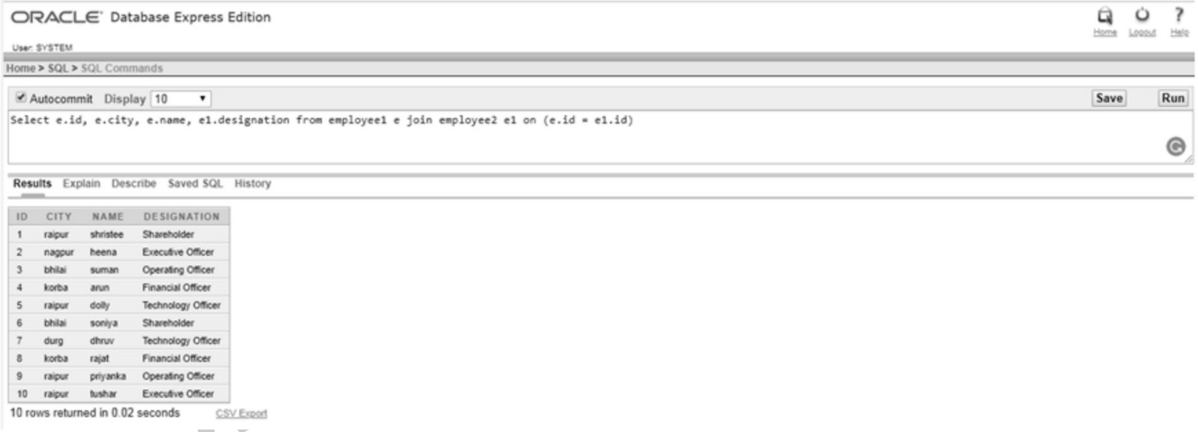
```
Select name, city from employee1 where id in ( select id from employee2 where designation='Shareholder')
```

NAME	CITY
shristee	raipur
soniya	bhilai

2 rows returned in 0.00 seconds

Example 2

Query: Select e.id, e.city, e.name, e1.designation from employee1 e join employee2 e1 on (e.id = e1.id)



```
Select e.id, e.city, e.name, e1.designation from employee1 e join employee2 e1 on (e.id = e1.id)
```

ID	CITY	NAME	DESIGNATION
1	raipur	shristee	Shareholder
2	nagpur	heena	Executive Officer
3	bhilai	suman	Operating Officer
4	korba	arun	Financial Officer
5	raipur	dolly	Technology Officer
6	bhilai	soniya	Shareholder
7	durg	dhruv	Technology Officer
8	korba	rajat	Financial Officer
9	raipur	priyanka	Operating Officer
10	raipur	tushar	Executive Officer

10 rows returned in 0.02 seconds

FACULTY OF MANAGEMENT

M.B.A III - Semester Examination

Model Paper – I

DATA BASE MANAGEMENT SYSTEMS

Time : 3 Hours] **[Max. Marks : 60**

Note : Answer **all** the questions

PART - A (5 × 2 = 10 Marks)
[Short Answer Type]

Answers

1. Data Independence **(Unit-I, SQA-2)**
2. Relational algebra **(Unit-II, SQA-1)**
3. Secondary Indexing **(Unit-III, SQA-8)**
4. Non-serial Schedule **(Unit-IV, SQA-4)**
5. DDL commands. **(Unit-V, SQA-2)**

PART - B (5 × 10 = 50 Marks)
[Essay Answer type]

Answer all the questions using the internal choice

6. (a) Explain briefly about ER Model. **(Unit-I, Q.No. 8)**

(OR)

(b) Explain briefly about Data Manipulation Operations. **(Unit-I, Q.No. 18)**

7. (a) Explain briefly about commands DDL constructs with examples. **(Unit-II, Q.No. 4)**

(OR)

(b) Explain briefly about (SQL SERVER). **(Unit-II, Q.No. 13)**

8. (a) What is query processing? Explain about the steps in query processing. **(Unit-III, Q.No. 1)**

(OR)

(b) Explain various methods of query optimization Algorithms in DBMS. **(Unit-III, Q.No. 6)**

9. (a) What is concurrency control? Explain the problems of concurrency control. **(Unit-IV, Q.No. 1)**

(OR)

(b) What are ACID properties? Explain. **(Unit-IV, Q.No. 3)**

10. (a) Explain the advantages of SQL disadvantages of SQL. **(Unit-V, Q.No. 4)**

(OR)

(b) Discuss about Date function in SQL. **(Unit-V, Q.No. 23)**

FACULTY OF MANAGEMENT

M.B.A III - Semester Examination

Model Paper – II

DATA BASE MANAGEMENT SYSTEMS

Time : 3 Hours] **[Max. Marks : 60**

Note : Answer **all** the questions

PART - A (5 × 2 = 10 Marks)
[Short Answer Type]

Answers

1. Explain briefly about ER Model. **(Unit-I, SQA-4)**
2. DDL **(Unit-II, SQA-5)**
3. Query Equivalance **(Unit-III, SQA-3)**
4. Atomicity **(Unit-IV, SQA-2)**
5. DCL commands **(Unit-V, SQA-5)**

PART - B (5 × 10 = 50 Marks)
[Essay Answer type]

Answer all the questions using the internal choice

6. (a) Discuss about Codd Rules in DBMS. **(Unit-I, Q.No. 15)**

(OR)

(b) What is the Network Model in DBMS? Explain. **(Unit-I, Q.No. 10)**

7. (a) What is the Oracle database? Give the brief introduction about it. **(Unit-II, Q.No. 9)**

(OR)

(b) Give the brief introduction about SQL3. **(Unit-II, Q.No. 3)**

8. (a) Explain about Query Optimization in Relational Algebra. **(Unit-III, Q.No. 2)**

(OR)

(b) What is Indexing in DBMS? Explain. **(Unit-III, Q.No. 7)**

9. (a) Explain about types of lock based protocols. **(Unit-IV, Q.No. 7)**

(OR)

(b) What are locks in DBMS? Explain about different types of locks. **(Unit-IV, Q.No. 6)**

10. (a) Explain the Process of SQL. **(Unit-V, Q.No. 3)**

(OR)

(b) Explain ALTER Command in SQL. **(Unit-V, Q.No. 12)**

FACULTY OF MANAGEMENT

M.B.A III - Semester Examination

Model Paper – III

DATA BASE MANAGEMENT SYSTEMS

Time : 3 Hours] **[Max. Marks : 60**

Note : Answer **all** the questions

PART - A (5 × 2 = 10 Marks)
[Short Answer Type]

		Answers
1.	Integrity constraints	**(Unit-I, SQA-9)**
2.	User-Defined Variable	**(Unit-II, SQA-8)**
3.	Query Optimization	**(Unit-III, SQA-4)**
4.	Concurrency control	**(Unit-IV, SQA-1)**
5.	Foreign Key	**(Unit-V, SQA-9)**

PART - B (5 × 10 = 50 Marks)
[Essay Answer type]

Answer all the questions using the internal choice

6. (a) Explain about different types of DBMS architecture. **(Unit-I, Q.No. 1)**

(OR)

(b) Explain DML commands with its syntax and example. **(Unit-I, Q.No. 7)**

7. (a) Discuss about DML constructs with an examples. **(Unit-II, Q.No. 5)**

(OR)

(b) What is DB2? Explain about it. **(Unit-II, Q.No. 12)**

8. (a) Write about equivalence rules in query optimization. **(Unit-III, Q.No. 4)**

(OR)

(b) Explain about B-Trees in DBMS. **(Unit-III, Q.No. 8)**

9. (a) Explain Multi-Version Schemes of Con-currency Control with example. **(Unit-IV, Q.No. 9)**

(OR)

(b) What is user Authentication? Explain different types of user authentication techniques. **(Unit-IV, Q.No. 13)**

10. (a) Explain the History of SQL. **(Unit-V, Q.No. 2)**

(OR)

(b) Explain insert statement with syntax and example. **(Unit-V, Q.No. 7)**

BUSINESS ANALYTICS

SYLLABUS

UNIT - I

INTRODUCTION TO BUSINESS ANALYTICS:

Definition of Business Analytics, Categories of Business Analytical methods and models, Business Analytics in practice, Big Data - Overview of using Data, Types of Data. Business Decision Modeling

UNIT - II

DESCRIPTIVE ANALYTICS:

Over view of Description Statistics (Central Tendency, Variability), Data Visualization-Definition, Visualization Techniques - Tables, Cross Tabulations, charts, Data Dash boards using Advanced Ms-Excel or SPSS.

UNIT - III

PREDICTIVE ANALYTICS:

Trend Lines, Regression Analysis - Linear & Multiple, Predictive modeling, forecasting Techniques, Data Mining - Definition, Approaches in Data Mining - Data Exploration & Reduction, Data mining and business intelligence, Data mining for business Classification, Association, Cause Effect Modelling.

UNIT - IV

PRESCRIPTIVE ANALYTICS

Overview of Linear Optimization, Non Linear Programming Integer Optimization, Cutting Plane algorithm and other methods, Decision Analysis - Risk and uncertainty methods - Text analytics, Web analytics.

UNIT - V

PROGRAMMING USING R

R Environment, R packages, Reading and Writing data in R, R functions, Control Statements, Frames and Subsets, Managing and Manipulating data in R.

Contents

Topic No. **Page No.**

UNIT - I

UNIT - II

Topic **Page No.**

UNIT - III

UNIT - IV

Topic **Page No.**

UNIT - V

Important Questions

UNIT - I

1. **Explain the different business analytical methods.**

Ans :

Refer Unit-I, Q.No. 3

2. **Explain the different models in Business Analytics?**

Ans :

Refer Unit-I, Q.No. 4

3. **Discuss briefly about role of business analytics in current business environment.**

Ans :

Refer Unit-I, Q.No. 5

4. **Explain the role of Business Analytics in Best Practices.**

Ans :

Refer Unit-I, Q.No. 7

5. **Explain the relationship of big data with other areas?**

Ans :

Refer Unit-I, Q.No. 11

6. **Explain the life cycle of big data.**

Ans :

Refer Unit-I, Q.No. 14

UNIT - II

1. **Explain briefly about Measures of Central Tendency.**

Ans :

Refer Unit-II, Q.No. 3

2. **Explain the various ways of Measure of variability.**

Ans :

Refer Unit-II, Q.No. 4

3. **What is Data visualization? Explain the importance of Data Visualization.**

Ans :

Refer Unit-II, Q.No. 5

4. What are the techniques of Data Visualization?

Ans :

Refer Unit-II, Q.No. 8

5. How "data visualization technique –Tables" can display data analysis reports using Ms.Excel?

Ans :

Refer Unit-II, Q.No. 9

6. Explain briefly about "Cross tabulations charts" by using Ms. Excel?

Ans :

Refer Unit-II, Q.No. 10

7. Explain briefly about Gantt Chart.

Ans :

Refer Unit-II, Q.No. 13

8. Explain briefly about pivot charts.

Ans :

Refer Unit-II, Q.No. 14

9. What are the steps to create interactive Excel Dash Board?

Ans :

Refer Unit-II, Q.No. 16

UNIT - III

1. Explain the concept of regression analysis.

Ans :

Refer Unit-III, Q.No. 4

2. Explain the assumptions of simple linear regressions ?

Ans :

Refer Unit-III, Q.No. 7

3. Explain the concept of simple linear regression by using MS Excel.

Ans :

Refer Unit-III, Q.No. 8

4. What is predictive modeling? Explain different types of Predictive Modeling.

Ans :

Refer Unit-III, Q.No. 11

5. Explain briefly about Forecasting techniques.

Ans :

Refer Unit-III, Q.No. 13

6. Explain the various techniques are used in data mining.

Ans :

Refer Unit-III, Q.No. 17

7. Explain the various approaches for data mining with Micro Strategy.

Ans :

Refer Unit-III, Q.No. 19

8. What is data exploration ? Explain the Steps of Data Exploration and Preparation.

Ans :

Refer Unit-III, Q.No. 20

9. Define Business Intelligence (BI). Discuss Characteristics, Need and Stages of BI.

Ans :

Refer Unit-III, Q.No. 25

10. Explain the History and Evolution of Business Intelligence.

Ans :

Refer Unit-III, Q.No. 26

11. Explain the Classification of Data Mining.

Ans :

Refer Unit-III, Q.No. 28

12. Explain the process of cause and effect analysis.

Ans :

Refer Unit-III, Q.No. 32

UNIT - IV

1. State the assumptions and applications of LPP.

Ans :

Refer Unit-IV, Q.No. 3

2. Write the computational procedure for simplex method.

Ans :

Refer Unit-IV, Q.No. 6

3. Explain briefly about Non Linear Programming?

Ans :

Refer Unit-IV, Q.No. 7

4. Explain briefly about the Cutting Plane Method.

Ans :

Refer Unit-IV, Q.No. 8

5. Explain briefly about the term decision analysis?

Ans :

Refer Unit-IV, Q.No. 11

6. Explain about decision making under uncertainty.

Ans :

Refer Unit-IV, Q.No. 13

7. Explain various Techniques of Text Analytics.

Ans :

Refer Unit-IV, Q.No. 16

8. Discuss the various of tools of web analytics.

Ans :

Refer Unit-IV, Q.No. 21

UNIT - V

1. Explain the various types of operators in R program.

Ans :

Refer Unit-V, Q.No. 4

2. What is R package?

Ans :

Refer Unit-V, Q.No. 6

3. How the data can be read and write in R?

Ans :

Refer Unit-V, Q.No. 8

4. Explain different types of functions?

Ans :

Refer Unit-V, Q.No. 10

5. Explain briefly about control statements.

Ans :

Refer Unit-V, Q.No. 11

6. Explain briefly about loop statement?

Ans :

Refer Unit-V, Q.No. 15

7. Explain briefly about data frame in R?

Ans :

Refer Unit-V, Q.No. 18

8. How data can be managed in R?

Ans :

Refer Unit-V, Q.No. 20

UNIT I

INTRODUCTION TO BUSINESS ANALYTICS:

Definition of Business Analytics, Categories of Business Analytical methods and models, Business Analytics in practice, Big Data - Overview of using Data, Types of Data. Business Decision Modeling

1.1 Definition of Business Analytics

Q1. What is Business analytics ?

Ans :

Business analytics is the practice of iterative, methodical exploration of an organization's data, with an emphasis on statistical analysis. Business analytics is used by companies committed to data-driven decision-making.

Definitions

(i) **According to Schaer (2018),** "allows your business to make predictive analysis rather than reacting to changes in data".

(ii) **According to Gabelli School of Business (2018),** "involves applying models, methods, and tools to data, producing insights that lead to informed business decisions"

(iii) **According to Wells (2008),** "the application of logic and mental processes to find meaning in data"

(iv) **According to Lynda (2018),** "allows us to learn from the past and make better predictions for the future".

Business analytics (BA) refers to the skills, technologies, practices for continuous iterative exploration and investigation of past business performance to gain insight and drive business planning. Business analytics focuses on developing new insights and understanding of business performance based on data and statistical methods. In contrast, business intelligence traditionally focuses on using a consistent set of metrics to both measure past performance and guide business planning, which is also based on data and statistical methods. Business analytics makes extensive use of statistical analysis, including explanatory and predictive modeling, and fact-based management to drive decision making. It is therefore closely related to management science. Analytics may be used as input for human decisions or may drive fully automated decisions. Business intelligence is querying, reporting, online analytical processing (OLAP), and "alerts."

In other words, querying, reporting, OLAP, and alert tools can answer questions such as what happened, how many, how often, where the problem is, and what actions are needed. Business analytics can answer questions like why is this happening, what if these trends continue, what will happen next (predict), and what is the best outcome that can happen (optimize).

Business analytics. Abbreviated as BA, business analytics is the combination of skills, technologies, applications and processes used by organizations to gain insight in to their business based on data and statistics to drive business planning

Q2. What is Data for business Analytics?

Ans :

A business analytics is used to gain insights that inform business decisions and can be used to automate and optimize business processes. Data-driven companies treat their data as a corporate asset and leverage it for a competitive advantage. Successful business analytics depends on data quality, skilled analysts who understand the technologies and the business, and an organizational commitment to data-driven decision-making.

Business analytics examples

- Business analytics techniques break down into two main areas. The first is basic business intelligence. This involves examining historical data to get a sense of how a business department, team or staff member performed over a particular time. This is a mature practice that most enterprises are fairly accomplished at using.
- The second area of business analytics involves deeper statistical analysis. This may mean doing predictive analytics by applying statistical algorithms to historical data to make a prediction about future performance of a product, service or website design change. Or, it could mean using other advanced analytics techniques, like cluster analysis, to group customers based on similarities across several data points. This can be helpful in targeted marketing campaigns, for example.

  Business analytics tools come in several different varieties:

  1. Data visualization tools
  2. Business intelligence reporting software
  3. Self-service analytics platforms
  4. Statistical analysis tools
  5. Big data platforms
- Self-service has become a major trend among business analytics tools. Users now demand software that is easy to use and doesn't require specialized training. This has led to the rise of simple-to-use tools from companies such as Tableau and Qlik, among others. These tools can be installed on a single computer for small applications or in server environments for enterprise-wide deployments. Once they are up and running, business analysts and others with less specialized training can use them to generate reports, charts and web portals that track specific metrics in data sets
- Once the business goal of the analysis is determined, an analysis methodology is selected and data is acquired to support the analysis. Data acquisition often involves extraction from one or more business systems, data cleansing and integration into a single repository, such as a data warehouse or data mart. The analysis is typically performed against a smaller sample set of data.
- Analytics tools range from spreadsheets with statistical functions to complex data mining and predictive modeling applications. As patterns and relationships in the data are uncovered, new questions are asked, and the analytical process iterates until the business goal is met.
- Deployment of predictive models involves scoring data records - typically in a database - and using the scores to optimize real-time decisions within applications and business processes. BA also supports tactical decision-making in response to unforeseen events. And, in many cases, the decision-making is automated to support real-time responses.

1.2 Categories of Business Analytical Methods and Models

Q3. Explain the different business analytical methods.

Ans : **(Imp.)**

There are four types in business analytics:

(i) Prescriptive: This type of analysis reveals what actions should be taken. This is the most valuable kind of analysis and usually results in rules and recommendations for next steps.

(ii) Predictive: An analysis of likely scenarios of what might happen. The deliverables are usually a predictive forecast.

(iii) Diagnostic : A look at past performance to determine what happened and why. The result of the analysis is often an analytic dashboard.

(iv) Descriptive : What is happening now based on incoming data. To mine the analytics, you typically use a real-time dashboard and/or email reports.

1. **Prescriptive analytics**

It is really valuable, but largely not used. Where big data analytics in general sheds light on a subject, prescriptive analytics gives you a laser-like focus to answer specific questions. For example, in the health care industry, you can better manage the patient population by using prescriptive analytics to measure the number of patients who are clinically obese, then add filters for factors like diabetes and LDL cholesterol levels to determine where to focus treatment. The same prescriptive model can be applied to almost any industry target group or problem.

2. **Predictive analytics**

It use big data to identify past patterns to predict the future. For example, some companies are using predictive analytics for sales lead scoring. Some companies have gone one step further use predictive analytics for the entire sales process, analyzing lead source, number of communications, types of communications, social media, documents, CRM data, etc. Properly tuned predictive analytics can be used to support sales, marketing, or for other types of complex forecasts.

3. **Diagnostic analytics**

They are used for discovery or to determine why something happened. For example, for a social media marketing campaign, you can use descriptive analytics to assess the number of posts, mentions, followers, fans, page views, reviews, pins, etc. There can be thousands of online mentions that can be distilled into a single view to see what worked in your past campaigns and what didn't.

4. **Descriptive analytics**

Descriptive analysis (or) data mining are at the bottom of the big data value chain, but they can be valuable for uncovering patterns that offer insight. A simple example of descriptive analytics would be assessing credit risk; using past financial performance to predict a customer's likely financial performance. Descriptive analytics can be useful in the sales cycle, for example, to categorize customers by their likely product preferences and sales cycle.

Q4. Explain the different models in Business Analytics?

Ans : **(Imp.)**

An analytical model is simply a mathematical equation that describes relationships among variables in a historical data set. The equation either estimates or classifies data values. In essence, a model draws a "line" through a set of data points that can be used to predict outcomes. What is a business analysis model?

Simply put, a business analysis model outlines the steps a business takes to complete a specific process, such as ordering a product or on boarding a new hire. Process modeling (or mapping) is key to improving process efficiency, training, and even complying with industry regulations.

Because there are many different kinds of processes, organizations, and functions within a business, BAs employ a variety of visual models to map and analyze data.

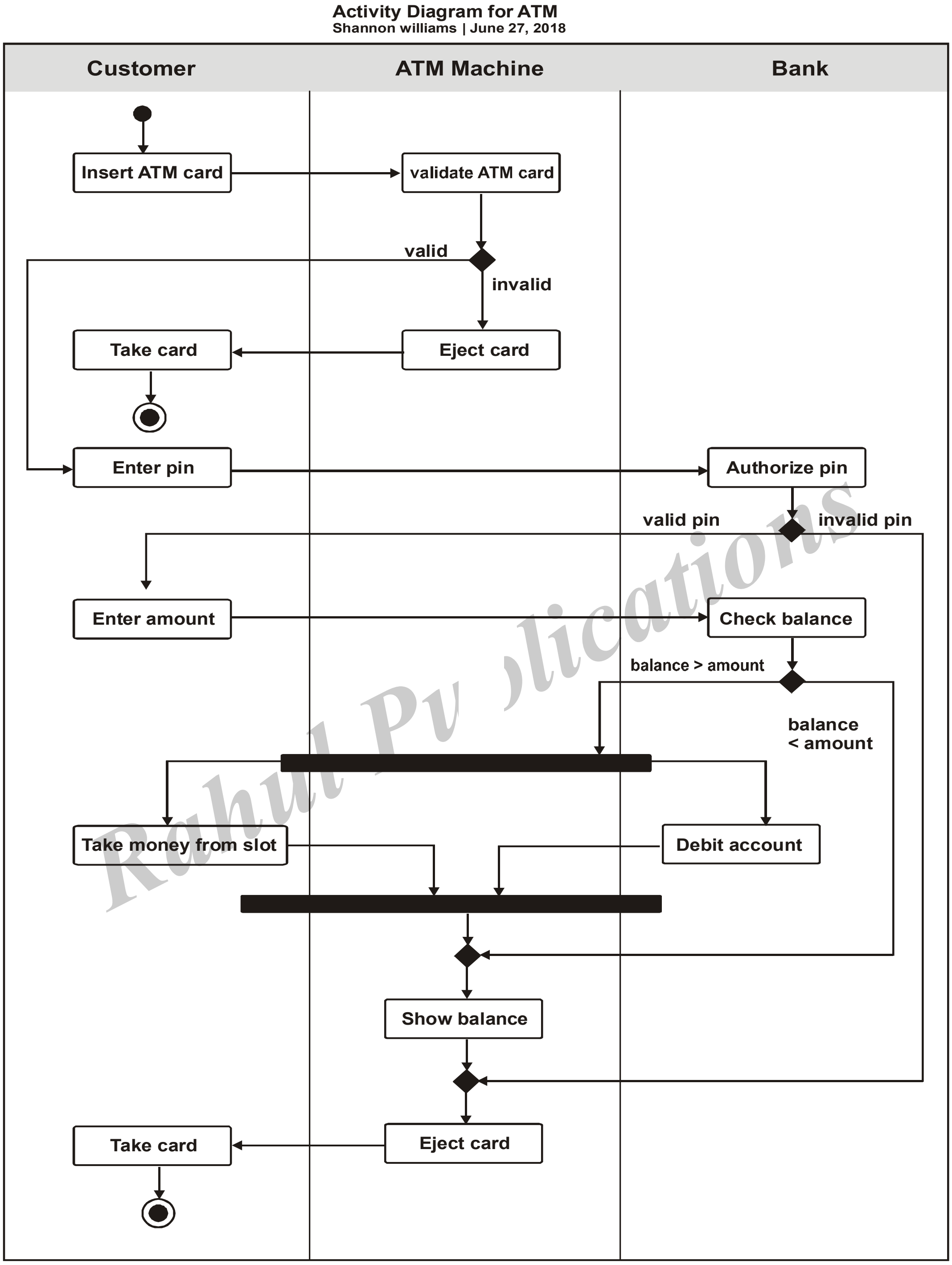
The following the different models:

1. **Activity diagrams**

Activity diagrams are a type of UML behavioural diagram that describes what needs to happen in a system. They are particularly useful for communicating process and procedure to stakeholders from both the business and development teams.

A Business analytical might use an activity diagram to map the process of logging in to a website or completing a transaction like withdrawing or depositing money

Activity diagram for ATM

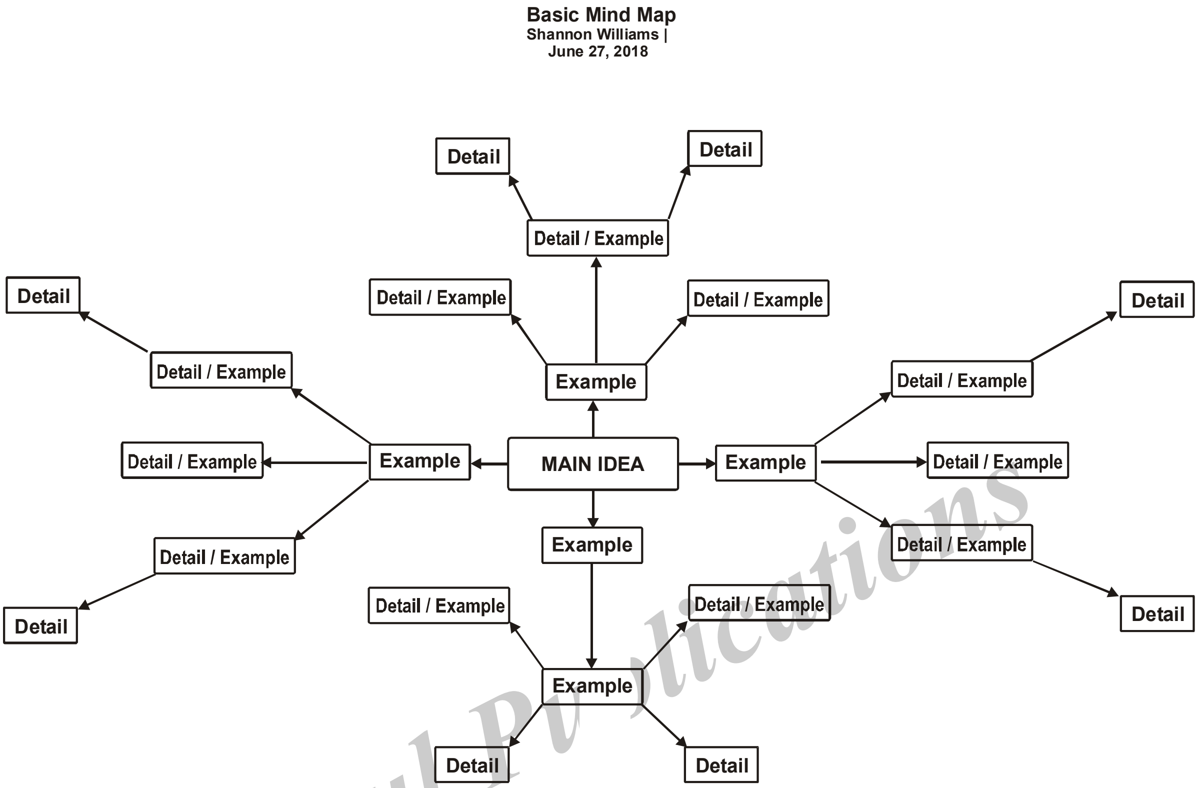


2. Feature mind maps

Business diagrams aren't just for late-stage analysis or documentation. They are also useful during a project's initial brainstorming phase. Feature mind maps help BAs organize the sometimes messy brainstorm process so that ideas, concerns, and requests are clearly captured and categorized.

This visual ensures initial details and ideas don't fall through the cracks so you can make informed decisions about project direction, goals, and scope down the line.

Basic mind map



3. Product roadmaps

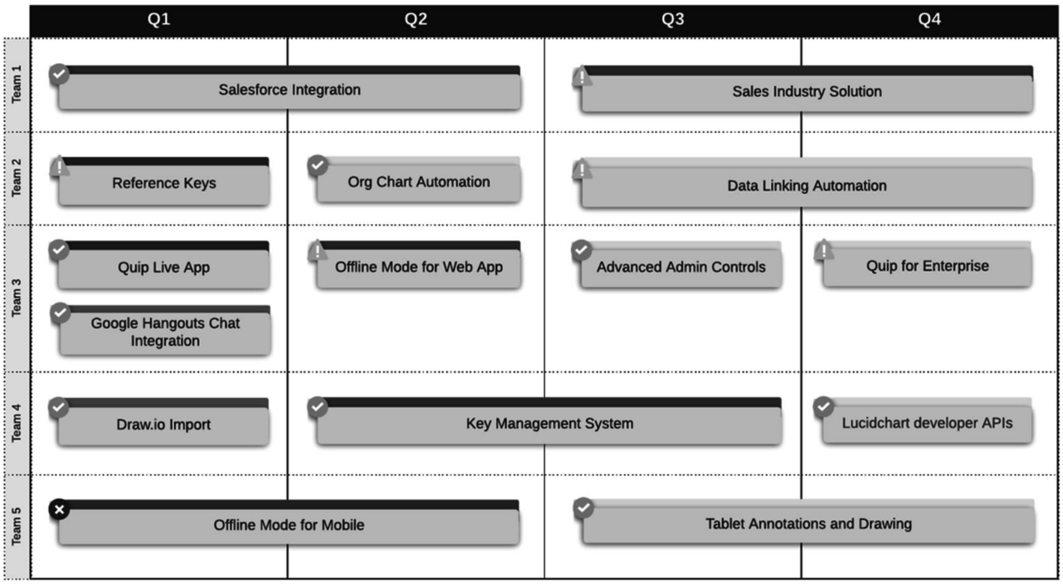
Product (or feature) roadmaps outline the development and launches of a product and its features. They are a focused analysis of a product's evolution, which helps developers and other stakeholders focus on initiatives that add direct value to the user.

The beauty of product roadmaps lies in their flexibility and range of applications. BAs can create different product roadmaps to illustrate different information, including:

- Maintenance and bug fixes
- Feature releases
- High-level strategic product goals

While product roadmaps are commonly used internally by development teams, they are also useful resources for other groups like sales.

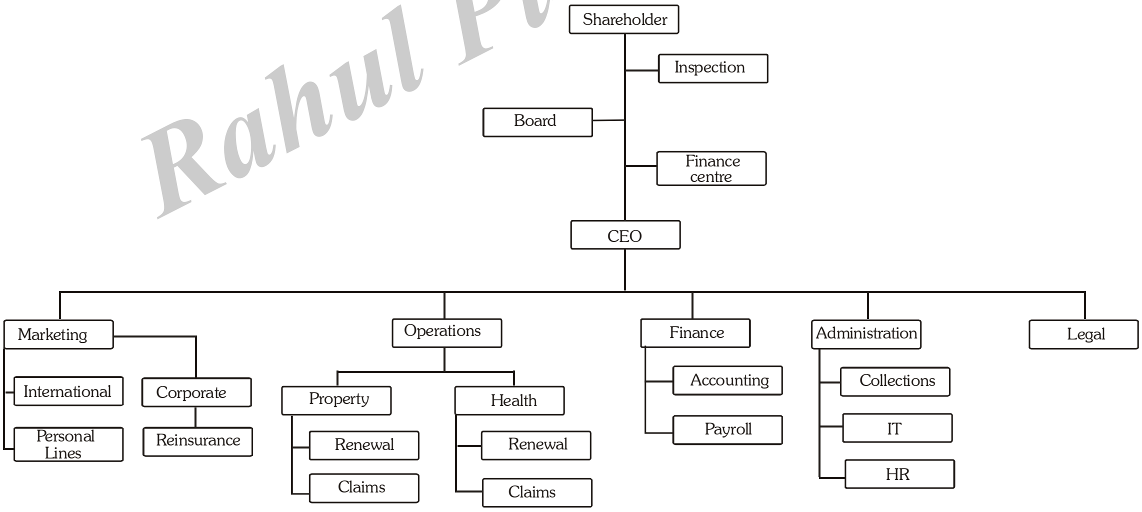
A defined product outline and schedule helps sales stay on the same page as the developers so they can deliver accurate, updated information to their prospects and clients. Because of their versatility and broad applications across teams and organizations, product roadmaps are a core part of an analyst's toolbox.



4. Organizational charts

Organizational charts outline the hierarchy of a business or one of its departments or teams. They are especially helpful reference charts for employees to quickly understand how the company is organized and identify key stakeholders and points of contact for projects or queries.

Additionally, organizational charts prove useful for stakeholder analysis and modeling new groupings and teams following organizational shifts.



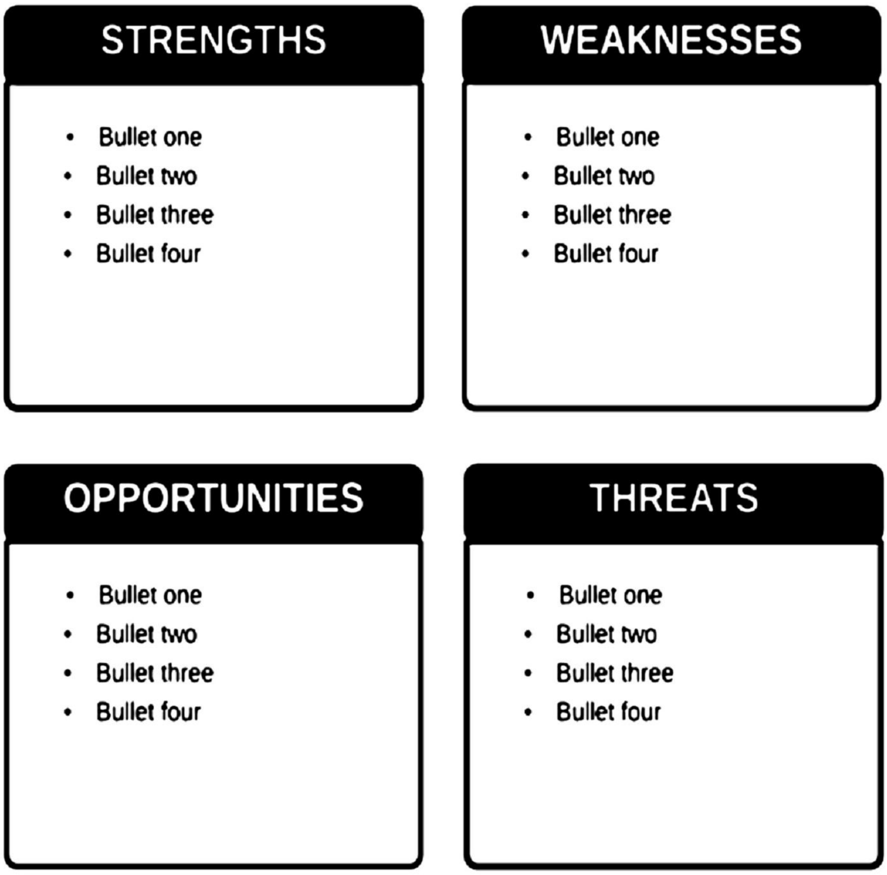
5. SWOT analysis

The SWOT analysis is a fundamental tool in a Business analytics. SWOT stands for strengths, weaknesses, opportunities, and threats. A SWOT analysis evaluates a business's strengths and weaknesses and identifies any opportunities or threats to that business.

SWOT analysis helps stakeholders make strategic decisions regarding their business. The goal is to capitalize on strengths and opportunities while reducing the impact of internal or external threats and weaknesses.

From a visual modeling perspective, SWOT analysis is fairly straight forward. A typical model will have four boxes or quadrants-one for each category-with bulleted lists outlining the respective results.

SWOT Analysis



6. User interface wireframe

Another essential business diagram is the UI wireframe. Software development teams use wireframes (also called mockups or prototypes) to visually outline and design a layout for a specific screen. In other words, wireframes are the blueprints for a website or software program. They help stakeholders assess navigational needs and experience for a successful practical application.

Wireframes range from low-fidelity to high-fidelity prototypes. Low-fidelity wireframes are the most basic outlines, showing only the bare-bones layout of the screen. High-fidelity wireframes are typically rendered in the later planning stages and will include specific UI elements (e.g., buttons, drop-down bars, text fields, etc.) and represent how the final implementation should look on the screen.

Website Design Wireframe (Click on image to modify online)

7. Process flow diagram

A process flow diagram (PFD) is typically used in chemical and process engineering to identify the basic flow of plant processes, but it can also be used in other fields to help stakeholders understand how their organization operates.

A PFD is best used to:

- Document a process.
- Study a process to make changes or improvements.
- Improve understanding and communication between stakeholders.

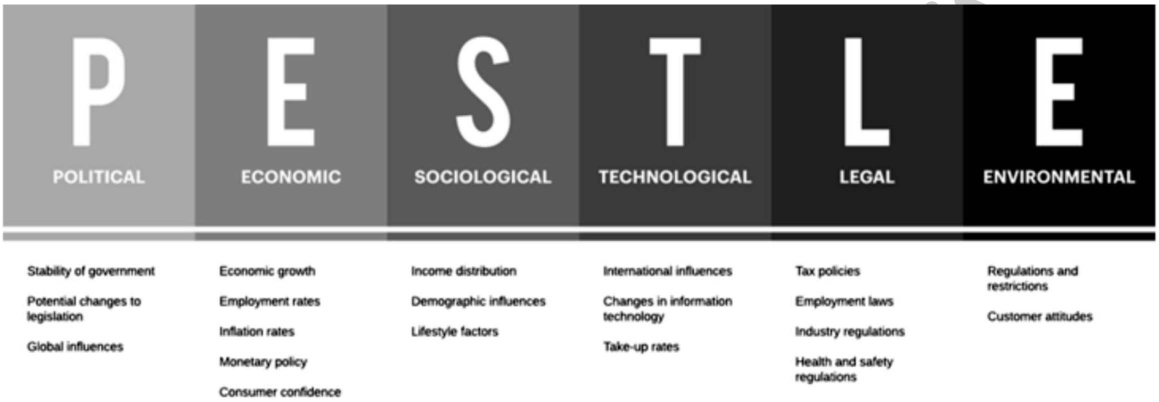
These diagrams focus on broad, high-level systems rather than annotating minor process details.

8. PESTLE analysis

A PESTLE analysis often goes hand-in-hand with a SWOT analysis. PESTLE evaluates external factors that could impact business performance. This acronym stands for six elements affecting business: political, economic, technological, environmental, legal, and sociological.

PESTLE analysis assesses the possible factors within each category, as well as their potential impact, duration of effect, type of impact (i.e., negative or positive), and level of importance.

This type of business analysis helps stakeholders manage risk, strategically plan and review business goals and performance, and potentially gain an advantage over competitors.



POLITICAL	ECONOMIC	SOCIOLOGICAL	TECHNOLOGICAL	LEGAL	ENVIRONMENTAL
Stability of government	Economic growth	Income distribution	International influences	Tax policies	Regulations and restrictions
Potential changes to legislation	Employment rates	Demographic influences	Changes in information technology	Employment laws	Customer attitudes
Global influences	Inflation rates	Lifestyle factors	Take-up rates	Industry regulations	
	Monetary policy			Health and safety regulations	
	Consumer confidence				

9. Entity-relationship diagram

An entity-relationship diagram (ER diagram) illustrates how entities (e.g., people, objects, or concepts) relate to one another in a system. For example, a logical ER diagram visually shows how the terms in an organization's business glossary relate to one another.

ER diagrams comprise three main parts:

- Entities
- Relationships
- Attributes

Attributes apply to the entities, describing further details about the concept. Relationships are where the key insights from ER diagrams arise. In a visual model, the relationships between entities are illustrated either numerically or via crow's foot notation.

These diagrams are most commonly used to model database structures in software engineering and business information systems and are particularly valuable tools for Business analytics in those fields.

Q5. Discuss briefly about role of business analytics in current business environment.

Ans : **(Imp.)**

1. Financial Analytics

Organizations use predictive models for forecasting future financial performance for constructing financial instruments like derivatives and assessing the risk involved in investment projects and portfolios. They also use prescriptive models for creating optimal capital budgeting plans for constructing optimal portfolios of investments and allocating assets. Addition to this, simulation is also used for ascertaining risk in the financial sector.

Example : GE Asset Management utilizes optimization models of analytics to make investment decisions of cash received from various sources. The approximate benefit obtained from using optimization models over a five-year period was $ 75 million.

2. Marketing Analytics

Business analytics is used in marketing for obtaining a better understanding of consumer behaviours by using the scanned data and social networking data. It leads to efficient use of advertising budgets, improved demand forecasting, effective pricing strategies, increased product line management and improved customer loyalty and satisfaction. Marketing analytics has gained much interest due to the data generated from social media.

Example : NBC Universal utilizes a predictive model every year to aid the annual up front market. An upfront market is a period in ending of May when every TV network sells most of the on-air advertisements for the upcoming season of television. The results of forecasting model are utilized by more than 200 NBC sales for supporting sales and pricing decisions.

3. Human Resource (HR) Analytics

HR function utilizes analytics to ensure that the organization consists of the employees with required skills to meet its needs, to ensure that it achieves its diversity goals and to ensure that it is hiring talent of the highest quality and also offering an environment which retains it.

Example : Sears Holding Corporation (SHC) owners of Roebuck Company, retailers Kmart and Sears. They made a team of HR analytics inside the corporate HR function. They apply predictive and descriptive analytics for tracking and influencing retention of employees and for supporting employee hiring.

4. Health Care Analytics

Health care organizations utilize analytics to provide more effective treatment and control cost. They use descriptive, prescriptive and predictive analytics for improving patient flow, staff and facility scheduling, purchasing and control of inventory. However, prescriptive analytics is specially used for the purpose of treatment and diagnosis. It is the most important proven utility of analytics.

Example : Memorial Sloan-Kettering Cancer Center along with Georgia Institute of Technology created a real-time prescriptive model for determining the optimal placement of radio active seeds for prostate cancer treatment. The results led to requirements of 20-30% lesser seeds and less invasive and faster procedure.

5. Supply Chain Analytics

Analytics is used by logistics and supply chain management to achieve efficiency. The entire spectrum of analytics is utilized by them. Various organizations such as UPS and FedEx apply analytics for efficient delivery of goods. Analytics helps them in optimal sorting of goods, staff and vehicle scheduling and vehicle routing, which helps in increasing the profitability. Analytics enable better processing control, inventory and more effective supply chains.

Example : ConAgra Foods utilized the prescriptive and predictive analysis for a better plan capacity utilization by incorporation of inherent uncertainty in pricing of commodities. ConAgra Foods attained a 100% return on investment in just three months.

6. **Analytics for Government and Non-Profit Organizations**

Government and non-profit organizations apply analytics for driving out inefficiencies and increasing the accountability and effectiveness of programs. During the period of Word War II, advanced analytics was first applied by the English and U.S. Military. Analytics applicability is very extensive in government agencies from elections to tax collections. Non-profit organizations utilize analytics for ensuring the accountability and effectiveness to their clients and donors.

Examples : The New York State Department incorporated with IBM for using prescriptive analytics in developing a more efficient tax collection approach.

Catholic Relief Services (CRS) is a non-profit organization which is the official international humanitarian agency of the U.S. Catholic community. This offer helps to the victims of both human-made and natural disasters. It also offers various other services through its agricultural, educational and health programs. It utilizes analytical spread sheet model for helping in the annual budget allocation based on the effects of its relief programs and efforts in various countries.

7. **Sports Analytics**

Analytical applicability in area of sports became popular when a renowned author Michael Lewis published Money ball in the year 2003. The book explained how the athletics of Oakland applied an analytical approach for evaluating players for assembling a competitive team with a limited budget. Analytics is used for evaluation of on-field strategy which is a common thing in professional sports. Analytics is also used in off-the-field decisions to ensure customer satisfaction.

Example : Professional sports teams utilize analytics for assessing players for the amateur drafts and for decision making of contract negotiations offered to the players. Various franchises across many major sports utilize prescriptive analytics for adjusting the ticket prices throughout the season for reflecting the potential demand and relative attractiveness for every game.

8. **Web Analytics**

The analysis of internet activity including visits of users to social media sites like Linkedln and Facebook and other websites is called web analytics. It plays a vital role in sales or promotions of products and services. Through internet various leading companies utilize advanced and descriptive analytics by applying them to the data gathered from online experiments for determining the best way for configuration of ads, websites and proper utilization of social networks to promote products and services. Online experimentation is exposing various group of visitors to varied versions of a website and then tracking the results. These experiments can be conducted without risking the overall business disruption of the company due to several number of internet users. But these experiments have proven to be invaluable as they enable the company for using trial-and-error method of determine statistically the reasons for differences in the sales and website traffic.

Q6. Explain the various challenges in business analytics.

Ans :

- **Executive Ownership:** Business Analytics requires buy-in from senior leadership and a clear corporate strategy for integrating predictive models.
- **IT Involvement:** Technology infrastructure and tools must be able to handle the data and Business Analytics processes.
- **Available Production Data vs. Cleansed Modeling Data :** Watch for technology

infrastructure that restrict available data for historical modeling, and know the difference between historical data for model development and real-time data in production.

- **Project Management Office (PMO)** : The correct project management structure must be in place in order to implement predictive models and adopt an agile approach.
- **End user Involvement and Buy-In :** End users should be involved in adopting Business Analytics and have a stake in the predictive model.
- **Change Management :** Organizations should be prepared for the changes that Business Analytics bring to current business and technology operations.
- **Explain ability vs. the "Perfect Lift"**: Balance building precise statistical models with being able to explain the model and how it will produce results.

1.3 Business Analytics in Practice

Q7. Explain the role of Business Analytics in Best Practices.

Ans : **(Imp.)**

Adopting and implementing Business Analytics is not something a company can do overnight. But, if a company follows some best practices for Business Analytics, they will get the levels of insight they seek and become more competitive and successful. We list some of the most important best practices for Business Analytics here, though your organization will need to determine which best practices are most fitting for there needs.

- Know the objective for using Business Analytics. Define the business use case and the goal ahead of time.
- Define the criteria for success and failure.
- Select the methodology and be sure to know the data and relevant internal and external factors
- Validate models using to predefined success and failure criteria

Business Analytics is critical for remaining competitive and achieving success. When they get BA best practices in place and get buy-in from all stakeholders, the organization will benefit from data-driven decision making.

1.4 Big Data - Overview of Data

Q8. What is Big Data?

Ans :

- Big Data is a phrase used to mean a massive volume of both structured and unstructured data that is so large it is difficult to process using traditional database and software techniques. In most enterprise scenarios the volume of data is too bigor it moves too fast or it exceeds current processing capacity.
- Big data is a term that is used to describe data that is high volume, high velocity, and/or high variety; requires new technologies and techniques to capture, store, and analyze it; and is used to enhance decision making, provide insight and discovery, and support and optimize processes.

Technologies in big data

We can categories them into two (storage and Querying/Analysis).

- Apache Hadoop. Apache Hadoop is a Java based free software framework that can effectively store large amount of data in a cluster. ...
- Microsoft HDInsight. ...
- NoSQL. ...
- Hive. ...
- Sqoop. ...
- PolyBase. ...
- Big data in EXCEL. ...
- Presto

- Some of the most common of those big data challenges include the following:
- Dealing with data growth. ...
- Generating insights in a timely manner. ...
- Recruiting and retaining big data talent. ...
- Integrating disparate data sources. ...
- Validating data. ...
- Securing big data. ...
- Organizational resistance

Q9. Explain the evolution of Big Data.

Ans :

The evolution of big data is discussed below,

(i) 1970s and before

(ii) 1980s and 1990s

(iii) 2000s and beyond

(i) 1970s and before : The data generation and storage of 1970s and before is fundamentally primitive and structured. This era is termed as the era of mainframes, as it stores the basic data.

(ii) 1980s and 1990s : In 1980s and 1990s the evolution of relational data bases took please. The relational data utilization is complex and thus this era comprises of data intensive applications.

(iii) 2000s and beyond : The World Wide Web (www) and the Internet of Things (IOT) have an aggression of structured, unstructured and multimedia data. The data driven is complex and unstructured.

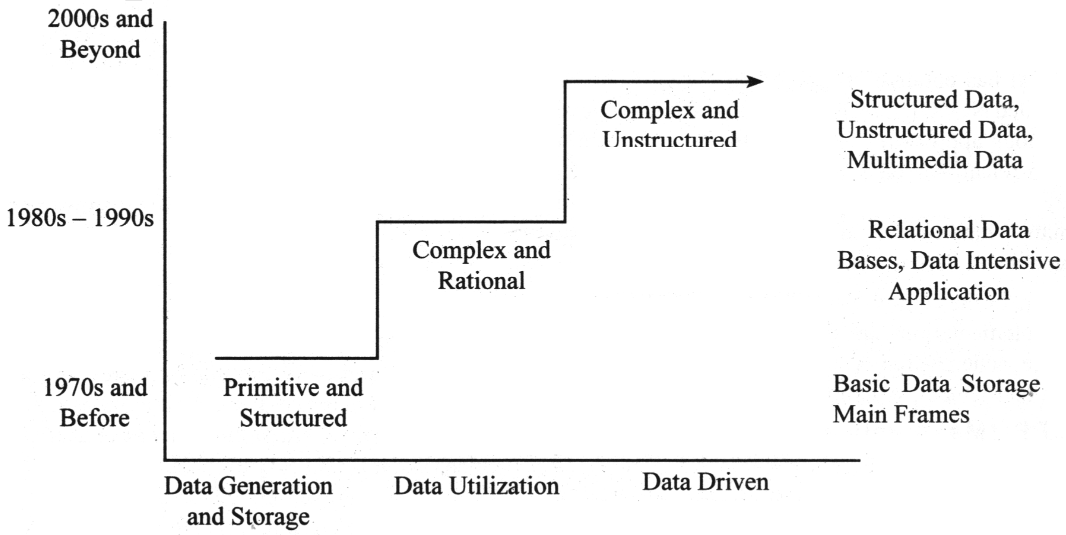

2000s and Beyond
1980s – 1990s
1970s and Before
Primitive and Structured
Complex and Rational
Complex and Unstructured
Structured Data, Unstructured Data, Multimedia Data
Relational Data Bases, Data Intensive Application
Basic Data Storage Main Frames
Data Generation and Storage
Data Utilization
Data Driven

Fig.: The Evolution of Big Data

Q10. Explain the dimensions of big data?

Ans :

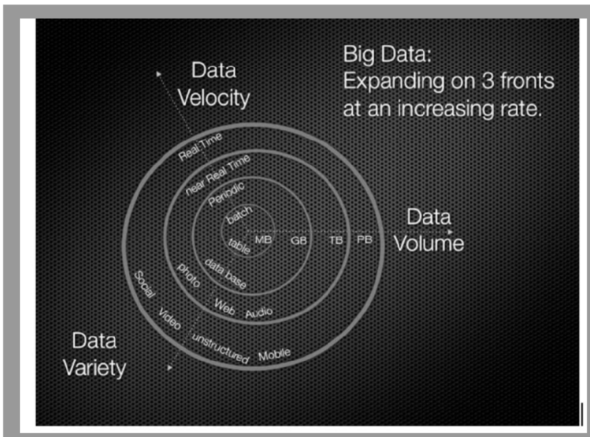
Big data refers to datasets whose size is beyond the ability of typical database software tool to capture, store, managed and analyze. Big data is data that goes beyond the traditional limits of data along four dimensions:

i) Data Volume

ii) Data Variety

iii) Data Velocity

iv) Variability



(i) Data Volume

Data Volume can be measured by quality of transactions, events and amount of history. Big Data isn't just a description of raw volume. The real challenge is identifying or developing most cost-effective and reliable methods for extracting value from all the terabytes and petabytes of data now available. That's where Big Data analytics become necessary.

(ii) Data Variety

It is the assortment of data. Traditionally data, especially operational data, is structured as it is put into a database based on the type of data (i.e., character, numeric, floating point, etc.). Wide variety of data: Internet data (Social media, Social Network-Twitter, Face book),

(iii) Data Velocity

It is the measure of how fast the data is coming in. Remember our Facebook example. 250 billion images may seem like a lot. But if you want your mind blown, consider this: Facebook users upload more than 900 million photos a day. So that 250 billion number from last year will seem like a drop in the bucket in a few months. Facebook has to handle a tsunami of photographs every day. It has to ingest it all, process it, file it, and somehow, later, be able to retrieve it.

(iv) Variability

The increase in the range of values typical of a large data set - and value, which addresses the need for valuation of enterprise data.

Q11. Explain the relationship of big data with other areas?

Ans : **(Imp.)**

Big data models Improve Operational Efficiencies Increase Revenues Achieve Competitive Differentiation Reduce risks and costs Sell to microtrends Offer new services Save time Enable self service Seize market share Lower complexity Improve customer experience Incubate new ventures Enable self service Detect fraud

i) Digital Marketing.

ii) Financial Services

iii) Big data and Advances in health care

iv) Advertising

(i) Digital Marketing

Introduction Database Marketers, Pioneers of Big Data Big Data & New School of Marketing Cross Channel Life cycle Marketing Social and Affiliate Marketing Empowering marketing with Social Intelligence Introduction Digital marketing encompasses using any sort of online media (profit or non-profit) for driving people to a website, a mobile app etc. and retaining them and interacting with them to understand what consumers really want.

Digital \marketing is easy when consumers interact with corporate‘ primary platform (i.e.. The one the corporate own) because corporate get good information about them. But corporate get very little information once people start interacting with other platforms (e.g., Face book, Twitter, Google +). One of the problems that have to be dealt with in a digital existence, is that corporate do not have access to data for all consumers (i.e., There is very little visibility about people when they interact in social networking sites). So corporate lose control of their ability to access the data they need, in order to make smart, timely decisions. Big data on the web will completely transform a company's ability to understand the effectiveness of its marketing and the ability to understand how its competitors are behaving. Rather than a few people making big decisions, the organizations will be able to make hundreds and thousands of smart decisions every day.

(ii) Financial Services

Fraud & Big Data - Fraud is intentional deception made for personal gain or to damage another individual. - One of the most common forms of fraudulent activity is credit card fraud. - Social media and mobile phones are forming new frontiers fraud. - Capegemini financial services team believes that due to the nature of data streams and processing required BIG Data Technologies provide an optimal technology solution based on the following three Vs : 1. High volume: Years of consumer records and transactions (150 billion + 2. records per year). 3. High velocity: Dynamic transactions and social media info. 4. High variety: Social media plus other unstructured data such as customer E-mails, call center conversations as well as transactional structured data.

(iii) Big data and Healthcare

Big data promises enormous revolution in healthcare, with the advancements in everything from the management of chronic disease to the delivery of personalized medicine. In addition to saving and improving lives, Big Data has the potential to transform the entire healthcare system by replacing guesswork and institution with objective data-driven science.-The healthcare industry now has huge amount of data: from biological data such as gene expression, Special Needs Plans (SNPs), proteomics, metabolomics, and next-generation gene sequence data etc. The exponential growth in data is further accelerated by the digitization of patient level data stored in Electronic Medical Records (EMRs) or Electronic Health Records (EHRs) and Health Information Exchanges (HIEs) enhanced with data from imaging and test results, medical and prescription claims and personal health devices. - In addition to saving and improving lives, Big Data has the potential to transform the entire health care system by replacing guesswork and intuition with objective, data-driven science

(iv) Big Data is changing the way advertisers address three related needs. (i) How much to spend on advertisements. (ii) How to allocate amount across all the marketing communication touch points. (iii) How to optimize advertising effectiveness. Given these needs advertisers need to measure their advertising end to end in terms of Reach, Response & Reaction. Reach, Resonance, and Reaction Reach: First part of reach is to identify the people who are most volumetrically responsive to their advertising and then answer questions such as what do those people watch? What do they do online? How to develop media plan against intended audience. The second part of reach is delivering advertisements to the right audience. That is, to understand if we are actually reaching our desired audience. If we think about the online world, it's a world where we can deliver 100 million impressions but we never really know for sure who our campaign was actually delivered to. If our intended audience is women aged 18 to 35, of our 100 million impressions, what percentage of impressions were actually delivered to the intended audience? What was the reach, what was the frequency, what was the delivery against the intended audience? Resonance: If we know whom we want to reach and we're reaching them efficiently with your media spend, the next question is, are our ads breaking through? Do people know they're from our brand? Are they changing attitudes? Are they making consumers more likely to want to buy our brand? This is what is called resonance. Reaction: Advertising must drive a behavioral reaction or it isn't really working. We have to measure the actual behavioral impact.

Q12. What are the categories which come under Big Data?

Ans :

Big data works on the data produced by various devices and their applications. Below are some of the fields that are involved in the umbrella of Big Data.

1. Black Box Data

It is an incorporated by flight crafts, which stores a large sum of information, which includes the conversation between crew members and any other communications (alert messages or any order passed) by the technical grounds duty staff.

2. Social Media Data

Social networking sites such as Face book and Twitter contains the information and the views posted by millions of people across the globe.

3. Stock Exchange Data

It holds information (complete details of in and out of business transactions) about the 'buyer' and 'seller' decisions in terms of share between different companies made by the customers.

4. Power Grid Data

The power grid data mainly holds the information consumed by a particular node in terms of base station.

5. Transport Data

It includes the data's from various transport sectors such as model, capacity, distance and availability of a vehicle.

6. Search Engine Data

Search engines retrieve a large amount of data from different sources of database.

Q13. Explain the importance of big data.

Ans :

The importance of big data is how you utilize the data which you own. Data can be fetched from any source and analyze it to solve that enable us in terms of

1. Cost reductions
2. Time reductions,
3. New product development and optimized offerings, and
4. Smart decision making.

Advantage of Big data Business Models:

Big data models Improve Operational Efficiencies Increase Revenues Achieve Competitive Differentiation Reduce risks and costs Sell to microtrends Offer new services Save time Enable self service Seize market share Lower complexity Improve customer experience Incubate new ventures Enable self service Detect fraud.

Industry Examples of Big data:

i) Digital Marketing.

ii) Financial Services

iii) Big data and Advances in health care

iv) Pioneering New Frontiers in medicine

v) Advertising

(i) Digital Marketing

Introduction Database Marketers, Pioneers of Big Data Big Data & New School of Marketing Cross Channel Life cycle Marketing Social and Affiliate Marketing Empowering marketing with Social Intelligence Introduction Digital marketing encompasses using any sort of online media (profit or non-profit) for driving people to a website, a mobile app etc. and retaining them and interacting with them to understand what consumers really want. Digital \marketing is easy when consumers interact with corporate' primary platform (i.e., The one the corporate own) because corporate get good information about them. But corporate get very little information once people start interacting with other platforms (e.g., Face book, Twitter, Google +). One of the problems that have to be dealt with in a digital existence, is that corporate do not have access to data for all consumers (i.e., There is very little visibility about people when they interact in social networking sites). So corporate lose control of their ability to access the data they need, in order to make smart, timely decisions. Big data on the web will completely transform a company's ability to understand the effectiveness of its marketing and the ability to understand how its competitors are behaving. Rather than a few people making big decisions, the organizations will be able to make hundreds and thousands of smart decisions every da

(ii) Financial Services

i) Fraud & Big Data - Fraud is intentional deception made for personal gain or to damage another individual. - One of the most common forms of fraudulent activity is credit card fraud. - Social media and mobile phones are forming new frontiers fraud. - Capegemini financial services team believes that due to the nature of data streams and processing required BIG Data Technologies provide an optimal technology solution based on the following three Vs : 1. High volume: Years of consumer records and transactions (150 billion + 2. records per year). 3. High velocity: Dynamic transactions and social media info. 4. High variety: Social media plus other unstructured data such as customer E-mails, call center conversations as well as transactional structured data.

(iii) Big data and Healthcare

Big data promises enormous revolution in healthcare, with the advancements in everything from the management of chronic disease to the delivery of personalized medicine. - In addition to saving and improving lives, Big Data has the potential to transform the entire healthcare system by replacing guesswork and institution with objective data-driven science. - The healthcare industry now has huge amount of data: from biological data such as gene expression, Special Needs Plans (SNPs), proteomics, metabolomics, and next-generation gene sequence data etc. The exponential growth in data is further accelerated by the digitization of patient level data stored in Electronic Medical Records (EMRs) or Electronic Health Records (EHRs) and Health Information Exchanges (HIEs) enhanced with data from imaging and test results, medical and prescription claims and personal health devices. - In addition to saving and improving lives, Big Data has the potential to transform the entire health care system by replacing guesswork and intuition with objective, data-driven science

(iv) Advertising and Big Data

Big Data is changing the way advertisers address three related needs. (i) How much to spend on advertisements. (ii) How to allocate amount across all the marketing communication touch points. (iii) How to optimize advertising effectiveness. Given these needs advertisers need to measure their advertising end to end in terms of Reach, Response & Reaction. Reach, Resonance, and Reaction Reach: First part of reach is to identify the people who are most volumetrically responsive to their advertising and then answer questions such as what do those people watch? What do they do online? How to develop media plan against intended audience. The second part of reach is delivering advertisements to the right audience. That is, to understand if we are actually reaching our desired audience. If we think about the online world, it's a world where we can deliver 100 million impressions but we never really know for sure who our campaign was actually delivered to. If our intended audience is women aged 18 to 35, of our 100 million impressions, what percentage of impressions were actually delivered to the intended audience? What was the reach, what was the frequency, what was the delivery against the intended audience? Resonance: If we know whom we want to reach and we're reaching them efficiently with your media spend, the next question is, are our ads breaking through? Do people know they're from our brand? Are they changing attitudes? Are they making consumers more likely to want to buy our brand? This is what is called resonance. Reaction: Advertising must drive a behavioral reaction or it isn't really working. We have to measure the actual behavioral impact.

Q14. Explain the life cycle of big data.

Ans : **(Imp.)**

Big Data Life Cycle

In today's big data context, the previous approaches are either incomplete or suboptimal. For example, the SEMMA methodology disregards completely data collection and pre-processing of different data sources. These stages normally constitute most of the work in a successful big data project.

A Big Data Analytics Cycle can be described by the following stages:

1. Business Problem Definition
2. Research
3. Human Resources Assessment
4. Data Acquisition
5. Data Mugging
6. Data Storage
7. Exploratory Data Analysis
8. Data Preparation for Modeling and Assessment
9. Modeling
10. Implementation

1. Business Problem Definition

This is a point common in traditional BI and big data analytics lifecycle. Normally, it is a non-trivial stage of a big data project to define the problem and evaluate correctly how much potential gain it may have for an organization. It seems obvious to mention this, but it has to be evaluated what are the expected gains and costs of the project.

2. Research

Analyze what other companies have done in the same situation. This involves looking for solutions that are reasonable for your company, even though it involves adapting other solutions to the resources and requirements that your company has. In this stage, a methodology for the future stages should be defined.

3. Human Resources Assessment

Once the problem is defined, it is reasonable to continue analyzing if the current staff is able to complete the project successfully. Traditional BI teams might not be capable to deliver an optimal solution to all the stages. So, it should be considered before starting the project if there is a need to outsource a part of the project or hire more people.

4. Data Acquisition

This section is key in a big data life cycle; it defines which type of profiles would be needed to deliver the resultant data product. It is a non-trivial step of the process; it normally involves gathering unstructured data from different sources. To give an example, it could involve writing a crawler to retrieve reviews from a website. This involves dealing with text, perhaps in different languages normally requiring a significant amount of time to be completed.

5. Data Mugging

Once the data is retrieved, for example, from the web, it needs to be stored in an easy-to-use format. To continue with the reviews examples, let's assume the data is retrieved from different sites where each has a different display of the data.

Suppose one data source gives reviews in terms of rating in stars. Therefore, it is possible to read this as a mapping for the response variable $y \in \{1, 2, 3, 4, 5\}$. Another data source gives reviews using two arrows system, one for up voting and the other for down voting. This would imply a response variable of the form $y \in$ {positive, negative}.

In order to combine both the data sources, a decision has to be made in order to make these two response representations equivalent. This can involve converting the first data source response representation to the second form, considering one star as negative and five stars as positive. This process often requires a large time allocation to be delivered with good quality.

6. Data Storage

Once the data is processed, it sometimes needs to be stored in a database. Big data technologies offer plenty of alternatives regarding this point. The most common alternative is using the Hadoop File System for storage that provides users a limited version of SQL, known as HTVE Query Language. This allows most analytics task to be done in similar ways as would be done in traditional BI data warehouses, from the user perspective. Other storage options to be considered are Mongo DB, Redis and SPARK.

This stage of the cycle is related to the human resources knowledge in terms of their abilities to implement different architectures. Modified versions of traditional data warehouses are still being used in large-scale applications. For example, Teradata and IBM offer SQL databases that can handle terabytes of data; open source solutions such as postgre SQL and MySQL are still being used for large-scale applications.

Even though there are differences in how the different storages work in the background, from the client side, most solutions provide a SQL API. Hence, having a good understanding of SQL is still a key skill to have for big data analytics.

This stage a priori seems to be the most important topic; in practice, this is not true. It is not even an essential stage. It is possible to implement a big data solution that would be working with real-time data. So, in this case, we only need to gather data to develop the model and then implement it in real time. So, there would not be a need to formally store the data at all.

7. Exploratory Data Analysis

Once the data has been cleaned and stored in a way that insights can be retrieved from it, the data exploration phase is mandatory. The objective of this stage is to understand the data. This is normally done with statistical techniques and also plotting the data. This is a good stage to evaluate whether the problem definition makes sense or is feasible.

8. Data Preparation for Modeling and Assessment

This stage involves reshaping the cleaned data retrieved previously and using statistical pre-processing for missing values imputation, outlier detection, normalization, feature extraction and feature selection.

9. Modelling

The prior stage should have produced several data sets for training and testing, e.g., a predictive model. This stage involves trying different models and looking forward to solving the business problem at hand. In practice, it is normally desired that the model would give some insight into the business. Finally, the best model or combination of models is selected evaluating its performance on a left-out data set.

10. Implementation

In this stage, the data product developed is implemented in the data pipeline of the company. This involves setting up a validation scheme while the data product is working in order to track its performance. For example, in case of implementing a predictive model, this stage would involve applying the model to new data and once the response is available, evaluate the model.

Q15. Explain the users of big data?

Ans :

The people who are using Big Data know better that, what Big Data is. Let's look at some such industries:

- **Healthcare:** Big Data has already started to create a huge difference in the healthcare sector. With the help of predictive analytics, medical professionals and HCPs are now able to provide personalized healthcare services to individual patients. Apart from that, fitness wearables, telemedicine, remote monitoring - all powered by Big Data and AI - are helping change lives for the better.
- **Academia:** Big Data is also helping enhance education today. Education is no more limited to the physical bounds of the classroom - there are numerous online educational courses to learn from. Academic institutions are investing in digital courses powered by Big Data technologies to aid the all-round development of budding learners.
- **Banking:** The banking sector relies on Big Data for fraud detection. Big Data tools can efficiently detect fraudulent acts in real-time such as misuse of credit/debit cards, archival of inspection tracks, faulty alteration in customer stats, etc.
- **Manufacturing:** According to TCS 2013 Global Trend Study, the most significant benefit of Big Data in manufacturing is improving the supply strategies and product quality. In the manufacturing sector, Big Data helps create a transparent infrastructure, thereby predicting uncertainties and incompetencies that can affect the business adversely.
- **IT:** One of the largest users of Big Data, IT companies around the world are using Big Data to optimize their functioning, enhance employee productivity and minimize risks in business operations. By combining Big Data technologies with ML and AI, the IT sector is continually powering innovation to find solutions even for the most complex of problems.

1.5 Data - Types of Data

Q16. What is Data and Explain the various types of data ?

Ans :

Meaning

Data are basic values or facts. Computers use many different types of data stored in digital format, such as text, numbers and multimedia. Data are organized in database tables, and database management systems are used to work with large databases. Data is a set of values of subjects with respect to qualitative or quantitative variables. Data and information are often used interchangeably; however data becomes information when it is viewed in context or in post-analysis.

(i) Structured Data

By structured data, we mean data that can be processed, stored and retrieved in a fixed format. It refers to highly organized information that can be readily and seamlessly stored and accessed from a database by simple search engine algorithms. For instance, the employee table in a company database will be structured as the employee details, their job positions, their salaries, etc. will be present in an organized manner.

(ii) Unstructured Data

Unstructured data refers to the data that lacks any specific form or structure whatsoever. This makes it very difficult and time-consuming to process and analyze unstructured data. Email is an example of unstructured data.

	Structured Data	**Unstructured Data**
Characteristics	➢ Pre-defined data models	➢ No pre-defined data model
	➢ Usually text only	➢ May be text, images, sound, video or other formats
	➢ Easy to search	➢ Difficult to search
Resides in	➢ Relational databases	➢ Applications
	➢ Data warehouses	➢ NoSQL databases
		➢ Data warehouses
		➢ Data lakes
Generated by	**Humans or Machines**	**Humans or Machines**
Typical applications	➢ Airline reservation systems	➢ Word processing
	➢ Inventory control	➢ Presentation software
	➢ CRM systems	➢ Email clients
	➢ ERP systems	➢ Tools for viewing or editing media
Examples	➢ **Dates**	➢ **Text Files**
	➢ Phone numbers	➢ Presentation software
	➢ Social security numbers	➢ Email messages
	➢ Credit card numbers	➢ Audio files
	➢ Customer names	➢ Video files
	➢ Addresses	➢ Images
	➢ Product names and numbers	➢ Surveillance imagery
	➢ Transaction information	

iii) Semi-structured

Semi-structured data pertains to the data containing both the formats mentioned above, i.e., structured and unstructured data. To be precise, it refers to the data that although has not been classified under a particular repository (database), yet contains vital information or tags that segregate individual elements within the data.

1.6 Business Decision Modeling

Q17. What is Decision Modeling?

Ans :

Decision modeling is a structured process that predicts the outcome of certain scenarios, offering valuable insights to business users. Decision models are a forecasting tool that provide an overview of all the potential possibilities of specific actions.

Decision modeling helps teams streamline their decision-making processes so they can prioritize their top business objectives. Even if they don't have a ton of information at their fingertips, managers can still use decision models to lay the groundwork for their decision networks and alter them accordingly.

Q18. Explain the advantages of Decision Modeling.

Ans :

Here are some of the major advantages of using decision models in your business.

1. **Make Decisions About New Products and Campaigns**

   Marketers are tasked with building brands, creating demand, promoting sales, and helping companies increase customer loyalty. To achieve this, they use decision models to gather real-time information about consumer behavior, including their preferences and spending patterns.

   This model helps them address questions such as:

   - Which product lines you should focus your market support?
   - How much money should you spend on your marketing campaign?
   - Which features of the product you should highlight in your marketing efforts?

2. **Decide Who's Best for the Job**

   If you want to assign specific tasks with tight deadlines to an employee, you can use a decision model to identify which employee performed best on similar tasks in the past, indicating who would be the best fit.

Q19. Discuss the various techniques of Business Decision Modeling.

Ans :

Different decision models act as reliable tools that facilitate the decision-making process. The most common decision models include:

1. **Rational Decision Model**

   This model requires you to follow a series of steps to find the best solution. It involves analyzing multiple solutions at the same time to come up with one that offers the best possible outcome. The steps are:

   1. Begin by defining the problem
   2. Identify the method you will use to evaluate possible solutions identified
   3. Determine the importance of each method
   4. Compile a list of all possible alternatives
   5. Select the best option or solution

2. **Bounded Rationality Model**

   Bounded reality decision modeling is focused on decisions that are "good enough" rather than perfect. That means that the selected decision is enough to address the current situation, but doesn't maximize the potential value in the situation a process known as satisficing.

   This model is used in incidents where quick decisions are required, as it takes less time and provides satisfying results. It's an ideal model when you're limited on time and/ or context.

3. **Recognition-Primed Model**

   This model uses prior experience and quick thinking to make decisions, often in a fast-paced decision-making environment. It involves the following:

   - Identifying any type of pattern in the information provided
   - Selecting a way that you think you can take action on it, then running it through your head
   - Creating possible solutions

Short Question and Answers

1. What is Business analytics?

Ans :

Business analytics is the practice of iterative, methodical exploration of an organization's data, with an emphasis on statistical analysis. Business analytics is used by companies committed to data-driven decision-making.

Definitions

(i) **According to Schaer (2018),** "allows your business to make predictive analysis rather than reacting to changes in data".

(ii) **According to Gabelli School of Business (2018),** "involves applying models, methods, and tools to data, producing insights that lead to informed business decisions"

(iii) **According to Wells (2008),** "the application of logic and mental processes to find meaning in data"

(iv) **According to Lynda (2018),** "allows us to learn from the past and make better predictions for the future".

2. Different business analytical methods.

Ans :

(i) **Prescriptive:** This type of analysis reveals what actions should be taken. This is the most valuable kind of analysis and usually results in rules and recommendations for next steps.

(ii) **Predictive:** An analysis of likely scenarios of what might happen. The deliverables are usually a predictive forecast.

(iii) **Diagnostic:** A look at past performance to determine what happened and why. The result of the analysis is often an analytic dashboard.

(iv) **Descriptive:** What is happening now based on incoming data. To mine the analytics, you typically use a real-time dashboard and/or email reports.

3. What is Big Data?

Ans :

- Big Data is a phrase used to mean a massive volume of both structured and unstructured data that is so large it is difficult to process using traditional database and software techniques. In most enterprise scenarios the volume of data is too bigor it moves too fast or it exceeds current processing capacity.
- Big data is a term that is used to describe data that is high volume, high velocity, and/or high variety; requires new technologies and techniques to capture, store, and analyze it; and is used to enhance decision making, provide insight and discovery, and support and optimize processes.

4. Dimensions of big data.

Ans :

(i) Data Volume

Data Volume can be measured by quality of transactions, events and amount of history. Big Data isn't just a description of raw volume. The real challenge is identifying or developing most cost-effective and reliable methods for extracting value from all the terabytes and petabytes of data now available. That's where Big Data analytics become necessary.

(ii) Data Variety

It is the assortment of data. Traditionally data, especially operational data, is structured as it is put into a database based on the type of data (i.e., character, numeric, floating point, etc.). Wide variety of data: Internet data(Social media ,Social Network-Twitter, Face book),

(iii) Data Velocity

It is the measure of how fast the data is coming in. Remember our Facebook example. 250 billion images may seem like a lot. But if you want your mind blown, consider this: Facebook users upload more than 900 million photos a day. So that 250 billion number from last year will seem like a drop in the bucket in a few months. Facebook has to handle a tsunami of photographs every day. It has to ingest it all, process it, file it, and somehow, later, be able to retrieve it.

(iv) Variability

The increase in the range of values typical of a large data set - and value, which addresses the need for valuation of enterprise data.

5. Various Challenges in Business Analytics

Ans :

- **Executive Ownership:** Business Analytics requires buy-in from senior leadership and a clear corporate strategy for integrating predictive models.
- **IT Involvement:** Technology infrastructure and tools must be able to handle the data and Business Analytics processes.
- **Available Production Data vs. Cleansed Modeling Data :** Watch for technology infrastructure that restrict available data for historical modeling, and know the difference between historical data for model development and real-time data in production.
- **Project Management Office (PMO)** : The correct project management structure must be in place in order to implement predictive models and adopt an agile approach.
- **End user Involvement and Buy-In :** End users should be involved in adopting Business Analytics and have a stake in the predictive model.
- **Change Management :** Organizations should be prepared for the changes that Business Analytics bring to current business and technology operations.
- **Explainability vs. the "Perfect Lift"**: Balance building precise statistical models with being able to explain the model and how it will produce results.

6. Digital Marketing.

Ans :

Introduction Database Marketers, Pioneers of Big Data Big Data & New School of Marketing Cross Channel Life cycle Marketing Social and Affiliate Marketing Empowering marketing with Social Intelligence Introduction Digital marketing encompasses using any sort of online media (profit or non-profit) for driving people to a website, a mobile app etc. and retaining them and interacting with them to understand what consumers really want.

Digital \marketing is easy when consumers interact with corporate' primary platform (ie. The one the corporate own) because corporate get good information about them. But corporate get very little information once people start interacting with other platforms (eg. Face book, Twitter, Google +). One of the problems that have to be dealt with in a digital existence, is that corporate do not have access to data for all consumers (ie. There is very little visibility about people when they interact in social networking sites). So corporate lose control of their ability to access the data they need, in order to make smart, timely decisions. Big data on the web will completely transform a company's ability to understand the effectiveness of its marketing and the ability to understand how its competitors are behaving. Rather than a few people making big decisions, the organizations will be able to make hundreds and thousands of smart decisions every day.

7. Predictive analytics.

Ans :

It use big data to identify past patterns to predict the future. For example, some companies are using predictive analytics for sales lead scoring. Some companies have gone one step further use predictive analytics for the entire sales process, analyzing lead source, number of communications, types of communications, social media, documents, CRM data, etc. Properly tuned predictive analytics can be used to support sales, marketing, or for other types of complex forecasts.

8. Descriptive analytics.

Ans :

Descriptive analysis (or) data mining are at the bottom of the big data value chain, but they can be valuable for uncovering patterns that offer insight. A simple example of descriptive analytics would be assessing credit risk; using past financial performance to predict a customer's likely financial performance. Descriptive analytics can be useful in the sales cycle, for example, to categorize customers by their likely product preferences and sales cycle.

9. Structured Data.

Ans :

By structured data, we mean data that can be processed, stored and retrieved in a fixed format. It refers to highly organized information that can be readily and seamlessly stored and accessed from a database by simple search engine algorithms. For instance, the employee table in a company database will be structured as the employee details, their job positions, their salaries, etc. will be present in an organized manner.

10. Unstructured Data.

Ans :

Unstructured data refers to the data that lacks any specific form or structure whatsoever. This makes it very difficult and time-consuming to process and analyze unstructured data. Email is an example of unstructured data.

Choose the Correct Answers

1. Facebook Tackles Big Data with _________ based on Hadoop. [a]
   (a) Project Prism (b) Prism
   (c) Project Data (d) Project Bid

2. All of the following accurately describe Hadoop, EXCEPT: [b]
   (a) Open Source (b) Real-time
   (c) Java-based (d) Distributed computing approach

3. What are the main components _________ of Big Data? [d]
   (a) Map Reduce (b) HDFS
   (c) YARN (d) All of these

4. _________ has the world's largest Hadoop cluster. [c]
   (a) Apple (b) Datamatics
   (c) Facebook (d) None of the mentioned

5. According to analysts, for what can traditional IT systems provide a foundation when they are integrated with Big Data technologies like Hadoop? [a]
   (a) Big Data management and data mining
   (b) Data warehousing and business intelligence
   (c) Management of Hadoop clusters
   (d) Collecting and storing unstructured data

6. What are the five V's of Big Data? [d]
   (a) Volume (b) Velocity
   (c) Variety (d) All the above

7. What are the different features of Big Data Analytics? [d]
   (a) Open Source (b) Scalability
   (c) Data Recovery (d) All the above

8. _________ Data refers to the data that lacks any specific form [b]
   (a) Structured data (b) Unstructured data
   (c) Both (d) None of the above

9. _________ is the last stage is Big data life cycle. [a]
   (a) Implementation (b) Datastorage
   (c) Data Mugging (d) Research

10. _________ analyze what other companies have done in the same situations. [d]
    (a) Implementation (b) Datastorage
    (c) Data Mugging (d) Research

Fill in the blanks

1. __________ refers to the skills, technologies, practices for continuous iterative exploration and investigation of past business performance to gain insight and drive business planning.
2. __________ tools range from spreadsheets with statistical functions to complex data mining and predictive modeling applications.
3. __________ analytics it is really valuable, but largely not used.
4. An __________ model is simply a mathematical equation that describes relationships among variables in a historical data set.
5. __________ analytics can be useful in the sales cycle,
6. __________ (or feature) roadmaps outline the development and launches of a product and its features.
7. __________ outline the hierarchy of a business or one of its departments or teams.
8. PFD stands for __________.
9. __________ can be measured by quality of transactions, events and amount of history.
10. __________ models Improve Operational Efficiencies Increase Revenues Achieve Competitive Differentiation Reduce risks and costs

Answers

1. Business analytics (BA)
2. Analytics
3. Prescriptive
4. Analytical
5. Descriptive
6. Product
7. Organizational charts
8. Process flow Diagram

9 Data Volume

10. Big data

UNIT II

DESCRIPTIVE ANALYTICS:

Over view of Description Statistics (Central Tendency, Variability), Data Visualization-Definition, Visualization Techniques - Tables, Cross Tabulations, charts, Data Dash boards using Advanced Ms-Excel or SPSS.

2.1 Overview of Description Statistics

Q1. What is Statistics?

Ans :

Meaning

Statistics is a branch of mathematics that deals with collecting, interpreting, organization and interpretation of data.

According to Prof Horace Secrist, Statistics may be defined as the aggregate of facts affected to a marked extent by multiplicity of causes, numerically expressed enumerated or estimated according to a reasonable standard of accuracy, collected in a systematic manner, for a predetermined purpose and placed in relation to each other.

Descriptive statistics employs a set of procedures that make it possible to meaningfully and accurately summarize and describe samples of data. In order for one to make meaningful statements about psychological events, the variable or variables involved must be organized, measured, and then expressed as quantities. Such measurements are often expressed as measures of central tendency and measures of variability.

Q2. Explain briefly about Descriptive Statistics.

Ans :

Meaning

Descriptive statistics is used to summarize data and make sense out of the raw data collected during the research. Since the data usually represents a sample, then the descriptive statistics is a quantitative description of the sample.

The level of measurement of the data affects the type of descriptive statistics. Nominal and ordinal type data (often termed together as categorical type data) will differ in the analysis from interval and ratio type data (often termed together as continuous type data).

Descriptive statistics for categorical data

Contingency tables (or frequency tables) are used to tabulate categorical data. A contingency table shows a matrix or table between independent variables at the top row versus a dependent variable on the left column, with the cells indicating the frequency of occurrence of possible combination of levels. (check SPSS for examples).

Descriptive statistics for continuous data

There are two the two aspects of descriptive statistics used for continuous type data. They are;

- Central tendency
- Variability of the data

2.1.1 Measures of Central Tendency

Q3. Explain briefly about Measures of Central Tendency.

Ans : **(Imp.)**

It refers to a number (statistic) that best characterizes the group as a whole" (Sommer & Sommer, 1997). It is generally referred to as the average. The three measures of central tendency, the mean, median, and mode, describe a distribution of data and are an index of the average, or typical, value of a distribution of scores

The three types of averages are:

1. Mean
2. Median
3. Mode

1. MEAN (M)

It is the arithmetic average (sum of all score divided by the number of cases) The mean, the arithmetic average of all scores under consideration, is computed by dividing the sum of the scores by the number of scores.

The sample mean of the values

2. MEDIAN

It is the midpoint of a distribution of data. Half the scores fall above and half below the median. The three measures of central tendency, the mean, median, and mode, describe a distribution of data and are an index of the average, or typical, value of a distribution of scores.

The median is the point at which 50% of the observations fall below and 50% above or, in other words, the middle number of a set of numbers arranged in ascending or descending order. (If the list includes an even number of categories, the median is the arithmetic average of the middle two numbers.) Based on the data in Table , the full list of each student's study hours would be written 10, 9, 9, 9, 8, 8, 8, 8, and so on. If the list were written out in full, it would be clear that the middle two numbers of the 40 entries are 6 and 6, which average 6. So the median of the hours studied is 6.

3. MODE

It is the single score that occurs most often in a distribution of data. The mode is the number that appears most often. Based on the data in Table , the mode of the number of hours studied is also 6 (8 students studied for 6 hours, so 6 appears 8 times in the list, more than any other number).

2.1.2 Measures of Variability

Q4. Explain the various ways of Measure of variability.

Ans : **(Imp.)**

There are many ways to describe variability including :

(i) Range

(ii) Interquartile Range (IQR)

(iii) Variance

(iv) Standard Deviation

(i) Range

Range = Maximum – Minimum

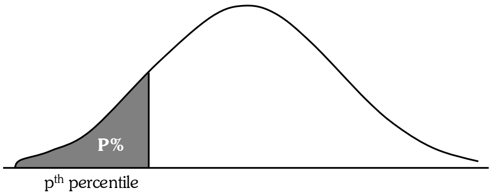
(a) Easy to calculate

(b) Very much affected by extreme values (ranges is not a resistant measure of variability).

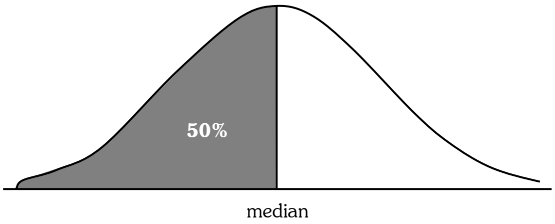
(ii) Interquartile Range (IQR)

In order to talk about interquartile range, we need to first talk about percentiles.

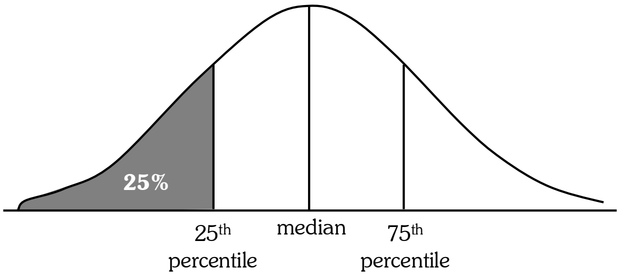
The path percentile of the data set is a measurement such that after the data are ordered from smallest to largest, at most, p% of the data are at or below this value and at most, (100 – p)% at or above it.



Thus, the median is the 50th percentile. Fifty percent or the data values fall at or below the median.



Also, Q_1 = lower quartile = the 25th percentile and Q_3 = upper quartile = the 75th percentile.



Interquartile Range

It is the difference between upper and lower quartiles and denoted as IQR.

IQR = $Q_3 - Q_1$ = upper quartile - lower quartile = 75th percentile - 25th percentile.

Details about how to compute IQR will be given in Lesson 2.3.

Note: IQR is not affected by extreme values. It is thus a resistant measure of variability.

(iii) Variance

Two vending machines A and B drop candies when a quarter is inserted. The number of pieces of candy one gets is random. The following data are recorded for six trials at each vending machine:

Pieces of candy from vending machine A:

1, 2, 3, 3, 5, 4

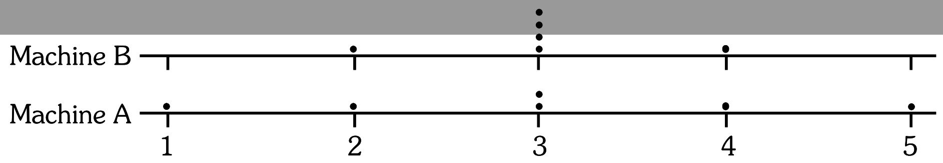
Mean = 3, Median = 3, Mode = 3

Pieces of candy from vending machine B:

2, 3, 3, 3, 3, 4

Mean = 3, Median = 3, Mode = 3

Dotplots for the pieces of candy from vending machine A and vending machine B:



They have the same center, but what about their spreads? One way to compare their spreads is to compute their standard deviations. In the following section, we are going to talk about how to compute the sample variance and the sample standard deviation for a data set.

Variance is the average squared distance from the mean.

Population variance is defined as:

$\alpha^2 = \Sigma i = 1N (yi - \mu) / 2N$

In this formula μ is the population mean and the summation is over all possible values of the population. N is the population size.

The sample variance that is computed from the sample and used to estimate α^2 is:

$s2 = \Sigma i = 1n\ (yi - \bar{y}\ )2n - 1$

Why do we divide by n – 1 instead of by n? Since μ is unknown and estimated by $\bar{y}$, the y_i's tend to be closer to $\bar{y}$ than to μ. To compensate, we divide by a smaller number, n – 1.

Sample Variance

It is the common default calculations used by software. When asked to calculate the variance or standard deviation of a set of data, assume - unless otherwise instructed - this is sample data and therefore calculating the sample variance and sample standard deviation.

Examples:

Let's find S2 for the data set from vending machine A: 1, 2, 3, 3, 4, 5

$\bar{y} = 1 + 2 + 3 + 3 + 4 + 56 = 3$

$s_2 = (y1 - \bar{y}\ )2 + +(yn - \bar{y}\ )\ 2n - 1$

$= (1 - 3)2 + (2 - 3)2 + (3 - 3)2 +(3 - 3)2 + (4 - 3)2 + (5 - 3)\ 26$

$-1 = 2$

Calculate S^2 for the data set from vending machine B yourself and check that it is smaller than the S^2 for data set A. Work out your answer first, then click the graphic to compare answers.

(iv) Standard Deviation

The population standard deviation is notated by σ and found by $\sigma = \sigma^2 - \sqrt{}$ has the same unit as y_i's. This is a desirable property since one may think about the spread in terms of the original unit.

σ is estimated by the sample standard deviation *s* :

$s = s2 - \sqrt{}$

For the data set *A*,

s = 2 – Š = 1.414 pieces of candy.

2.2 Data Visualization

Q5. What is Data visualization? Explain the importance of Data Visualization.

Ans : **(Imp.)**

It is a general term that describes any effort to help people understand the significance of data by placing it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data visualization software.

- Today's data visualization tools go beyond the standard charts and graphs used in Microsoft Excel spreadsheets, displaying data in more sophisticated ways such as info graphics, dials and gauges, geographic maps, spark lines, heat maps, and detailed bar, pie and fever charts. The images may include interactive capabilities, enabling users to manipulate them or drill into the data for querying and analysis. Indicators designed to alert users when data has been updated or predefined conditions occur can also be included.

Importance

- Data visualization has become the de facto standard for modern business intelligence (BI). The success of the two leading vendors in the BI space, Tableau and Qlike - both of which heavily emphasize visualization - has moved other vendors toward a more visual approach in their software. Virtually all BI software has strong data visualization functionality.
- Data visualization tools have been important in democratizing data and analytics and making data-driven insights available to workers throughout an organization. They are typically easier to operate than traditional statistical analysis software or earlier versions of BI software. This has led to a rise in lines of business implementing data visualization tools on their own, without support from IT.
- Data visualization software also plays an important role in big data and advanced analytics projects. As businesses accumulated massive troves of data during the early years of the big data trend, they needed a way to quickly and easily get an overview of their data. Visualization tools were a natural fit.
- Visualization is central to advanced analytics for similar reasons. When a data scientist is writing advanced predictive analytics or machine learning algorithms, it becomes important to visualize the outputs to monitor results and ensure that models are performing as intended. This is because visualizations of complex algorithms are generally easier to interpret than numerical outputs.

Examples:

Data visualization tools can be used in a variety of ways. The most common use today is as a BI reporting tool. Users can set up visualization tools to generate automatic dash boards that track company performance across key performance indicators and visually interpret the results.

Many business departments implement data visualization software to track their own initiatives. For example, a marketing team might implement the software to monitor the performance of an email campaign, tracking metrics like open rate, click-through rate and conversion rate.

Q6. How data visualization works?

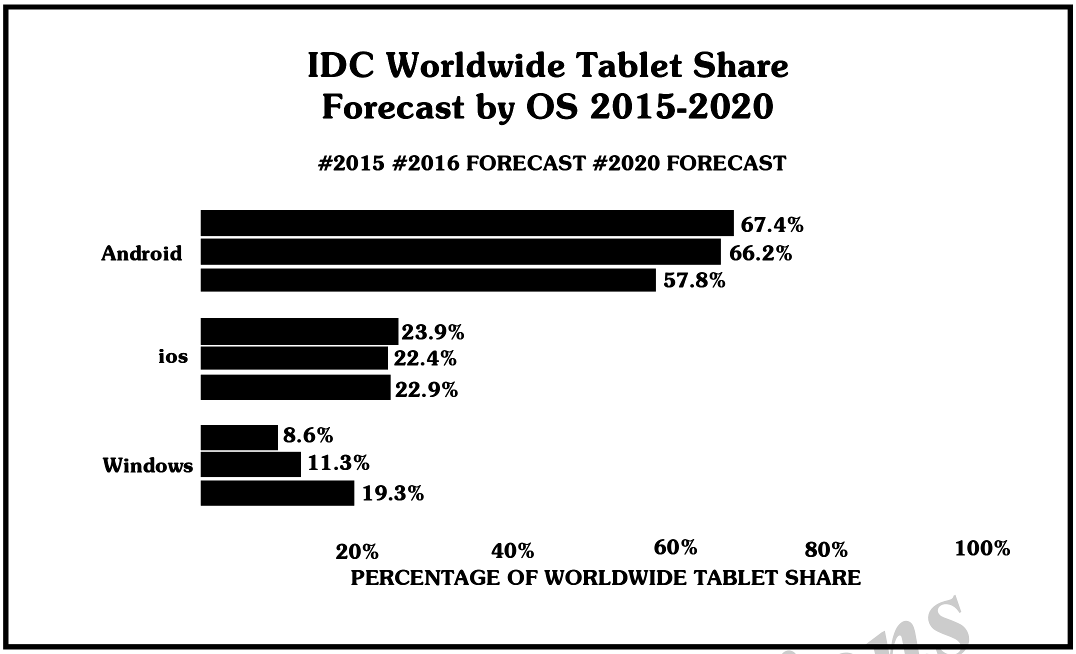
Ans :

Most of today's data visualization tools come with connectors to popular data sources, including the most common relational databases, Hadoop and a variety of cloud storage platforms. The visualization software pulls in data from these sources and applies a graphic type to the data.

Data visualization software allows the user to select the best way of presenting the data, but, increasingly, software automates this step. Some tools automatically interpret the shape of the data and detect correlations between certain variables and then place these discoveries into the chart type that the software determines is optimal.

Typically, data visualization software has a dashboard component that allows users to pull multiple visualizations of analyses into a single interface, generally a web portal.

Below is a chart forecasting tablet sales by operating system.



Q7. Explain the uses of data visualization.

Ans :

- By using data visualization, it became easier for business owners to understand their large data in a simple format.
- The visualization method is also time saving. So, businesses does not have to spend much time to make a report or solve a query. They can easily do it in a less time and in a more appealing way.
- Visual analytics offers a story to the viewers. By using charts and graphs or images, a person can easily exposure the whole concept as well the viewers will be able to understand the whole thing in an easy way.
- The most complicated data will look easy when it gets through the process of visualization. Complicated data report gets converted into a simple format. And it helps people to understand the concept in an easy way.
- With the visualization process, it gets easier to the business owners to understand their product growth and market competition in a better way.

2.3 Data Visualization Techniques

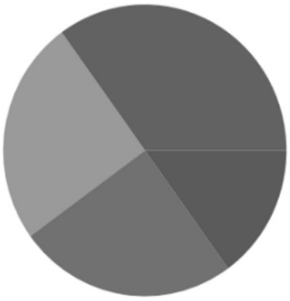
Q8. What are the techniques of Data Visualization?

Ans : **(Imp.)**

The data visualization techniques are Diagrams, charts, graphs.

Most widely used forms of data visualization are presented below:

1. Pie Chart

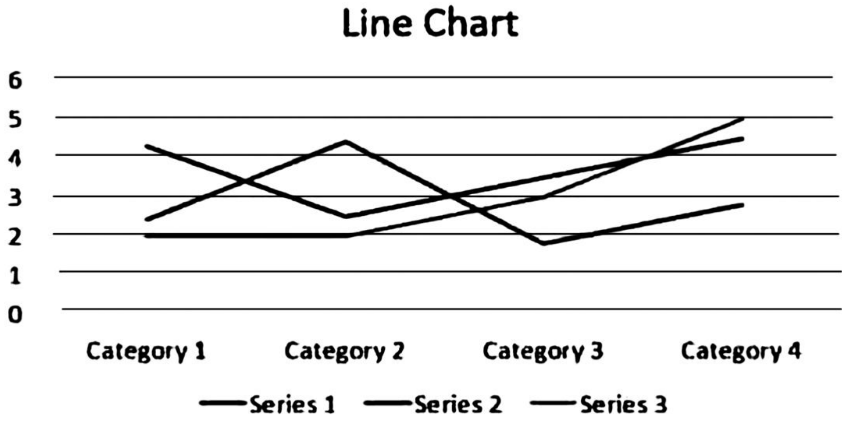


Pie Chart

Category 1 Category 2 Category 3 Category 4

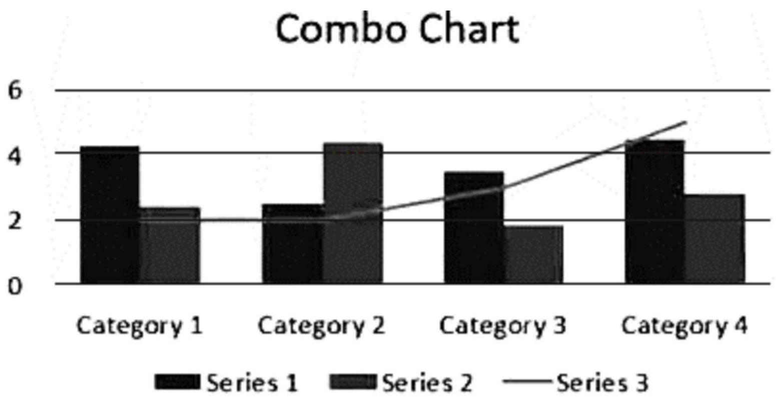
Pie Charts : Pie Charts are one of the common popular techniques. It also comes under data visualization techniques in excel. However, to some people, it can be hard to understand the chart while comparing to the line and bar type chart.

2. Line Chart



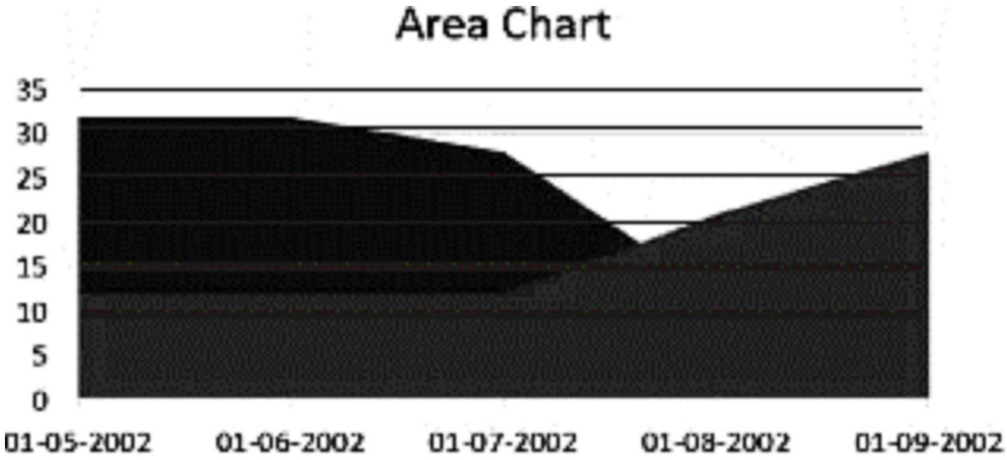
To make your data simple and more appealing you can simply use the line charts technique. Line chart basically displays the relationship between two patterns. Also, it is one of the most used techniques world wide.

3. Combo Chart



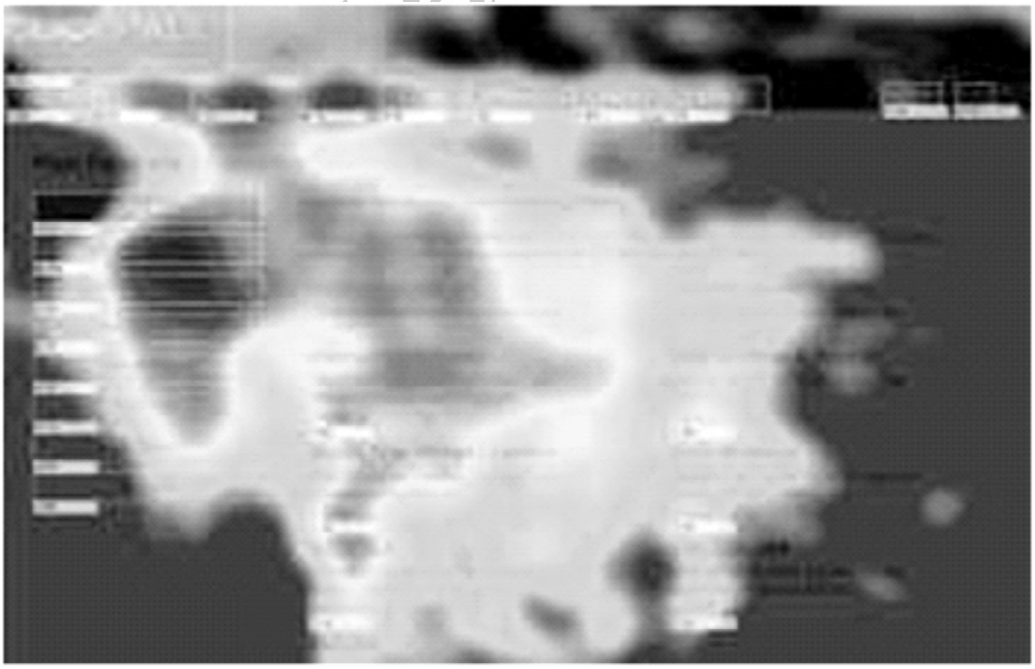
Bars charts are also one of the most commonly used techniques when it comes to comparing two different patterns. The bar charts can display the data in a horizontal way or in a vertical way. It all depends on your needs.

4. Area Chart



An area chart or area graph is similar to a line chart but provides graphically quantitative data. The areas can be filled with colour, hatch, pattern. This chart is generally used when comparing quantities which is depicted by area.

5. Heat Map

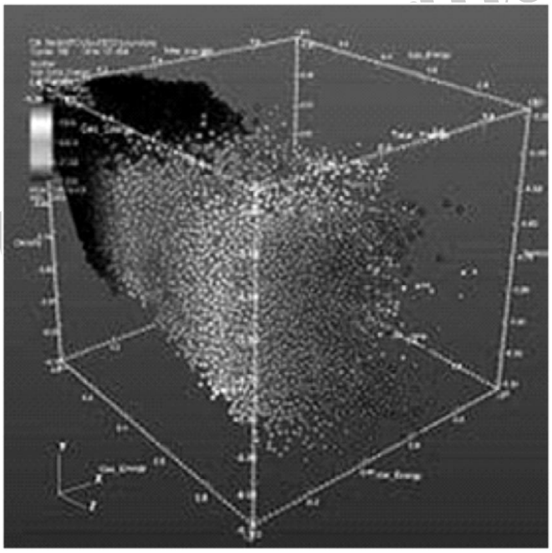


This type of chart is widely used by websites, mobile application makers, research institutes etc. These maps shows the concentration of activity/entity over a particular area.

6. Network Diagram

This is a powerful tool for finding out connections & correlations. It highlights and bridges the gaps. Shows strongly one activity is connected to other.

7. Scattered 3 D Plot



As the image shows it shows the distribution of entity in a 3 dimensional nature. It can be considered as showing location and concentration of gases in a box with different colours assigned to each gas.

2.3.1 Tables

Q9. How "data visualization technique –Tables" can display data analysis reports using Ms.Excel?

Ans : **(Imp.)**

Data analysis reports using Ms.Excel can display in a number of ways. However, if the data analysis results can be visualized as charts that highlight the notable points in the data, the audience can quickly grasp what they want to project in the data. It also leaves a good impact on the presentation style.

Here you will get to know how to use Excel charts and Excel formatting features on charts that enable you to present your data analysis results with emphasis.

Visualizing Data with Charts

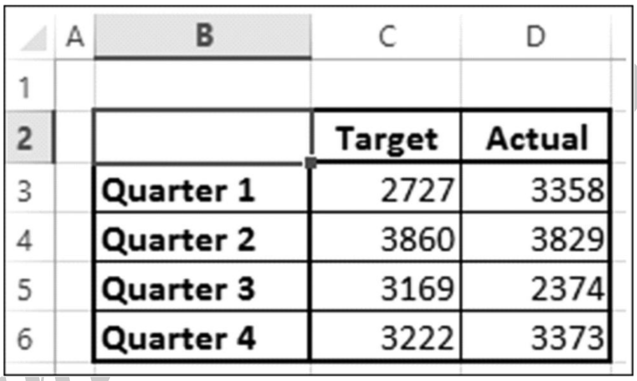
In Excel, charts are used to make a graphical representation of any set of data. A chart is a visual representation of the data, in which the data is represented by symbols such as bars in a Bar Chart or lines in a Line Chart. Excel provides you with many chart types and you can choose one that suits your data or you can use the Excel Recommended Charts option to view charts customized to your data and select one of those.

Refer to the Tutorial Excel Charts for more information on chart types.

In this chapter, you will understand the different techniques that you can use with the Excel charts to highlight your data analysis results more effectively.

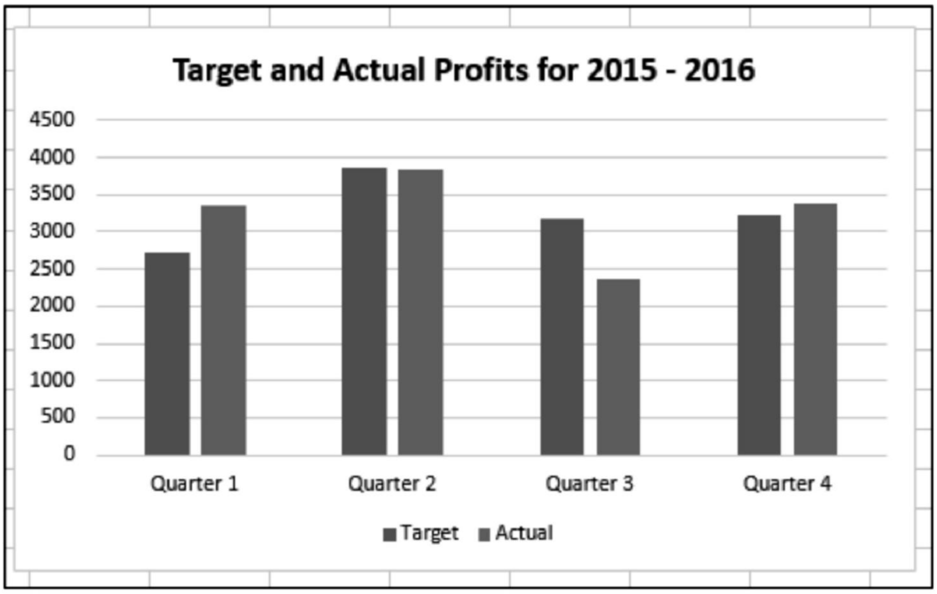
Creating Combination Charts

Suppose you have the target and actual profits for the fiscal year 2015-2016 that you obtained from different regions.



	A	B	C	D
1				
2			**Target**	**Actual**
3		**Quarter 1**	2727	3358
4		**Quarter 2**	3860	3829
5		**Quarter 3**	3169	2374
6		**Quarter 4**	3222	3373

We will create a Clustered Column Chart for these results.

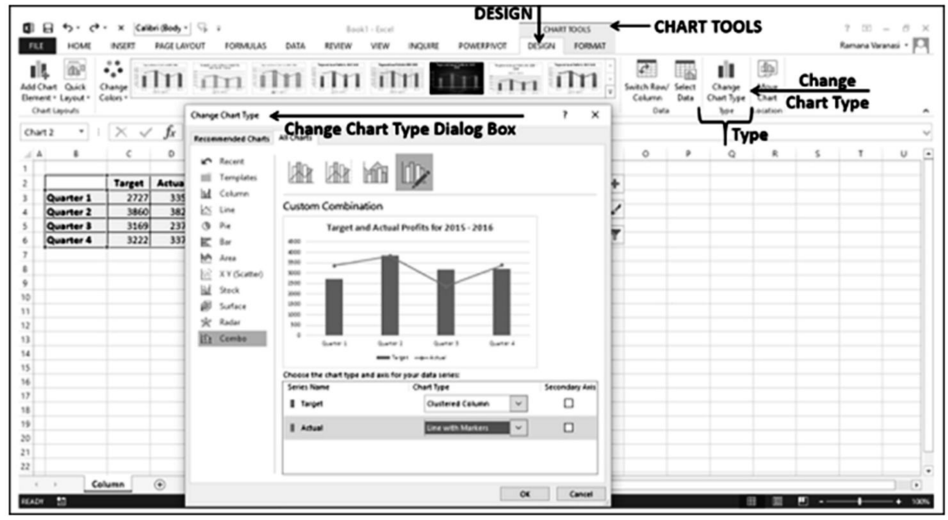


As you observe, it is difficult to visualize the comparison quickly between the targets and actual in this chart. It does not give a true impact on your results.

A better way of distinguishing two types of data to compare the values is by using Combination Charts. In Excel 2013 and versions above, you can use Combo charts for the same purpose.

Use Vertical Columns for the target values and a Line with Markers for the actual values.

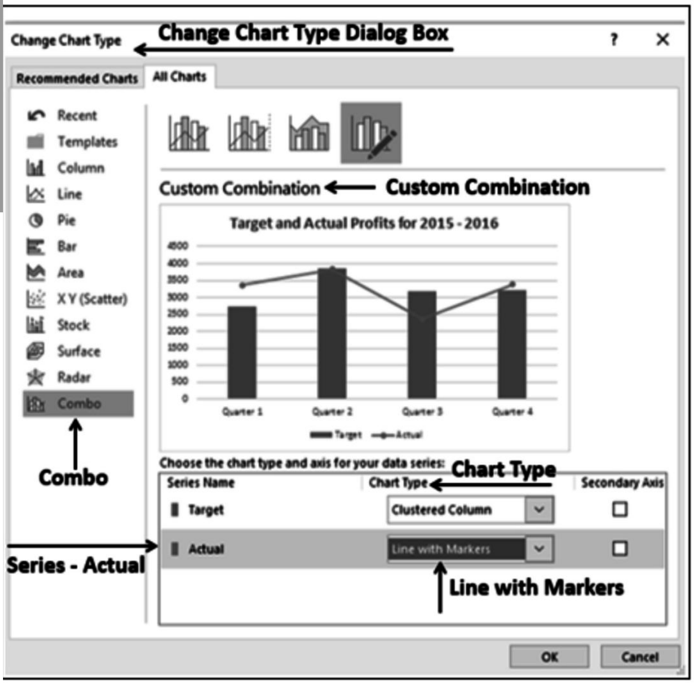
- Click the DESIGN tab under the CHART TOOLS tab on the Ribbon.
- Click Change Chart Type in the Type group. The Change Chart Type dialog box appears.


DESIGN
CHART TOOLS
Change Chart Type
Type
Change Chart Type Dialog Box
Custom Combination
Target and Actual Profits for 2015 - 2016

	Target	Actual
Quarter 1	2727	335
Quarter 2	3860	382
Quarter 3	3169	237
Quarter 4	3222	337

Series Name	Chart Type	Secondary Axis
Target	Clustered Column	□
Actual	Line with Markers	□

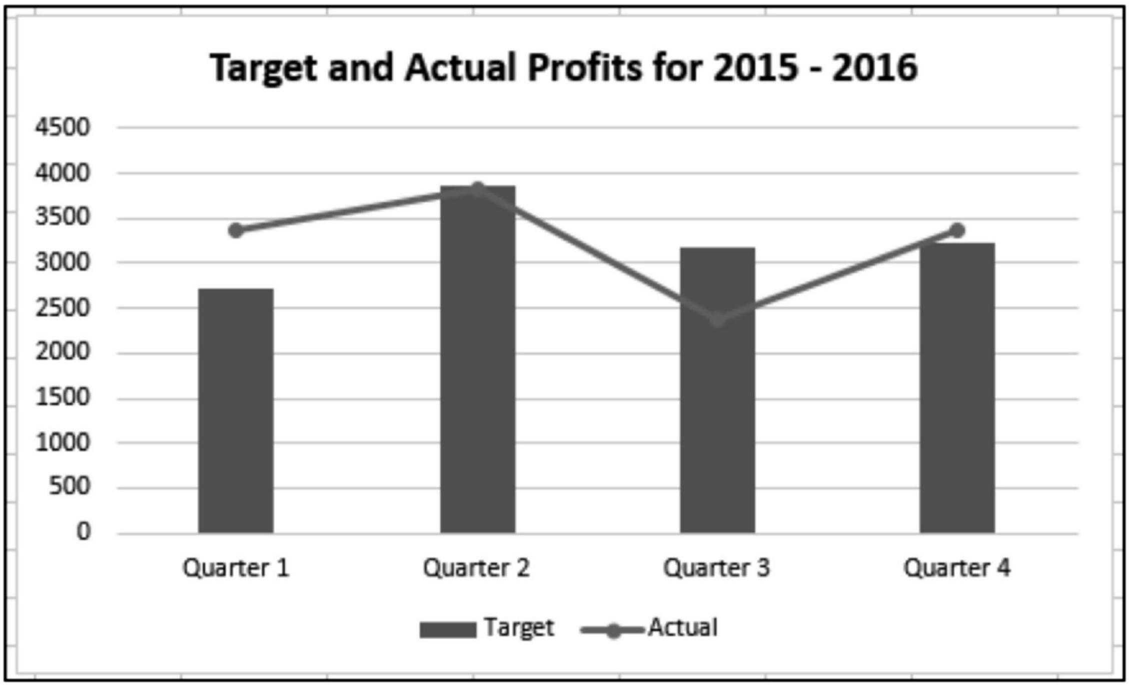
- Click Combo.
- Change the Chart Type for the series Actual to Line with Markers. The preview appears under Custom Combination.
- Click OK.



Choose the chart type and axis for your data series:

Series Name	Chart Type	Secondary Axis
Target	Clustered Column	□
Actual	Line with Markers	□

Your Customized Combination Chart will be displayed.

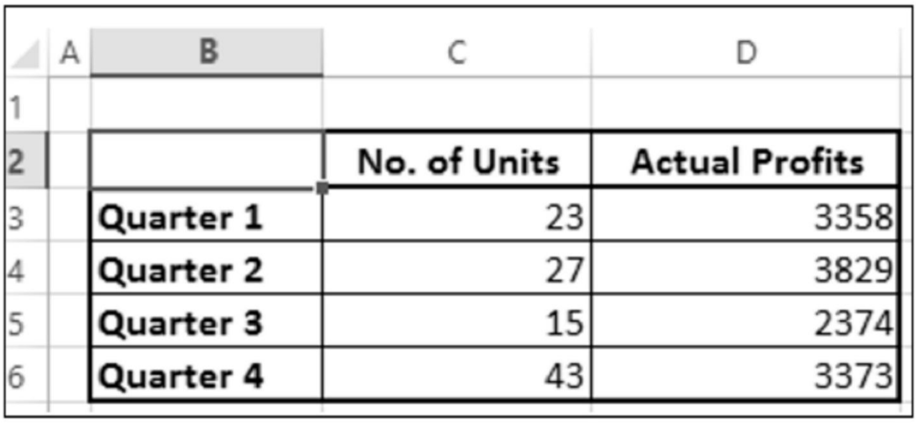


As you observe in the chart, the Target values are in Columns and the Actual values are marked along the line. The data visualization has become better as it also shows you the trend of your results.

However, this type of representation does not work well when the data ranges of your two data values vary significantly.

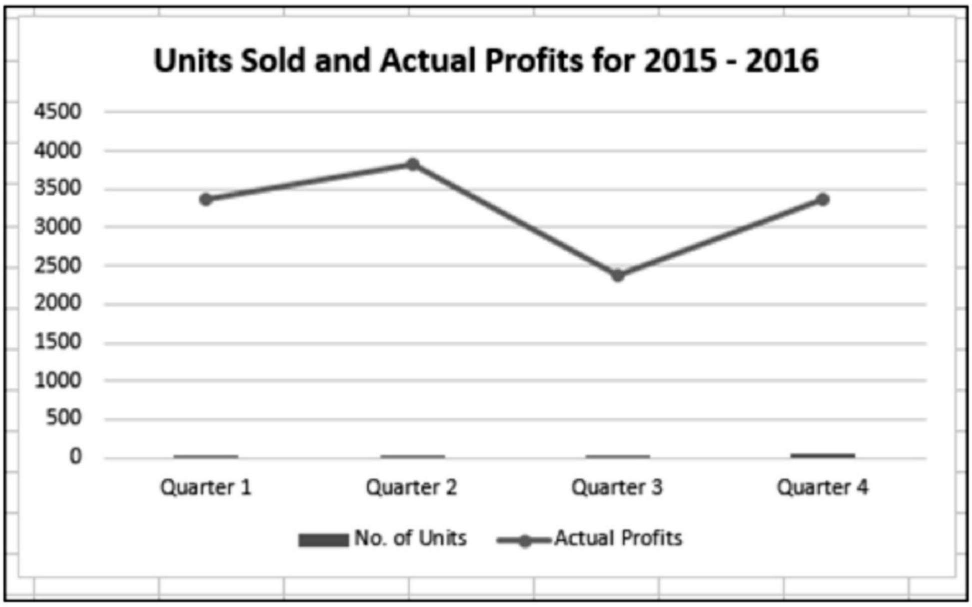
Creating a Combo Chart with Secondary Axis

Suppose you have the data on the number of units of your product that was shipped and the actual profits for the fiscal year 2015-2016 that you obtained from different regions.



	No. of Units	Actual Profits
Quarter 1	23	3358
Quarter 2	27	3829
Quarter 3	15	2374
Quarter 4	43	3373

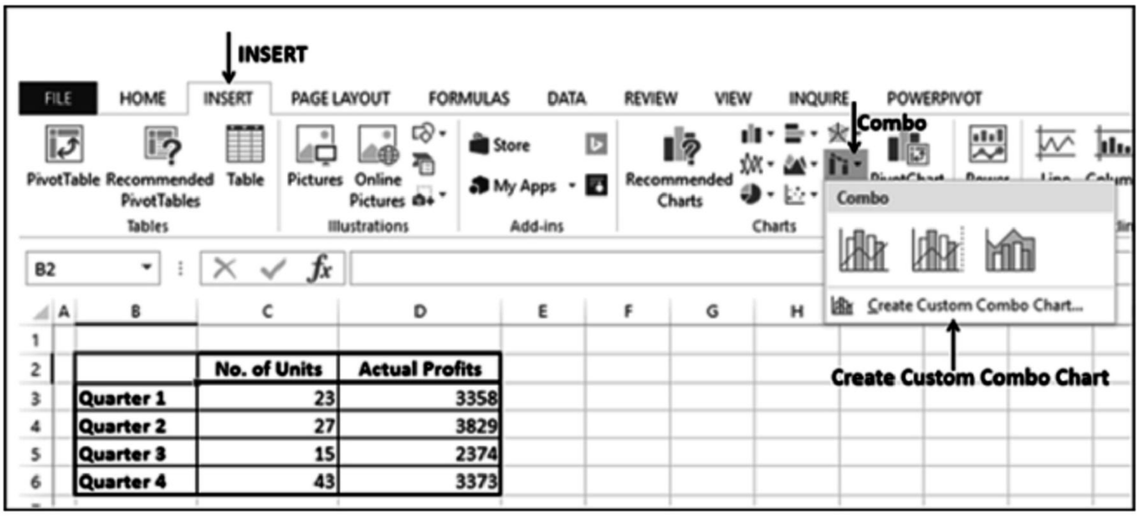
If you use the same combination chart as before, you will get the following:



In the chart, the data of **No. of Units** is not visible as the data ranges are varying significantly.

In such cases, you can create a combination chart with secondary axis, so that the primary axis displays one range and the secondary axis displays the other.

- ➢ Click the INSERT tab.
- ➢ Click Combo in Charts group.
- ➢ Click Create Custom Combo Chart from the drop-down list.



	No. of Units	Actual Profits
Quarter 1	23	3358
Quarter 2	27	3829
Quarter 3	15	2374
Quarter 4	43	3373

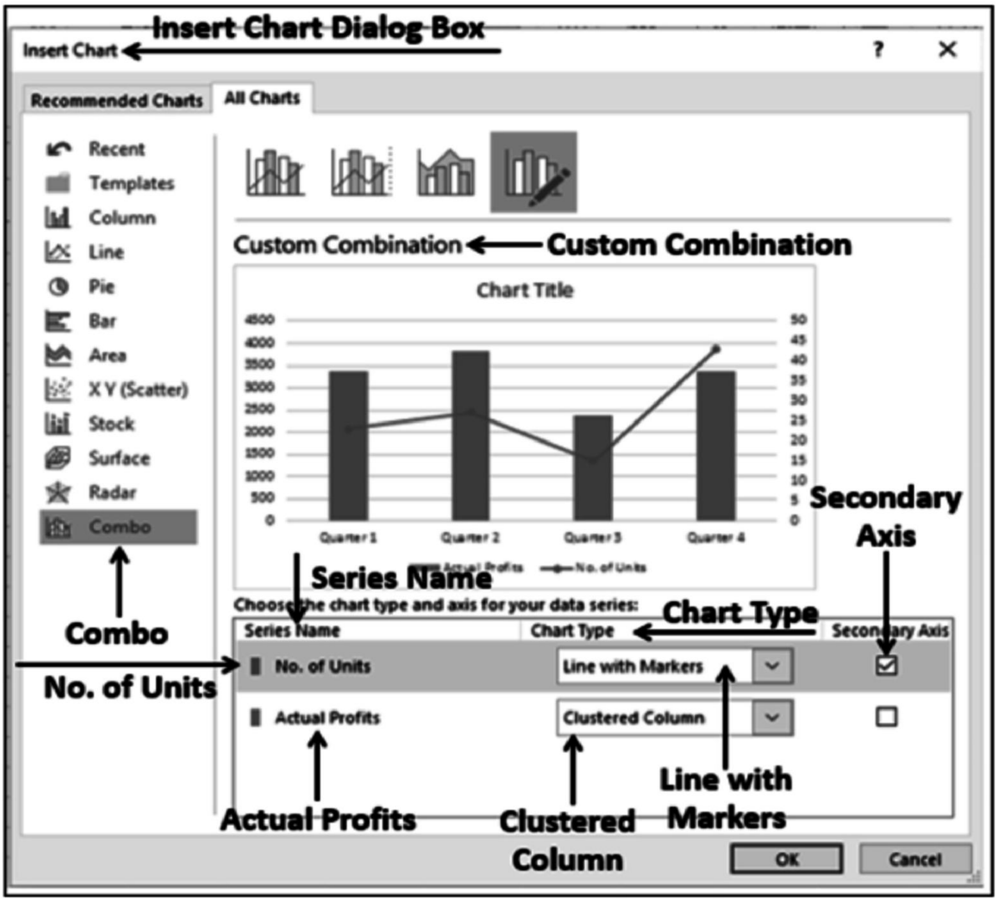
The Insert Chart dialog box appears with Combo highlighted.

For Chart Type, choose

- ➢ Line with Markers for the Series No. of Units
- ➢ Clustered Column for the Series Actual Profits

- Check the Box Secondary Axis to the right of the Series No. of Units and click OK.

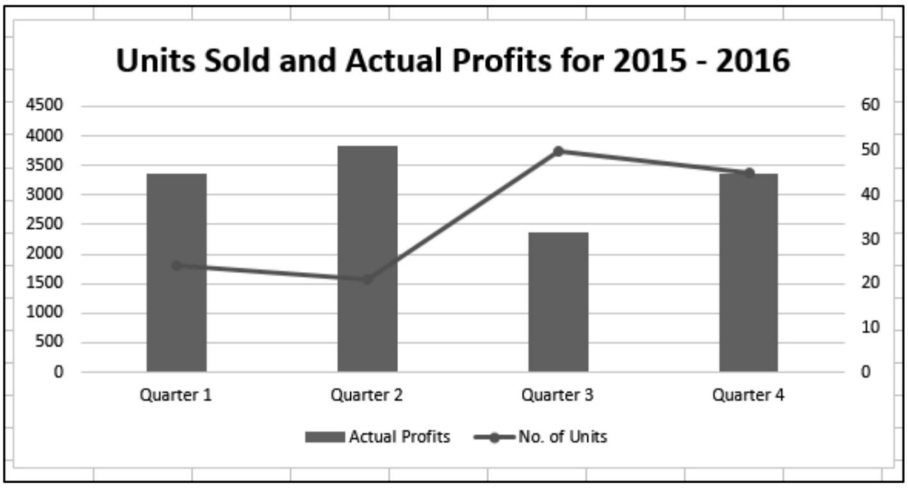
A preview of your chart appears under Custom Combination.



Choose the chart type and axis for your data series:

Series Name	Chart Type	Secondary Axis
No. of Units	Line with Markers	☑
Actual Profits	Clustered Column	□

Your Combo chart appears with Secondary Axis.

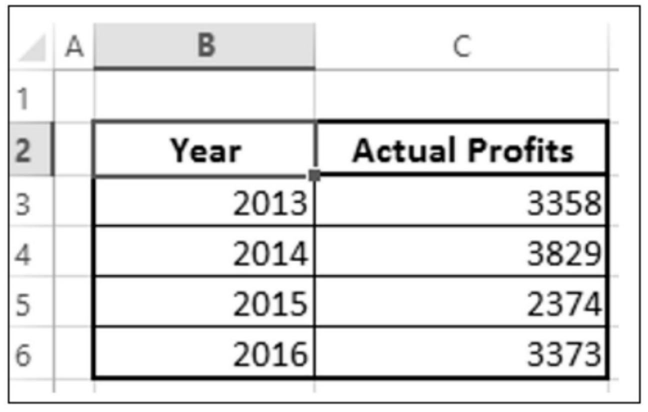


You can observe the values for Actual Profits on the primary axis and the values for No. of Units on the secondary axis.

A significant observation in the above chart is for Quarter 3 where No. of Units sold is more, but the Actual Profits made are less. This could probably be assigned to the promotion costs that were incurred to increase sales. The situation is improved in Quarter 4, with a slight decrease in sales and a significant rise in the Actual Profits made.

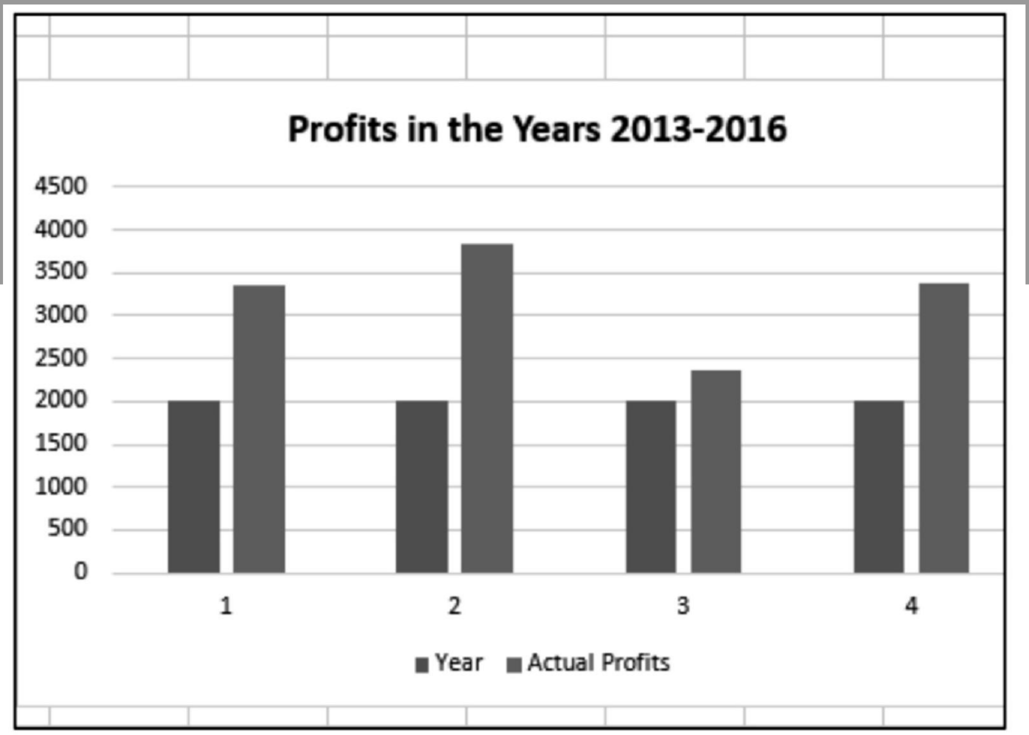
Discriminating Series and Category Axis

Suppose you want to project the Actual Profits made in Years 2013-2016.



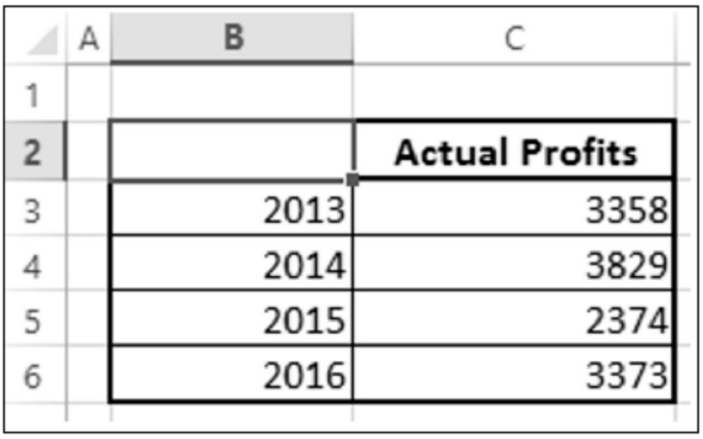
	A	B	C
1			
2		Year	Actual Profits
3		2013	3358
4		2014	3829
5		2015	2374
6		2016	3373

Create a clustered column for this data.



As you observe, the data visualization is not effective as the years are not displayed. You can overcome this by changing year to category.

Remove the header year in the data range.



	A	B	C
1			
2			Actual Profits
3		2013	3358
4		2014	3829
5		2015	2374
6		2016	3373

Now, year is considered as a category and not a series. Your chart looks as follows -

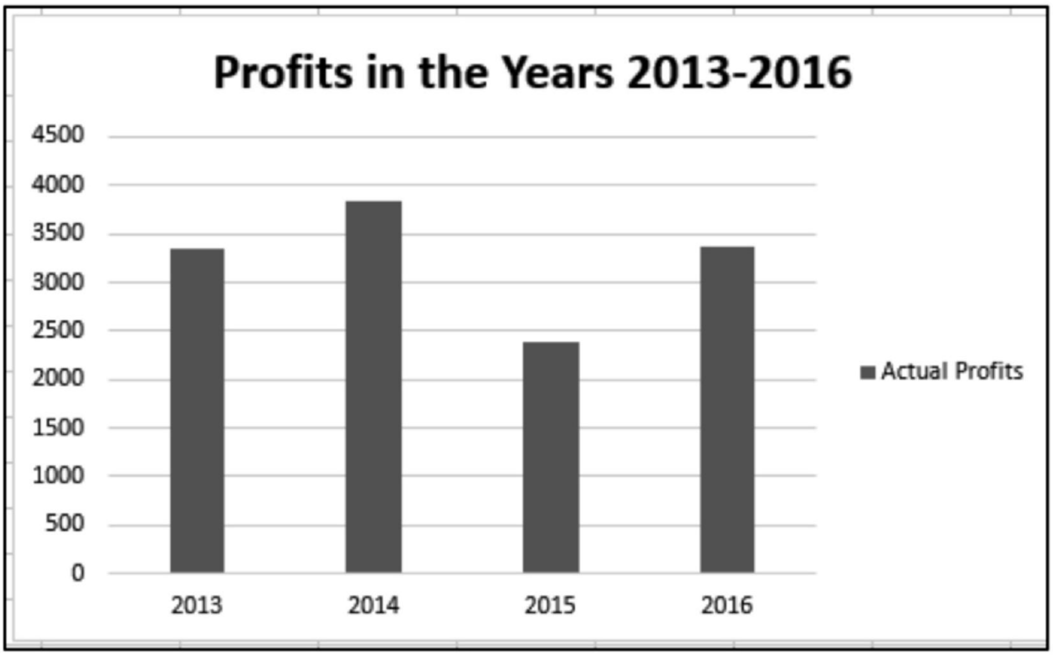


Chart Elements and Chart Styles

Chart Elements give more descriptions to your charts, thus helping visualizing your data more meaningfully.

- Click the Chart

Three buttons appear next to the upper-right corner of the chart “

- ✚ Chart Elements
- Chart Styles
- Chart Filters

For a detailed explanation of these, refer to Excel Charts tutorial.

- Click Chart Elements.
- Click Data Labels.

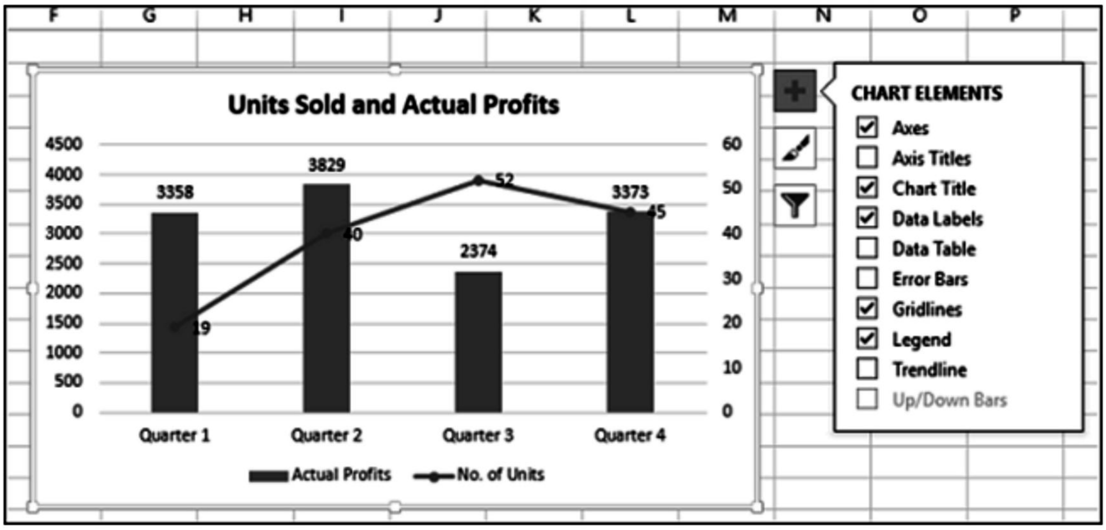
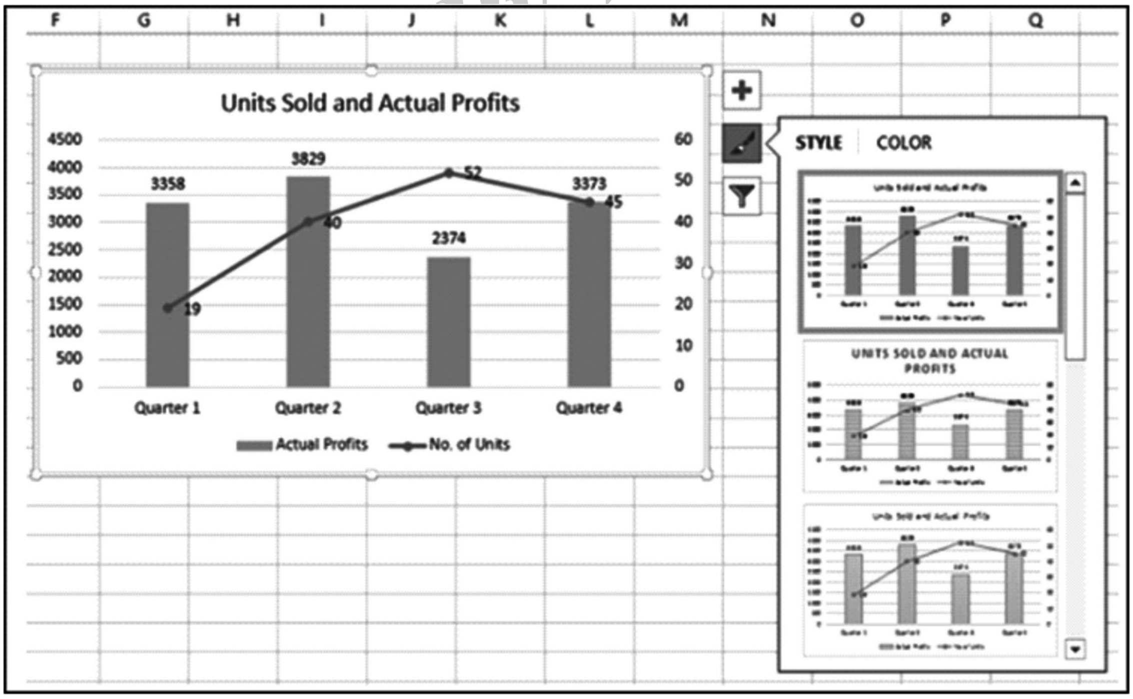


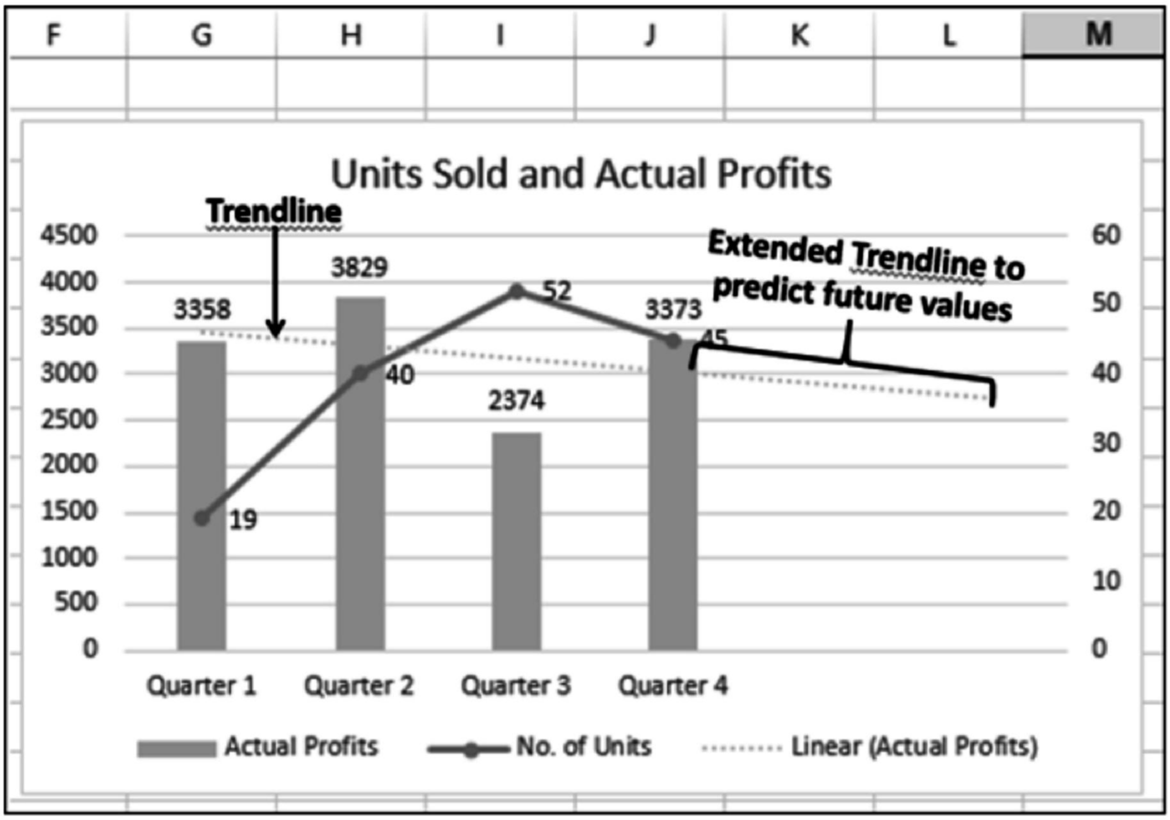
CHART ELEMENTS

☑ Axes
☐ Axis Titles
☑ Chart Title
☑ Data Labels
☐ Data Table
☐ Error Bars
☑ Gridlines
☑ Legend
☐ Trendline
☐ Up/Down Bars

- Click Chart Styles
- Select a Style and Color that suits your data.



You can use Trendline to graphically display trends in data. You can extend a Trendline in a chart beyond the actual data to predict future values.



2.3.2 Cross Tabulations

Q10. Explain briefly about "Cross tabulations charts" by using Ms. Excel?

Ans : **(Imp.)**

Cross tabulation is usually performed on categorical data that can be divided into mutually exclusive groups.

An example of categorical data is the region of sales for a product. Typically, region can be divided into categories such as geographic area (North, South, Northeast, West, etc) or state (Andhra Pradesh, Rajasthan, Bihar, etc). The important thing to remember about categorical data is that a categorical data point cannot belong to more than one category.

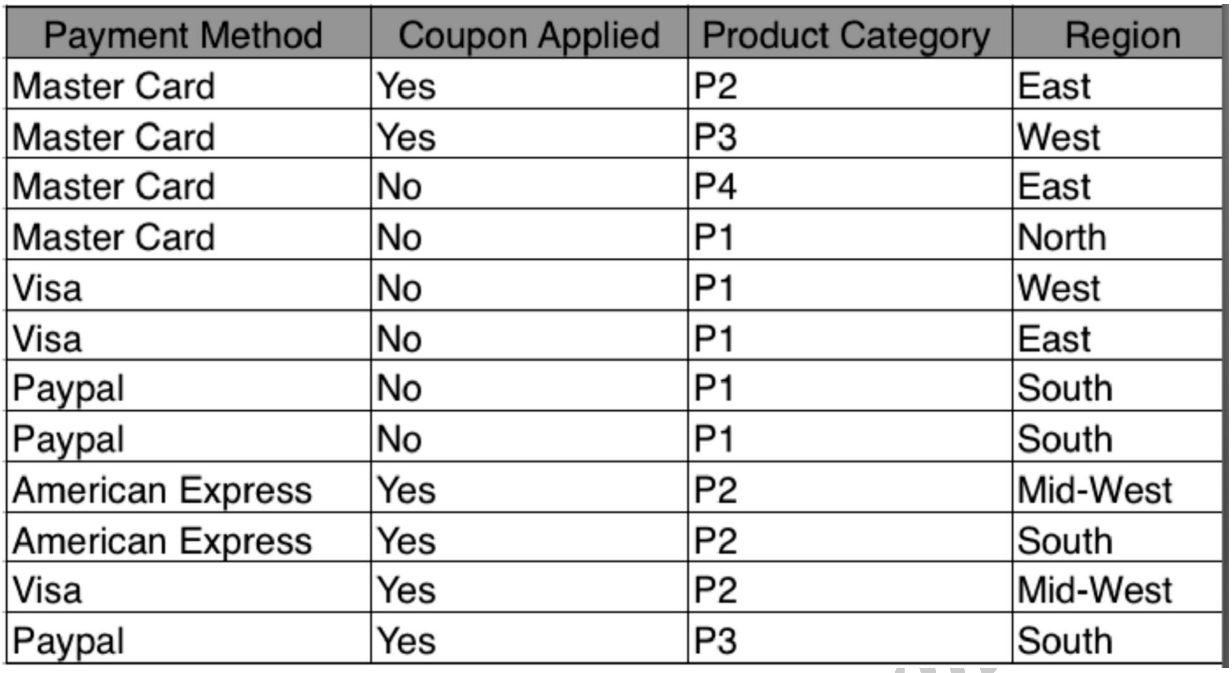
Cross tabulations are used to examine relationships within data that may not be readily apparent. Cross tabulation is especially useful for studying market research or survey responses. Cross tabulation of categorical data can be done with through tools such as SPSS, SAS, and Microsoft Excel.

An example of cross tabulation

"No other tool in Excel gives you the flexibility and analytical power of a pivot table." –Bill Jalen

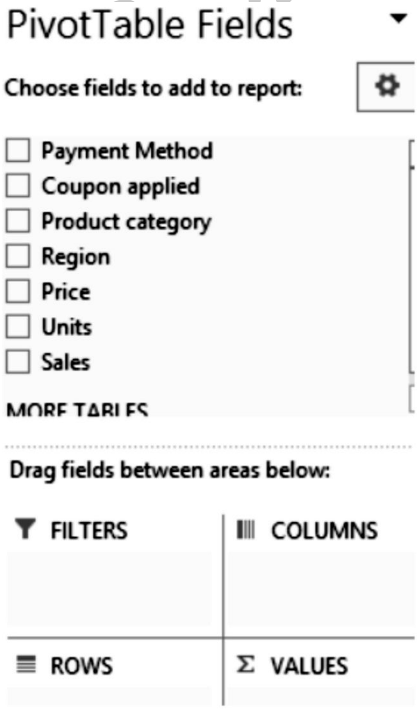
One simple way to do cross tabulations is Microsoft Excel's pivot table feature. Pivot tables are a great way to search for patterns as they help in easily grouping raw data.

Consider the below sample data set in Excel. It displays details about commercial transactions for four product categories. Let's use this data set to show cross tabulation in action.



Payment Method	Coupon Applied	Product Category	Region
Master Card	Yes	P2	East
Master Card	Yes	P3	West
Master Card	No	P4	East
Master Card	No	P1	North
Visa	No	P1	West
Visa	No	P1	East
Paypal	No	P1	South
Paypal	No	P1	South
American Express	Yes	P2	Mid-West
American Express	Yes	P2	South
Visa	Yes	P2	Mid-West
Paypal	Yes	P3	South

This data can be converted to pivot table format by selecting the entire table and inserting a pivot table in the Excel file. The table can correlate different variables row-wise, column-wise, or value-wise in either table format or chart format.



PivotTable Fields

Choose fields to add to report:

□ Payment Method
□ Coupon applied
□ Product category
□ Region
□ Price
□ Units
□ Sales

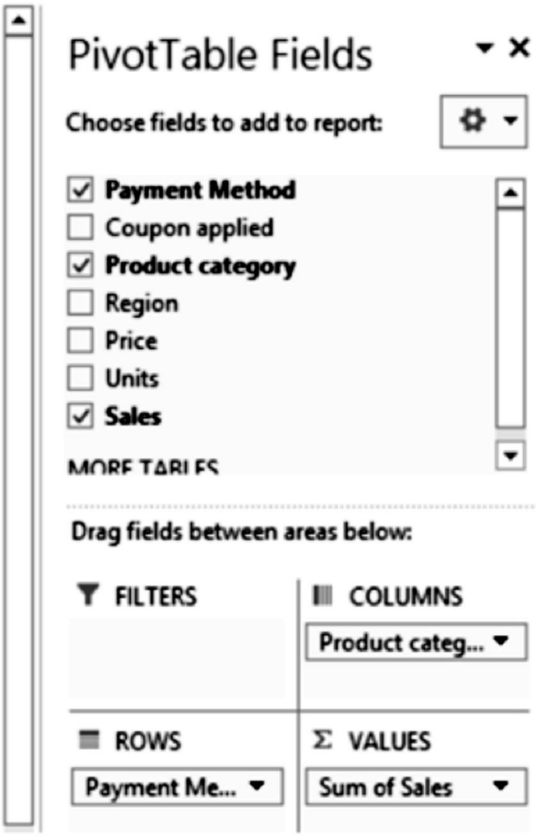
MORE TABLES

Drag fields between areas below:

FILTERS COLUMNS

ROWS Σ VALUES

Let's use cross tabulation to check the relation between the type of payment method (i.e. visa, MasterCard, PayPal, etc) and the product category with respect to the region of sales. We can select these three categories in the pivot table.



PivotTable Fields

Choose fields to add to report:

☑ **Payment Method**
□ Coupon applied
☑ **Product category**
□ Region
□ Price
□ Units
☑ **Sales**

MORE TABLES

Drag fields between areas below:

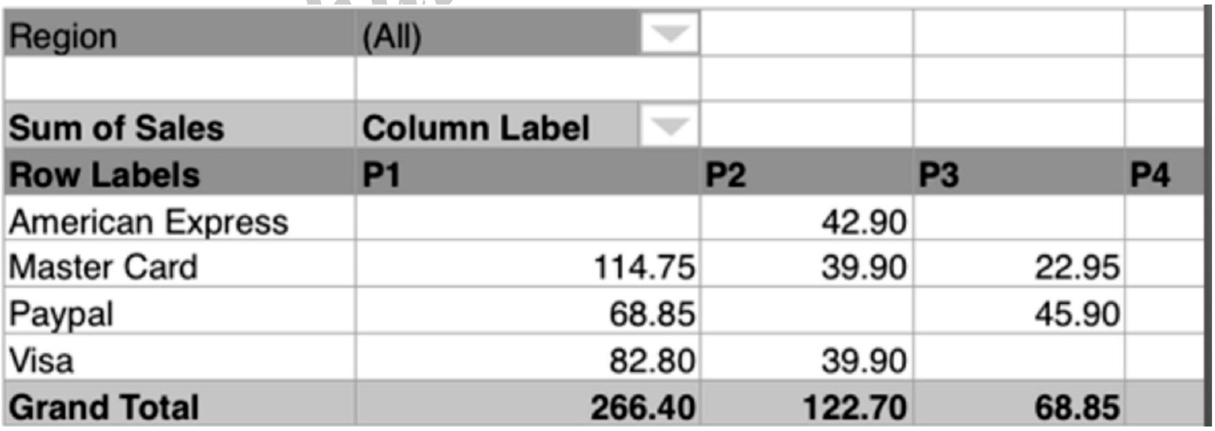
FILTERS

COLUMNS: Product categ...

ROWS: Payment Me...

VALUES: Sum of Sales

Then the results appear in a pivot table:



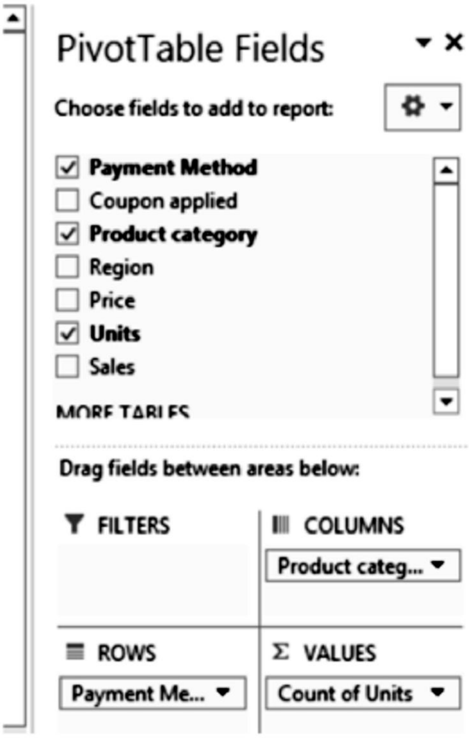
Region	(All)			
Sum of Sales	**Column Label**			
Row Labels	**P1**	**P2**	**P3**	**P4**
American Express		42.90		
Master Card	114.75	39.90	22.95	
Paypal	68.85		45.90	
Visa	82.80	39.90		
Grand Total	**266.40**	**122.70**	**68.85**	

Cross tabulation 1: Relation between payment method and the total amount of sales in product category with respect to region in which products sold

It is now clear that the highest sales were done for P1 using Master Card. Therefore, we can conclude that the MasterCard payment method and product P1 category is the most profitable combination.

Similarly, we can use cross tabulation and find the relation between the product category and the payment method type with regard to the number of transactions.

This can be done by grouping the payment method, product category, and units sold:



PivotTable Fields

Choose fields to add to report:

☑ Payment Method
□ Coupon applied
☑ Product category
□ Region
□ Price
☑ Units
□ Sales

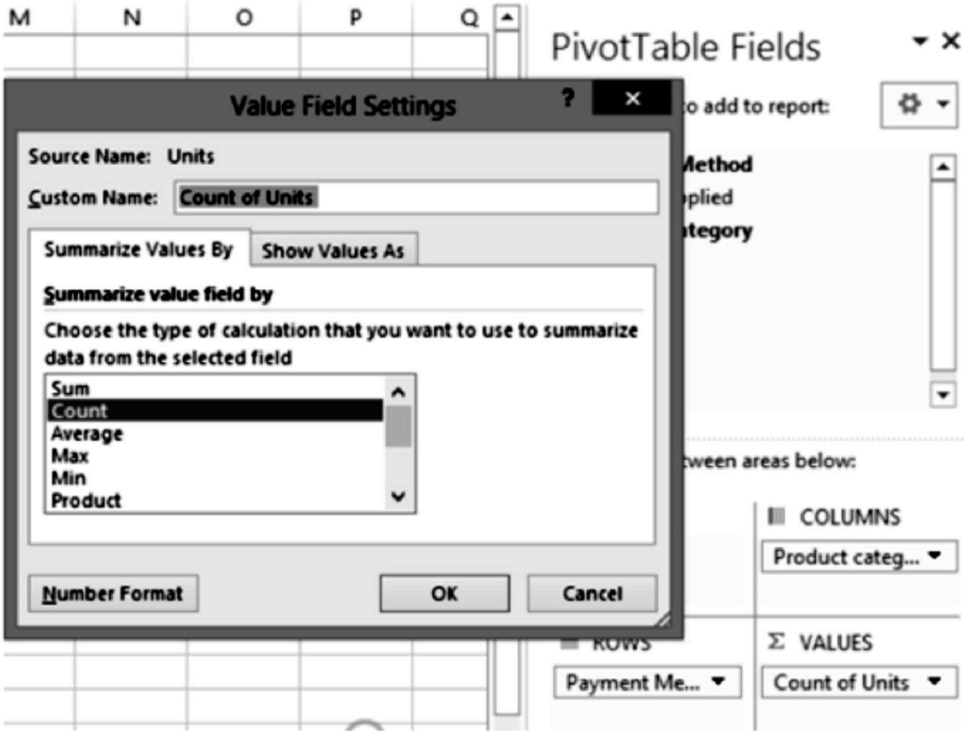
MORE TABLES

Drag fields between areas below:

FILTERS	COLUMNS
	Product categ...

ROWS	VALUES
Payment Me...	Count of Units

By default, Excel's pivot table aggregates values as a sum. Summing the units will give us the total number of units sold. Since we want to compare the number of transactions instead of the number of units sold, we need to change the Value Field Setting from Sum to Count for Units.



Value Field Settings

Source Name: Units

Custom Name: Count of Units

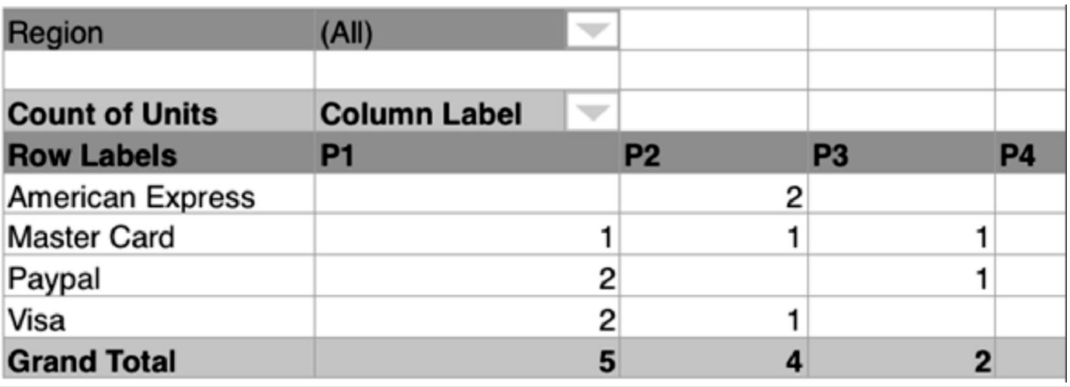
Summarize Values By | Show Values As

Summarize value field by

Choose the type of calculation that you want to use to summarize data from the selected field

- Sum
- Count
- Average
- Max
- Min
- Product

The results of this pivot table mapping is as shown below. This is a cross tabulation analysis of 3 variables - it analyses the correlation between the payment method and payment category according to the number of transactions.



Region	(All) ▾			
Count of Units	**Column Label** ▾			
Row Labels	**P1**	**P2**	**P3**	**P4**
American Express		2		
Master Card	1	1	1	
Paypal	2		1	
Visa	2	1		
Grand Total	**5**	**4**	**2**	

Cross tabulation 2: Relation between payment method and total number of transactions in the product category with respect to region of sales

For all regions, we can observe that the highest selling category of products was P1 and the highest number of transactions was done using Master Card. We can also see the preferred payment method in each of the product categories. For example, American Express is the preferred card for P2 products.

Q11. Explain the benefits of cross tabulation.

Ans :

(i) Eliminates confusion while interpreting data

Raw data can be difficult to interpret. Even for small data sets, it is all too easy to derive wrong results by just looking at the data. Cross tabulation offers a simple method of grouping variables, which minimizes the potential for confusion or error by providing clear results.

(ii) Helps in deriving innumerable insights

As we observed in our example, cross tabulation can help us derive great insights from raw data. These insights are not easy to see when the raw data is formatted as a table. Since cross tabulation clearly maps out relations between categorical variables, researchers can gain better and deeper insights — insights that otherwise would have been overlooked or would have taken a lot of time to decode from more complicated forms of statistical analysis.

(iii) Offers data points to chart out a course of action

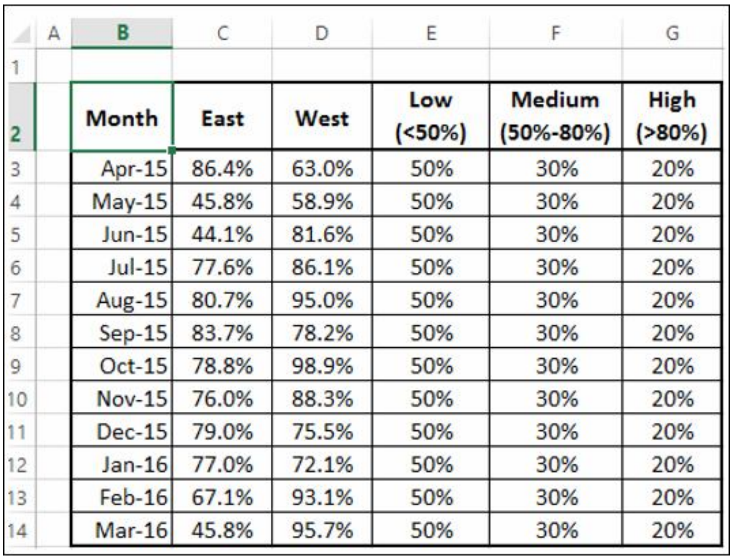
Cross tabulation makes it easier to interpret data, which is beneficial for researchers who have limited knowledge of statistical analysis. With cross tabulation, people do not need statistical programming to correlate categorical variables. The clarity offered by cross tabulation helps professionals evaluate their current work and chart out future strategies.

Q12. Explain briefly about band chart.

Ans :

You might have to present customer survey results of a product from different regions. Band Chart is suitable for this purpose. A Band Chart is a Line Chart with an added shaded area to display the upper and lower boundaries of groups of data.

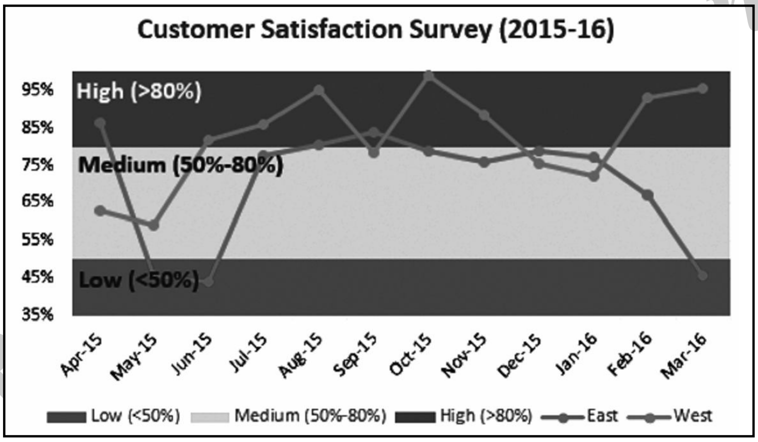
Suppose your customer survey results from the east and west regions, month-wise are:



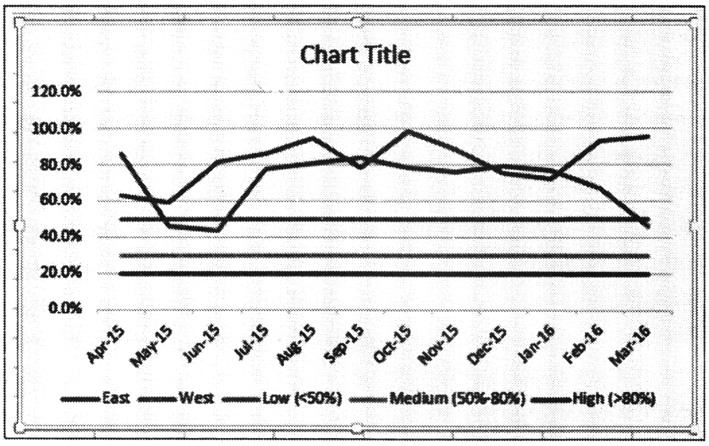
	A	B	C	D	E	F	G
1							
2		Month	East	West	Low (<50%)	Medium (50%-80%)	High (>80%)
3		Apr-15	86.4%	63.0%	50%	30%	20%
4		May-15	45.8%	58.9%	50%	30%	20%
5		Jun-15	44.1%	81.6%	50%	30%	20%
6		Jul-15	77.6%	86.1%	50%	30%	20%
7		Aug-15	80.7%	95.0%	50%	30%	20%
8		Sep-15	83.7%	78.2%	50%	30%	20%
9		Oct-15	78.8%	98.9%	50%	30%	20%
10		Nov-15	76.0%	88.3%	50%	30%	20%
11		Dec-15	79.0%	75.5%	50%	30%	20%
12		Jan-16	77.0%	72.1%	50%	30%	20%
13		Feb-16	67.1%	93.1%	50%	30%	20%
14		Mar-16	45.8%	95.7%	50%	30%	20%

Here, in the data < 50% is Low, 50% - 80% is Medium and > 80% is High.

With Band Chart, you can display your survey results as follows:

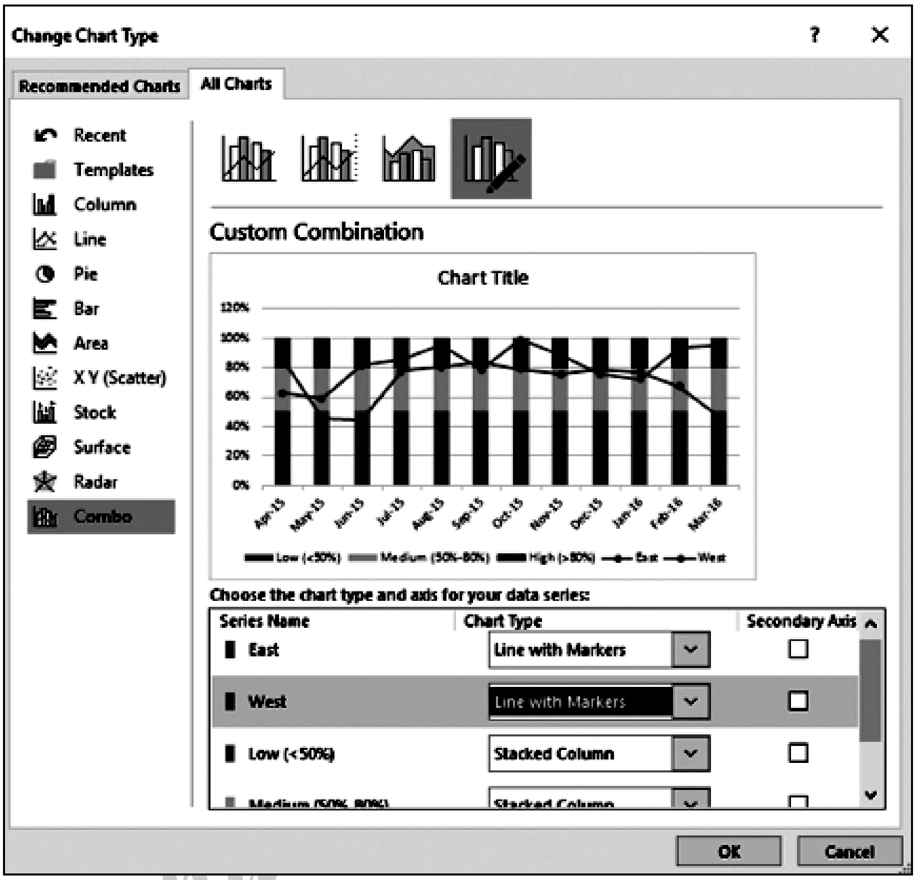


Create a Line Chart from your data.



Change the chart type to:

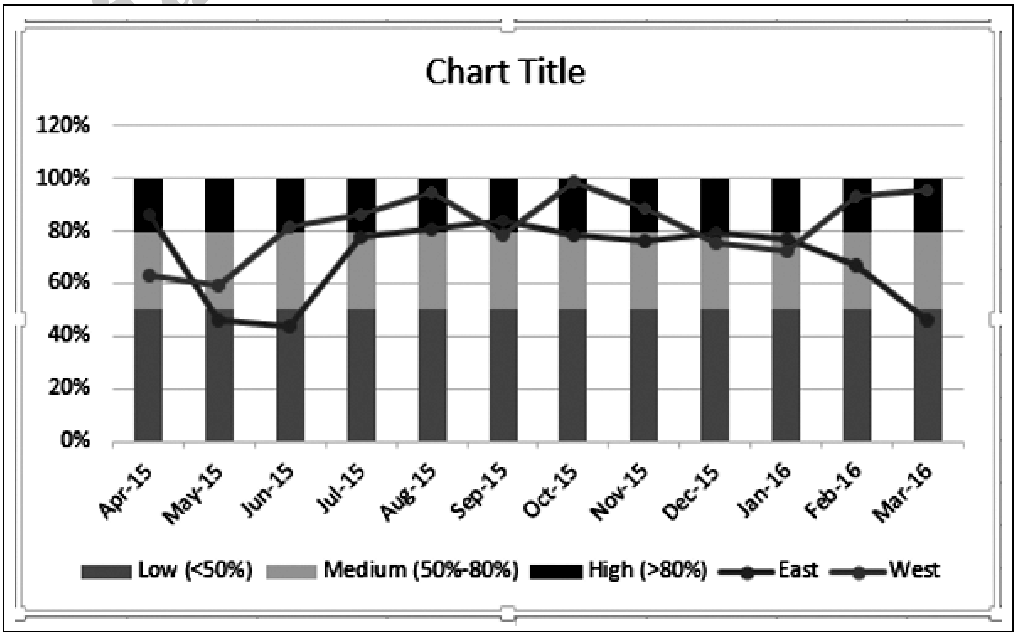
- ➢ East and West Series to Line with Markers.
- ➢ Low, Medium and High Series to Stacked Column.



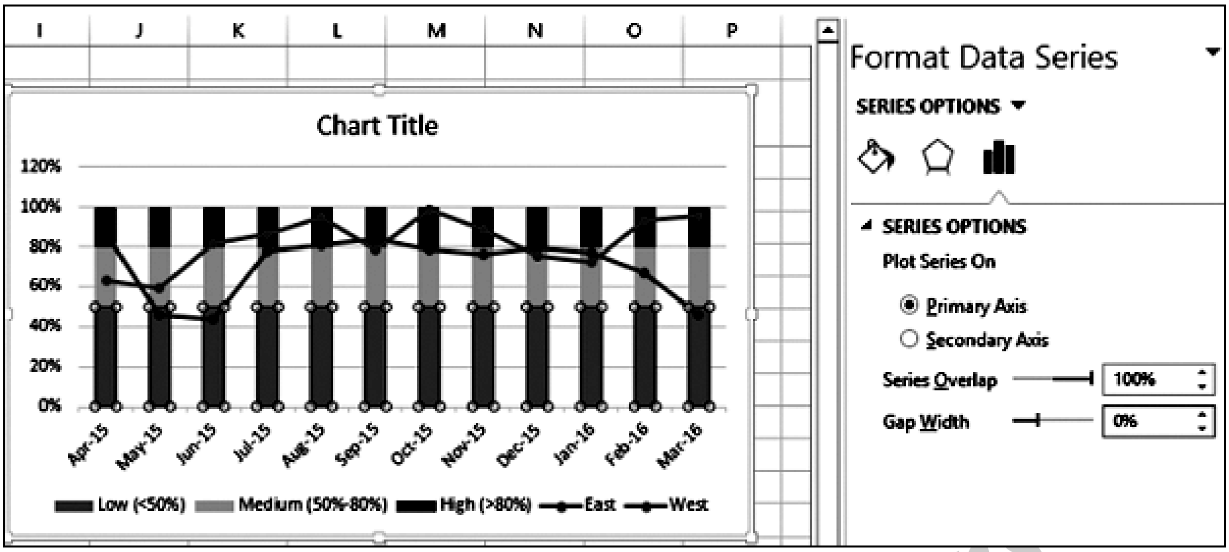
Choose the chart type and axis for your data series:

Series Name	Chart Type	Secondary Axis
East	Line with Markers	□
West	Line with Markers	□
Low (<50%)	Stacked Column	□
Medium (50%-80%)	Stacked Column	□

Your chart looks as follows:



- ➢ Click on one of the columns.
- ➢ Change gap width to 0% in Format Data Series.



Format Data Series

SERIES OPTIONS ▾

◢ SERIES OPTIONS

Plot Series On

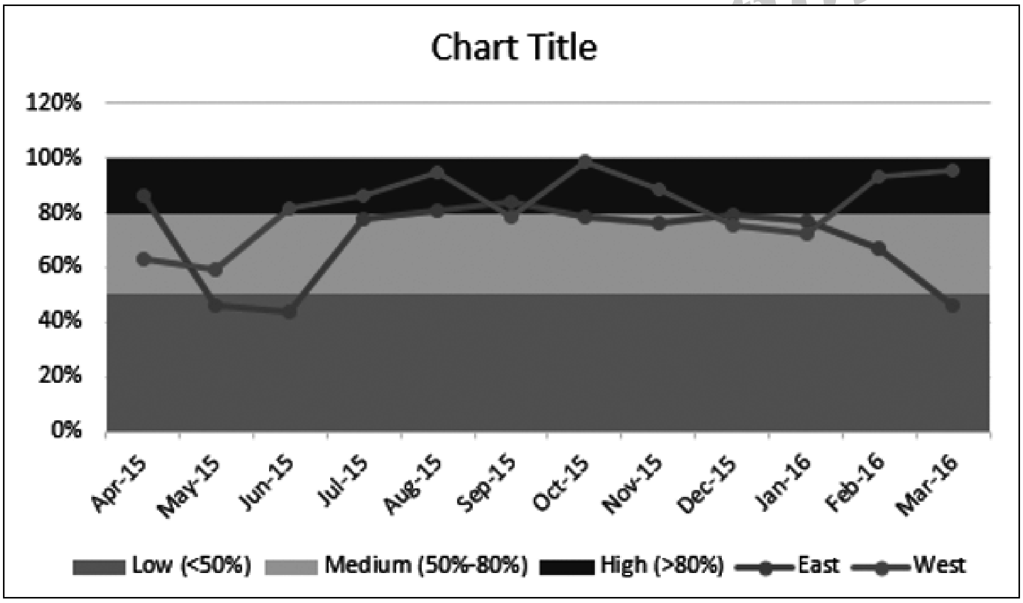
● Primary Axis

○ Secondary Axis

Series Overlap 100%

Gap Width 0%

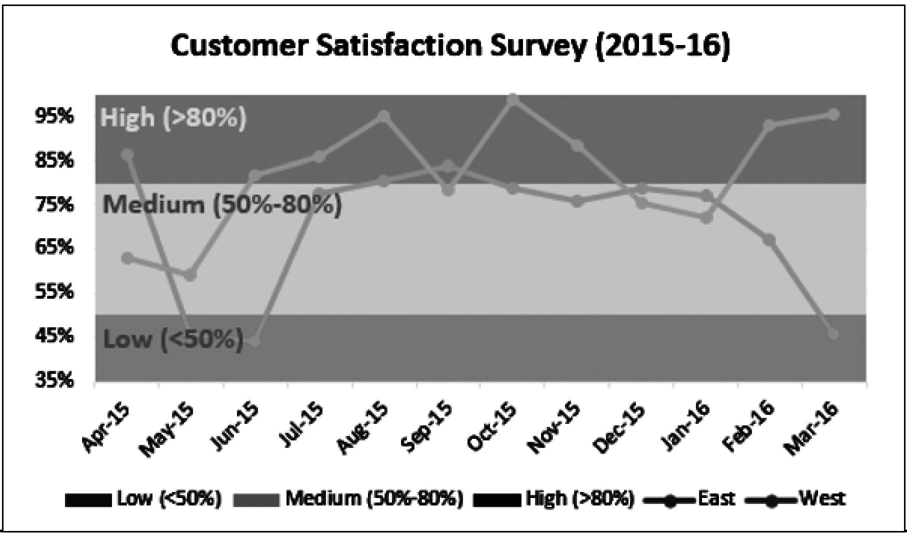
You will get Bands instead of columns.



To make the chart more presentable:

- ➢ Add Chart Title.
- ➢ Adjust Vertical Axis range.
- ➢ Change the colors of the bands to Green-Yellow-Red.
- ➢ Add Labels to bands.

The final result is the Band Chart with the defined boundaries and the survey results represented across the bands. One can quickly and clearly make out from the chart that while the survey results for the region West are satisfactory, those for the region East have a decline in the last quarter and need attention.



Q13. Explain briefly about Gantt Chart.

Ans : **(Imp.)**

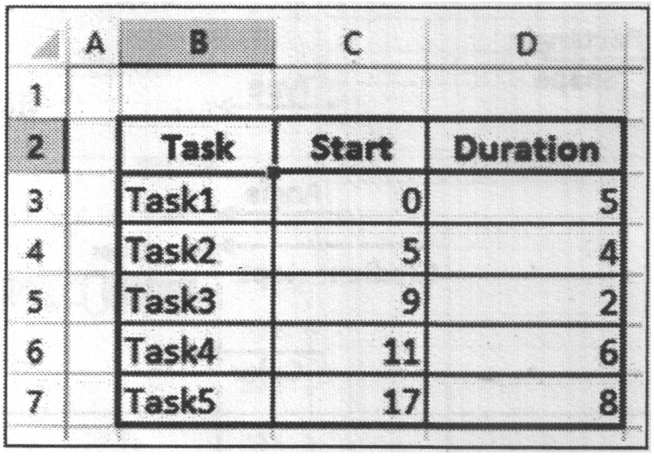
A Gantt Chart is a chart in which a series of horizontal lines shows the amount of work done in certain periods of time in relation to the amount of work planned for those periods.

In Excel, you can create a Gantt Chart by customizing a Stacked Bar Chart type so that it depicts tasks, task duration and hierarchy. An Excel Gantt Chart typically uses days as the unit of time along the horizontal axis.

Consider the following data where the column:

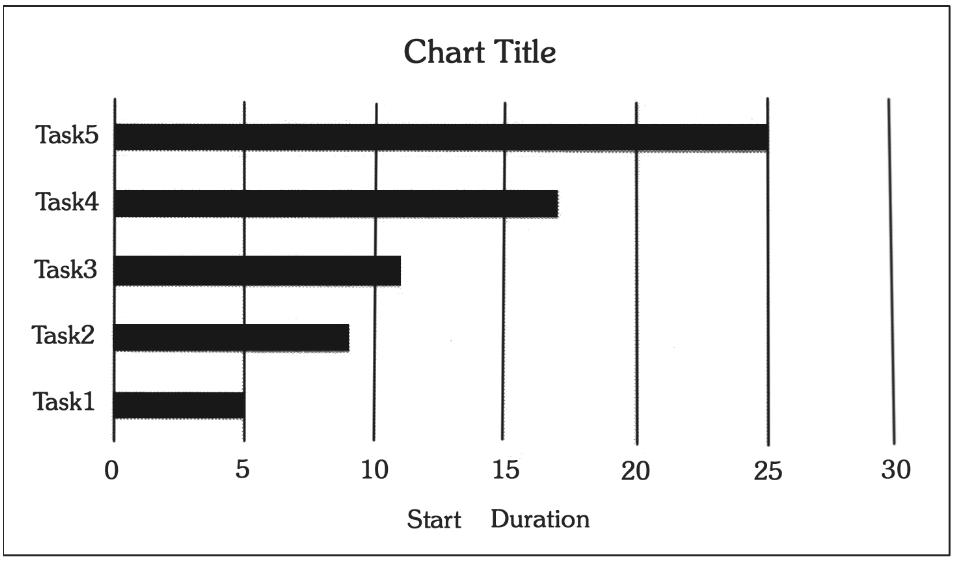
- Task represents the Tasks in the project
- Start represents number of days from the Start Date of the project
- Duration represents the duration of the Task

Note that Start of any Task is Start of previous Task + Duration. This is the case when the Tasks are in hierarchy.

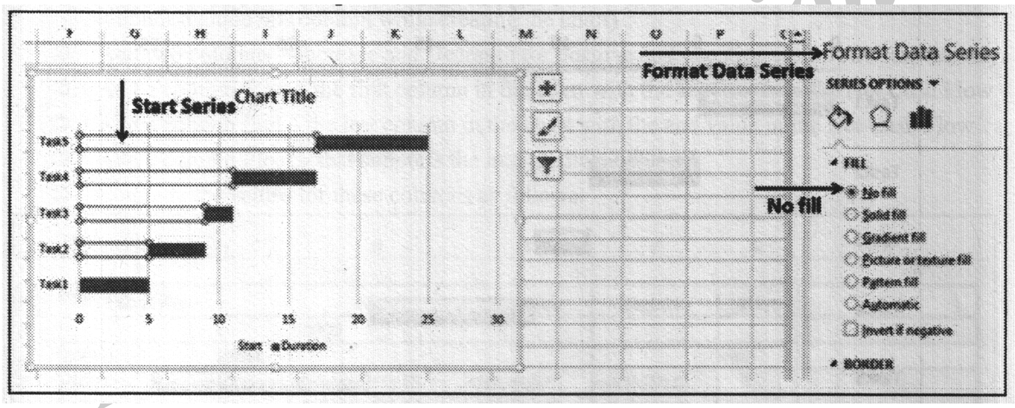


	A	B	C	D
1				
2		Task	Start	Duration
3		Task1	0	5
4		Task2	5	4
5		Task3	9	2
6		Task4	11	6
7		Task5	17	8

- Select the data
- Create Stacked Bar Chart



- Right-click on Start Series
- In Format Data Series options, select No fill



Format Data Series

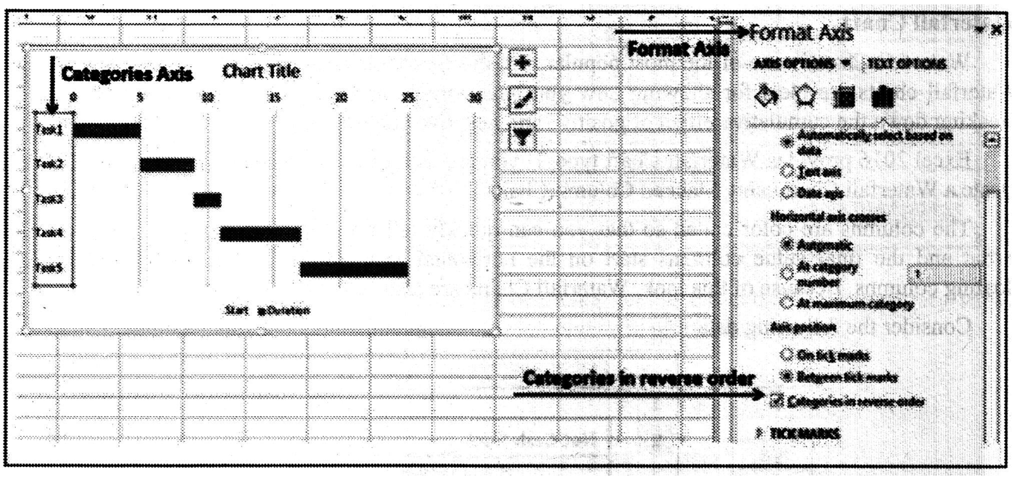
SERIES OPTIONS ▾

◢ FILL

● No fill
○ Solid fill
○ Gradient fill
○ Picture or texture fill
○ Pattern fill
○ Automatic
□ Invert if negative

◢ BORDER

- Right-click on Categories Axis
- In Format Axis options, select Categories in reverse order.



Format Axis

AXIS OPTIONS ▾ TEXT OPTIONS

● Automatically select based on data
○ Text axis
○ Date axis

Horizontal axis crosses

● Automatic
○ At category number
○ At maximum category

Axis position

○ On tick marks
● Between tick marks
☑ Categories in reverse order

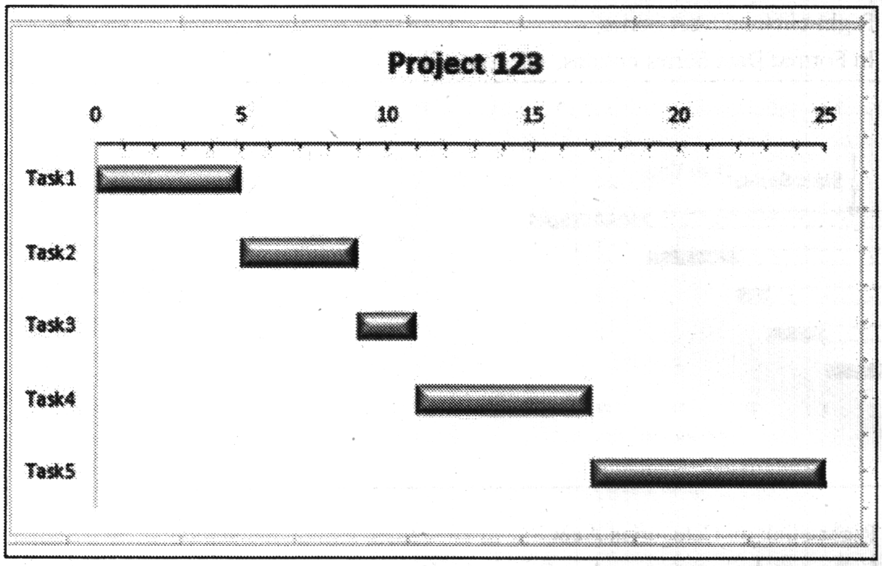
▸ TICK MARKS

In Chart Elements, deselect :

- Legend
- Gridlines

Format the Horizontal Axis to :

- Adjust the range
- Major Tick Marks at 5 day intervals
- Minor Tick Marks at 1 day intervals Format Data Series to make it look impressive Give a Chart Title.



Q14. Explain briefly about pivot charts.

Ans : **(Imp.)**

PivotCharts are used to graphically summarize data and explore complicated data.

A PivotChart shows data series, categories and chart axes the same way a standard chart does. Additionally, it also gives you interactive filtering controls right on the chart so that you can quickly analyze a subset of your data.

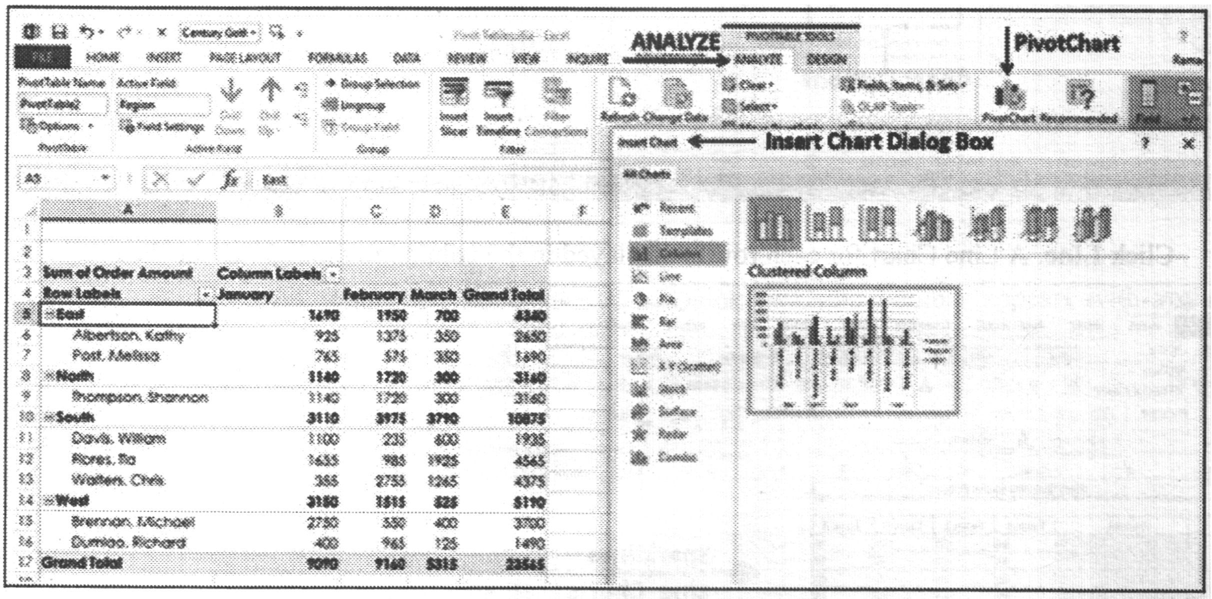
PivotCharts are useful when you have data in a huge PivotTable, or many complex worksheet data that includes text and numbers. A PivotChart can help you make sense of this data.

You can create a PivotChart from:

- A PivotTable
- A Data Table as a standalone without Pivot Table Pivot Chart from Pivot Table

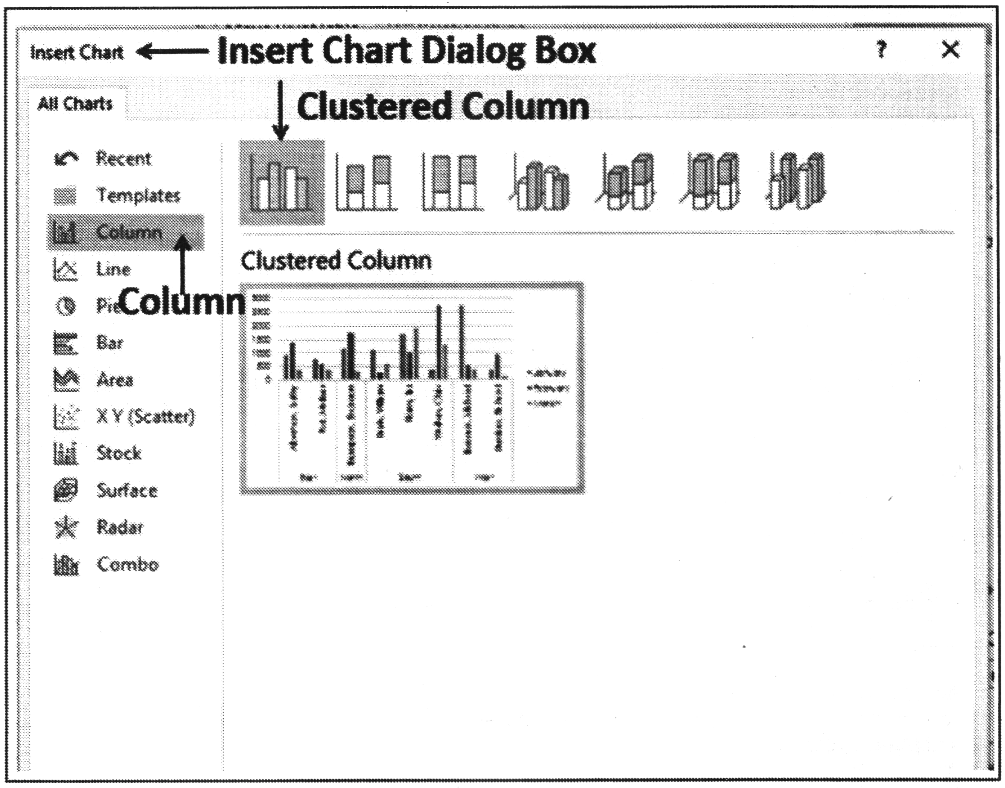
To create a PivotChart, follow the steps given below:

- Click the PivotTable.
- Click ANALYZE under PIVOTTABLE TOOLS on the Ribbon.
- Click on PivotChart. The Insert Chart dialog box appears.



Sum of Order Amount	Column Labels			
Row Labels	January	February	March	Grand Total
East	1690	1950	700	4340
Albertson, Kathy	925	1375	350	2650
Post, Melissa	765	575	350	1690
North	1140	1720	300	3160
Thompson, Shannon	1140	1720	300	3160
South	3110	3975	3790	10875
Davis, William	1100	235	600	1935
Flores, Tia	1635	985	1925	4565
Walters, Chris	355	2755	1265	4375
West	3150	1515	525	5190
Brennan, Michael	2750	550	400	3700
Dumlao, Richard	400	965	125	1490
Grand Total	9090	9160	5315	23565

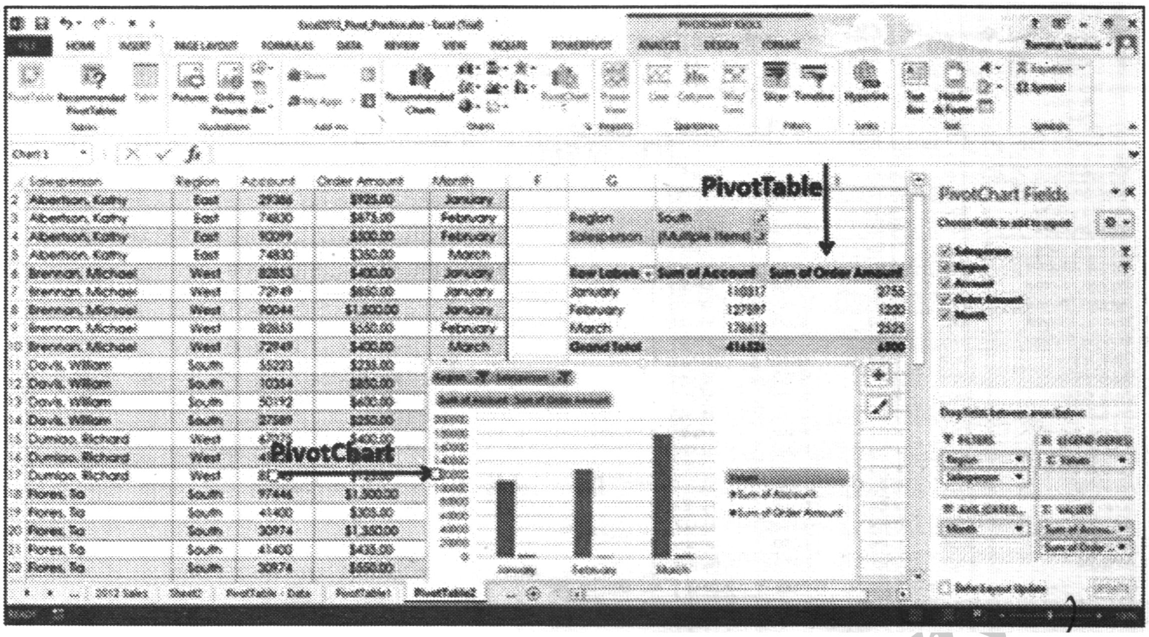
Select Clustered Column from the option Column.



Click OK. The PivotChart is displayed.

The PivotChart has three filters - Region, Salesperson and Month.

- Click the Region Filter Control option. The Search Box appears with the list of all Regions. Check boxes appear next to Regions.



	Salesperson	Region	Account	Order Amount	Month
2	Albertson, Kathy	East	29386	$925.00	January
3	Albertson, Kathy	East	74830	$875.00	February
4	Albertson, Kathy	East	90099	$500.00	February
5	Albertson, Kathy	East	74830	$350.00	March
6	Brennan, Michael	West	82853	$400.00	January
7	Brennan, Michael	West	72949	$850.00	January
8	Brennan, Michael	West	90044	$1,500.00	January
9	Brennan, Michael	West	82853	$550.00	February
10	Brennan, Michael	West	72949	$400.00	March
11	Davis, William	South	55223	$235.00	
12	Davis, William	South	10354	$850.00	
13	Davis, William	South	50192	$400.00	
14	Davis, William	South	27589	$250.00	
15	Dumlao, Richard	West	47925	$400.00	
16	Dumlao, Richard	West	[illegible]	[illegible]	
17	Dumlao, Richard	West	[illegible]	[illegible]	
18	Flores, Tia	South	97446	$1,300.00	
19	Flores, Tia	South	41400	$305.00	
20	Flores, Tia	South	30974	$1,350.00	
21	Flores, Tia	South	41400	$435.00	
22	Flores, Tia	South	30974	$550.00	

Region: South
Salesperson: (Multiple Items)

Row Labels	Sum of Account	Sum of Order Amount
January	138817	2755
February	127897	1220
March	178612	2525
Grand Total	416326	6500

2.3.3 Data Dashboards using Advanced MS. Excel (or) Spss

Q15. What is Data dashboard and explain different types dashboards?

Ans :

A data dashboard is an information management tool that visually tracks, analyzes and displays key performance indicators (KPI), metrics and key data points to monitor the health of a business, department or specific process. They are customizable to meet the specific needs of a department and company. Behind the scenes, a dashboard connects to your files, attachments, services and API's, but on the surface displays all this data in the form of tables, line charts, bar charts and gauges. A data dashboard is the most efficient way to track multiple data sources because it provides a central location for businesses to monitor and analyze performance. Real-time monitoring reduces the hours of analyzing and long line of communication that previously challenged businesses.

In the present terms, a dashboard can be defined as a data visualization tool that displays the current status of metrics and key performance indicators (KPIs) simplifying complex data sets to provide users with at a glance awareness of current performance.

Dashboards consolidate and arrange numbers and metrics on a single screen. They can be tailored for a specific role and display metrics of a department or an organization on the whole.

Dashboards can be static for a one-time view, or dynamic showing the consolidated results of the data changes behind the screen. They can also be made interactive to display the various segments of large data on a single screen.

Types

Dashboards can be categorized based on their utility as follows:

1. Strategic Dashboards
2. Analytical Dashboards
3. Operational Dashboards
4. Informational Dashboards

1. Strategic Dashboards

Strategic dashboards support managers at any level in an organization for decision-making. They provide the snapshot of data, displaying the health and opportunities of the business, focusing on the high level measures of performance and forecasts.

- ➢ Strategic dashboards require to have periodic and static snapshots of data (e.g., daily, weekly, monthly, quarterly and annually). They need not be constantly changing from one moment to the next and require an update at the specified intervals of time.
- ➢ They portray only the high level data not necessarily giving the details.
- ➢ They can be interactive to facilitate comparisons and different views in case of large data sets at the click of a button. But, it is not necessary to provide more interactive features in these dashboards.

The following screenshot shows an example of an executive dashboard, displaying goals and progress.

2. Analytical Dashboards

Analytical dashboards include more context, comparisons, and history. They focus on the various facets of data required for analysis.

Analytical dashboards typically support interactions with the data, such as drilling down into the underlying details, and hence should be interactive.

Examples of analytical dashboards include Finance Management dashboard and Sales Management dashboard.

3. Operational Dashboards

Operational dashboards are for constant monitoring of operations. They are often designed differently from strategic or analytical dashboards and focus on monitoring of activities and events that are constantly changing and might require attention and response at a moment's notice. Thus, operational dashboards require live and up-to-date data available at all times, and hence should be dynamic.

An example of an operation dashboard could be a support-system dashboard, displaying live data on service tickets that require an immediate action from the supervisor on high-priority tickets.

4. Informational Dashboards

Informational dashboards are just for displaying figures, facts and/or statistics. They can be either static or dynamic with live data but not interactive. For example, flights arrival/departure information dashboard in an airport.

Q16. What are the steps to create interactive Excel Dash Board?

Ans : **(Imp.)**

Create Interactive Excel Dashboard

Most of us probably rely on our trusted MS Excel dashboard for the day to day running of our businesses, but like many, we struggle to turn that data into something that will actually interest people and want them to know more about it. It is a comprehensive as well as complete visual report or analysis of your project which can be shared with other people concerned. Creating a excel dashboard can be tedious, time consuming as well as difficult if you do not have the proper knowledge about how to go about doing it. But fret now, that's where we enter.

Dashboards are not native to Excel, as they can be created on PowerPoint as well. Excel Dashboards offer a more dynamic approach to presenting data as compared to the more liner as well as un moving nature of PowerPoint dashboards. An interactive dashboard in excel is basically slices of visualization which enables your data to tell a story. A dashboard is only useful if they are dynamic, easy to use as well as compatible with the PC you are using. Before making a dashboard, you need to consider the decisions the end user will make based on the data, the look and feel of it. You will also need to keep in mind how much they are familiar with the data and how much context they have. For example, a monthly report for your boss who is already familiar with everything is going to look very different from the one which you make to pitch a new idea to a potential client.

Another thing to remember is that the data should be the star of the excel dashboard. There is no need to clutter the screen with unnecessary components, so keeping it simple is the best way to go. You will also want to strike a perfect balance between making it look striking (so that it holds your audience's attention), but not so stylized so that it takes away from the data to be presented. When we tell a story, we must always consider the tastes and distastes of the audience and adapt our presentation accordingly. For example, if you are presenting to a very formal organisation, you should do your best to keep the excel dashboard as simple as possible, without compromising on subdued attractiveness.

Armed with the right knowledge about how to go on about creating a stunning excel dashboard, you can create a excel dashboard of your own without it being tedious or difficult! We provide you with a step by step analysis below:

1. Bringing in data

Sure, Excel is very useful and flexible. But to create a excel dashboard you cannot just paste some data and add a few charts. You need to maintain it, update it and you must impose some kind of structure to that data. Usually you don't have to enter the data directly into the spreadsheet. You can copy paste the data but the best option is to bring in data via an external source. You can use it to connect your excel dashboard to Access or Oracle. A good practice is to limit the amount of data you bring in. As we've seen before, data can be brought in with two basic structures: a flat file and a pivot table. A flat file is generally smaller, where as a pivot table is a large file (As a thumb rule). Both have their pros and cons which one must figure out only through experience.

2. Select a background

Select an appropriate background which will bring your excel dashboard appear attractive without taking focus away from the data. Your data should be the star. You can go with subdued shades like blue, grey and black or you can take it up a notch like orange, green and purple. It is your choice, but keep in mind the audience you will be presenting it to. I suggest you stick to subdued hues if it is for official purposes.

3. Manage your data and link it to your excel dashboard

If you are using a pivot table, use the GETPIVOTDATA function. If you use a flat file, there are a number of formulae you can use like DSUM, DGET, VLOOKUP, MATCH, INDEX or even a dew math formulas like SUM, SUMIF, etc

But be careful here, do not punch in formula after formula. Fewer formulas mean a safer and a more reliable excel dashboard which is also easier to maintain. You can automatically reduce the formula number by using pivot tables.

Also, another important point is that you should name all your ranges. Always, always document your work. Simplify your work by making your excel dashboard formulas cleaner.

4. Use Dynamic Charting

Dashboards that a user can't interact with don't make much sense. All your excel dashboards should have controls which will enable you to change the markets, product details as well as other knitty critters. What is most important is that the user must be able to be in complete charge of his or her own excel dashboard and make changes whenever and wherever they want.

If you are creating interactive charts, you will need dynamic ranges. You can do this by using the OFFSET() function. You can also add a few cool things to your excel dashboard like greeting the user and selecting the corresponding profile when they open the excel dashboard. All this can be done using macros. All you need to do is record a macro, add a FOR NEXT or a FOR EACH loop. If you have never recorded a macro before, there are a large number of sites online which give you perfectly tailored macros as per your needs.

5. Design your excel dashboard report

If you are still using Excel 2003 or 2007, their default charts are not very attractive so I suggest you avoid them like the plague, but make sure to use acceptable formats. Excel 2010 and 2013 are a lot better but they still need some work. Keep this in mind, a chart is used to discover actionable patterns in the data and you should do your best to bring out most of it. This also means that you should remove all the jazzy, glittery stuff which adds no value to your excel dashboard. What you can do instead is create a hierarchy of focus and contextual data that is relevant, and create a form of basic interact if not much.

6. Dashboard Storytelling

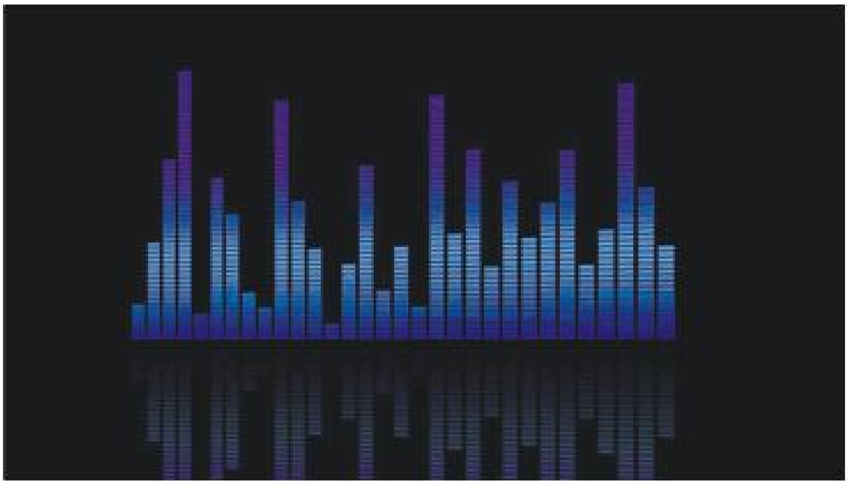
Storytelling which is pregnant with data is the best kind that there is. With better access to data and better tools to make a point, we are able to recover a lot of data types. However, even though data is good, it is great, but you must not reveal all of it at once. When deciding how to make a excel dashboard, start by reviewing the purpose of the said dashboard. The goal shouldn't be to overwhelm the audience with data, but to provide data in such a form so that it gives them the insight you want them to have. I think this is true for all data based projects.

Let your audience explore the data on their own by offering them their own filters and controls. This is where interactive visuals come in the picture. If you are a newcomer to interactive excel dashboards, you can still spot trends and learn how to build up a stunning dashboard. If you are a pro at it, you can drill down deeper into the data for better charts.

7. Select the right kind of chart type

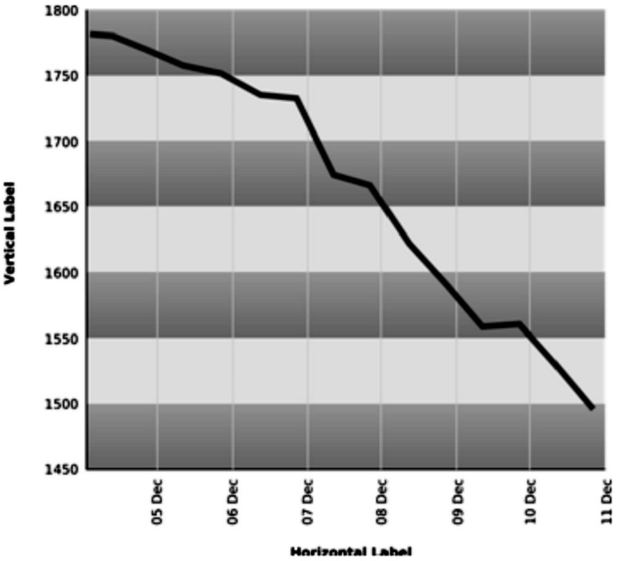
Before we decide which chart to use in our excel dashboard, let us have a review of all the charts used in dashboards and when to use what.

(a) **Bar Charts:** Bar charts as we all know are bars on the x axis. One of the most common misgiving about excel dashboards is that the more is better; the truth is, that is seldom true. Bar charts are simple and very effective. They are particularly useful to compare one concept to another as well as trends.



(b) Pie Charts: These charts, in my personal opinion, should be used very carefully and sparingly. Well, no matter how you feel about pie charts, you should only use them when you need a graph representing proportions of a whole. Use with extreme frugality.

(c) Line Charts: These are one of my favorites. They are so simplistic. These charts include a serious of data points that are connected by a line. These are best used to show developments over a certain period of time.



(d) **Tables:** Tables are great if you have detailed information with different measuring units, which may be difficult to represent through other charts or graphs.

(e) **Area charts:** Area charts are very useful for multiple data series, which may or may not be related to each other (partially or wholly). They are also useful for an individual series that represents a physically countable set.

So choose wisely, and you will be good.

8. Colour theory

Colours in a excel dashboard make it livelier as opposed to the drab and overused grey, black and white. I could write an entire book on how colour theory works, but well, that's already dine. You must know which colours work together and which do not. For example, you cannot pair bright pink and red together unless you want an assault on the eyes. One thing you must keep in mind while selecting a colour coding, that 8% of men and 0.5% or women are colour blind.

Most people can perceive a colour, but cannot correctly distinguish between two shades of the same colour. These people can perceive changes in brightness though, just like me and you. Avoid having shades that overlap, like the example I gave above. That would not only look ugly, but also be completely useless for users we discussed above.

Q17. What are the benefits of data dash boards?

Ans :

Dashboards allow managers to monitor the contribution of the various departments in the organization. To monitor the organization's overall performance, dashboards allow you to capture and report specific data points from each of the departments in the organization, providing a snapshot of current performance and a comparison with earlier performance.

Benefits of dashboards include the following:

- Visual presentation of performance measures
- Ability to identify and correct negative trends
- Measurement of efficiencies/inefficiencies
- Ability to generate detailed reports showing new trends
- Ability to make more informed decisions based on collected data
- Alignment of strategies and organizational goals
- Instant visibility of all systems in total
- Quick identification of data outliers and correlations
- Time-saving with the comprehensive data visualization as compared to running multiple reports.

Short Question and Answers

1. What is Statistics?

Ans :

Statistics is a branch of mathematics that deals with collecting, interpreting, organization and interpretation of data.

According to Prof Horace Secrist, Statistics may be defined as the aggregate of facts affected to a marked extent by multiplicity of causes, numerically expressed enumerated or estimated according to a reasonable standard of accuracy, collected in a systematic manner, for a predetermined purpose and placed in relation to each other.

Descriptive statistics employs a set of procedures that make it possible to meaningfully and accurately summarize and describe samples of data. In order for one to make meaningful statements about psychological events, the variable or variables involved must be organized, measured, and then expressed as quantities. Such measurements are often expressed as measures of central tendency and measures of variability.

2. Descriptive Statistics.

Ans :

Descriptive Statistics

Descriptive statistics is used to summarize data and make sense out of the raw data collected during the research. Since the data usually represents a sample, then the descriptive statistics is a quantitative description of the sample.

The level of measurement of the data affects the type of descriptive statistics. Nominal and ordinal type data (often termed together as categorical type data) will differ in the analysis from interval and ratio type data (often termed together as continuous type data).

Descriptive statistics for categorical data

Contingency tables (or frequency tables) are used to tabulate categorical data. A contingency table shows a matrix or table between independent variables at the top row versus a dependent variable on the left column, with the cells indicating the frequency of occurrence of possible combination of levels. (check SPSS for examples).

Descriptive statistics for continuous data

There are two the two aspects of descriptive statistics used for continuous type data. They are;

- Central tendency
- Variability of the data.

3. What is Data visualization?

Ans :

It is a general term that describes any effort to help people understand the significance of data by placing it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data visualization software.

- Today's data visualization tools go beyond the standard charts and graphs used in Microsoft Excel spreadsheets, displaying data in more sophisticated ways such as info graphics, dials and gauges, geographic maps, spark lines, heat maps, and detailed bar, pie and fever charts. The images may include interactive capabilities, enabling users to manipulate them or drill into the data for querying and analysis. Indicators designed to alert users when data has been updated or predefined conditions occur can also be included.

4. Cross Tabulations.

Ans :

Cross tabulation is usually performed on categorical data - data that can be divided into mutually exclusive groups.

An example of categorical data is the region of sales for a product. Typically, region can be divided into categories such as geographic area (North, South, Northeast, West, etc) or state (Andhra Pradesh, Rajasthan, Bihar, etc). The important thing to remember about categorical data is that a categorical data point cannot belong to more than one category.

Cross tabulations are used to examine relationships within data that may not be readily apparent. Cross tabulation is especially useful for studying market research or survey responses. Cross tabulation of categorical data can be done with through tools such as SPSS, SAS, and Microsoft Excel.

5. Benefits of cross tabulation.

Ans :

(i) Eliminates confusion while interpreting data

Raw data can be difficult to interpret. Even for small data sets, it is all too easy to derive wrong results by just looking at the data. Cross tabulation offers a simple method of grouping variables, which minimizes the potential for confusion or error by providing clear results.

(ii) Helps in deriving innumerable insights

As we observed in our example, cross tabulation can help us derive great insights from raw data. These insights are not easy to see when the raw data is formatted as a table. Since cross tabulation clearly maps out relations between categorical variables, researchers can gain better and deeper insights — insights that otherwise would have been overlooked or would have taken a lot of time to decode from more complicated forms of statistical analysis.

(iii) Offers data points to chart out a course of action

Cross tabulation makes it easier to interpret data, which is beneficial for researchers who have limited knowledge of statistical analysis. With cross tabulation, people do not need statistical programming to correlate categorical variables. The clarity offered by cross tabulation helps professionals evaluate their current work and chart out future strategies.

6. What is data dashboard?

Ans :

A data dashboard is an information management tool that visually tracks, analyzes and displays key performance indicators (KPI), metrics and key data points to monitor the health of a business, department or specific process. They are customizable to meet the specific needs of a department and company. Behind the scenes, a dashboard connects to your files, attachments, services and API's, but on the surface displays all this data in the form of tables, line charts, bar charts and gauges. A data dashboard is the most efficient way to track multiple data sources because it provides a central location for businesses to monitor and analyze performance. Real-time monitoring reduces the hours of analyzing and long line of communication that previously challenged businesses.

7. Importance of data visualization.

Ans :

- Data visualization has become the de facto standard for modern business intelligence (BI). The success of the two leading vendors in the BI space, Tableau and Qlike - both of which heavily emphasize visualization - has moved other vendors toward a more visual approach in their software. Virtually all BI software has strong data visualization functionality.
- Data visualization tools have been important in democratizing data and analytics and making data-driven insights available to workers throughout an organization. They are typically easier to operate than traditional statistical analysis software or earlier versions of BI software. This has led to a rise in lines of business implementing data visualization tools on their own, without support from IT.
- Data visualization software also plays an important role in big data and advanced analytics projects. As businesses accumulated massive troves of data during the early years of the big data trend, they needed a way to quickly and easily get an overview of their data. Visualization tools were a natural fit.

- Visualization is central to advanced analytics for similar reasons. When a data scientist is writing advanced predictive analytics or machine learning algorithms, it becomes important to visualize the outputs to monitor results and ensure that models are performing as intended. This is because visualizations of complex algorithms are generally easier to interpret than numerical outputs.

8. Explain the uses of data visualization.

Ans :

- By using data visualization, it became easier for business owners to understand their large data in a simple format.
- The visualization method is also time saving. So, businesses does not have to spend much time to make a report or solve a query. They can easily do it in a less time and in a more appealing way.
- Visual analytics offers a story to the viewers. By using charts and graphs or images, a person can easily exposure the whole concept as well the viewers will be able to understand the whole thing in an easy way.
- The most complicated data will look easy when it gets through the process of visualization. Complicated data report gets converted into a simple format. And it helps people to understand the concept in an easy way.
- With the visualization process, it gets easier to the business owners to understand their product growth and market competition in a better way.

9. Gantt Chart.

Ans :

A Gantt Chart is a chart in which a series of horizontal lines shows the amount of work done in certain periods of time in relation to the amount of work planned for those periods.

In Excel, you can create a Gantt Chart by customizing a Stacked Bar Chart type so that it depicts tasks, task duration and hierarchy. An Excel Gantt Chart typically uses days as the unit of time along the horizontal axis.

Consider the following data where the column:

- Task represents the Tasks in the project
- Start represents number of days from the Start Date of the project
- Duration represents the duration of the Task.

10. What are the benefits of data dash boards?

Ans :

Benefits of dashboards include the following:

- Visual presentation of performance measures
- Ability to identify and correct negative trends
- Measurement of efficiencies/inefficiencies
- Ability to generate detailed reports showing new trends
- Ability to make more informed decisions based on collected data
- Alignment of strategies and organizational goals
- Instant visibility of all systems in total
- Quick identification of data outliers and correlations
- Time-saving with the comprehensive data visualization as compared to running multiple reports.

Choose the Correct Answers

1. An ideal measure of central tendency is ________. [a]

   (a) Arithmetic mean (b) Moving average

   (c) Median (d) Harmonic Mean

2. Mathematical average is called [a]

   (a) Arithmetic mean (b) Geometric mean

   (c) Mode (d) None of these

3. Sum of deviations of the items is zero from [a]

   (a) Mean (b) Median

   (c) Mode (d) Geometric mean

4. A ________ is a characteristic that takes different values at different times, places or situations. [d]

   (a) Attributes (b) Data

   (c) Statistics (d) Variable

5. Measure of central tendency ________. [d]

   (a) Mean (b) Mode

   (c) Median (d) All

6. Which of the following measures of central tendency will always change if a single value in the data changes? [a]

   (a) Mean (b) Median

   (c) Mode (d) All of these

7. If a positively skewed distribution has a median of 50, which of the following statement is true? [c]

   (a) Mean is greater than 50 (b) Mean is less than 50

   (c) Both (a) and (c) (d) Mode is greater than 50

8. If the variance of a dataset is correctly computed with the formula using (n - 1) in the denominator, which of the following option is true? [c]

   (a) Dataset is a sample

   (b) Dataset is a population

   (c) Dataset could be either a sample or a population

   (d) Dataset is from a census

9. The difference between the highest and the lowest value of the observations in a data is called: [b]

   (a) Mean (b) Range

   (c) Total frequency (d) Sum of observation

10. The range of the data: 6,14,20,16,6,5,4,18,25,15, and 5 is [b]

    (a) 4 (b) 21

    (c) 25 (d) 20

Fill in the blanks

1. _______ is a branch of mathematics that deals with collecting, interpreting, organization and interpretation of data.
2. _______ statistics is used to summarize data and make sense out of the raw data collected during the research.
3. _______ It is the midpoint of a distribution of data.
4. _______ stands for business integer.
5. _______ reports using Ms.Excel can display in a number of ways.
6. _______ is usually performed on categorical data that can be divided into mutually exclusive groups.
7. _______ Charts are used to graphically summarize data and explore complicated data.
8. KPI stands for _______.
9. _______ Represent the information in Rows and Columns.
10. _______ allow managers to monitor the contribution of the various departments in the organization.

Answers

1. Statistics
2. Descriptive
3. Median
4. BI
5. Data analysis
6. Cross tabulation
7. Pivot
8. Key perofrmance indicators
9. Tables
10. Dashboards

UNIT III

Predictive Analytics:

Trend Lines, Regression Analysis - Linear & Multiple, Predictive modeling, forecasting Techniques, Data Mining - Definition, Approaches in Data Mining - Data Exploration & Reduction, Data mining and business intelligence, Data mining for business Classification, Association, Cause Effect Modelling.

3.1 Trend Lines

Q1. What is predictive analysis ?

Ans :

Definition

Predictive Analytics is the practice of extracting information from existing data sets in order to determine patterns and predict future outcomes and trends. It does not tell you what will happen in the future.

Predictive analytics is the branch of the Advanced Analytics which is used to make predictions about unknown future events. It uses many techniques from data mining, statistics, modeling, machine learning, and artificial intelligence to analyze current data to make predictions about future.

Predictive Analytics uses historical data to predict future events. Typically, historical data is used to build a mathematical model that captures important trends. That predictive model is then used on current data to predict what will happen next, or to suggest actions to take for optimal outcomes.

Q2. How predictive analysis works ?

Ans :

Predictive Analytics is the process of using data analytics to make predictions based on data. This process uses data along with analysis, statistics and machine learning techniques to create a predictive model for forecasting future events.

The term "predictive analytics" describes the application of a statistical or machine learning technique to create a quantitative prediction about the future. Frequently, supervised machine learning techniques are used to predict a future value (How long can this machine run before requiring maintenance?) or to estimate a probability (How likely is this customer to default on a loan?).

Predictive Analytics starts with a business goal to use data to reduce waste, save time or cut costs. The process harnesses heterogeneous, often massive, data sets into models that can generate clear, actionable outcomes to support achieving that goal, such as less material waste, less stocked inventory, and manufactured product that meets specifications.

Q3. Explain briefly about Trend Analysis.

Ans :

A trend line is a straight line connecting a number of points on a graph. It is used to analyze the specific direction of a group of values set in a presentation. There are two kinds of trend lines, an uptrend with values going higher, and a downtrend where the direction of the line gradually drops to the lower values.

(i) Predicting the Future

Trend lines allow businesses to see the difference in various points over a period of time. This helps foretell the possible path the values will take in the future. This can help reveal perform Predictive Trend Line. The Predictive Trend Line add-on displays linear, exponential, and logarithmic trend lines in your dashboard charts. The trend line allows an end user to predict future values/metrics based on historical data. See the referenced wiki page on the regression analysis for more information business departments, such as sales.

By knowing how to add a trend line to your presentation, you can create a graphical representation of the values you have computed. This will enable the user to easily comprehend and analyze the message you are trying to imply.

ii) Predictive Trend Line

The Predictive Trend Line add-on displays linear, exponential, and logarithmic trend lines in your dashboard charts. The trend line allows an end user to predict future values/ metrics based on historical data. See the referenced wiki page on the regression analysis for more information.

3.2 Regression Analysis

Q4. Explain the concept of regression analysis.

Ans : **(Imp.)**

Regression takes its name from studies made by Sir Francis Galton. He compared the heights of persons to the heights of their parents. His major conclusion was that the off-springs of unusually tall persons tend to be shorter than their parents while children of usually short parents tend to be shorter than their parents.

In a sense, the successive generations of off-springs from tall persons "regress" downward toward the height of the population. While the reverse is true originally for short families. But the distribution of heights for the total population, continues to have the same variability from generation to generation.

Regression Variables

(i) Independent Variable (Regressor or Predictor or Explanatory)

The variable which influences the values of the other variable or which is used for prediction of the value of the other variable is called independent variable.

(ii) Dependent Variable (Regressed or Explained Variable)

The variable whose value is influenced or is to be predicted is called dependent variable.

Regression test generates lines of regression of the two variables which helps in estimating the values. Lines of regression of y on x is the line which gives the best estimate for the value of y for any specified value of x. Similarly, line of regression of x on y is the line which gives the best estimate for the value of x for any specified value of y.

Types of Regression

(a) Simple Regression

The regression analysis confined to the study of only two variables at a time is termed as simple regression.

(b) Multiple Regression

The regression analysis for studying more than two variables at a time is termed as multiple regression.

(c) Linear Regression

If the regression curve is a straight line, the regression is termed as linear regression. The equation of such a curve is the equation of a straight line i.e., first degree equation in variables x and y.

(d) Nonlinear Regression

If the curve of the regression is not a straight line, the regression is termed as curved or non-linear regression. The regression equation will be a functional relation between variables x and y involving terms in x and y of degree more than one.

Applications / Utility of Regression Test

Regression lines or equations are useful in the predictions of values of one variable for a specified value of the other variable.

Example

(i) For pharmaceutical firms which are interested in studying the effect of new drugs in patients, regression test helps in such predictions.

(ii) When price and demand are related, we can estimate or predict the future demand for a specified price.

(iii) When crop yield depends on the amount of rainfall, then regression test can predict crop yield for a particular amount of rainfall.

(iv) If advertising expenditure and sales are related, then regression analysis helps in estimating the advertising expenditure for a required amount of sales (or) sales expected for a particular advertising expenditure.

(v) When capital employed and profits earned are related, the test can be used to predict profits for a specified amount of capital invested.

Q5. Explain the limitations of regression analysis.

Ans :

Some of the limitations of regression analysis are as follows :

(i) Regression analysis assumes that linear relationship exists among the related variables. But in the 'area of social sciences, linear relationship may not exist among the related variables.

(ii) When regression analysis is used to evaluate the value of dependent variable based on independent variable, it is assumed that the static conditions of relationship exist between 'them. These statistic conditions do not exist in social sciences, so, this assumption "minimizes the use' of regression analysis in social science.

(iii) The value of dependent variable can be evaluated based on independent variable by using regression analysis but only upto some limits. If the circumstances go beyond the limits, then resists would be inaccurate.

Even though, there are many limitations of regression 'technique, it is still regarded as a very useful statistical tool for estimating or predicting the value of dependent variable.

Q6. Explain about regression equation.

Ans :

Regression is mainly concerned with the estimation of unknown value of one variable from the known value of other variable of the given observations. For doing so, there must be a relation between two variables. This relationship is mathematically expressed in the form of equation known as "Regression Equation " or " Estimating Equation".

The regression equation which states and explains the linear relationship between two variables is known as 'Linear Regression Equation'. Basically, as there are two regression lines, there would be two regression equations i.e.,

1. Regression equation of K on X and
2. Regression equation of X on Y.

The regression equation of Y on X is considered for predicting the value of Y when a specific value of X is given. Whereas the regression equation of X on Y is used for predicting the unknown value of X when a specific value of Y is given.

Formation of Regression Equations

There are two ways of forming regression equations as follows,

(a) Normal equation and

(b) Regression coefficient.

Formation of Regression Equation through Normal Equation

Generally, the situations where perfect linear relationship exists between the two variables X and Y, usually there would be two regression lines and when there are two regression lines, there would be two regression equations as follows,

1. The regression equation of Y on X is denoted as Y. = a + bX.
2. The regression equation of X on Y is denoted as X = a + bY.

In the above equations 'a' and 'b' are two unknown constants which ascertains the positions of the regression line. Therefore, these constants are known as parameters of the regression lines.

The parameter 'a' ascertains the level of a fitted line, whereas 'b' ascertains the slope of the line. Y_C and X_C are the symbols stating and showing the values of Y and X calculated from the relationship for given X or Y.

Regression Equation of Y on X

$Y = a + bX$

By applying the least square principle, the values of 'a' and 'b' are determined in such a way $Y_C = a + bX$ is minimum.

The normal equation for determining the value of a and b are,

$\Sigma y = Na + b\Sigma x$ ...(1)

$\Sigma xy = a\Sigma x + b\Sigma x^2$ ...(2)

Regression Equation of X on Y

$Y_c = a + by$

The normal equation for obtaining the values of a and b are,

$\Sigma x = Na + b\Sigma y$ ...(1)

$\Sigma xy = a\Sigma y + b\Sigma y^2$ ... (2)

After calculating the values of N, Σx, Σy, Σx^2, "Σy^2, substitute them in regression equation Y on X and X on Y for ascertaining the values of a and b. Lastly, by substituting the values of a and b in regression equation, the required best fitting straight line is obtained.

(b) Regression Coefficients

To estimate values of population parameter β_0 and β_1, under certain assumptions, the fitted or estimated regression equation representing the straight line regression model is written as :

$$\hat{y} = a + bx$$

where

y = estimated average (mean) value of dependent variable y for a given value of independent variable x.

a or b_0 = y - intercept that represents average value of $\hat{y}$

b = slope of regression line that represents the expected change in the value of y for unit change in the value of x

To determine the value of $\hat{y}$ for a given value of x, this equation requires the determination of two unknown constants a (intercept) and b (also called regression coefficient). Once these constants are calculated, the regression line can be used to compute an estimated value of the dependent variable y for a given value of independent variable x.

The particular values of *a* and *b* define a specific linear relationship between x and y based on sample data. The coefficient '*a*' represents the level of fitted line (i.e., the distance of the line above or below the origin) when x equals zero, whereas coefficient '*b*' represents the slope of the line (a measure of the change in the estimated value of y for a one-unit change in).

The regression coefficient 'b' is also denoted as :

- b_{yx} (regression coefficient of y on x) in the regression line, y = a + bx
- b_{xy} (regression coefficient of x on y) in the regression line, x = c + dy.

Q6. Discuss briefly about simple regression.

Ans :

Simple regression represents the relationship between two variables where one of them is independent variables 'X' and other variable is dependent variable 'Y'.

The relationship between two variables can be of three types. They are,

(i) Linear Relationship

The graph of linear relationship between two variables looks as follows,

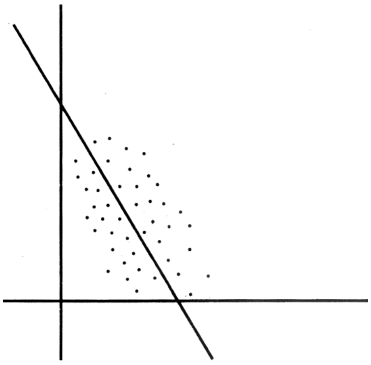


Fig. : Linear Relationship

ii) Non-Linear Relationship

The graph of non-linear relationship between two variables looks as follows,

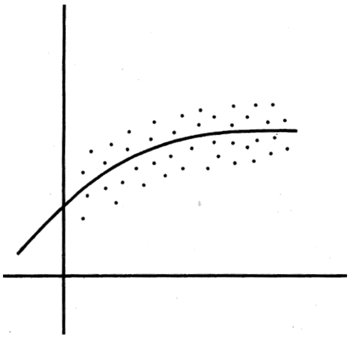


Fig. : Non-linear Relationship

iii) No Relationship

The graph of no relationship between two variables looks as follows,

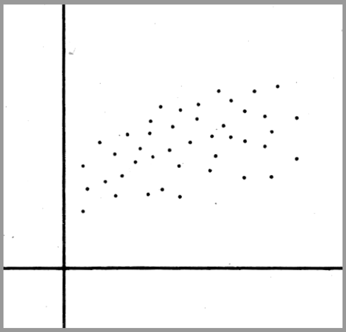


Fig. : No Relationship

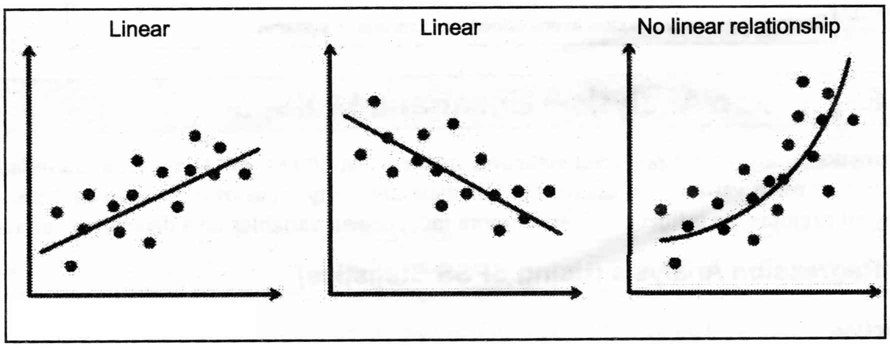
3.2.1 Linear Regression

Q7. Explain the assumptions of simple linear regressions ?

Ans : **(Imp.)**

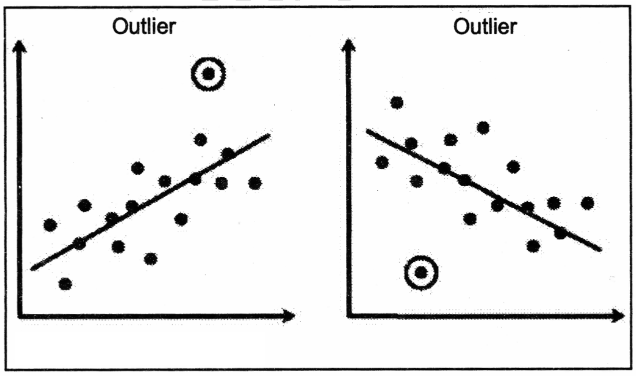
1. Two variables should be measured at the continuous level (i.e., they are either interval or ratio variables). Examples of continuous variables include revision time (measured in hours), intelligence (measured using IQ score), exam performance (measured from 0 to 100), weight (measured in kg), and so forth. You can learn more about interval and ratio variables in our article: Types of Variable.

2. There needs to be a linear relationship between the two variables. Whilst there are a number of ways to check whether a linear relationship exists between your two variables, we suggest creating a scatterplot using SPSS Statistics where you can plot the dependent variable against your independent variable and then visually inspect the scatterplot to check for linearity. Your scatteiplot may look something like one of the following:



If the relationship displayed in your scatterplot is not linear, you will have to either run a non-linear regression analysis, perform a polynomial regression or "transform" your data, which you can do using SPSS Statistics. We show you how to: (a) create a scatterplot to check for linearity when carrying out linear regression using SPSS Statistics; (b) interpret different scatterplot results; and (c) transform your data using SPSS Statistics if there is not a linear relationship between your two variables.

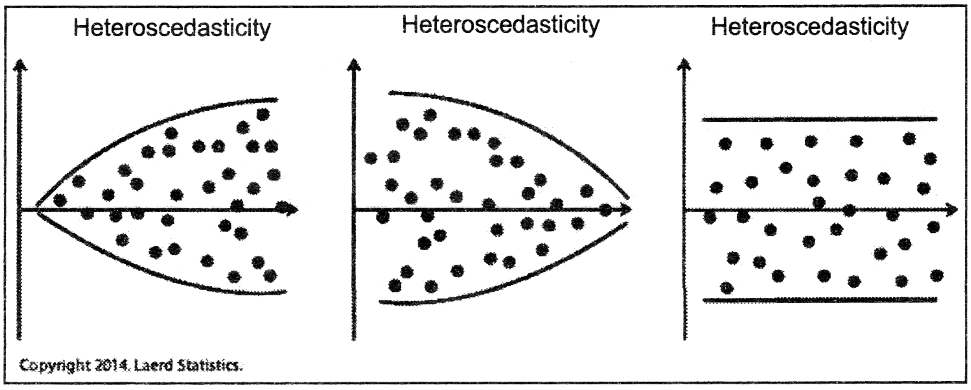
3. There should be no significant outliers. An outlier is an observed data point that has a dependent variable value that is very different to the value predicted by the regression equation. As such, an outlier will be a point on a scatterplot that is (vertically) far away from the regression line indicating that it has a large residual, as highlighted below:



The problem with outliers is that they can have a negative effect on the regression analysis (e.g., reduce the fit of the regression equation) that is used to predict the value of the dependent (outcome) variable based on the independent (predictor) variable. This will change the output that SPSS Statistics produces and reduce the predictive accuracy of your results. Fortunately, when using SPSS Statistics to run a linear regression on your data, you can easily include criteria to help you detect possible outliers. We: (a) show you how to detect outliers using "case-wise diagnostics", which is a simple process when using SPSS Statistics; and (b) discuss some of the options you have in order to deal with outliers.

4. We should have independence of observations, which you can easily check using the Durbin-Watson statistic, which is a simple test to run using SPSS Statistics.

5. Your data needs to show homoscedasticity, which is where the variances along the line of best fit remain similar as you move along the line. Whilst we explain more about what this means and how to assess the homoscedasticity of your data take a look at the three scatterplots below, which provide three simple examples: two of data that fail the assumption (called heteroscedasticity) and one of data that meets this assumption (called homoscedasticity):



Whilst these help to illustrate the differences in data that meets or violates the assumption of homoscedasticity, real-world data can be a lot more messy and illustrate different patterns of heteroscedasticity. Therefore, we explain: (a) some of the things you will need to consider when interpreting your data; and (b) possible ways to continue with your analysis if your data fails to meet this assumption.

6. Finally, you need to check that the residuals (errors) of the regression line are approximately normally distributed Two common methods to check this assumption include using either a histogram (with a superimposed normal curve) or a Normal P-P Plot. Again, we: (a) show you how to check this assumption using SPSS Statistics, whether you use a histogram (with superimposed normal curve) or Normal P-P Plot; (b) explain how to interpret these diagrams; and (c) provide a possible solution if your data fails to meet this assumption.

You can check assumptions #2, #3, #4, #5 and #6 using SPSS Statistics. Assumptions #2 should be checked first, before moving onto assumptions #3, #4, #5 and #6. We suggest testing the assumptions in this order because assumptions #3, #4, #5 and #6 require you to run the linear regression procedure in SPSS Statistics first. So, it is easier to deal with these after checking assumption #2. Just remember that if you do not run the statistical tests on these assumptions correctly, the results you get when running a linear regression might not be valid.

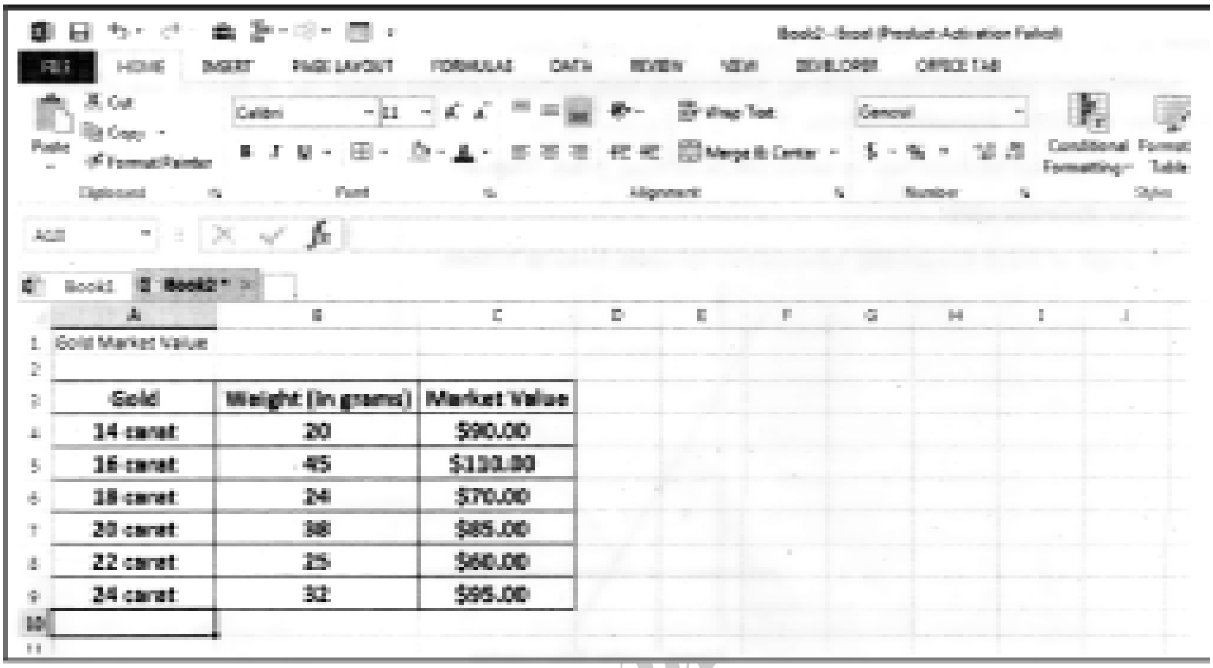
Q8. Explain the concept of simple linear regression by using MS Excel.

Ans : **(Imp.)**

In Microsoft Excel, the information regarding statistical properties of regression analysis are provided by the software tools of regression analysis. The regression tool can be used not only for simple regression but, also for multiple regression.

The steps to be followed for generating regression analysis output are as follows,

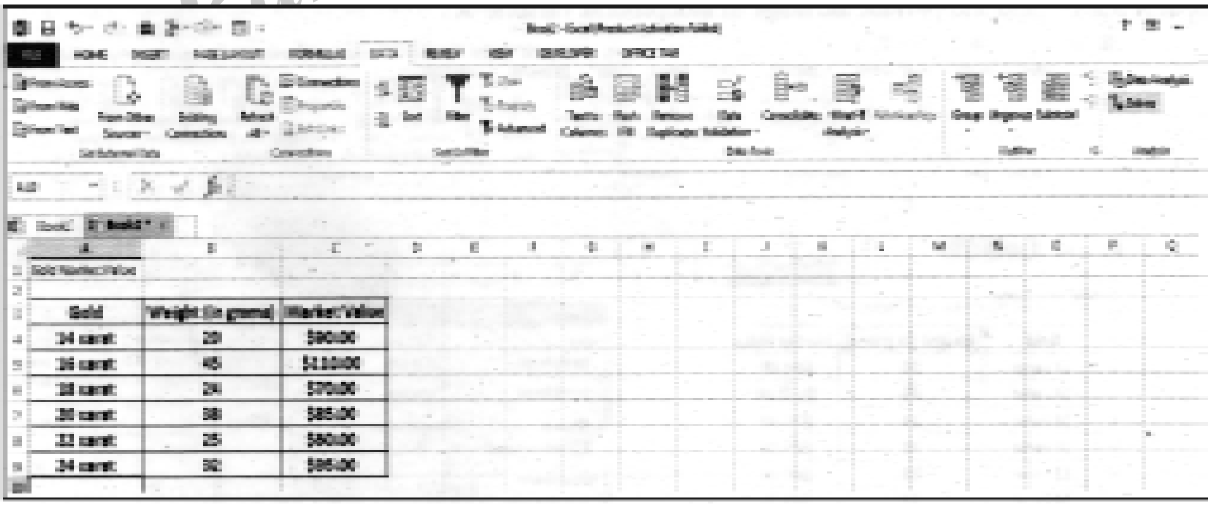
1. Select the data wherein user want to apply regression.



Gold Market Value

Gold	Weight (in grams)	Market Value
14 carat	20	$90.00
16 carat	45	$110.00
18 carat	24	$70.00
20 carat	38	$85.00
22 carat	25	$60.00
24 carat	32	$95.00

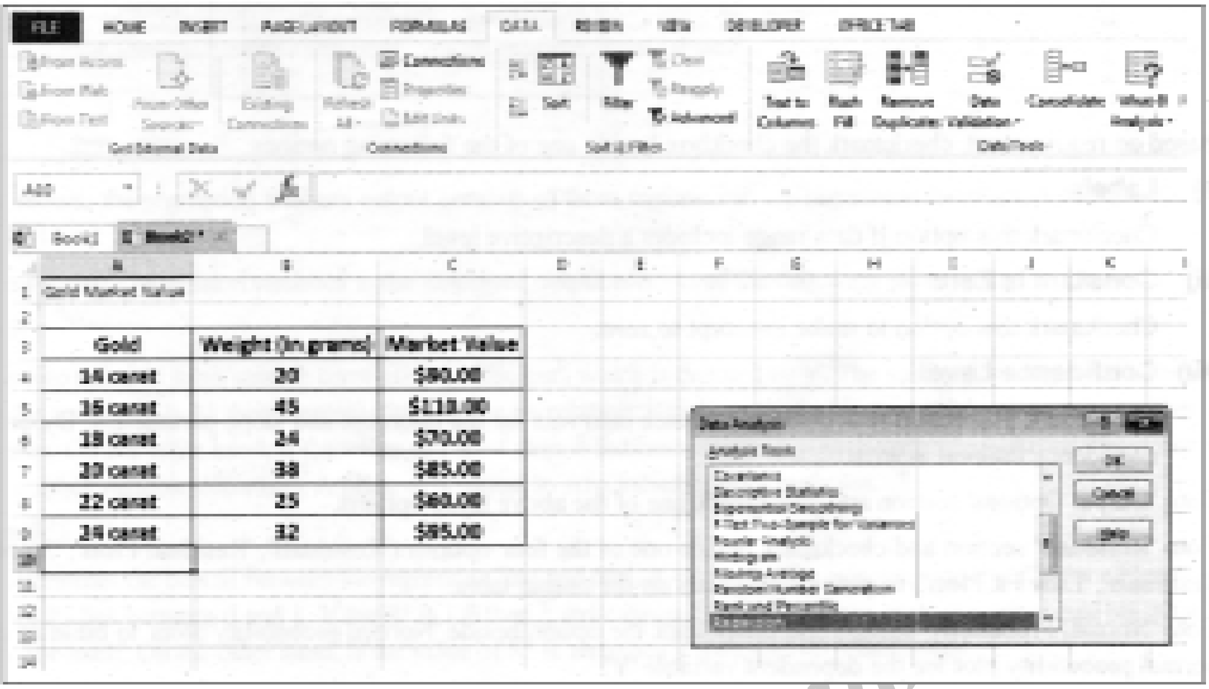
2. Click on 'Data' tab present on excel ribbon and the click 'Data Analysis' command present under' Analysis' group.



Gold Market Value

Gold	Weight (in grams)	Market Value
14 carat	20	$90.00
16 carat	45	$110.00
18 carat	24	$70.00
20 carat	38	$85.00
22 carat	25	$60.00
24 carat	32	$95.00

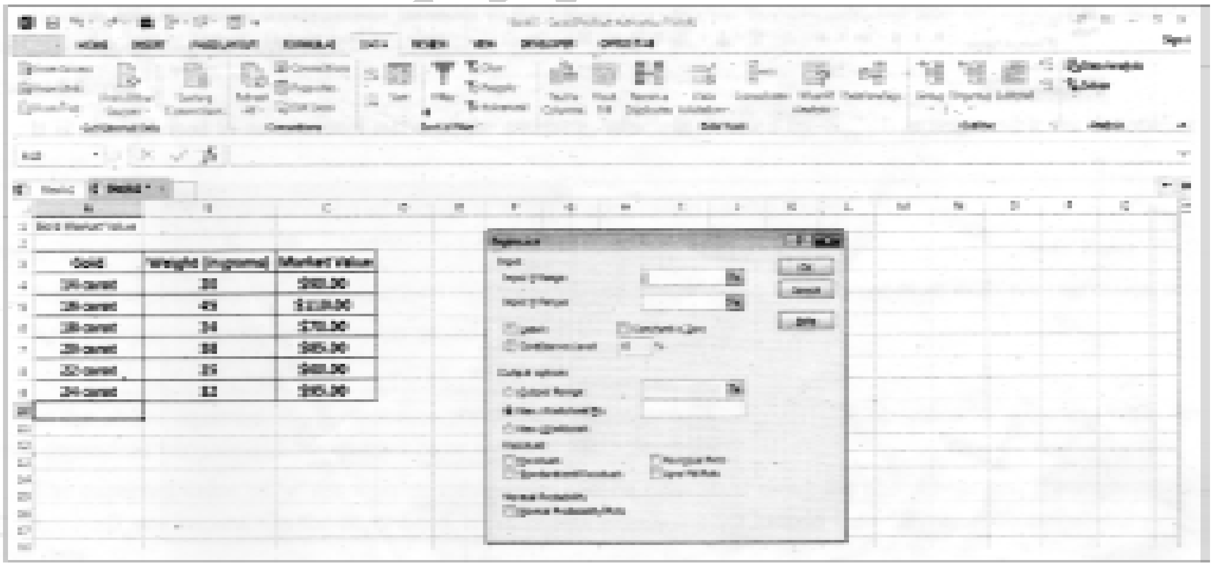
3. As a result, 'Data Analysis' dialog box will be displayed. Goto 'Analysis Tools' section and select 'Regression' option from the menu list and then click "OK" button.



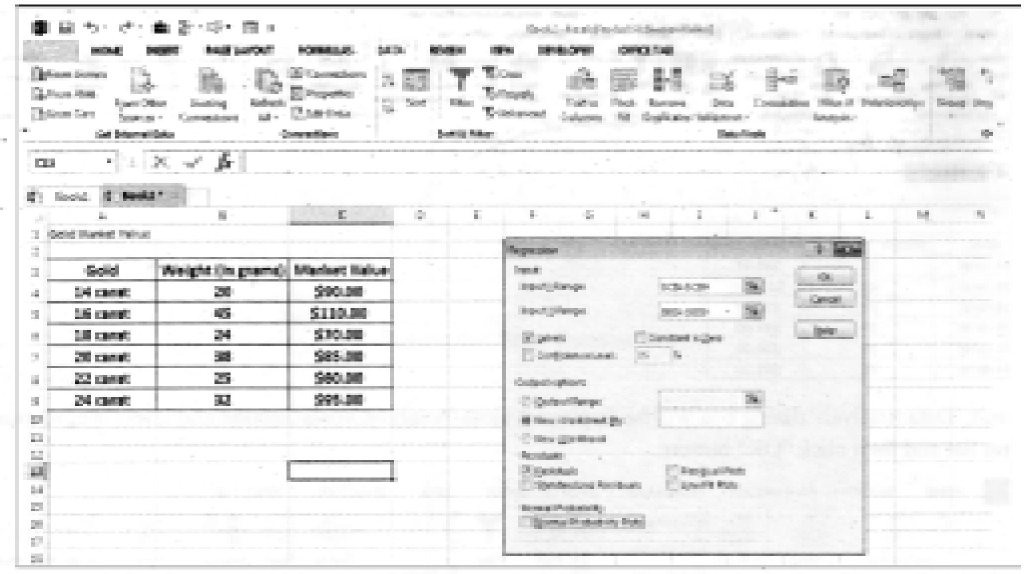
Gold Market Value

Gold	Weight (in grams)	Market Value
14 carat	20	$90.00
16 carat	45	$110.00
18 carat	24	$70.00
20 carat	38	$85.00
22 carat	25	$60.00
24 carat	32	$95.00

4. As a result, ‘Regression’ window appears on screen.



5. In the ‘Regression’ dialog box, goto ‘Input Y Range’ field and provide the range of dependent variable ‘Y’. Similarly, Goto ‘Input X Range’ field and provide the range of independent variable ‘X’.



Gold Market Value

Gold	Weight (in grams)	Market Value
14 carat	20	$90.00
16 carat	45	$110.00
18 carat	24	$70.00
20 carat	38	$85.00
22 carat	25	$60.00
24 carat	32	$95.00

6. Based on requirement, checkmark the checkbox beside one of the following options.

   i) Labels : Checkmark this option if data range includes a descriptive level.

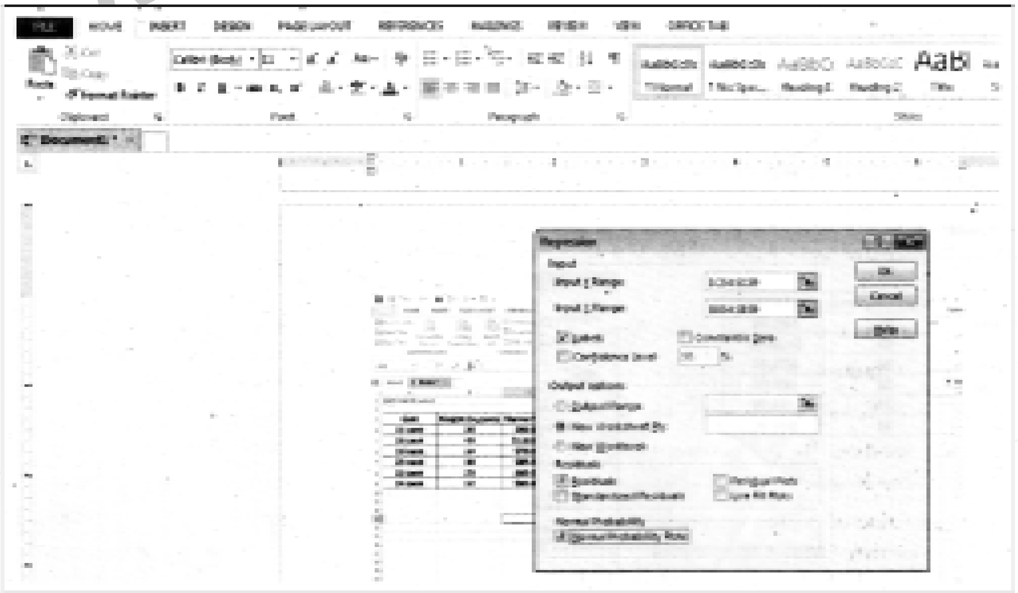
   ii) Constant is Zero : Checkmark this option to make intercept to zero.

   iii) Confidence Level : Checkmark this option to include confidence intervals for the intercept and slope parameters. By default, the confidence interval is 95%.

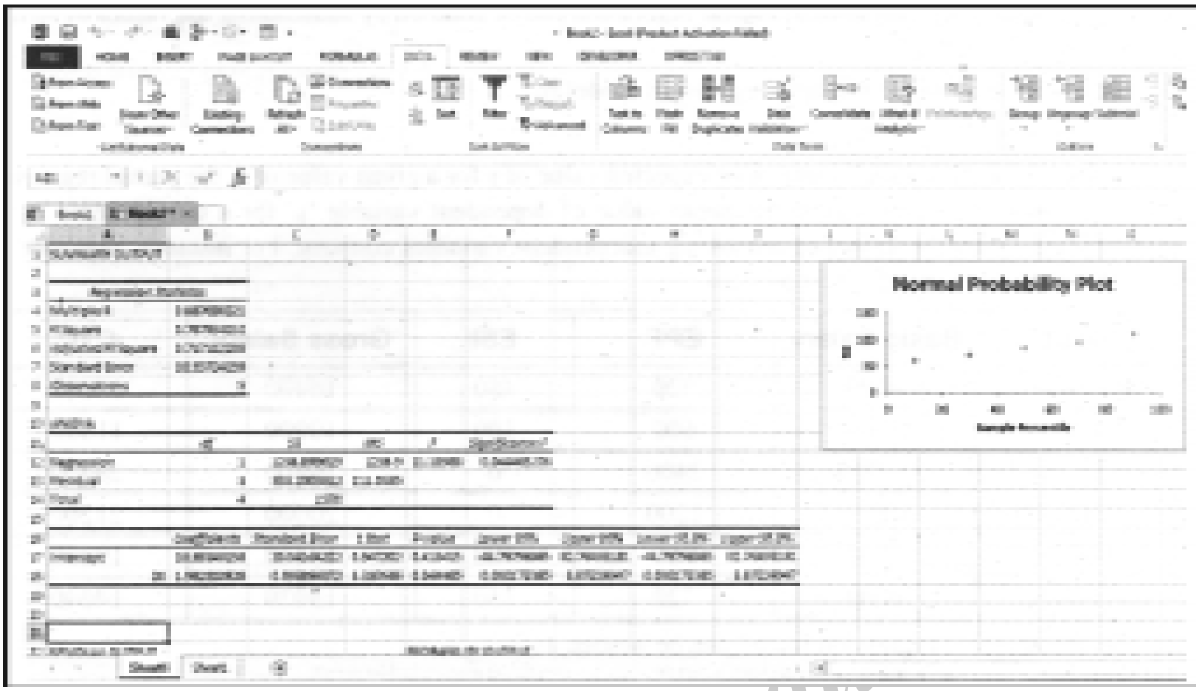
7. Goto 'Output Options' section and checkmark one of the above three options.

8. Goto 'Residuals' section and checkmark beside one of the four options ('Residuals', 'Residual Plots', 'Standardized Residuals', 'Line Fit Plots') to provide residuals on the output table.

9. Goto 'Normal Probability' section and checkmark the option beside 'Normal probability plots' to build or construct normal probability plot for the dependent variable 'Y'.



10. Click on “OK” button. As a result, the regression analysis output will be displayed on the screen.



SUMMARY OUTPUT								
Regression Statistics								
Multiple R	[illegible]							
R Square	[illegible]							
Adjusted R Square	[illegible]							
Standard Error	[illegible]							
Observations	5							
ANOVA								
	df	SS	MS	F	Significance F			
Regression	1	[illegible]	[illegible]	[illegible]	[illegible]			
Residual	3	[illegible]	[illegible]					
Total	4	[illegible]						
	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

As shown, the regression analysis output consists of three regions namely regression statistics, Annova and unlabelled section.

The region of regression statistics in the displayed output consists of the following parameters,

(i) Multiple R

It is also referred to as ‘sample correlation coefficient’, which is denoted by ‘r’. The value of multiple R’ lies between - 1 and +1. If the value of r is +1 then it represents positive correlation, which means that if one variable increases another variable also increases. On the other hand, if the value of r is -1 then it indicates negative correlation, which means that if one variable decreases another variable decreases. A value of ‘r’ equal to zero indicates no correlation.

(ii) R-Square(R^2)

It determines the best fit between the regression line and data. R-sqaure is also referred to as ‘coefficient of determination’. The value of R^2 lies between 0 and 1. If the R^2 is 1.0 then it indicates perfect fit where in each and every data point falls on the regression line itself. On the other hand, if the value of R^2 is zero then it indicates no relationship.

(iii) Adjusted R Square

It refers to a statistical measure that includes in the model not only the sample size, but also the number of independent variables for modifying the value of R^2.

(iv) Standard Error

It is also referred to as ‘standard error’ of the estimate, which is denoted by ‘S_{YX}’. It is responsible for describing the variability in ‘Y’.

3.2.2 Multiple Regression

Q9. Define multiple regression ? Explain its assumptions.

Ans :

Meaning

Multiple regression also allows you to determine the overall fit (variance explained) of the model and me relative contribution of each of the predictors to the total variance explained. For example, you might want to know how much of the variation in exam performance can be explained by revision time, test anxiety, lecture attendance and gender "as a whole", but also the "relative contribution" of each independent variable in explaining the variance.

Assumptions

1. Your dependent variable should be measured on a continuous scale (i.e., it is either an interval or ratio variable). Examples of variables that meet this criterion include revision time (measured in hours), intelligence (measured using IQ score), exam performance (measured from 0 to 100), weight (measured in kg), and so forth. You can learn more about interval and ratio variables in our article: Types of Variable. If your dependent variable was measured on an ordinal scale, you will need to carry out ordinal regression rather than multiple regression.

   Examples of ordinal variables include Likert items (e.g., a 7-point scale from "strongly agree" through to "strongly disagree"), amongst other ways of ranking categories (e.g., a 3-point scale explaining how much a customer liked a product, ranging from "Not very much" to "Yes, a lot"). You can access our SPSS Statistics guide on ordinal regression here.

2. You have two or more independent variables which can be either continuous (i.e., an interval or ratio variable) or categorical (i.e., an ordinal or nominal variable). For examples of continuous and ordinal variables, see the bullet above.

   Examples of nominal variables include gender (e.g., 2 groups: male and female), ethnicity (e.g., 3 groups: Caucasian, African American and Hispanic), physical activity level (e.g., 4 groups: sedentary, low, moderate and high), profession (e.g., 5 groups: surgeon, doctor, nurse, dentist, therapist), and so forth. Again, you can learn more about variables in our article: Types of Variable. If one of your independent variables is dichotomous and considered a moderating variable, you might need to run a Dichotomous moderator analysis.

3. You should have independence of observations (i.e., independence of residuals), which you can easily check using the Durbin-Watson statistic, which is a simple test to run using SPSS Statistics. We explain how to interpret the result of the Durbin-Watson statistic, as well as showing you the SPSS Statistics procedure required, in our enhanced multiple regression guide.

4. There needs to be a linear relationship between: (a) the dependent variable and each of your independent variables, and (b) the dependent variable and the independent variables collectively. Whilst there are a number of ways to check for these linear relationships, we suggest creating scatterplots and partial regression plots using SPSS Statistics, and then visually inspecting these scatterplots and partial regression plots to check for linearity.

   If the relationship displayed in your scatterplots and partial regression plots are not linear, you will have to either run a non-linear regression analysis or "transform" your data, which you can do using SPSS Statistics. In our enhanced multiple regression guide, we show you how to: (a) create scatterplots and partial regression plots to check for linearity when carrying out multiple regression using SPSS Statistics; (b) interpret different scatterplot and partial regression plot results; and (c) transform your data using SPSS Statistics if you do not have linear relationships between your variables.

5. Your data needs to show homoscedasticity, which is where the variances along the line of best fit remain similar as you move along the line. We explain more about what this means and how to assess the homoscedasticity of your data in our enhanced multiple regression guide. When you analyze your own data, you will need to plot the studentized residuals against the unstandardized predicted values. In our enhanced multiple regression guide, we explain: (a) how to test for homoscedasticity using SPSS Statistics; (b) some of the things you will need to consider when interpreting your data; and (c) possible ways to continue with your analysis if your data fails to meet this assumption.

6. Your data must not show multicollinearity, which occurs when you have two or more independent variables that are highly correlated with each other. This leads to problems with understanding which independent variable contributes to the variance explained in the dependent variable, as well as technical issues in calculating a multiple regression model.

   Therefore, in our enhanced multiple regression guide, we show you: (a) how to use SPSS Statistics to detect for multicollinearity through an inspection of correlation coefficients and Tolerance/VIF values; and (b) how to interpret these correlation coefficients and Tolerance/VIF values so that you can determine whether your data meets or violates this assumption.

7. There should be no significant outliers, high leverage points or highly influential points. Outliers, leverage and influential points are different terms used to represent observations in your data set that are in some way unusual when you wish to perform a multiple regression analysis. These different classifications of unusual points reflect the different impact they have on the regression line. An observation can be classified as more than one type of unusual point.

   However, all these points can have a very negative effect on the regression equation that is used to predict the value of the dependent variable based on the independent variables. This can change the output that SPSS Statistics produces and reduce the predictive accuracy of your results as well as the statistical significance. Fortunately, when using SPSS Statistics to run multiple regression on your data you can detect possible outliers, high leverage points and highly influential points.

   In our enhanced multiple regression guide, we: (a) show you how to detect outliers using "case- wise diagnostics" and "studentized deleted residuals", which you can do using SPSS Statistics, and discuss some of the options you have in order to deal with outliers; (b) check for leverage points using SPSS Statistics and discuss what you should do if you have any; and (c) check for influential points in SPSS Statistics using a measure of influence known as Cook's Distance, before presenting some practical approaches in SPSS Statistics to deal with any influential points you might have.

8. Finally, you need to check that the residuals (errors) are approximately normally distributed (we explain these terms in our enhanced multiple regression guide). Two common methods to check this assumption include using: (a) a histogram (with a superimposed normal curve) and a Normal P-P Plot; or (b) a Normal Q-Q Plot of the studentized residuals.

   Again, in our enhanced multiple regression guide, we: (a) show you how to check this assumption using SPSS Statistics, whether you use a histogram (with superimposed normal curve) and Normal P-P Plot, or Normal Q-Q Plot; (b) explain how to interpret these diagrams; and (c) provide a possible solution if your data fails to meet this assumption.

Q10. Discuss briefly the concept of multiple regression using excel.

Ans :

The regression analysis for studying more than two variables at a time is termed as multiple regression.

A linear regression model that contains multiple independent variables is referred to as multiple regression model. An equation for multiple regression model is given by,

$$\boxed{y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \in}$$

In the above equation, β_0, β_1 specifies population parameters, X_1, X_2, ... X_p specifies independent-variables, Y defines dependent variable and '$\in$' defines error term.

The expected value of 'y' for a given value of V can be calculated using the above equation if parameter values of β_0, β_1, . . ., β_q are known. On the other hand, if parameter values are not known then they must be calculated using the sample data.

The estimated regression equation for multiple linear regression can be attained by substituting the values of sample statistics b_0, b_1, ... , b_p in β_0, β_1, ... , β_p.

The estimated regression equation in multiple regression model is,

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + ... + b_p x_p$$

In the above equation, y refers to point estimator of expected value of y for a given value of x, the partial regression coefficients b_0, b_1, ... ,b_p indicates the change in the mean value of dependent variable 'y' for a unit increase in the independent variables, while holding the values of remaining independent variables constant. For instance, consider the following excel file containing salary details of employees.

Employee	Dept	Basic Salary	EPF	ESI	Gross Salary	CTC
Divya	IT	8000	920	480	16400	17800
Sushanth	CSE	5000	600	300	10900	11800
Keerthi	ECE	12000	2400	0	26400	28400
Jyoshna	MECH	10000	1200	0	20000	21200
Praveen	ECE	8500	960	480	18440	19880
Anusha	EE	6000	720	350	13070	14040

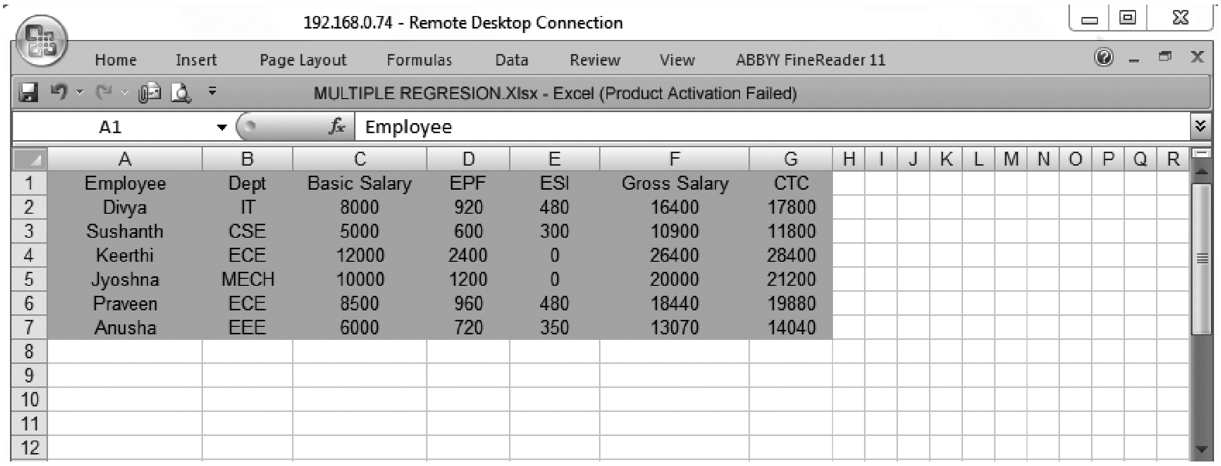
In the above table, the multiple regression model can be written as,

$$\text{CTC} = b_0 + b_x \text{ Basic Salary} + b_2 \text{ EPF} + b_3 \text{ ESI} + b_4 \text{ Gross Salary}$$

Therefore, b, indicates the change in the mean value of CTC for a unit increase in the associated independent variable 'EPF' while holding all remaining independent variables 'Basic Salary', 'EPF', 'ESI' and 'Gross Salary Constant like simple linear regression, multiple linear regression also follows the least squares technique for estimating both intercept and slope coefficients.

The steps to be followed for generating regression analysis output in case of multiple linear regression are given below,

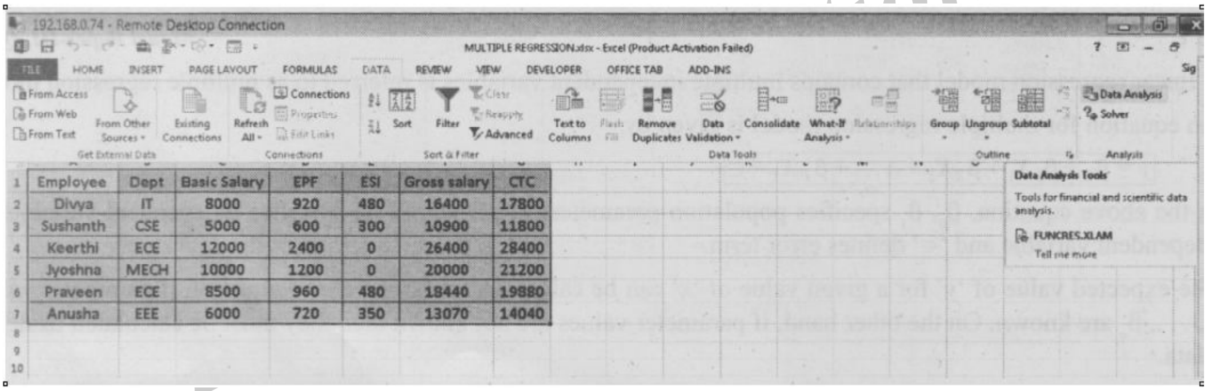
1. Select the data wherein user want to apply regression.



192.168.0.74 - Remote Desktop Connection
Home
Insert
Page Layout
Formulas
Data
Review
View
ABBYY FineReader 11
MULTIPLE REGRESION.Xlsx - Excel (Product Activation Failed)
A1
Employee

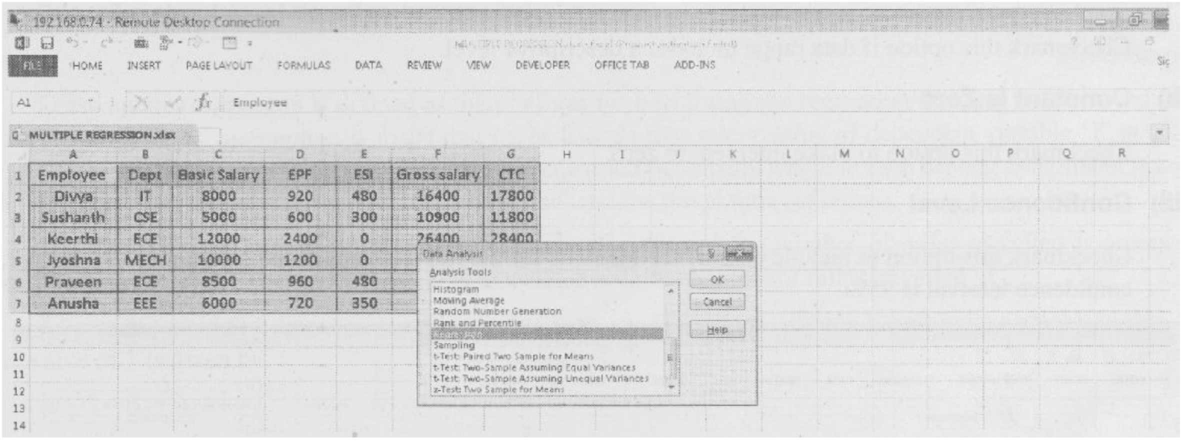
	A	B	C	D	E	F	G
1	Employee	Dept	Basic Salary	EPF	ESI	Gross Salary	CTC
2	Divya	IT	8000	920	480	16400	17800
3	Sushanth	CSE	5000	600	300	10900	11800
4	Keerthi	ECE	12000	2400	0	26400	28400
5	Jyoshna	MECH	10000	1200	0	20000	21200
6	Praveen	ECE	8500	960	480	18440	19880
7	Anusha	EEE	6000	720	350	13070	14040

2. Click on 'Data' tab present on excel ribbon and the click 'Data Analysis' command present under 'Analysis' group.



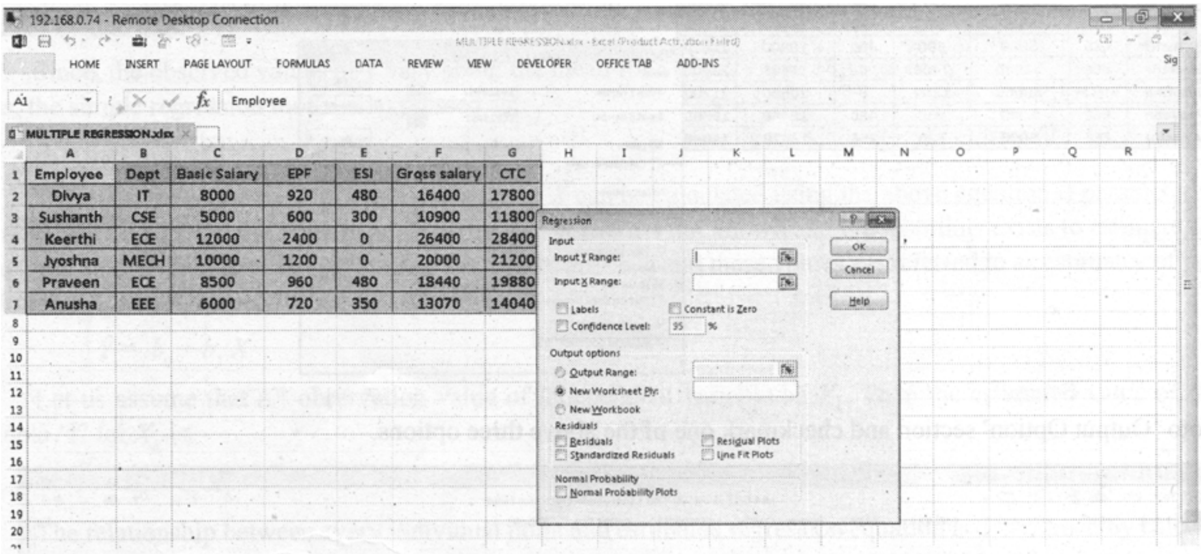
1	Employee	Dept	Basic Salary	EPF	ESI	Gross salary	CTC
2	Divya	IT	8000	920	480	16400	17800
3	Sushanth	CSE	5000	600	300	10900	11800
4	Keerthi	ECE	12000	2400	0	26400	28400
5	Jyoshna	MECH	10000	1200	0	20000	21200
6	Praveen	ECE	8500	960	480	18440	19880
7	Anusha	EEE	6000	720	350	13070	14040

3. As a result, 'Data Analysis' dialog box will be displayed. Goto 'Analysis Tools' section and select 'Regression' option from the menu list and then click "OK" button.



	A	B	C	D	E	F	G
1	Employee	Dept	Basic Salary	EPF	ESI	Gross salary	CTC
2	Divya	IT	8000	920	480	16400	17800
3	Sushanth	CSE	5000	600	300	10900	11800
4	Keerthi	ECE	12000	2400	0	26400	28400
5	Jyoshna	MECH	10000	1200	0		
6	Praveen	ECE	8500	960	480		
7	Anusha	EEE	6000	720	350		

4. As a result, 'Regression' window appears on screen.


192.168.0.74 - Remote Desktop Connection
HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER OFFICE TAB ADD-INS
MULTIPLE REGRESSION.xlsx
OK
Cancel
Help

A1 fx Employee

	A	B	C	D	E	F	G
1	Employee	Dept	Basic Salary	EPF	ESI	Gross salary	CTC
2	Divya	IT	8000	920	480	16400	17800
3	Sushanth	CSE	5000	600	300	10900	11800
4	Keerthi	ECE	12000	2400	0	26400	28400
5	Jyoshna	MECH	10000	1200	0	20000	21200
6	Praveen	ECE	8500	960	480	18440	19880
7	Anusha	EEE	6000	720	350	13070	14040

Regression

Input

Input Y Range:

Input X Range:

□ Labels □ Constant is Zero

□ Confidence Level: 95 %

Output options

○ Output Range:

● New Worksheet Ply:

○ New Workbook

Residuals

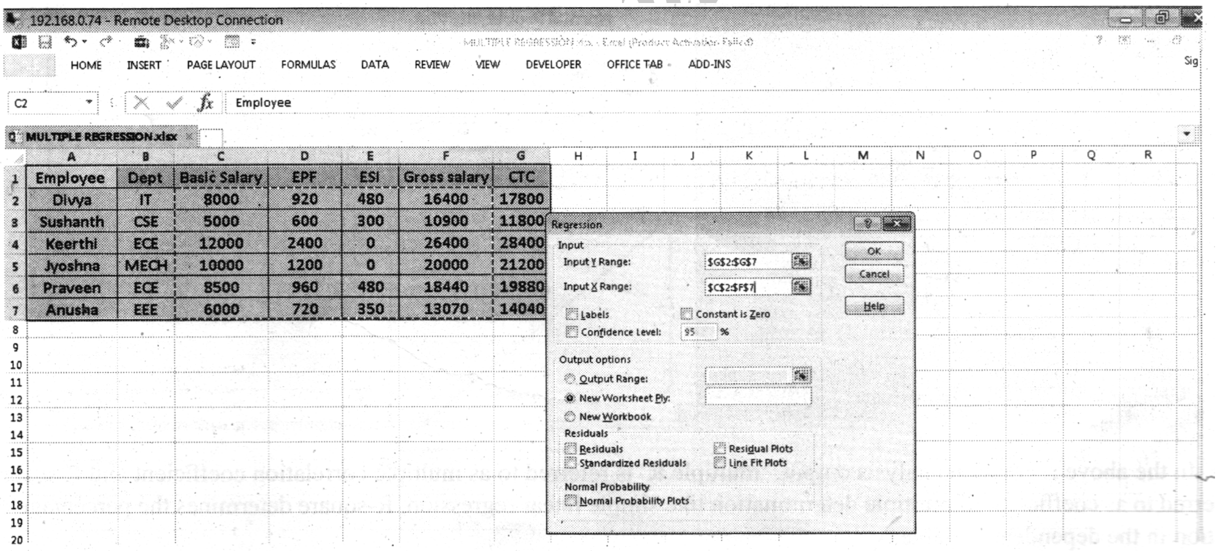
□ Residuals □ Residual Plots

□ Standardized Residuals □ Line Fit Plots

Normal Probability

□ Normal Probability Plots

5. In the 'Regression', dialog box, Goto 'Input Y Range' field and provide the range of dependent variable Y. Similarly, Goto 'Input X Range' field and provide the entire range of independent variable ˡJC.


192.168.0.74 - Remote Desktop Connection
HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER OFFICE TAB ADD-INS
MULTIPLE REGRESSION.xlsx
OK
Cancel
Help

C2 fx Employee

	A	B	C	D	E	F	G
1	Employee	Dept	Basic Salary	EPF	ESI	Gross salary	CTC
2	Divya	IT	8000	920	480	16400	17800
3	Sushanth	CSE	5000	600	300	10900	11800
4	Keerthi	ECE	12000	2400	0	26400	28400
5	Jyoshna	MECH	10000	1200	0	20000	21200
6	Praveen	ECE	8500	960	480	18440	19880
7	Anusha	EEE	6000	720	350	13070	14040

Regression

Input

Input Y Range: $G$2:$G$7

Input X Range: $C$2:$F$7

□ Labels □ Constant is Zero

□ Confidence Level: 95 %

Output options

○ Output Range:

● New Worksheet Ply:

○ New Workbook

Residuals

□ Residuals □ Residual Plots

□ Standardized Residuals □ Line Fit Plots

Normal Probability

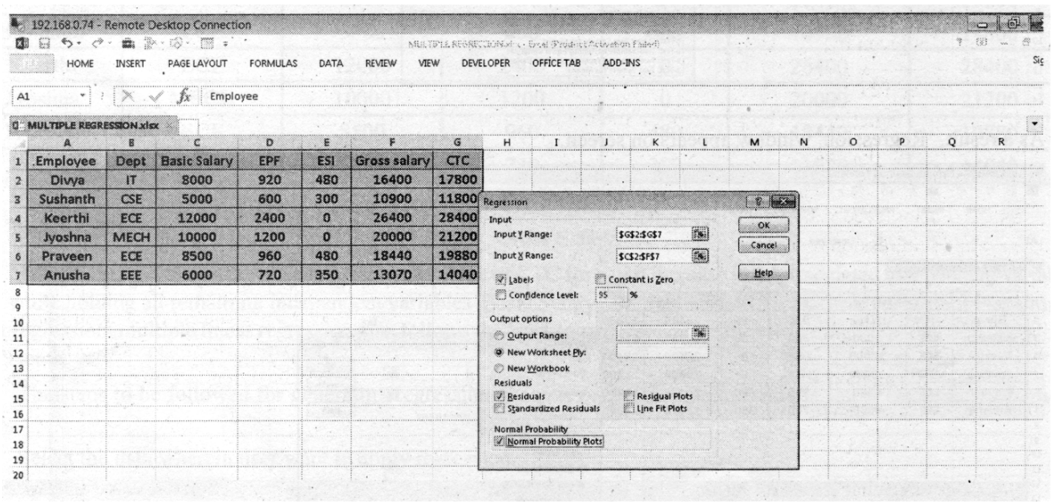
□ Normal Probability Plots

6. Based on requirement, checkmark the checkbox beside one of the following options.

(i) Labels: Checkmark this option if data range includes a descriptive level.

(ii) Constant is Zero: Checkmark this option to make intercept to zero.

(iii) Confidence Level: Checkmark this option to include confidence intervals for the intercept and slope parameters. By default, the confidence interval is 95%.



192.168.0.74 - Remote Desktop Connection
MULTIPLE REGRESSION.xlsx - Excel (Product Activation Failed)
HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER OFFICE TAB ADD-INS
A1
MULTIPLE REGRESSION.xlsx

Employee

	A	B	C	D	E	F	G
1	Employee	Dept	Basic Salary	EPF	ESI	Gross salary	CTC
2	Divya	IT	8000	920	480	16400	17800
3	Sushanth	CSE	5000	600	300	10900	11800
4	Keerthi	ECE	12000	2400	0	26400	28400
5	Jyoshna	MECH	10000	1200	0	20000	21200
6	Praveen	ECE	8500	960	480	18440	19880
7	Anusha	EEE	6000	720	350	13070	14040

Regression

Input

Input Y Range: $G$2:$G$7

Input X Range: $C$2:$F$7

☑ Labels ☐ Constant is Zero

☐ Confidence Level: 95 %

Output options

○ Output Range:

● New Worksheet Ply:

○ New Workbook

Residuals

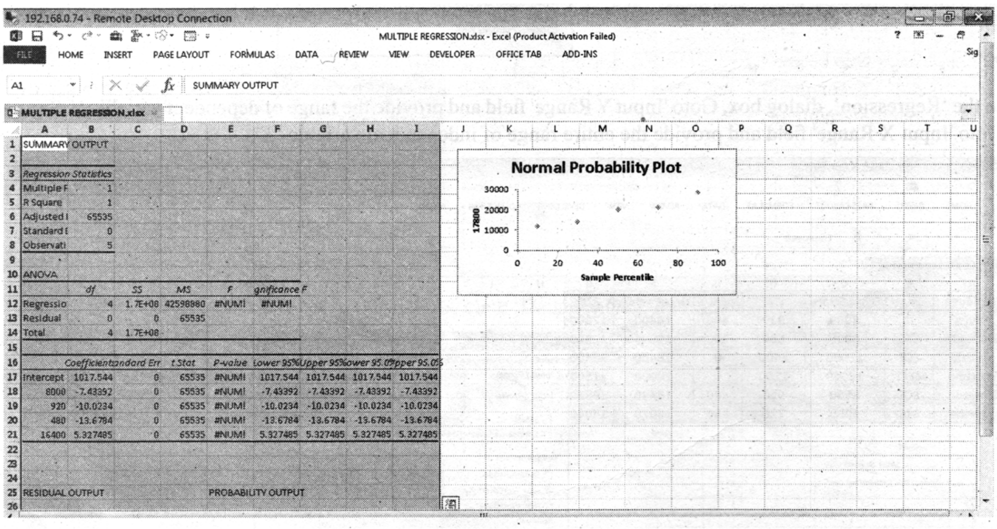
☑ Residuals ☐ Residual Plots

☐ Standardized Residuals ☐ Line Fit Plots

Normal Probability

☑ Normal Probability Plots

7. Goto 'Output Option' section and checkmark one of the above three options.



192.168.0.74 - Remote Desktop Connection
MULTIPLE REGRESSION.xlsx - Excel (Product Activation Failed)
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER OFFICE TAB ADD-INS
A1
MULTIPLE REGRESSION.xlsx
Normal Probability Plot
30000
20000
10000
0
17800
0
20
40
60
80
100
Sample Percentile

SUMMARY OUTPUT

	A	B	C	D	E	F	G	H	I
1	SUMMARY OUTPUT								
2									
3	Regression Statistics								
4	Multiple F	1							
5	R Square	1							
6	Adjusted I	65535							
7	Standard E	0							
8	Observati	5							
9									
10	ANOVA								
11		df	SS	MS	F	gnificance F			
12	Regressio	4	1.7E+08	42598980	#NUM!	#NUM!			
13	Residual	0	0	65535					
14	Total	4	1.7E+08						
15									
16		Coefficientsandard Err		t Stat	P-value	Lower 95%	Upper 95%	ower 95.0%	pper 95.0%
17	Intercept	1017.544	0	65535	#NUM!	1017.544	1017.544	1017.544	1017.544
18	8000	-7.43392	0	65535	#NUM!	-7.43392	-7.43392	-7.43392	-7.43392
19	920	-10.0234	0	65535	#NUM!	-10.0234	-10.0234	-10.0234	-10.0234
20	480	-13.6784	0	65535	#NUM!	-13.6784	-13.6784	-13.6784	-13.6784
21	16400	5.327485	0	65535	#NUM!	5.327485	5.327485	5.327485	5.327485
22									
23									
24									
25	RESIDUAL OUTPUT				PROBABILITY OUTPUT				
26									

In the above regression analysis output, 'multiple R' is referred to as multiple correlation coefficient and R square is referred to as coefficient of multiple determination like simple linear regression, R-square determines the percentage of variation in the dependent variable.

3.3 Predictive Modeling

Q11. What is predictive modeling? Explain different types of Predictive Modeling.

Ans : **(Imp.)**

Meaning

Predictive modelling is a statistical technique using machine learning and data mining to predict and forecast likely future outcomes with the aid of historical and existing data. It works by analysing current and historical data and projecting what it learns on a model generated to forecast likely outcomes. Predictive modelling can be used to predict just about anything, from TV ratings and a customer's next purchase to credit risks and corporate earnings.

A predictive model is not fixed; it is validated or revised regularly to incorporate changes in the underlying data. In other words, it's not a one-and-done prediction. Predictive models make assumptions based on what has happened in the past and what is happening now. If incoming, new data shows changes in what is happening now, the impact on the likely future outcome must be recalculated, too. For example, a software company could model historical sales data against marketing expenditures across multiple regions to create a model for future revenue based on the impact of the marketing spend.

Most predictive models work fast and often complete their calculations in real time. That's why banks and retailers can, for example, calculate the risk of an online mortgage or credit card application and accept or decline the request almost instantly based on that prediction.

Types

1. **Classification model:** It categorizes data for simple and direct query response.
2. **Clustering model:** This model nests data together by common attributes. It works by grouping things or people with shared characteristics or behaviours and plans strategies for each group at a larger scale. An example is in determining credit risk for a loan applicant based on what other people in the same or a similar situation did in the past.
3. **Forecast model:** This is a very popular model, and it works on anything with a numerical value based on learning from historical data. For example, in answering how much lettuce a restaurant should order next week or how many calls a customer support agent should be able to handle per day or week, the system looks back to historical data.
4. **Outliers model:** This model works by analyzing abnormal or outlying data points. For example, a bank might use an outlier model to identify fraud by asking whether a transaction is outside of the customer's normal buying habits or whether an expense in a given category is normal or not.
5. **Time series model:** This model evaluates a sequence of data points based on time. For example, the number of stroke patients admitted to the hospital in the last four months is used to predict how many patients the hospital might expect to admit next week, next month or the rest of the year. A single metric measured and compared over time is thus more meaningful than a simple average.

Q12. Explain various common predictive algorithms.

Ans :

Some of the more common predictive algorithms are:

(i) Random Forest

This algorithm is derived from a combination of decision trees, none of which are related, and can use both classification and regression to classify vast amounts of data.

(ii) Generalized Linear Model (GLM) for Two Values

This algorithm narrows down the list of variables to find "best fit." It can work out tipping points and change data capture and other influences, such as categorical predictors, to determine the "best fit" outcome, thereby overcoming drawbacks in other models, such as a regular linear regression.

(iii) Gradient Boosted Model

This algorithm also uses several combined decision trees, but unlike Random Forest, the trees are related. It builds out one tree at a time, thus enabling the next tree to correct flaws in the previous tree. It's often used in rankings, such as on search engine outputs.

(iv) K-Means

A popular and fast algorithm, K-Means groups data points by similarities and so is often used for the clustering model. It can quickly render things like personalized retail offers to individuals within a huge group, such as a million or more customers with a similar liking of lined red wool coats.

(v) Prophet

This algorithm is used in time-series or forecast models for capacity planning, such as for inventory needs, sales quotas and resource allocations. It is highly flexible and can easily accommodate heuristics and an array of useful assumptions.

3.4 Forecasting Techniques

Q13. Explain briefly about Forecasting techniques.

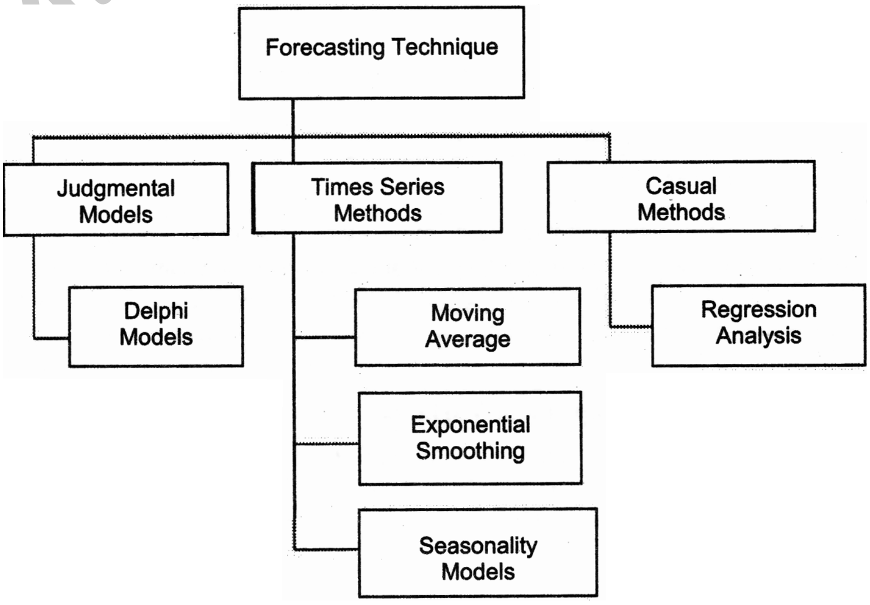
Ans : **(Imp.)**

Forecasting Analytics is a Quantitative forecasting, which focuses on data for generating numerical forecasts, is an important component of decision making in a wide range of areas and across many business functions, including economic forecasting, workload projections, sales forecasts and transportation demand.

Predictive modeling is a process that uses data mining and probability to forecast outcomes. Each model is made up of a number of predictors, which are variables that are likely to influence future results. Once data has been collected for relevant predictors, a statistical model is formulated.

Forecasting is the process of making predictions of the future based on past and present data and most commonly by analysis of trends. A commonplace example might be estimation of some variable of interest at some specified future date. Prediction is a similar, but more general term. Both might refer to formal statistical methods employing time series, cross-sectional or longitudinal data, or alternatively to less formal judgmental methods. Usage can differ between areas of application: for example, in hydrology the terms "forecast" and "forecasting" are sometimes reserved for estimates of values at certain specific future times, while the term "prediction" is used for more general estimates, such as the number of times floods will occur over a long period.

Risk and uncertainty are central to fore-casting and prediction; it is generally considered good practice to indicate the degree of uncertainty attaching to forecasts. In any case, the data must be up to date in order for the forecast to be as accurate as possible. In some cases the data used to predict the variable of interest is itself forecasted.



The forecasting method you select is a function of multiple qualities about your item. Is demand steady, cyclical or sporadic? Are there seasonal trends? Are trends strong or limited? Is the item new? Each item being forecast has a somewhat unique history (and future), and therefore an optimal method. A method that accurately forecasts one data set might prove inaccurate for another.

Determining the optimal forecast method is a rather complex science, especially across a large product line. This may be nearly impossible using only spreadsheets. However, sophisticated forecasting software can within seconds test multiple methods for each item to determine which method will give you the most accurate results.

Specific Forecasting Methods

1. Moving Averages
2. Exponential Smoothing
3. Regression Analysis Models
4. Hybrid Forecasting Methods

1. Moving Averages

Moving average methods take the average of past actuals and project it forward. These methods assume that the recent past represents the future. As a result, they work best for products with relatively little change — steady demand, no seasonality, limited trends or cycles and no significant demand shifts. Many companies apply this method because it is simple and easy to use. However, since few products actually behave in this way, it tends to be less useful than more specialized methods.

2. Exponential Smoothing

Exponential smoothing is a more advanced form of time series forecasting. Unlike moving averages, exponential smoothing methods can capture trends and recurring patterns. They accomplish this by:

- Emphasizing the more recent data (as opposed to a moving average which weights all data equally), and
- Smoothing out fluctuations, which are often caused by pure randomness in the data (or “noise” in the system).

Forecasters determine the forecast weights, controlling how fast or slow the model responds to demand changes in your actuals. Not all exponential smoothing methods can handle seasonality or other recurring patterns.

Exponential smoothing forecasting methods include:

(i) Simple exponential smoothing

(ii) Holt’s linear method

(iii) Winters’ multiplicative season

(iv) Winters’ additive season

3. Regression Analysis Models

Many companies use regression models to determine the relationship between demand and demand drivers. They are especially useful for seeing trends and seasonality.

Regression analysis methods include:

(i) Linear regression

(ii) Hyperbolic trend

(iii) Logarithmic trend

(iv) Square root trend

(v) Quadratic trend

4. Hybrid Forecasting Methods

Hybrid forecasting methods combine regression, data smoothing and other techniques to produce forecasts that can compensate for the weaknesses of individual methods. For example, some forecasting methods are great at short-term forecasting, but cannot capture seasonality.

Hybrid forecasting methods include:

- Vanguard Dampened Trend - a powerful hybrid model that simultaneously detects all trends, cycles and seasonality in historical data and responds with the most accurate exponential smoothing method. Vanguard Dampened Trend is available across all Vanguard business forecasting applications
- Log Theta
- Theta

Causal / Econometric Forecasting Methods

Some forecasting methods try to identify the underlying factors that might influence the variable that is being forecast. For example, including information about climate patterns might improve the ability of a model to predict umbrella sales. Forecasting models often take account of regular seasonal variations. In addition to climate, such variations can also be due to holidays and customs: for example, one might predict that sales of college football apparel will be higher during the football season than during the off season.

Several informal methods used in causal forecasting do not rely solely on the output of mathematical algorithms, but instead use the judgment of the forecaster. Some forecasts take account of past relationships between variables: if one variable has, for example, been approximately linearly related to another for a long period of time, it may be appropriate to extrapolate such a relationship into the future, without necessarily understanding the reasons for the relationship.

Causal methods include

- Regression analysis includes a large group of methods for predicting future values of a variable using information about other variables. These methods include both parametric (linear or non-linear) and non-parametric techniques.
- Autoregressive moving average with exogenous inputs

Judgmental Methods

Judgmental forecasting methods incorporate intuitive judgement, opinions and subjective probability estimates. Judgmental forecasting is used in cases where there is lack of historical data or during completely new and unique market conditions. Artificial intelligence methods

- Artificial neural networks
- Group method of data handling
- Support vector machines

Often these are done today by specialized programs loosely labelled :

- Data mining
- machine learning
- Pattern recognition

Other methods

- Simulation
- Prediction market
- Probabilistic forecasting and Ensemble forecasting.

3.5 Data Mining

3.5.1 Definition of Datamining

Q14. Explain briefly about data mining.

Ans : **(Imp.)**

Definition

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use.

- Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD.Aside from the raw analysis step, it also involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.
- The difference between data analysis and data mining is that data analysis is to summarize the history such as analyzing the effectiveness of a marketing campaign, in contrast, data mining focuses on using specific machine learning and statistical models to predict the future and discover the patterns among data.
- The term "data mining" is in fact a misnomer, because the goal is the extraction of patterns and knowledge from large amounts of data,

not the extraction (mining) of data itself. It also is a buzzword and is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence (e.g., machine learning) and business intelligence.

- The book Data mining: Practical machine learning tools and techniques with Java (which covers mostly machine learning material) was originally to be named just Practical machine learning, and the term data mining was only added for marketing reasons. Often the more general terms (large scale) data analysis and analytics – or, when referring to actual methods, artificial intelligence and machine learning – are more appropriate.

Q15. Explain the steps involved in data mining.

Ans :

Steps

1. Problem Definition

A data mining project starts with the understanding of the business problem. Data mining experts, business experts, and domain experts work closely together to define the project objectives and the requirements from a business perspective. The project objective is then translated into a data mining problem definition.

In the problem definition phase, data mining tools are not yet required.

2. Data Exploration

Domain experts understand the meaning of the metadata. They collect, describe, and explore the data. They also identify quality problems of the data. A frequent exchange with the data mining experts and the business experts from the problem definition phase is vital.

In the data exploration phase, traditional data analysis tools, for example, statistics, are used to explore the data.

3. Data Preparation

Domain experts build the data model for the modeling process. They collect, cleanse, and format the data because some of the mining functions accept data only in a certain format. They also create new derived attributes, for example, an average value.

In the data preparation phase, data is tweaked multiple times in no prescribed order. Preparing the data for the modeling tool by selecting tables, records, and attributes, are typical tasks in this phase. The meaning of the data is not changed.

4. Modeling

Data mining experts select and apply various mining functions because you can use different mining functions for the same type of data mining problem. Some of the mining functions require specific data types. The data mining experts must assess each model.

In the modeling phase, a frequent exchange with the domain experts from the data preparation phase is required.

The modeling phase and the evaluation phase are coupled. They can be repeated several times to change parameters until optimal values are achieved. When the final modeling phase is completed, a model of high quality has been built.

Evaluation : Data mining experts evaluate the model. If the model does not satisfy their expectations, they go back to the modeling phase and rebuild the model by changing its parameters until optimal values are achieved. When they are finally satisfied with the model, they can extract business explanations and evaluate the following questions:

- Does the model achieve the business objective?
- Have all business issues been considered?

At the end of the evaluation phase, the data mining experts decide how to use the data mining results.

5. **Deployment**

Data mining experts use the mining results by exporting the results into database tables or into other applications, for example, spreadsheets.

The Intelligent Miner products assist you to follow this process. You can apply the functions of the Intelligent Miner products independently, iteratively, or in combination.

The following figure shows the phases of the Cross Industry Standard Process for data mining (CRISP DM) process model.

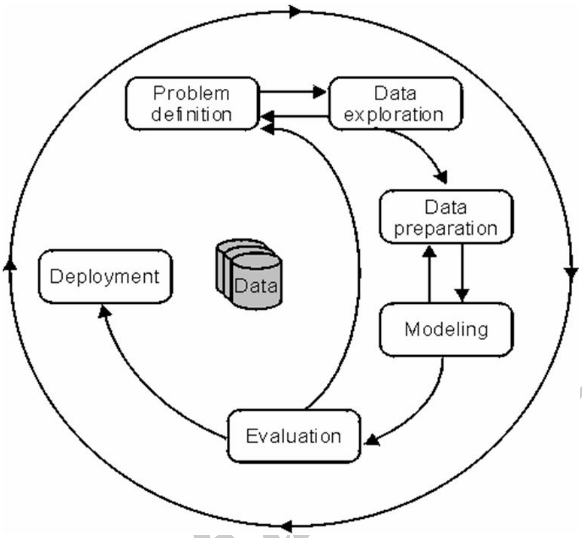


Fig. : The CRISP DM process model

IM Modeling helps you to select the input data, explore the data, transform the data, and mine the data. With IM Visualization you can display the data mining results to analyze and interpret them. With IM Scoring, you can apply the model that you have created with IM Modeling.

Q16. Explain the scope of data mining ?

Ans :

1. Data mining process the work in such a manner that it allows business to more proactive to grow substantially.
2. It optimizes large database within the short time and works business intelligence which is more vital to organizational growth.
3. It represents the data in some logical order or may be in pattern to identify the sequential way of processing of data.
4. It includes tree-shaped structure to understand the hierarchy of data and representation of the set of information described in the database.
5. Brings a genetic way of classification of different sets of data items to view the data in the quick glance.

Q17. Explain the various techniques are used in data mining.

Ans : **(Imp.)**

1. **Classification**

This analysis is used to retrieve important and relevant information about data and metadata. This data mining method helps to classify data in different classes.

2. **Clustering**

Clustering analysis is a data mining technique to identify data that are like each other. This process helps to understand the differences and similarities between the data.

3. **Regression**

Regression analysis is the data mining method of identifying and analyzing the relationship between variables. It is used to identify the likelihood of a specific variable, given the presence of other variables.

4. **Association Rules**

This data mining technique helps to find the association between two or more items. It discovers a hidden pattern in the data set.

5. **Outer Detection**

This type of data mining technique refers to observation of data items in the data set which do not match an expected pattern or expected behavior. This technique can be used in a variety of domains, such as intrusion, detection, fraud or fault detection, etc. Outer detection is also called Outlier Analysis or Outlier Mining.

6. Sequential Patterns

This data mining technique helps to discover or identify similar patterns or trends in transaction data for certain period.

7. Prediction

Prediction has used a combination of the other data mining techniques like trends, sequential patterns, clustering, classification, etc. It analyzes past events or instances in a right sequence for predicting a future event.

Q18. Explain merits and demerits of data mining.

Ans :

Advantages

- Data mining technique helps companies to get knowledge-based information.
- It helps organizations to make the profitable adjustments in operation and production.
- It is a cost-effective and efficient solution compared to other statistical data applications.
- It helps with the decision-making process.
- It facilitates automated prediction of trends and behaviors as well as automated discovery of hidden patterns.
- It can be implemented in new systems as well as existing platforms.
- It is the speedy process which makes it easy for the users to analyze huge amount of data in less time.

Disadvantages

- There are chances of companies selling useful information of their customers to other companies for money. For example, American Express has sold credit card purchases of their customers to the other companies.
- Many data mining analytics software is difficult to operate and requires advance training to work on.
- Different data mining tools work in different manners due to different algorithms employed in their design. Therefore, the selection of correct data mining tool is a very difficult task.
- The data mining techniques are not accurate. Hence, it can cause serious consequences in certain conditions.

3.5.2 Approaches in Data Mining

Q19. Explain the various approaches for data mining with Micro Strategy.

Ans : **(Imp.)**

(i) Scoring the database: Records are scored in batches and saved as tables or columns.

(ii) Database does the scoring: The database scores records in response to queries.

(iii) MicroStrategy does the scoring: MicroStrategy scores records using metrics and reports.

While MicroStrategy supports all three approaches, each has positive and negative aspects. The next sections describe each approach in detail.

(i) Scoring the Database

In this approach, records are scored and inserted into the database either as new tables or as new columns in existing tables. Most often, a third-party scoring engine receives a result set and scores the records. Then the scores are added to the database. Once they are part of the database, MicroStrategy attributes or metrics can reference those scores, just like any other data in the database. Historically, this approach has been the most common. Its pros and cons are described below.

Pros

- Since an external scoring engine performs the scoring calculation, model complexity and performance is hidden within the scoring engine. Thus, the scoring process does not require any database resources and does not impact other business intelligence work.
- At run time, data is simply read from the database without having to calculate

the score on the fly. Scoring on the fly can slow analysis especially if millions of scores are involved.

- MicroStrategy can use this approach by just creating metrics or attributes for the scored data.

Cons

- This approach requires database space and the support of a database administrator.
- New records that are inserted after the batch scoring are not scored.
- Updating the model or scores requires more database and database administrator overhead.
- In many companies, adding or updating information in the enterprise data warehouse is not done easily or whenever desired. The cross functional effort required to score the database limits the frequency of scoring and prevents the vast majority of users from trying new models or changing existing ones.

This approach is really no different than adding other entities to a MicroStrategy project. For more information, see the Project Design Guide.

(ii) Database does the scoring

In this approach, data mining features of the database system are used to perform the scoring. Nearly all major databases have the ability to score data mining models. The most common approach persists the model in the database and then generates scores by using extensions to the SQL queries processed by the database to invoke the model. A key feature of this approach is that the model can be scored in a system that is different from the data mining tool that developed the model.

- The model can be saved in the database as a Predictive Model Markup Language (PMML) object, or, less frequently, in some form of executable code. For more information on PMML, see PMML overview.
- Persisting the model in this way is possible since the sophisticated algorithms needed to create the model are not required to score them. Scoring simply involves mathematical calculations on a set of inputs to generate a result.
- The ability to represent the model and score it outside of the model creation tool is relatively new, but more companies are adopting this approach. Its advantages and disadvantages are described below.

Pros

- Scores can be calculated on the fly even if new records are added.
- Updating the model is easier than in the Score the database option.
- This approach requires less database space than the score the database option.
- When the database supports accessing its data mining features via SQL, MicrovStrategy can take advantage of this approach using its SQL Engine.

Cons

- This approach requires support from a database administrator and application knowledge of the database's data mining tool. However, the database administrator usually does not have this knowledge.
- The database data mining tool is typically an additional cost.

(iii) MicroStrategy does the scoring

- In this approach, predictive models are applied from within the Business Intelligence platform environment, without requiring support from the database and from database administrators to implement data mining models. This direct approach reduces the time required, the potential for data inconsistencies, and cross-departmental dependencies.

- MicroStrategy Data Mining Services uses enterprise data resources without significantly increasing the overhead. MicroStrategy Data Mining Services allows sophisticated data mining techniques to be applied directly within the business intelligence environment. Just as the other approaches, it also has advantages and disadvantages, as described below:

Pros

- MicroStrategy stores the predictive model in its metadata as a predictive metric that can be used just like any other metric.
- Scores can be done on the fly even if new records are added.
- The predictive model can be viewed in MicroStrategy Developer.
- The predictive model is easily updated using MicroStrategy Developer.
- This approach does not require database space or support from a database administrator.
- MicroStrategy can take advantage of this approach by using the Analytical Engine.

Cons

- This approach does not take advantage of the database data mining features.
- Predictor inputs need to be passed from the database to Intelligence Server. For large result sets, databases typically handle data operations more efficiently than moving data to MicroStrategy and scoring it there.

3.6 Data Exploration and Reduction

Q20. What is data exploration ? Explain the Steps of Data Exploration and Preparation.

Ans : **(Imp.)**

Meaning

Data exploration is an informative search used by data consumers to form true analysis from the information gathered. Often, data is gathered in a non-rigid or controlled manner in large bulks. For true analysis, this unorganized bulk of data needs to be narrowed down. This is where data exploration is used to analyze the data and information from the data to form further analysis.

Data often converges in a central warehouse called a data warehouse. This data can come from various sources using various formats. Relevant data is needed for tasks such as statistical reporting, trend spotting and pattern spotting. Data exploration is the process of gathering such relevant data.

Steps of Data Exploration and Preparation

Remember the quality of your inputs decide the quality of your output. So, once you have got your business hypothesis ready, it makes sense to spend lot of time and efforts here. With my personal estimate, data exploration, cleaning and preparation can take up to 70% of your total project time.

Below are the steps involved to understand, clean and prepare your data for building your predictive model:

1. Variable Identification
2. Univariate Analysis
3. Bi-variate Analysis
4. Missing values treatment
5. Outlier treatment
6. Variable transformation
7. Variable creation

Finally, we will need to iterate over steps 4 – 7 multiple times before we come up with our refined model.

1. Variable Identification

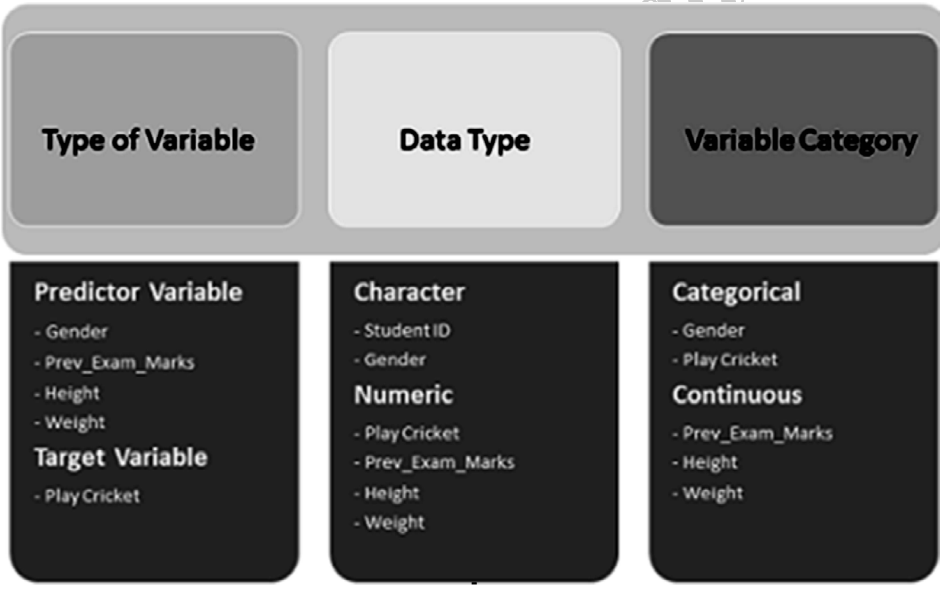
First, identify Predictor (Input) and Target (output) variables. Next, identify the data type and category of the variables.

Let's understand this step more clearly by taking an example.

Example

Suppose, we want to predict, whether the students will play cricket or not (refer below data set). Here you need to identify predictor variables, target variable, data type of variables and category of variables. Below, the variables have been defined in different category:

Student ID	Gender	Prev Exam Marks	Height (cm)	Weight Caregory (kgs)	Play Cricket
S001	M	65	178	61	1
S002	F	75	174	56	0
S003	M	45	163	62	1
S004	M	57	175	70	0
S005	F	59	162	67	0



Type of Variable

Predictor Variable
- Gender
- Prev_Exam_Marks
- Height
- Weight

Target Variable
- Play Cricket

Data Type

Character
- Student ID
- Gender

Numeric
- Play Cricket
- Prev_Exam_Marks
- Height
- Weight

Variable Category

Categorical
- Gender
- Play Cricket

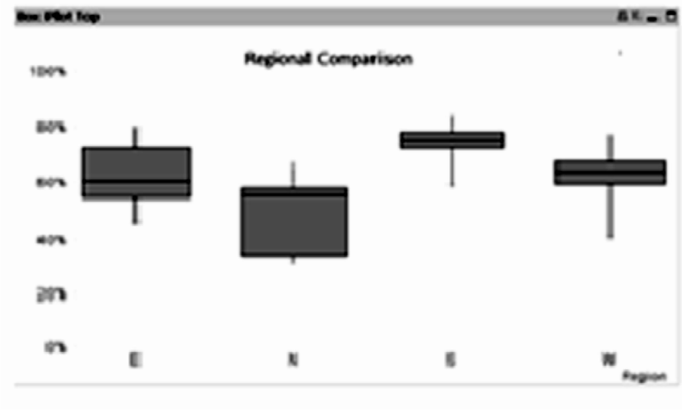
Continuous
- Prev_Exam_Marks
- Height
- Weight

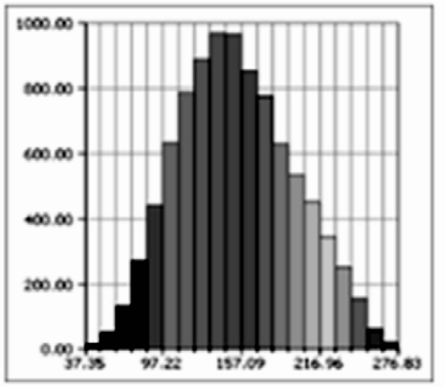
2. Univariate Analysis

At this stage, we explore variables one by one. Method to perform uni-variate analysis will depend on whether the variable type is categorical or continuous. Let's look at these methods and statistical measures for categorical and continuous variables individually:

(i) Continuous Variables: In case of continuous variables, we need to understand the central tendency and spread of the variable. These are measured using various statistical metrics visualization methods as shown below:

Central Tendency	Measure of Dispersion	Visualization Methods
Mean	Range	Histogram
Median	Quartile	Box Plot
Mode	IQR	
Min	Variance	
Max	Standard Deviation	
	Skewness and Kurtosis	





Note: Univariate analysis is also used to highlight missing and outlier values. In the upcoming part of this series, we will look at methods to handle missing and outlier values. To know more about these methods, you can refer course descriptive statistics from Udacity.

(ii) Categorical Variables: For categorical variables, we'll use frequency table to understand distribution of each category. We can also read as percentage of values under each category. It can be be measured using two metrics, Count and Count% against each category. Bar chart can be used as visualization.

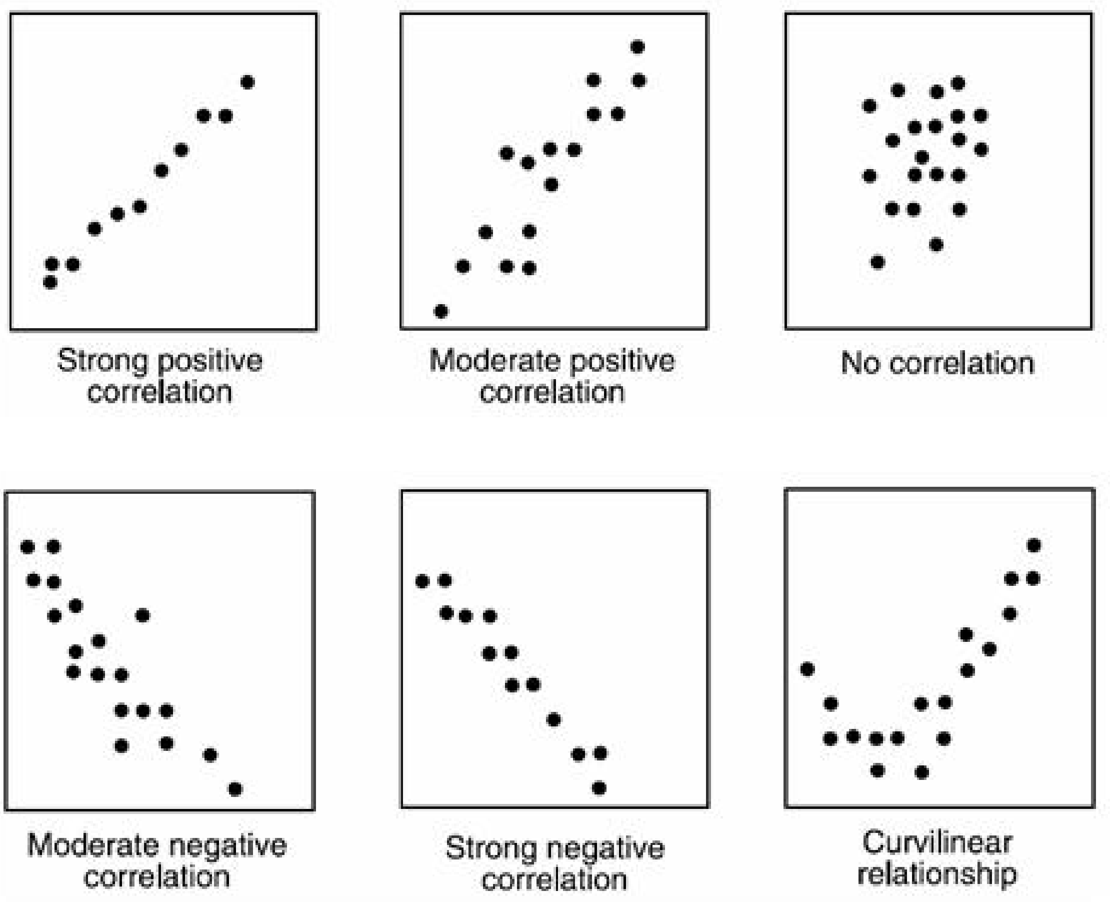
Bi-variate Analysis

Bi-variate Analysis finds out the relationship between two variables. Here, we look for association and disassociation between variables at a pre-defined significance level. We can perform bi-variate analysis for any combination of categorical and continuous variables. The combination can be: Categorical & Categorical, Categorical & Continuous and Continuous & Continuous. Different methods are used to tackle these combinations during analysis process.

Let's understand the possible combinations in detail:

Continuous and Continuous

While doing bi-variate analysis between two continuous variables, we should look at scatter plot. It is a nifty way to find out the relationship between two variables. The pattern of scatter plot indicates the relationship between variables. The relationship can be linear or non-linear.



Strong positive correlation

Moderate positive correlation

No correlation

Moderate negative correlation

Strong negative correlation

Curvilinear relationship

Scatter plot shows the relationship between two variable but does not indicates the strength of relationship amongst them. To find the strength of the relationship, we use Correlation. Correlation varies between -1 and +1.

- –1: perfect negative linear correlation
- +1:perfect positive linear correlation and
- 0: No correlation

Correlation can be derived using following formula:

Correlation = Covariance(X,Y) / SQRT(Var(X)* Var(Y))

Various tools have function or functionality to identify correlation between variables. In Excel, function CORREL() is used to return the correlation between two variables and SAS uses procedure PROC CORR to identify the correlation. These function returns Pearson Correlation value to identify the relationship between two variables:

X	65	72	78	65	72	70	65	68
Y	72	69	79	69	84	75	60	73

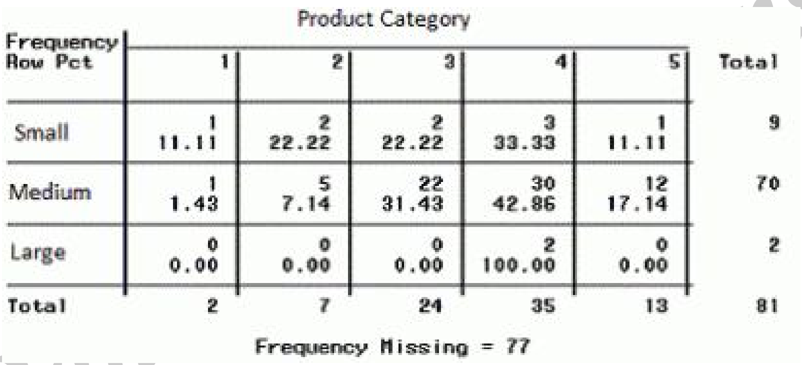
Metrics	Formula	Value
Co-Variance (X,Y)	= COVAR(E6:L6,E7:L7)	18.77
Variance (X)	= VAR.P(E6:L6)	18.48
Variance (Y)	= VAR.P(E7:L7)	45.23
Correlation	= G10/SQRT(G11*G12)	0.65

In above example, we have good positive relationship(0.65) between two variables X and Y.

Categorical and Categorical

To find the relationship between two categorical variables, we can use following methods:

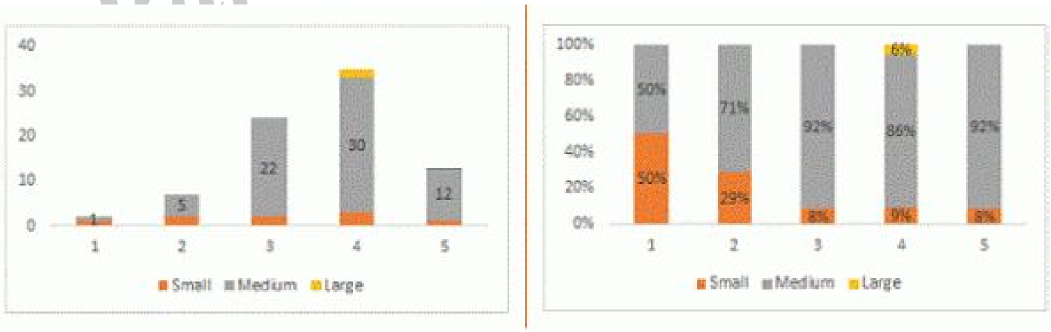
- **Two-way table:** We can start analyzing the relationship by creating a two-way table of count and count%. The rows represents the category of one variable and the columns represent the categories of the other variable. We show count or count% of observations available in each combination of row and column categories.
- **Stacked Column Chart:** This method is more of a visual form of Two-way table.



Product Category

Frequency Row Pct	1	2	3	4	5	Total
Small	1 11.11	2 22.22	2 22.22	3 33.33	1 11.11	9
Medium	1 1.43	5 7.14	22 31.43	30 42.86	12 17.14	70
Large	0 0.00	0 0.00	0 0.00	2 100.00	0 0.00	2
Total	2	7	24	35	13	81

Frequency Missing = 77



Chi-Square Test

This test is used to derive the statistical significance of relationship between the variables. Also, it tests whether the evidence in the sample is strong enough to generalize that the relationship for a larger population as well. Chi-square is based on the difference between the expected and observed frequencies in one or more categories in the two-way table. It returns probability for the computed chi-square distribution with the degree of freedom.

Probability of 0: It indicates that both categorical variable are dependent

Probability of 1: It shows that both variables are independent.

Probability less than 0.05: It indicates that the relationship between the variables is significant at 95% confidence. The chi-square test statistic for a test of independence of two categorical variables is found by:

$X^2 = \Sigma(O - E)^2 / E$ where O represents the observed frequency. E is the expected frequency under the null hypothesis and computed by:

$$E = \frac{\text{row total} \times \text{column total}}{\text{Sample size}}$$

From previous two-way table, the expected count for product category 1 to be of small size is 0.22. It is derived by taking the row total for Size (9) times the column total for Product category (2) then dividing by the sample size (81). This is procedure is conducted for each cell. Statistical Measures used to analyze the power of relationship are:

- Cramer's V for Nominal Categorical Variable
- Mantel-Haenszed Chi-Square for ordinal categorical variable.

Different data science language and tools have specific methods to perform chi-square test. In SAS, we can use Chisq as an option with Procfreq to perform this test.

Categorical and Continuous

While exploring relation between categorical and continuous variables, we can draw box plots for each level of categorical variables. If levels are small in number, it will not show the statistical significance. To look at the statistical significance we can perform Z-test, T-test or ANOVA.

Z-Test/ T-Test

Either test assess whether mean of two groups are statistically different from each other or not. If the probability of Z is small then the difference of two averages is more significant. The T-test is very similar to Z-test but it is used when number of observation for both categories is less than 30.

$$Z = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{S^2\left(\frac{1}{N_1} + \frac{1}{N_2}\right)}}$$

$$S^2 = \frac{(N_1 - 1)S_1^2 + (N_2 - 1)S_3^2}{N_1 + N_2 - 2}$$

Where

- $\bar{X}_1, \bar{X}_{2\text{: Averages}}$
- $S_1^2, S^2_{2\text{:Variances}}$
- $N_1, N_{2\text{: Counts}}$
- t : has t distribution with $N_1 + N_2 - 2$ degree of freedom

ANOVA

It assesses whether the average of more than two groups is statistically different.

Example

Suppose, we want to test the effect of five different exercises. For this, we recruit 20 men and assign one type of exercise to 4 men (5 groups). Their weights are recorded after a few weeks. We need to find out whether the effect of these exercises on them is significantly different or not. This can be done by comparing the weights of the 5 groups of 4 men each.

Till here, we have understood the first three stages of Data Exploration, Variable Identification, Uni-Variate and Bi-Variate analysis. We also looked at various statistical and visual methods to identify the relationship between variables.

Now, we will look at the methods of Missing values Treatment. More importantly, we will also look at why missing values occur in our data and why treating them is necessary.

3.7 Data Reduction

Q21. Explain the Data Reduction in Data mining.

Ans :

Data reduction is the process of minimizing the amount of data that needs to be stored in a data storage environment. Data reduction can increase storage efficiency and reduce costs.

(a) Data reduction can be achieved using several different types of technologies. The best-known data reduction technique is data deduplication, which eliminates redundant data on storage systems.

(b) The duplication process typically occurs at the storage block level. The system analyzes the storage to see if duplicate blocks exist, and gets rid of any redundant blocks. The remaining block is shared by any file that requires a copy of the block. If an application attempts to modify this block, the block is copied prior to modification so that other files that depend on the block can continue to use the unmodified version, thereby avoiding file corruption. Some storage arrays track which blocks are the most heavily shared.

(c) Those blocks that are shared by the largest number of files may be moved to a memory- or flash storage-based cache so they can be read as efficiently as possible.

(d) Data reduction strategies applied on huge data set. Complex data and mining on huge amounts of data can take a long time, making such analysis impractical or infeasible.

(e) Data reduction techniques can be applied to obtain a reduces data should be more efficient yet produce the same analytical results.

Q22. Explain the Strategies for data reduction ?

Ans :

The following are the Strategies for reduction

1. **Data cube aggregation,** where aggregation operations are applied to the data in the construction of a data cube.

2. **Attribute subset selections**, where irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed,

3. **Dimensionality reduction**, where encoding mechanism are used to reduce the data set size.

4. **Numerosity reductions**, where the data are replaced or estimated by alternative, smaller data representations such as parametric models or non parametric method such as clustering, sampling, and the use of histograms.

5. **Discretization and concept hierarchy generation,** where raw data values for attributes are replaced by range or higher conceptual levels. Data discretization is a form of numerosity reduction that is very useful for the automatic generation of concept hierarchies. Discretization and concept hierarchy generation are powerful tools for data mining, in that they allow the mining of data at multiple levels of abstraction.

Q23. Explain the techniques used in Data Reduction.

Ans :

(i) Dimensionality Reduction

Dimensionality Reduction ensures the reduction of the number of attributesorrandom variables in the data set. Dimensionality reduction is a series of techniques in machine learning and statistics to reduce the number of random variables to consider. It involves feature selection and feature extraction. Dimensionality reduction makes analyzing data much easier and faster for machine learning algorithms without extraneous variables to process, making machine learning algorithms faster and simpler in turn.

(ii) Sample Numerosity Reduction

Replaces the original data by an alternative smaller data representation This is a technique of choosing smaller forms or data representation to reduce the volume of data.

These techniques may be parametric or nonparametric.

(a) Parametric

For parametric methods, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. (Outliers may also be stored.)

Example: Log-linear models, which estimate discrete multidimensional probability distributions.

(b) Nonparametric

Nonparametric methods are used for storing reduced representations of the data include histograms, clustering, and sampling.

Regression and Log-Linear Models

- ➢ Regression and log-linear models can be used to approximate the given data.
- ➢ In (simple) linear regression, the data are modeled to fit a straight line.
- ➢ Multiple linear regression is an extension of (simple) linear regression, which allows a response variable y to be modeled as a linear function of two or more predictor variables.
- ➢ Log-linear models approximate discrete multidimensional probability distributions.
- ➢ Log-linear models can be used to estimate the probability of each point in a multidimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations.
- ➢ This allows a higher-dimensional data space to be constructed from lower dimensional spaces.
- ➢ Log-linear models are therefore also useful for dimensionality reduction and data smoothing
- ➢ Regression and log-linear models can both be used on sparse data, although their application may be limited.
- ➢ While both methods can handle skewed data, regression does exceptionally well. Regression can be computationally intensive when applied to high dimensional data, whereas log-linear models show good scalability for up to 10 or so dimensions.

(iii) Cardinality Reduction

Transformations applied to obtain a reduced representation of the original data.

The term cardinality refers to the uniqueness of data values contained in a particular column (attribute) of a database table. The lower the cardinality, the more duplicated elements in a column. Thus, a column with the lowest possible cardinality would have the same value for every row. SQL databases use cardinality to help determine the optimal query plan for a given query.

3.8 Data Mining and Business Intelligence

Q24. What is business intelligence?

Ans :

The term 'Business Intelligence' has evolved from the decision support systems and gained strength with the technology and applications like data warehouses, Executive Information Systems and Online Analytical Processing (OLAP).

Business Intelligence System is basically a system used for finding patterns from existing data from operations.

Business Intelligence is the set of processes, technologies, and tools that help an organization to transform raw data into meaningful and useful information for business analysis (Identifying business needs and determining solutions to business problems).

Q25. Define Business Intelligence (BI). Discuss Characteristics, Need and Stages of BI.

Ans :

Meaning

Business Intelligence (BI) according to Larissa Terpeluk Moss and S. Atre is neither a product nor a system, it is an architecture and a collection of integrated operational as well as decision-support applications databases that offer the business entities easy access to business data.

Characteristics

1. It is created by procuring data and information for use in decision-making.
2. It is a combination of skills, processes, technologies, applications and practices.
3. It contains background data along with the reporting tools.
4. It is a combination of a set of concepts and methods strengthened by fact-based support systems.
5. It is an extension of Executive Support System or Executive Information System.
6. It collects, integrates, stores, analyzes, and provides access to business information
7. It is an environment in which business users get reliable, secure, consistent, compre-hensible, easily manipulated and timely information.
8. It provides business insights that lead to better, faster, more relevant decisions.

Need

1. Analyzing a large amount of data to provide historical, current and predictive views of the business operations.
2. To support business decisions from operations to strategic.
3. To process real-time data and develop effective business strategies.
4. Easy interpretation of unstructured data and using it to support decisions.

Stages

Below are the five stages of Big data Business Intelligence in any organization.

1. **Data Sourcing**

   Defining the data to be loaded into the system. Usually BI applications gathers data from a data warehouse (Data marts, OLTP or OLAP).

2. **ETL (Extract Transform Load)**

   Extracting the source data and transforming per business rules and loading into the Data Warehouses.

3. **Data Warehousing**

   Storing transformed data into various Data warehouses types and making it available for business analysis.

4. **Data Analysis**

   Applying various techniques like data mining, text mining, Process mining to identify trends and patterns in business operations.

5. **Decision Making**

   Based on the reports, dashboards and alerts from previous stage, making valuable business decisions and bench marking future growth.

Q26. Explain the History and Evolution of Business Intelligence.

Ans : **(Imp.)**

History

The term BI was coined by the Gartner Group in the mid-1990s. However, the concept is much older; it has its roots in the Management Information Systems (MIS) reporting systems of the 1970s.

During that period, reporting systems were static and two-dimensional and had no analytical capabilities. In the early 1980s, the concept of executive information systems (EIS) emerged. This concept expanded the computerized support to top-level managers and executives.

Some of the capabilities introduced were dynamic multidimensional (ad hoc or on-demand) reporting, forecasting and prediction, trend analysis, drill down to details, status access, and critical success factors (CSFs). These features appeared in dozens of commercial products until the mid-1990s. Then the same capabilities and some new ones appeared under the name BI.

Evolution

Today, a good Bl-based enterprise information system contains all the information executives need. So, the original concept of EIS was transformed into BI. By 2005, BI systems started to include artificial intelligence capabilities as well as powerful analytical capabilities. Figure illustrates the various tools and techniques that may be included in a BI system. It illustrates the evolution of BI as well. The tools shown in Figure provide the capabilities of BI. The most sophisticated BI products include most of these capabilities; others specialize in only some of them.

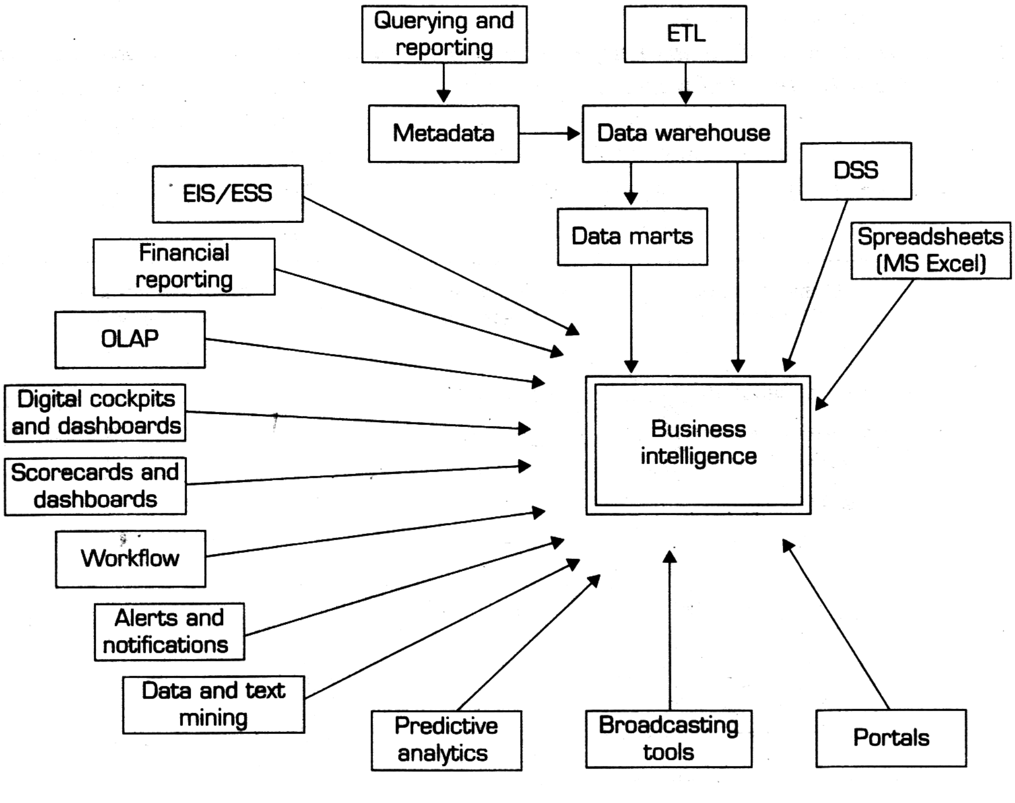


Figure : Evolution of BI

Q27. Discuss about Architecture of Business Intelligence.

Ans :

A BI system has four major components: a data warehouse, with its source data; **business analytics,** a collection of tools for manipulating, mining, and analyzing the data in the data warehouse; business performance management (BPM) for monitoring and analyzing performance; and a user interface {e.g., a dashboard). The relationship among these components is illustrated in figure.

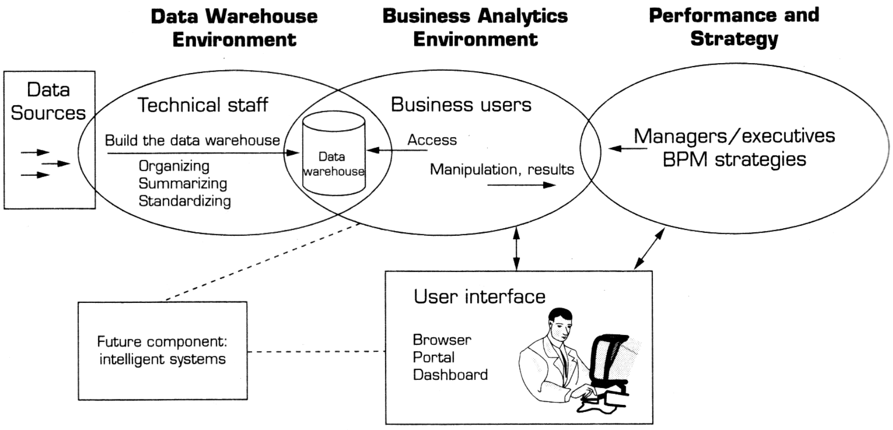


Figure : A High-Level Architecture of BI

Notice that the data warehousing environment is mainly the responsibility of technical staff, whereas the analytical environment (also known as business analytics) is the realm of business users. Any user can connect to the system via the user interface, such as a browser. Top managers may also use the BPM component and also a dashboard.

Components

1. Data Warehousing

The data warehouse and its variants are the cornerstone of any medium-to-large BI system. Originally, the data warehouse included only historical data that were organized and summarized, so end users could easily view or manipulate data and information. Today, some data warehouses include current data as well, so they can provide real-time decision support.

2. Business Analytics

End users can work with the data and information in a data ware-house by using a variety of tools and techniques. These tools and techniques fit into two major categories:

(a) **Reports and Queries:** Business analytics include static and dynamic reporting, all types of queries, discovery of information, multidimensional view, drill down to details, and so on.

(b) **Data, text, and Web mining and other sophisticated mathematical and statistical tools :** Data mining is a process of searching for unknown relationships or information in large databases or data warehouses, using intelligent tools such as neural computing, predictive analytics techniques, or advanced statistical methods.

3. Business Performance Management (BPM)

Business performance management is also called as corporate performance management. BPM includes the evolution and architecture of Business Intelligence (BI).

BPM introduced a new concept called as management and feedback that extends the measuring, monitoring and comparison of sales, cost, profit and profitability. It involves in various processes such as planning, forecasting and budgeting. The traditional Decision Support Systems (DSS), Executive Information System (E1S), Business Intelligence (BI) helps in the bottom-up extraction of information from data where as BPM provides/offers a top-down application of corporate wide strategy. Moreover, it is basically combines with the balanced scorecard methodology as well as dashboards.

4. User Interface

User interface includes dashboards and other information broadcasting tools such as dash boards, portal and browser. Using these tools, a user can connect to user interface. Dashboards are meaningful groups of corporate/marketing performance measures (key performance indicators), exceptions and trends. They are used to combine the information collected from various business areas and also provides a graph that shows the comparison between the actual performance and the required metrics. Hence, it shows view of the organization's health.

In addition to this, user interface includes some other information broadcasting tools such as digital cockpits, corporate portals and visualization tools that ranges from multidimensional cube presentation to virtual reality are known to be the integral parts of business intelligence (BI) systems. Since, BI is derived from EIS, BI acquires many visual aids for executives are converted into BI software. However, Geographical Information system (GIS) also plays an incremental role in DSS.

3.9 Data Classification

Q28. Explain the Classification of Data Mining.

Ans : **(Imp.)**

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

- Classification is a classic data mining technique based on machine learning. Basically, classification is used to classify each item in a set of data into one of a predefined set of classes or groups.
- Classification method makes use of mathematical techniques such as decision trees, linear programming, neural network and statistics. In classification, we develop the software that can learn how to classify the data items into groups.
- For example, we can apply classification in the application that given all records of employees who left the company, predict who will probably leave the company in a future period. In this case, we divide the records of employees into two groups that named "leave" and "stay". And then we can ask our data mining software to classify the employees into separate groups.
- In technical term, classification in data mining defines as assigning an object to a certain class based on its similarity to previous examples of other objects.
- The classification process comes under the predictive method. With classification, new samples of data are classified into known classes.
- The classification is the initial process of data mining and use algorithms like decision trees, Bayesian classifiers. For classification the data required must be already labeled one.

Examples of classification are:

1. A marketing manager of a company needs to analyze the customer with available profile that who will buy a new computer.
2. A bank officer wants to predict that which loan applicants are risky or which are safe.
   - A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time.
   - In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments, and so on.

- Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case.
- The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values:
- Example, high credit rating or low credit rating. Multi-class targets have more than two values: for example, low, medium, high, or unknown credit rating
- In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target.
- Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.
- Classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model.

Q29. Explain various issues relating to data classification.

Ans :

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities:

- **Data Cleaning:** Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.
- **Relevance Analysis:** Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and Reduction:** The data can be transformed by any of the following methods:

  (i) Normalization: The data is transformed using normalization. It involves scaling all values for given attribute in order to make them fall within a small specified range. It is used when in the learning step, the neural networks or the methods involving measurements are used.

  (ii) Generalization: The data can also be transformed by generalizing it to the higher concept. For this purpose, we can use the concept hierarchies.

3.10 Data Association

Q30. Explain the Association in Data Mining?

Ans :

- It is a data mining function that discovers the probability of the co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules. Association rules are often used to analyze sales transactions.
- Association analysis is about discovering relationship among huge data sets. Just like the famous market basket analysis which gives a relationship between {Diapers -> beer}. It says that whenever a person buys diapers he/ she also buys beer.
- Besides market basket , association rules can be applied to Bioinformatics , web mining and medical analysis.

There are two key issues that need to be addressed while applying this ;

- First detecting the pattern

- Some of the detected patterns can be spurious and may be happening only by chance.
- The strength of a association rule can be measured in terms of support and confidence.
- Support determines how often a rule is applicable to the data set while confidence determines how frequently items in Y appear in transactions that contain X.
- Use packages like a rules, a rules CBA , a rules Sequences in R

  Ex :

  - library ("a rules")
  - data ("Adult")

- rules <- apriori(Adult, parameter = list(supp = 0.5, conf = 0.9, target = "rules"))

3.11 Cause Effect Modeling

Q31. Explain cause effect modelling in Data mining.

Ans : **(Imp.)**

- Time series data can be used to extract delayed relationship between two variables, for example, "CO_2 emission occurring at a place might cause air pollution at another place after some delay". These lagged relationships signify the time lag between the cause–effect parameters..
- A system such as mechanical, biological or social-economic system consists of independent components. These components influence one another to maintain their activity for the existence of a system in order to achieve the goal of the system
- The system changes behavior when a component is changed or removed significantly. This motivates us to find the reason or cause behind fault and discover the cause parameters in explaining the interactions among the components of a system or process.
- The causal discovery indicates not only that the indicators are correlated, but also how changing a cause variable is expected to induce a change in an effect variable. For example, with analyzed cause–effect relationships, we can predict potential effects before taking any actions (causes), which is useful in preventing inaccurate decision or policy making in the social-economical system.
- Time series data can be used to extract delayed relationship between two variables, for example, "CO2 emission occurring at a place might cause air pollution at another place after some delay".
- These lagged relationships signify the time lag between the cause–effect parameters. Identifying lagged relationships between socioeconomic processes is challenging due to the presence of various complex dependencies in the data.
- This dependency among the various parameters has enabled us to identify relationships among different domain parameters in time series data.
- The cause–effect relationship for time series prediction is a step towards extracting the various existing causal relations between different domain, such as employment, education, agriculture and rural development etc.
- It has also emerged in economics and social sciences such as to improve the economic development and growth of a country and to study the impact of climate change.

Q32. Explain the process of cause and effect analysis.

Ans : **(Imp.)**

The following are the steps to solve a problem with Cause and Effect Analysis:

Step 1: Identify the Problem

First, write down the exact problem you face. Where appropriate, identify who is involved, what the problem is, and when and where it occurs.

Then, write the problem in a box on the left-hand side of a large sheet of paper, and draw a line across the paper horizontally from the box. This arrangement, looking like the head and spine of a fish, gives you space to develop ideas.

Example: In this simple example, a manager is having problems with an uncooperative branch office.

Step 2: Work Out the Major Factors Involved

Next, identify the factors that may be part of the problem. These may be systems, equipment, materials, external forces, people involved with the problem, and so on.

Try to draw out as many of these as possible. As a starting point, you can use models such as the McKinsey 7S Framework (which offers you Strategy, Structure, Systems, Shared Values, Skills, Style and Staff as factors that you can consider) or the 4Ps of Marketing (which offers Product, Place, Price, and Promotion as possible factors).

Brainstorm any other factors that may affect the situation.

Then draw a line off the "spine" of the diagram for each factor, and label each line.

Step 3: Identify Possible Causes

Now, for each of the factors you considered in Step 2, brainstorm possible causes of the problem that may be related to the factor.

Show these possible causes as shorter lines coming off the "bones" of the diagram. Where a cause is large or complex, then it may be best to break it down into sub-causes. Show these as lines coming off each cause line.

Example: For each of the factors he identified in step 2, the manager brainstorms possible causes of the problem, and adds these to his diagram.

Step 4: Analyze Your Diagram

By this stage, you should have a diagram showing all of the possible causes of the problem that you can think of.

Depending on the complexity and importance of the problem, you can now investigate the most likely causes further. This may involve setting up investigations, carrying out surveys, and so on. These will be designed to test which of these possible causes is actually contributing to the problem.

Example: The manager has now finished his analysis. If he had not looked at the problem this way, he might have dealt with it by assuming that people in the branch office were "being difficult"?

Short Question and Answers

1. What is predictive analysis ?

Ans :

Predictive Analytics is the practice of extracting information from existing data sets in order to determine patterns and predict future outcomes and trends. It does not tell you what will happen in the future.

Predictive analytics is the branch of the Advanced Analytics which is used to make predictions about unknown future events. It uses many techniques from data mining, statistics, modeling, machine learning, and artificial intelligence to analyze current data to make predictions about future.

Predictive Analytics uses historical data to predict future events. Typically, historical data is used to build a mathematical model that captures important trends. That predictive model is then used on current data to predict what will happen next, or to suggest actions to take for optimal outcomes.

2. Regression analysis.

Ans :

Regression takes its name from studies made by Sir Francis Galton. He compared the heights of persons to the heights of their parents. His major conclusion was that the off-springs of unusually tall persons tend to be shorter than their parents while children of usually short parents tend to be shorter than their parents.

In a sense, the successive generations of off-springs from tall persons "regress" downward toward the height of the population. While the reverse is true originally for short families. But the distribution of heights for the total population, continues to have the same variability from generation to generation.

Regression Variables

(i) **Independent Variable (Regressor or Predictor or Explanatory).** The variable which influences the values of the other variable or which is used for prediction of the value of the other variable is called independent variable.

(ii) **Dependent Variable (Regressed or Explained Variable).** The variable whose value is influenced or is to be predicted is called dependent variable.

Regression test generates lines of regression of the two variables which helps in estimating the values. Lines of regression of y on x is the line which gives the best estimate for the value of y for any specified value of x. Similarly, line of regression of x on y is the line which gives the best estimate for the value of x for any specified value of y.

3. Limitations of Regression Analysis

Ans :

Some of the limitations of regression analysis are as follows :

(i) Regression analysis assumes that linear relationship exists among the related variables. But in the 'area of social sciences, linear relationship may not exist among the related variables.

(ii) When regression analysis is used to evaluate the value of dependent variable based on independent variable, it is assumed that the static conditions of relationship exist between 'them. These statistic conditions do not exist in social sciences, so, this assumption "minimizes the use' of regression analysis in social science.

(iii) The value of dependent variable can be evaluated based on independent variable by using regression analysis but only upto some limits. If the circumstances go beyond the limits, then resists would be inaccurate.

4. Moving Averages

Ans :

Moving average methods take the average of past actuals and project it forward. These methods assume that the recent past represents the future. As a result, they work best for products with relatively little change - steady demand, no seasonality, limited trends or cycles and no significant demand shifts. Many companies apply this method because it is simple and easy to use. However, since few products actually behave in this way, it tends to be less useful than more specialized methods.

5. Data Mining

Ans :

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use.

- Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD.Aside from the raw analysis step, it also involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.
- The difference between data analysis and data mining is that data analysis is to summarize the history such as analyzing the effectiveness of a marketing campaign, in contrast, data mining focuses on using specific machine learning and statistical models to predict the future and discover the patterns among data.
- The term "data mining" is in fact a misnomer, because the goal is the extraction of patterns and knowledge from large amounts of data, not the extraction (mining) of data itself. It also is a buzzword and is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence (e.g., machine learning) and business intelligence.

6. Scope of Data Mining

Ans :

(i) Data mining process the work in such a manner that it allows business to more proactive to grow substantially.

(ii) It optimizes large database within the short time and works business intelligence which is more vital to organizational growth.

(iii) It represents the data in some logical order or may be in pattern to identify the sequential way of processing of data.

(iv) It includes tree-shaped structure to understand the hierarchy of data and representation of the set of information described in the database.

(v) Brings a genetic way of classification of different sets of data items to view the data in the quick glance.

7. Benefits of Data Mining

Ans :

- Data mining technique helps companies to get knowledge-based information.
- It helps organizations to make the profitable adjustments in operation and production.
- It is a cost-effective and efficient solution compared to other statistical data applications.
- It helps with the decision-making process.
- It facilitates automated prediction of trends and behaviors as well as automated discovery of hidden patterns.
- It can be implemented in new systems as well as existing platforms.
- It is the speedy process which makes it easy for the users to analyze huge amount of data in less time.

8. Disadvantages of Data Mining

Ans :

- There are chances of companies selling useful information of their customers to other companies for money. For example, American Express has sold credit card purchases of their customers to the other companies.

- Many data mining analytics software is difficult to operate and requires advance training to work on.
- Different data mining tools work in different manners due to different algorithms employed in their design. Therefore, the selection of correct data mining tool is a very difficult task.
- The data mining techniques are not accurate. Hence, it can cause serious consequences in certain conditions.

9. What is data exploration?

Ans :

Data exploration is an informative search used by data consumers to form true analysis from the information gathered. Often, data is gathered in a non-rigid or controlled manner in large bulks. For true analysis, this unorganized bulk of data needs to be narrowed down. This is where data exploration is used to analyze the data and information from the data to form further analysis.

Data often converges in a central warehouse called a data warehouse. This data can come from various sources using various formats. Relevant data is needed for tasks such as statistical reporting, trend spotting and pattern spotting. Data exploration is the process of gathering such relevant data.

10. Data Reduction

Ans :

Data reduction is the process of minimizing the amount of data that needs to be stored in a data storage environment. Data reduction can increase storage efficiency and reduce costs.

(a) Data reduction can be achieved using several different types of technologies. The best-known data reduction technique is data deduplication, which eliminates redundant data on storage systems.

(b) The duplication process typically occurs at the storage block level. The system analyzes the storage to see if duplicate blocks exist, and gets rid of any redundant blocks. The remaining block is shared by any file that requires a copy of the block. If an application attempts to modify this block, the block is copied prior to modification so that other files that depend on the block can continue to use the unmodified version, thereby avoiding file corruption. Some storage arrays track which blocks are the most heavily shared.

(c) Those blocks that are shared by the largest number of files may be moved to a memory- or flash storage-based cache so they can be read as efficiently as possible.

(d) Data reduction strategies applied on huge data set. Complex data and mining on huge amounts of data can take a long time, making such analysis impractical or infeasible.

11. Association in Data Mining

Ans :

- It is a data mining function that discovers the probability of the co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules. Association rules are often used to analyze sales transactions.
- Association analysis is about discovering relationship among huge data sets. Just like the famous market basket analysis which gives a relationship between {Diapers -> beer}. It says that whenever a person buys diapers he/she also buys beer.
- Besides market basket, association rules can be applied to Bioinformatics , web mining and medical analysis.

12. Business Intelligence

Ans :

The term ‘Business Intelligence’ has evolved from the decision support systems and gained strength with the technology and applications like data warehouses, Executive Information Systems and Online Analytical Processing (OLAP).

Business Intelligence System is basically a system used for finding patterns from existing data from operations.

Business Intelligence is the set of processes, technologies, and tools that help an organization to transform raw data into meaningful and useful information for business analysis (Identifying business needs and determining solutions to business problems).

13. Association in Data Mining

Ans :

- It is a data mining function that discovers the probability of the co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules. Association rules are often used to analyze sales transactions.
- Association analysis is about discovering relationship among huge data sets. Just like the famous market basket analysis which gives a relationship between {Diapers -> beer}. It says that whenever a person buys diapers he/ she also buys beer.
- Besides market basket , association rules can be applied to Bioinformatics , web mining and medical analysis.

14. Cause effect modelling

Ans :

- Time series data can be used to extract delayed relationship between two variables, for example, “CO_2 emission occurring at a place might cause air pollution at another place after some delay”. These lagged relationships signify the time lag between the cause–effect parameters..
- A system such as mechanical, biological or social-economic system consists of independent components. These components influence one another to maintain their activity for the existence of a system in order to achieve the goal of the system
- The system changes behavior when a component is changed or removed significantly. This motivates us to find the reason or cause behind fault and discover the cause parameters in explaining the interactions among the components of a system or process.
- The causal discovery indicates not only that the indicators are correlated, but also how changing a cause variable is expected to induce a change in an effect variable. For example, with analyzed cause–effect relationships, we can predict potential effects before taking any actions (causes), which is useful in preventing inaccurate decision or policy making in the social-economical system.
- Time series data can be used to extract delayed relationship between two variables, for example, “CO2 emission occurring at a place might cause air pollution at another place after some delay”.

Choose the Correct Answer

1. ___________ is an essential process where intelligent methods are applied to extract data patterns. [b]

   (a) Data warehousing (b) Data mining

   (c) Text mining (d) Data selection

2. Which of the following is not a data mining functionality? [c]

   (a) Characterization and Discrimination (b) Classification and regression

   (c) Selection and interpretation (d) Clustering and Analysis

3. ___________ is the process of finding a model that describes and distinguishes data classes or concepts. [b]

   (a) Data Characterization (b) Data Classification

   (c) Data discrimination (d) Data selection

4. Strategic value of data mining is ___________. [c]

   (a) Cost-sensitive (b) Work-sensitive

   (c) Time-sensitive (e) Technical-sensitive

5. ___________ is a comparison of the general features of the target class data objects against the general features of objects from one or multiple contrasting classes. [a]

   (a) Data Characterization (b) Data Classification

   (c) Data discrimination (d) Data selection

6. If the two series move in reverse directions and the variations in their values are always proportionate, it is said to be: [c]

   (a) Negative correlation (b) Positive correlation

   (c) Perfect negative correlation (d) Perfect positive correlation

7. If one item is fixed and unchangeable and the other item varies, the correlation coefficient will be: [c]

   (a) Positive (b) Negative

   (c) Zero (d) Undecided

8. A process by which we estimate the value of dependent variable on the basis of one or more independent variables is called: [b]

   (a) Correlation (b) Regression

   (c) Residual (d) Slope

9. The slope of the regression line of Y on X is also called the: [d]

   (a) Correlation coefficient of X on Y (b) Correlation coefficient of Y on X

   (c) Regression coefficient of X on Y (d) Regression coefficient of Y on X

10. In simple linear regression, the numbers of unknown constants are: [b]

    (a) One (b) Two

    (c) Three (d) Four

Fill in the blanks

1. ________ Analytics is the process of using data analytics to make predictions based on data.

2. The regression analysis confined to the study of only two variables at a time is termed as ________ regression.

3. The regression analysis for studying more than two variables at a time is termed as ________ regression.

4. ________ average methods take the average of past actuals and project it forward.

5. Exponential smoothing is a more advanced form of ________ forecasting.

6. ________ is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.

7. A data mining project starts with the understanding of the ________ problem.

8. ________ experts build the data model for the modeling process.

9. ________ is an informative search used by data consumers to form true analysis from the information gathered.

10. ________ Analytics is the practice of extracting information from existing data sets in order to determine patterns.

Answers

1. Predictive
2. Simple
3. Multiple
4. Moving
5. Time series
6. Data mining
7. Business
8. Domain
9. Data exploration
10. Predictive

UNIT IV

PRESCRIPTIVE ANALYTICS

Overview of Linear Optimization, Non Linear Programming Integer Optimization, Cutting Plane algorithm and other methods, Decision Analysis - Risk and uncertainty methods - Text analytics, Web analytics.

4.1 Overview of Linear Optimization

Q1. What is linear programming problem (LPP)? States the mathematical formulation of LPP.

Ans :

In 1947, George dantzig and his associates, while working in the U.S. department of Air Force, observed that a large of military programming and planning problems could be formulated as maximizing/ minimizing a linear form of profit/cost function whose variables were restricted to values satisfying a system of linear constraints.

A linear form is meant a thematical expression of the type $a_1x_1 + a_2x_2 + ... + a_nx_n$, where $a_1, a_2, ..., a_n$ are constants and $x_1, x_2, ..., x_n$ are variables. The term 'Programming' refers to the process of determining a particular programme or plan of action. So Linear Programming (L.P.) is one of the most important optimization (maximization/minimization) techniques developed in the field of Operations Research.

The general LPP calls for optimizing (maximizing/minimizing) a linear function of variables called the '*Objective Function*' subject to a set of linear equations and / or inequalities called the '*constraints*' or *restrictions*.

Mathematical Model of LPP

In order to find the values of n decision variables $x_1, x_2, ..., x_n$ to maximize or minimize the objective function

$$z = c_1x_1 + c_2x_2 + c_3x_3 + ... + c_nx_n$$

and also satisfy m-constraints :

$$\left.\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + ... + a_{1j}x_j + ... + a_{1n}x_n (\leq= \text{or} \geq)\ b_1 \\
a_{21}x_1 + a_{22}x_2 + ... + a_{2j}x_j + ... + a_{2n}x_n (\leq= \text{or} \geq)\ b_2 \\
\quad\vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\qquad\qquad \vdots \\
a_{i1}x_1 + a_{i2}x_2 + ... + a_{ij}x_j + ... + a_{in}x_n\ (\leq\ \ = \text{or} \geq)\ b_i \\
\quad\vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\qquad\qquad \vdots \\
a_{m1}x_1 + a_{m2}x_2 + ... + a_{mj}x_j + ... + a_{mn}x_n (\leq= \text{or} \geq)\ b_m
\end{array}\right\}$$

where constraints may be in the form of any inequality ($\leq$ or $\geq$) or even in the form of an equation (=) and finally satisfy the non-negativity restrictions

$$x_1 \geq 0,\ x_2 \geq 0,\ \ldots, x_j \geq 0,\ \ldots, x_n \geq 0.$$

Formulation of LPP

The formulation of linear programming problem as a mathematical model involves the following basic steps :

Step 1

Find the key-decision to be made from the study of the solution. (In this connection, looking for variables helps considerably).

Step 2

Identify the variables and assume symbols x_1, x_2 ... for variable quantities noticed in step 1.

Step 3

Express the possible alternatives mathematically in terms of variables. The set of feasible alternatives generally in the given situation is :

$$\{(x_1, x_2)\ ;\ x_1 > x_2 > 0\}$$

Step 4

Mention the objective quantitatively and express it as a linear function of variables.

Step 5

Express the constraints also as linear equalities / inequalities in terms of variables.

Q2. State the advantages and limitations of linear programming problem.

Ans :

Advantages

1. It helps in organization and study of the information in the same way that the scientific approach to the problem requires.
2. With LP the execute builds into his planning a true reflection of the limitations and restrictions under which he must operate.
3. Once a basic plan is arrived at through LP, it can be reevaluated for changing conditions.
4. Highlighting of bottlenecks in the production process is the striking advantages of this technique.
5. It provides flexibility in analyzing a variety of multidimensional problems.

Limitations of LP

Inspite of wide area of applications, some limitations are associated with linear programming techniques. These are stated below :

1. In some problems objective functions and constraints are not linear. Generally, in real life situations concerning business and industrial problems constraints are not linearly created to variables.
2. There is no guarantee of getting integer valued solutions, for example, in finding out how may men and machines would be required to perform a particular job, rounding off the solution to the nearest integer will not give an optimal solution. Integer programming deals with such problems.
3. Linear programming model does not take into consideration the effect of time and uncertainty. Thus the model should be defined in such a way that any change due to internal as well as external factors can be incorporated.
4. Sometimes large-scale problems cannot be solved with linear programming techniques even when the computer facility is available. Such difficulty may be removed by decomposing the main problem into several small problems and then solving them separately.
5. Parameters appearing in the model are assumed to be constant. But, in real life situations they are neither constant not deterministic.
6. Linear programming deals with only single objective, whereas in real life situations problems come across with multiobjectives.

Q3. State the assumptions and applications of LPP.

Ans : **(Imp.)**

Assumptions of LPP

1. Proportionality

A primary requirement of linear programming problem is that the objective function and every constraint function must be linear.

Roughly speaking, it simply means that if 1 kg of a product costs Rs. 2, then 10 kg will cost Rs. 20. If a steel mill can produce 200 tons in 1 hour, it can produce 1000 tons in 5 hours.

Intuitively, linearity implies that the product of variables such as x_1 x_2, powers of variables such as x_3^2, and combination of variables such as $a_1x_1 + a_2 \log x_2$, are not allowed.

2. **Additivity**

Additivity means if it takes t_1 hours on machine G to make product a and t_2 hours to make product B, then the time on machine G devoted to produce A and B both is $t_1 + t_2$, provided the time required to change the machine from product A to B is negligible.

Then additivity may not hold, in general. If we mix several liquids of different chemical composition, then the total volume of the mixture may not be the sum of the volume of individual liquids.

3. **Multiplicativity**

It requires :

(a) It takes one hour to make a single item on a given machine, it will take 10 hours to make 10 such items.

(b) The total profit from selling a given number of units is the unit profit times the number of units sold.

4. **Divisibility**

It means that the fractional levels of variables must be permissible besides integral values.

5. **Deterministic**

All the parameters in the linear programming models are assumed to be known exactly. While in actual practice, production may depend upon change also.

Applications of LPP

1. **Personnel Assignment Problem**

Suppose we are given m persons, n-jobs, and the expected productivity c_{ij} of *i*th person on the *j*th job. We want to find an assignment of persons $x_{ij} \geq 0$ for all i and j, to n jobs so that the average productivity of person assigned is maximum.

2. **Transportation Problem**

We suppose that m factories (called sources) supply n warehouses (called destinations) with a certain product. Factory F_i (i = 1, 2, ..., m) produces a_i units (total or per unit time) and warehouse W_j (j = 1, 2, 3 ..., n) requires b_j units. Let the decision variables x_{ij}, be the amount shipped from factory F_i to warehouse W_j. The objective is to determine the number of units transported from factory F_i to warehouse W_j. The objective is to determine the number of units transported from factory F_i to warehouse W_j so that the total transportation cost $\sum_{i=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij}$ is minimized.

3. **Efficiencing on Operation of System of Dams**

In this problem, we determine variations in water storage of dams which generate power so as to maximize the energy obtained from the entire system. The physical limitations of storage appear as inequalities.

4. **Agricultural Applications**

Linear programming can be applied in agricultural planning for allocating the limited resource such as acreage, labour, water, supply and working capital, etc. so as to maximize the net revenue.

5. **Military Applications**

These applications involve the problem of selecting an air weapon system against gorillas so as to keep them pinned down and simultaneously minimize the amount of aviation gasoline used, a variation of transportation problem that maximizes the total tonnage of bomb dropped on a set of targets, and the problem of community defence against disaster to find the number of defence units that should be used in the attack in order to provide the required level of protection at the lowest possible cost.

6. Marketing Management

Linear programming helps in analyzing the effectiveness of advertising campaign and time based on the available advertising media. It also helps travelling sales-man in finding the shortest route for his tour.

7. Manpower Management

Linear programming allows the personnel manager to analyses personnel policy combinations in terms of their appropriateness for maintaining a steady-state flow of people into through and out of the firm.

8. Physical Distribution

Linear programming determines the most economic and efficient manner of locating manufacturing plants and distribution centres for physical distribution.

Q4. What are the requirements of linear programming problem ?

Ans :

1. Decision variables and their relationship

The decision (activity variables refer to candidates (products, services, projects etc.) that are competiting with one another for sharing the given limited resources. These variables are usually inter-related in terms of utilization of resources and need simultaneous solutions. The relationship among these variables should be linear.

2. Well defined objective function

A linear programming problem must have a clearly defined objective function to optimize which may be either to maximize contribution by utilizing available resources, or it may be to produce at the lowest possible cost by using a limited amount of productive factors. It should be expressed as a linear function of decision variables.

3. Presence of constraints or restrictions

There must be limitations on resources (like production capacity, manpower, time, machines, markets, etc.) which are to be allocated among various competing activities. These must be capable of being expressed as linear equalities or inequalities in terms of decision variables.

4. Alternative courses of action

There must be alternative courses of action. For example, it must be possible to make a selection between various combinations of the productive factors such as men, machines, materials, markets, etc.

5. Non-negative restrictions

All decision variables must assume non-negative values as negative values of physical quantities is an impossible situation. If any of the variables is unrestricted in sign, a trick can be employed which enforces non-negativity changing the original information of the problem.

Example

Rahul and Co. manufacturers two brands of products namely Shivnath and Harinath. Both these models have to under go the operations on three machines lathe, milling and grinding. Each unit of Shivnath gives a profit of Rs. 45 and requires 2 hours on lathe, 3 hours on milling and 1 hour on grinding. Each unit of Harinath can give a profit of Rs. 70 and requires 3, 5, and 4 hours on lathe, milling and grinding respectively. Due to prior commitment, the use of lathe hours are restricted to a maximum of 70 hours in a week. The operators to operate milling machines are hired for 110 hours / week. Due to scarce availability of skilled man power for grinding machine, the grinding hours are limited to 100 hours / week. Formulate the data into an LPP.

Sol :

Step 1 : Selection of Variables

In the above problem, we can observe that the decision is to be taken on how many products of each brand is to be manufactured. Hence the quantities of products to be produced per week are the decision variables.

Therefore we assume that the number of units of product Shivnath brand produced per week = x_1.

The number of units of product of Harinath brand produced per week = x_2.

Step 2 : Setting Objective

In the given problem the profits on the brands are given.

Therefore objective function is to maximize the profits.

Now, the profit on each unit of Shivnath brand = Rs. 45.

Number of units of Shivnath to be manufactured = x_1

$\therefore$ The profit on x_1 units of Shivnath brand = $45\ x_1$

Similarly, the profit on each unit of Harinath brand = Rs. 70

Number of units of Harinath brand to be manufactured = x_2

$\therefore$ The profit on x_2 units of Harinath brand = $70\ x_2$

The total profit on both brands = $45\ x_1 + 70\ x_2$

This total profit (say z) is to be maximized

Hence, the objective function is to Maximize $z = 45x_1 + 70x_2$

Step 3 : Identification of Constraint Set

In the above problem, the constraints are the availability of machine hours.

1. Constraint on Lathe Machine

Each unit of Shivnath brand requires 2 hours / week

So x_1 units of Shivnath brand requires $2x_1$ hours / week.

Each unit of Harinath brand requires 3 hours / week and so x_2 units of Harinath brand require $3x_2$ hours / week.

Total lathe hours utilized for both the brands is $2x_1 + 3x_2$ and this cannot exceed 70 hours / week

$\therefore\quad 2x_1 + 3x_2 \leq 70$

(Constraint on availability of lathe hours due to prior commitment)

2. Constraint on Milling Machine

Milling hours required for each unit of Shivnath brand = 3 hours / week.

$\therefore$ For x_1 units = $3x_1$ hours / week

Milling hours required for each unit of Harinath brand = 5 hours / week.

$\therefore$ For x_2 units = $5x_2$

Total milling hours = $3x_1 + 5x_2$

This can not be more than 110

$\therefore\quad 3x_1 + 5x_2 \leq 110$

(Constraint on availability of milling machine hours due to hiring)

3. Constraint on Grinding Machine

One unit of Shivnath needs one hour / week and x_1 units need x_1 hours / week.

One unit of Harinath needs 4 hours / week and x_2 units need $4x_2$ hours / week.

Total grinding hours = $x_1 + 4x_2$ and this cannot be greater than 100 hours.

$\therefore \quad x_1 + 4x_2 \leq 100$

(Constraint on availability of grinding hours due to scarcity of skilled labour)

4. Writing Conditions of Variables

Both x_1 and x_2 are the number of products to be produced. There can not exist any negative production. Therefore x_1 and x_2 can not assume any negative values (i.e., non negative)

Mathematically

$x_1 \geq 0$ and $x_2 \geq 0$

Step 5 : Summary

Maximize $Z = 45x_1 + 70x_2$

Subject to $2x_1 + 3x_2 \leq 70$

$3x_1 + 5x_2 \leq 110$

$x_1 + 4x_2 \leq 100$

$x_1 \geq 0$ and $x_2 \geq 0.$

Q5. Describe the steps involved in graphical solution to linear programming models.

Ans :

Simple linear programming problems of two decision variables can be easily solved by graphical method. The outlines of graphical procedure are as follows :

Step1 : Consider each inequality-constraint a equation.

Step 2 : Plot each equation on the graph, as each one will geometrically represent a straight line.

Step 3 : Shade the feasible region. Every point on the line will satisfy the equation of the line. If the inequality-constraint corresponding to that lines is '$\leq$', then the region below the line lying in the first quadrant (due to non-negativity of variables) is shaded. For the inequality-constraint with '$\geq$' sign, the region above the line in the first quadrant is shaded. The points lying in common region will satisfy all the constraints simultaneously. The common region thus obtained is called the *feasible region.*

Step 4 : Choose the convenient value of z (say = 0) and plot the objective function line.

Step 5 : Pull the objective function line until the extreme points of the feasible region. In the maximization case, this line will stop farthest from the origin and passing through at least one corner of the feasible region. In the minimization case, this line will stop nearest to the origin and passing through at least one corner of the feasible region.

Step 6 : Read the coordinates of the extreme point(s) selected in *step 5* and find the maximum or minimum (as the case may be) value of z.

Q6. Write the computational procedure for simplex method.

(OR)

Explain the procedure for simplex method.

Ans : **(Imp.)**

Simplex algorithm was originally proposed by G.B Dantzig in 1948.

It starts at a basic level of the problem.

At each step it projects the improvement in the objective function over its previous step. Thus, the solution becomes optimum when no further improvement is possible on the objective function.

Simplex Algorithm

The algorithm goes as follows :

Step 1: Formulation of LPP

- Selection of decision variables
- Setting of objective function
- Identification of constraint set
- Writing the conditions of variables.

Step 2: Convert constraints into equality form.

- Add slack variable if constraints is $\leq$ type.
- Subtract surplus and add an artificial variable if the constraint is $\geq$ type.
- Add an artificial variable if constraint is exact (=) type.

Step 3: Find, Initial Basic Feasible Solution (IBFS)

- If m non identical equations have n variables (m < n) including all decision, slack / surplus and artificial variables, we get m number of variables basic and (n – m) variables non basic (i.e., equated to zero).
- First make all decision variables (and surplus) as non basic i.e., equate to zero to identity the IBFS.
- Find solution values for basic variables.

Step 4: Construct, Initial Simplex Tableau as given above with the following notations.

- C_B : Coefficient of basic variable in the objective functions
  (or contribution of basic variables)
- BV : Basic variabels (form IBFS)
- SV : Solution value (from IBFS)
- C_j : Contribution of jth variables or coefficient of each variable (jth) in objective function.
- $Z_j - C_j$: Net contribution.

C_B	BV	C_j SV	C_j x_i, S & A	Min-Ratio	Remarks
Contribution of basic variable in objective function	Basic variables	Solution variables	y_i Key Element KE	Most min ratio of SV/key co. vlaue	← Key row
		Z_j	Sum of products of C_B and y_i		
		$Z_j - C_j$	Most negative value		

Key column

Step 5: Find 'out going' and 'incoming' variables.

- Find Z_n by summation of products of C_B and y_i for each column
- Computed $Z_j - C_j$ value for each column
- To find key column use most negative value of $Z_j - C_j$
- Variable in key column is 'in coming variable' or 'entering' variable.
- The variable of key row is 'out going' or 'existing' variable.
- Find the minimum ratio of solution value to corresponding key column value to identify key row.
- The cross section of key column and key row is key element with which the next iteration is carried out.

Step 6: Re-write next tableau as per given set of rules.

- Replace the existing variable from the basis with the entering variable along with its coefficient (or contribution).
- You have to make key element as unity (i.e., 1) and other element in the key column as zeros.
- To make key element as unity, divide the whole key row by the key element. This is supposed as the new row in the place of key row in the next iteration table.
- To find other rows of next iteration table, use this new row. By appropriate adding or subtracting entire new row in the old rows, make other elements of the key column as zeros.

Step 7: Check whether all the values of $Z_j - C_j$ are positive. If all are positive, the optimal solution is reached. Write the solution values and find Z_{opt}. (i.e., Z_{max} or Z_{min} as the case may be).

If $Z_j - C_j$ values are still negative, again choose most negative among these and go to step 5 and repeat the iteration till all the values of $Z_j - C_j$ become positive.

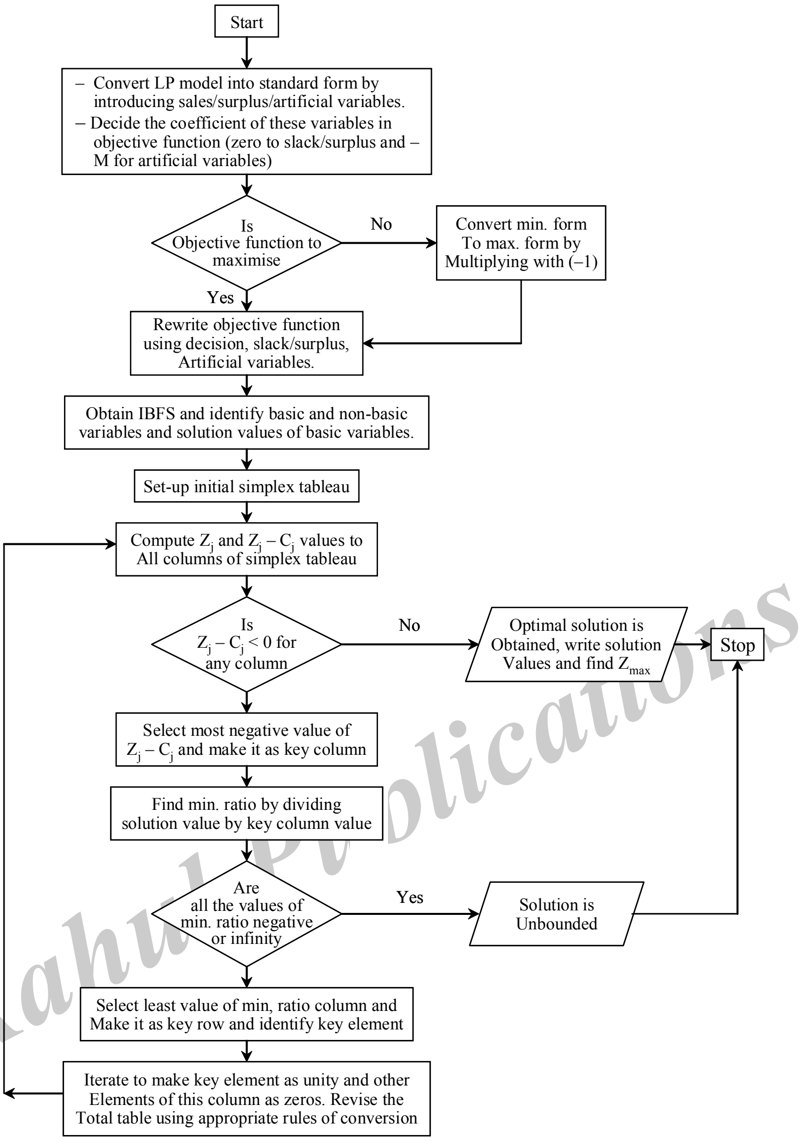

Start
– Convert LP model into standard form by introducing sales/surplus/artificial variables.
– Decide the coefficient of these variables in objective function (zero to slack/surplus and – M for artificial variables)
Is Objective function to maximise
No
Convert min. form To max. form by Multiplying with (–1)
Yes
Rewrite objective function using decision, slack/surplus, Artificial variables.
Obtain IBFS and identify basic and non-basic variables and solution values of basic variables.
Set-up initial simplex tableau
Compute Z_j and $Z_j – C_j$ values to All columns of simplex tableau
Is $Z_j – C_j < 0$ for any column
No
Optimal solution is Obtained, write solution Values and find Z_{max}
Stop
Select most negative value of $Z_j – C_j$ and make it as key column
Find min. ratio by dividing solution value by key column value
Are all the values of min. ratio negative or infinity
Yes
Solution is Unbounded
Select least value of min, ratio column and Make it as key row and identify key element
Iterate to make key element as unity and other Elements of this column as zeros. Revise the Total table using appropriate rules of conversion

Fig.: Flow Chart or Simplex Method

4.2 Non-linear Programming Integer Optimization

Q7. Explain briefly about Non Linear Programming?

Ans : **(Imp.)**

- Non Linear Programming (NLP). If an LP problem is feasible then, at least in theory, it can always be solved because. We know the solution is a "corner point": a point where lines or planes intersect.
- There are a finite number of possible solution points. The simplex algorithm will find that point. Also, a very informative sensitivity analysis is relatively easy to obtain for LP problems. But in many interesting, real-world problems, the objective function may not be a linear function, or some of the constraints may not be linear constraints

- Optimization problems that involve nonlinearities are called nonlinear programming (NLP) problems. Many NLPs do not have any constraints. They are called unconstrained NLPs. Solutions to NLPs are found using search procedures.
- Solutions are more difficult to determine, compared to LPs. One problem is difficulty in distinguishing between a local and global minimum or maximum point.
- Nonlinear programming is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear.
- An optimization problem is one of calculation of the extrema (maxima, minima or stationary points) of an objective function over a set of unknown real variables and conditional to the satisfaction of a system of equalities and inequalities, collectively termed constraints.
- It is the sub-field of mathematical optimization that deals with problems that are not linear.
- In mathematics, nonlinear programming is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear.
- It is the sub-field of mathematical optimization that deals with problems that are not linear.

General Form of Non-linear Programming Problems

Max f(x)

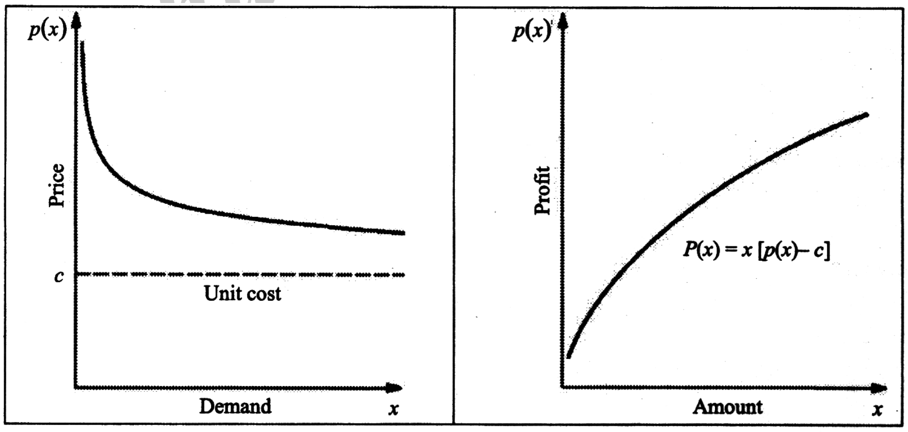
S.T. gi(x) $\le$ b_i for i = 1, ... , m

$x \ge 0$

No algorithm that will solve every specific problem fitting this format is available.

Example - The Product-Mix Problem with Price Elasticity

- The amount of a product that can be sold has an inverse relationship to the price charged, *i.e.*, the relationship between demand and price is an inverse curve.



The firm's profit from producing and selling x units is the sales revenue xp(x) minus the production costs, i.e., P(x) = xp(x) – cx.

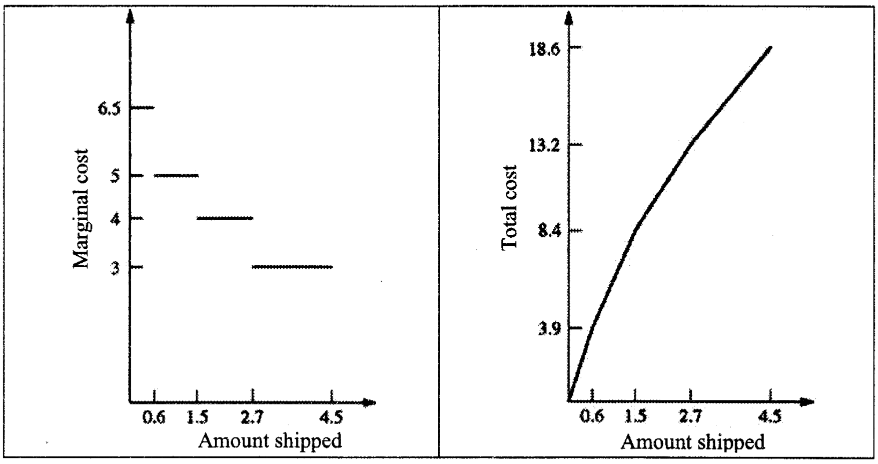
If each of the firm's products has a similar profit function, say, Pfixj) for producing and selling Xj units of product j, then the overall objective function is

$$f(x) = \sum_{j-1} p_j(x_j)\text{, a sum of non-linear functions.}$$

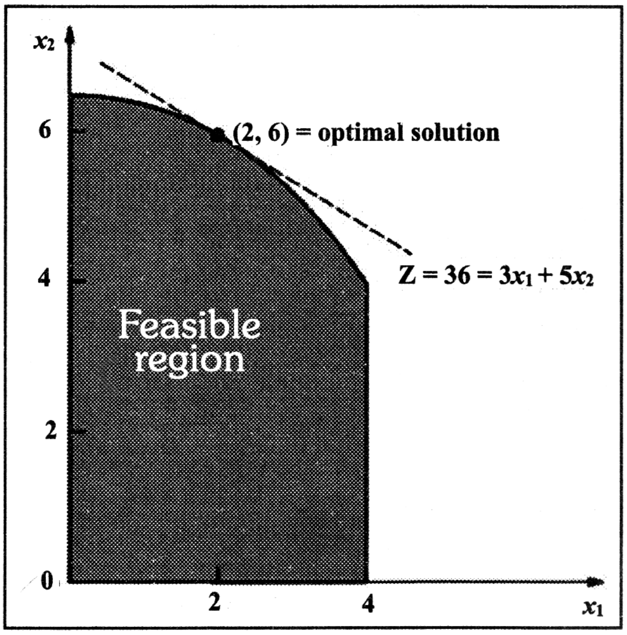
- Non-linearities also may arise in the g,(x) constraint function.

An Example - The Transportation Problem with Volume Discounts

- Determine an optimal plan for shipping goods from various sources to various destinations, given supply and demand constraints.
- In actuality, the shipping costs may not be fixed. Volume discounts sometimes are available for large shipments, which cause a piece-wise linear cost function.



Graphical Illustration of Non-linear Programming Problems



Max $Z = 3x_1 + 5x_2$

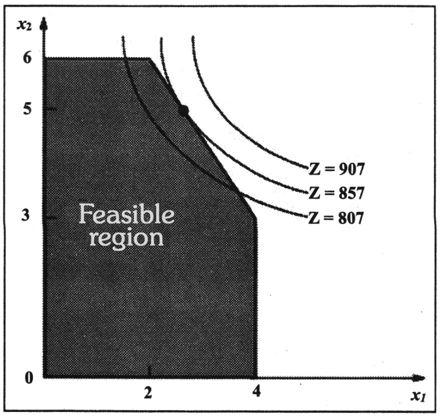
S.T. $x_1 \leq 4$

$9x_1^2 + 5x_2^2 \leq 216$

$X_1, X_2 \geq 0$

- The optimal solution is no longer a CPF anymore. (Sometimes, it is; sometimes, it isn't). But, it still lies on the boundary of the feasible region.

- We no longer have the tremendous simplification used in LP of limiting the search for an optimal solution to just the CPF solutions.
- What if the constraints are linear; but the objective function is not?



Max $Z = 126x_1 - 9x^2 + 182x - 13x^2$

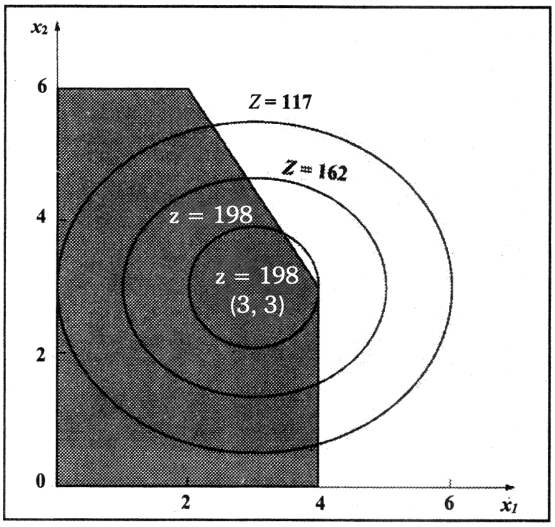
S.T. $x_1 \leq 4$

$2x_2 \leq 12$

$3x_1 + 2x_2 \leq 18$

$x_1, x_2 \geq 0$

What if we change the objective function to $54x_1 - 9x^2 + 78x - 13x^{27}$



- The optimal solution lies inside the feasible region.
- That means we cannot only focus on the boundary of feasible region. We need to look at the entire feasible region.

Constrained Optimization with Equality Constraints

- Consider the problem of finding the minimum or maximum of the function f(x), subject to the restriction that x must satisfy all the equations:

  $g_1(x) = b_1$

  …

  g m(x) = bm

Example

Max f $(x_1, x_2) = x_1^2 + 2x_2$

S.T. $g(x_1, x_2) = x_1^2 + x_2^2 = 1$

A classical method is the method of Lagrange multipliers.

- The Lagrangian function h(x, λ) = f(x) – $\sum_{i=1}^{m} \lambda_i [g_i(x) - b_i]$, where ($\lambda_1$, λ_2, λ_m) are called Lagrange multipliers.
- For the feasible values of x, gi(x) – b_i = 0 for all i, so h(x, λ) = f(x).
- The method reduces to analyzing h(x, X) by the procedure for unconstrained optimization.
- Set all partial derivative to zero
- Notice that the last m equations are equivalent to the constraints in the original problem. So, only feasible solutions are considered.

4.3 Cutting Planning Algorithm and Methods

Q8. Explain briefly about the Cutting Plane Method.

Ans : **(Imp.)**

- In mathematical optimization, the cutting-plane method is any of a variety of optimization
- Methods that iteratively refine a feasible set or objective function by means of linear inequalities, termed cuts. Such procedures are commonly used to find integer solutions to mixed integer linear programming (MILP) problems, as well as to solve general, not necessarily differentiable convex optimization problems. The use of cutting planes to solve MILP was introduced by Ralph E. Gomory.
- Cutting plane methods for MILP work by solving a non-integer linear program, the linear relaxation of the given integer program.
- The theory of Linear Programming dictates that under mild assumptions (if the linear program has an optimal solution, and if the feasible region does not contain a line), one can always find an extreme point or a corner point that is optimal.
- The obtained optimum is tested for being an integer solution. If it is not, there is guaranteed to exist a linear inequality that separates the optimum from the convex hull of the true feasible set.
- Finding such an inequality is the separation problem, and such an inequality is a cut. A cut can be added to the relaxed linear program.
- Then, the current non-integer solution is no longer feasible to the relaxation. This process is repeated until an optimal integer solution is found.

We start by solving the LP relaxation to get a lower bound for the minimum objective value.

We assume the final simplex tableau is given, the basic variables having columns with coefficient 1 in one constraint row and 0 in other rows. The solution can be read from this form: when the non - basic variables are 0, the basic variables have the values on right-hand side (RHS). The objective function row is of the same form, with its basic variable f.

If the LP solution is fractional, i.e., not integer, at least one of the RHS values is fractional. We proceed by appending to the model a constraint that cuts away a part of the feasible set so that no integer solutions are lost.

Take a row i from the final simplex tableau, with a fractional RHS d. Denote by x_{j0} the basic variable of this row and N the index set of non-basic variables.

Row i as an equation:

$$x_{jo} + \sum_{j \in N} w_{ij} x_j = d$$

Denote by Idm, the largest integer, i.e., #d (the whole part of d, if d is positive). Because all variables are non-negative,

$$\sum_{j\in N}[w_{ij}]x_j \# \sum_{j\in N} w_{ij}x_j$$

$$x_{jo} + \sum_{j\in N}[w_{ij}]x_j \# d$$

Left hand side is integer

$$x_{j0}\sum_{j\in N}[w_{ij}]x_j \# 1dm$$

From the first and last formula, it follows that

$$d - idm \# \sum_{j\in N}(w_{ij} - [w_{ij}])x_j \# idm$$

If we denote the fractional parts by symbols r = d – Idm and $f_{ij} = w_{ij}$ – IWijm,, we get a cut constraint or a cutting plane in the solution space:

$$\sum_{j\in N} f_{ij}x_j \$ r$$

Equation form, using a slack variable s_i:

$$-\sum_{j\in N} f_{ij}x_j + s_i = -r$$

This equation is of basic form, with basic variable $s_i = -r$.

The resulting simplex tableau is optimal but infeasible, and we apply the dual simplex method until all variables are non-negative.

The cut constraints do not cut out any feasible integer points and they pass through at least one integer point.

The next cutting plane algorithm operates with a simplex tableau.

Q9. Explain briefly about Cutting Plane Algorithm ?

Ans : **(Imp.)**

- Cutting-plane methods for general convex continuous optimization and variants are known under various names: Kelley's method, Kelley–Cheney–Goldstein method, and bundle methods.
- They are popularly used for non-differentiable convex minimization, where a convex objective function and its subgradient can be evaluated efficiently but usual gradient methods for differentiable optimization can not be used.
- This situation is most typical for the concave maximization of Lagrangian dual functions.
- Another common situation is the application of the Dantzig–Wolfe decomposition to a structured optimization problem in which formulations with an exponential number of variables are obtained.
- Generating these variables on demand by means of delayed column generation is identical to performing a cutting plane on the respective dual problem.

Q10. Explain the other methods in cutting plane algorithm.

Ans :

The following are the methods.

1. Cutting planes
2. Localization methods

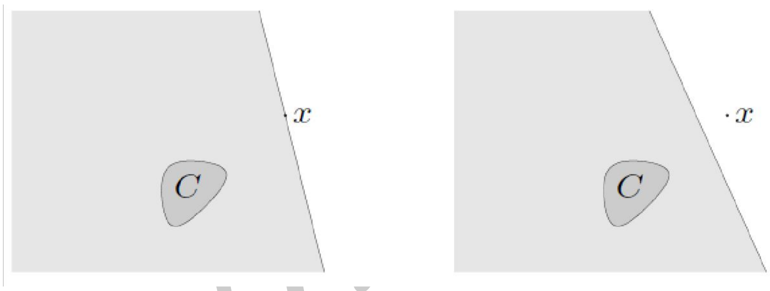
1. Cutting-plane

Oracle provides a black-box description of a convex set C

- When queried at x, oracle either asserts $x \in C$ or returns $a \neq 0$, b with

$$a^T x \geq b, \quad a^T z \leq b \quad \forall z \in C$$

- $a^T z = b$ defines a cutting plane, separating x and C
- Cut is neutral if $a^T x = b$: query point is on boundary of half-space
- Cut is deep if $a^T x > b$: query point in interior of half-spaces that is cut.



2. Localization method

- Goal: Find a point in convex set C described by cutting-plane oracle
- Algorithm: choose bounded set P_0 containing C; repeat for $k \geq 1$:
- Choose a point $x^{(k)}$ in P_{k-1} and query the cutting-plane oracle at $x^{(k)}$.
- if $x^{(k)} \in C$, return $x^{(k)}$; else, add cutting plane $a_k^T z \leq b_k$ to P_{k-1}:

$$P_k = P_{k-1} \cap \{z \mid a_k^T z \leq b_k\}$$

- Termine if $P_k = f$

Variation: to keep P_k simple, choose $P_k \supseteq P_{k-1} \cap \{z \mid a_k^T z \leq b_k\}$.

4.4 Decision Analysis

Q11. Explain briefly about the term decision analysis?

Ans : **(Imp.)**

The term decision analysis was coined in 1964 by Ronald A. Howard, professor of management science and engineering at Stanford University. Decision analysis refers to a systematic, quantitative and interactive approach to addressing and evaluating important choices confronted by organizations in the private and public sector. Decision analysis is interdisciplinary and draws on theories from the fields of psychology, economics, and management science. It utilizes a variety of tools which include models for decision-making under conditions of uncertainty or multiple objectives; techniques of risk analysis and risk assessment; experimental and descriptive studies of decision-making behaviour; economic analysis of competitive and strategic decisions; techniques for facilitating decision-making by groups; and computer modeling software and expert systems for decision support.

Example

If XYZ real estate development company were deciding whether or not to build a new shopping center in a location, they might examine several pieces of input to aid in their decision-making process. These might include traffic at the proposed location on various days of the week at different times, the popularity of similar shopping centers in the area, financial demographics and spending habits of the area population, local competition, and preferred shopping habits of the area population. All of these items could be put into a decision analysis program and different simulations could be run that would help XYZ company make their decision about the shopping center.

Q12. Explain Decision Making and criteria for decision making.

Ans :

- Normally in decision we have two or more then two alternative from which to be choose one.
- The consequence of the alternative depends upon the outcomes of some future random event (or states of nature).
- Outcomes of such decision are usually unexpected but possible to predict by information carry chosen one alternative.

Criteria For Decision Making

If a decision maker carryout information about an alternative upon it become easy to takes decision.

Like converting available information into a measure of desirability.

An appropriate one criteria for decision can be: ⇒ Probability estimates of future outcomes. And willingness of decision maker to take risk.

4.4.1 Decision Making under Risk and Uncertainity

Q13. Explain about decision making under uncertainty.

Ans : **(Imp.)**

In decision-making under uncertainty the probabilities associated with occurrence of the different states of nature are not given. The decision maker has to determine the expected payoff for the courses of action or strategies as the probabilities associated with the occurrence of states of nature are not given. The decision maker has number of criteria available and has to select one among them. The selection depends upon the attitude of the decision maker and the policy of an organization.

Decision-making under uncertainty has various criteria such as,

1. **Criterion of Pessimism or Maximin**

   Maximin initially identifies the worst possible outcome for each course of action i.e., maximum loss or minimum outcome that would occur under each decision alternative and then choosing the best out of the worst outcome (i.e., maximum payoff) is order to select the optimal course of action or strategy.

2. **Criterion of Optimism or Maximax**

   Maximax is totally reverse of maximin operations research criterion of pessimism. Maximax identifies the best possible outcome (maximum payoff associated with each course of action and then choose the maximum of the maximum value in order to select the optimal course of action or strategy.

3. **Minimax Regret Criterion**

   Minimax regret criterion is useful in identifying the regret (or opportunity loss) which is associated with each states of nature if a specific course of action is undertaken.

For each conditional profit or the cost value (payoff) regret value is calculated by taking the difference between the maximum payoff under a state of nature and the payoff resulting from each course of action under that state is calculated. It can be shown in an equational form as,

Regret payoff = Maximum payoff from a course of action-Payoff

Once the value is obtained, find the highest regret value for each course of action and then select that course of action with the minimum regret values. The regret is quite similar to the EOL (Expected Opportunity Loss) which is also known as conditional opportunity loss.

4. Hurwicz Criterion or Criterion of Realism

A rational decision maker should not be either completely optimistic or pessimistic. Hurwicz introduced the idea of coefficient of optimism. Let the coefficient of optimism be a then,

$$0 \leq \alpha \leq 1$$

a) If a is close to 1, the decision-maker is optimistic about the future.

b) If a is close to zero, the decision-maker is pessimistic about the future. According to Hurwicz, select the strategy that maximizes.

H = a (Maximum payoff in column)+ (1 - a) (Minimum payoff in column)

5. Criterion of Rationality or Baye's or Laplace Criterion

Laplace criterion is based on the principle of equal likelihood or insufficient reason. According to this principle, as probabilities of future states of nature is unknown, there is no reason to consider any one outcome more likely than the other i.e., all outcomes must be considered equally likely. With outcomes, each outcome will thus have a probability of *1ln*. With the help of these probabilities such a course of action must be chosen which has the highest expected loss.

As all the outcomes are weighted equally, the average outcome for each course of action must be calculated i.e., add for each course of action the payoffs for all outcomes (states of nature), and then divide it by the number of courses of action.

Example

A food product company is contemplating the introduction of a revolutionary new product with new packaging to replace the existing product at much price (S_1) or a moderate change in the composition of the existing product with a new packaging at a small increase in price (S_2) or a small change in the composition of the existing except the word 'new' with a negligible increase in price (S_3). The three possible states of nature of events are, (i) High increase in sales (N_2), (ii) No change in sales (N_2) and (iii) Decrease in sales (N_3). The marketing department of the company worked out the payoffs in terms of yearly net profits for each course of action for these events (expected sales). This is represented in the following table,

States of Nature	Courses of Action		
	S_1	S_2	S_3
N_1	7,00,000	5,00,000	3,00,000
N_2	3,00,000	4,50,000	3,00,000
N_3	1,50,000	0	3,00,000

Which strategy should the company choose on the basis of,

a) Maximin criterion

b) Maximax criterion

c) Minimax regret criterion

d) Laplace criterion.

Sol :

a) The given table is again reproduced as table (1) and includes an extra row indicating the worst or minimum outcome associated with each course of action (strategy).

States of Nature	Courses of Action		
	S_1	S_2	S_3
N_1	7,00,000	5,00,000	3,00,000
N_2	3,00,000	4,50,000	3,00,000
N_3	1,50,000	0	3,00,000
Minimum payoff	1,50,000	0	3,00,000

Table(1)

Since this is associated with the worst possible outcomes of decrease in sales worth Rs. 3,00,000 the optimal course of action (or strategy) is obtained S_3 applying the maximin criterion.

b) Including an extra row representing the maximum payoff associated with each course of action and then applying the criterion of maximax, the optimal course of action is S_1, since this associates with it a maximum outcomes of Rs. 7,00,000 as shown in table (2).

c) The minimum value among the maximum regret as shown in table (3) is zero and this corresponds to course of action S_1.

States of Nature	Courses of Action		
	S_1	S_2	S_3
N_1	7,00,000	5,00,000	3,00,000
N_2	3,00,000	4,50,000	3,00,000
N_3	1,50,000	0	3,00,000
Maximum payoff	7,00,000	5,00,000	3,00,000

Table (2)

State of Nature	Courses of Action		
	S_1	S_2	S_3
N_1	7,00,000 – 7,00,000 = 0	7,00,000 – 5,00,000 = 2,00,000	7,00,000 – 3,00,000 = 4,00,000
N_2	4,50,000-3,00,000 = 1,50,000	4,50,000-4,50,000 = 0	4,50,000-3,00,000 = 1,50,000
N_3	3,00,000-1,50,000	3,00,000-0 = 3,00,000	3,00,000-3,00,000 = 0
Maximum regret	1,50,000	3,00,000	4,00,000

Table (3)

d) Here it is assumed that each course of action has a probability of occurrence equal to 1/3. Therefore, expected returns can be obtained as shown in table (4).

Course of Action	Expected Return
S_1	1/3 (7,00,000 + 3,00,000 + 1,50,000) = 3,83,333.33
S_2	1/3 (5,00,000 + 4,50,000 + 0) = 3,16,666.66
S_3	1/3 (3,00,000 + 3,00,000 + 3,00,000) = 3,00,000

Table (4)

Thus, Laplace criterion suggest that the executive should choose the strategy 5.

Q14. Explain about decision making under risk.

Ans : **(Imp.)**

Decision-making under risk assumes the long-run relative frequency of the states of nature occurrence to be given and besides this it also enumerates several states of nature. The state of natures information is probabilistic in nature i.e., the decision maker cannot predict which outcome will occur as a result of selecting a particular course of action. As each course of action results in more than one outcome, it is not easy to calculate the exact monetary payoffs or outcomes for the various combination of courses of action and states of nature.

The decision maker with the help of the past records or experience assigns probabilities to the likely possible occurrence of each state of nature. Once the probability distribution of the states of nature is known, then the best course of action must be selected which yields the highest expected payoffs.

The most widely used criteria for evaluating the alternative courses of action, is the Expected Monetary Value (EMV) which is also called as expected utility. The objective of decision-making under this condition is to optimize the expected payoff.

Example

An electrical manufacturing company has seen its business expanded to the point where it needs to increase production beyond its existing capacity. It has narrowed the alternatives to two approaches to increase the maximum production capacity, (a) Expansion, at a cost of Rs. 8 million, or (b) Modernization at a cost of Rs. 5 million. Both approaches would require the same amount of time for implementation. Management believes that over the required payback period, demand will either be high or moderate. Since high demand is considered to be somewhat less likely than moderate demand, the probability of high demand has been setup at 0.35.

If the demand is high, expansion would gross an estimated additional Rs.12 million but modernization only an additional Rs. 6 million, due to lower maximum production capability. On the other hand, if the demand is moderate, the comparable figures would be Rs. 7 million for expansion and Rs. 5 million for modernization.

(a) Calculate conditional profit in relation to various action and outcome combinations and states of nature.

(b) If company wishes to maximize its expected monetary value, then it should modernize or expand?

(c) Calculate the EVPI.

(d) Construct the conditional opportunity loss table and also calculate EOL.

Sol :

a) Defining the state of nature of outcome (over which the company has no control) and course of action (company's possible decision).

Let,

States of nature : 0_1 = High demand, O_2 = Moderate demand

Courses of action : S_1 = Expand, S_2 = Modernize

Since the probability that the demand is high (outcome O_1) is estimated to 0.35, the probability of moderate demand (outcome O_2 must be (1 – 0.35) = 0.65. The calculations for conditional profit values are as follows,

	Course of Action	
State of Nature Oj	**S_1 (Expand)**	**S_2 (Modernize)**
O_1 (high demand)	12 – 8 = 4	6 – 5 = 1
O_2 (moderate demand)	7 – 8 = –1	5 – 5 = 0

Table (1) : Conditional Profit (Million Rs.)

b) The payoff table (1) can be rewritten as follows along with the given probabilities of states of nature.

		Course of Action	
State of Nature Oj	**Probability P(Oj)**	**S_1 (Expand)**	**S_2 (Modernize)**
O_1 (high demand)	0.35	4	1
O_2 (moderate demand)	0.65	–1	0

Table (2) : Conditional Profit (Million Rs.)

The calculation of EMVs for courses of action 5, and S., are given below,

EMV(S_1) = (0.35)(4) + (0.65)(–1) = 1.40 – 0.65 = Rs. 0.75 million

EMV(S_2) = (0.35)(1) + (0.65)(0) = 0.35 = 0.35 million

To maximize EMV, the company must expand course of action. The EMV of the optimal course of action is generally denoted by EMV*. Therefore,

EMV* = EMV(S_1) Rs. 0.75 million

c) To compute EVPI, we shall first calculate EPPL. For calculating EPPI, we choose the optimal course of action for each state of nature, multiply its conditional profit by the given probability to get weighted profit and then sum these weights as shown in the table (3).

State of Nature Oj	**Probability P(Oj)**	**Optimal Course of Action**	**Conditional Profit**	**Weighted Profits**
O_1	0.35	S_1	4	4 × 0.35 = 1.40
O_2	0.65	S_2	0	0 × 0.65 = 0
			EPPI	1.40

Table (3): Profit of Optimal Course of Action

The optimal EMV* is Rs. 0.75 million corresponding to the course of action S_1. Then,

EVPI = EPPI – EMV (S_1) = 1.40 – 0.75 = Rs. 0.65 million

Alternately, if the company could get a perfect information (for forecast) of demand (high or moderate) it should consider paying upto 0.65 million for an information.

The expected value of perfect information in business helps in getting and absolute upper bound on the amount that should be spent to get additional information on which a given decision is based.

d) The opportunity loss value are shown below.

State of Nature	Probability P(Oj)	Conditional Profit (Rs. million) Course of Action		Loss (Rs.million) Courses of Action	
O_j		S_1	S_2	S_1	S_2
O_1	0.35	4	1	0	3
O_2	0.65	–1	0	1	0

Table (4): Conditional Opportunity Loss Table

The conditional opportunity loss values may be explained as, if outcome O_1 occurred, then the maximum profit of Rs.4 million would be achieved by selecting course of action Thus, the choice of S_1 would result in zero opportunity loss, as it is the best decision if outcome O_1 occurs. If course of action S_2 were chosen with a payoff of one million, then this would result in a opportunity loss of 4 – 13 millions. If the outcome O_2 occurred, then the best course of action would be with zero loss. Thus, no opportunity loss would be associated with the choice of S_2. But, if S_1 were chosen, then the opportunity loss would be 0 – (–1) = Rs. 1 million. That is, the company would have Rs. 1 million worse off in that situation, if it had chosen course of action S_1.

Using the given forecast of probabilities associated with each state of nature P(O_1) = 0.35 and P(O_2) = 0.65, the expected opportunity losses for the two courses of action are,

EOL(S_1)= 0.35(0) + 0.65(1) = Rs. 0.65 million

EOL(S_2) = 0.35(3) + 0.65(0) = Rs. 0.05 million

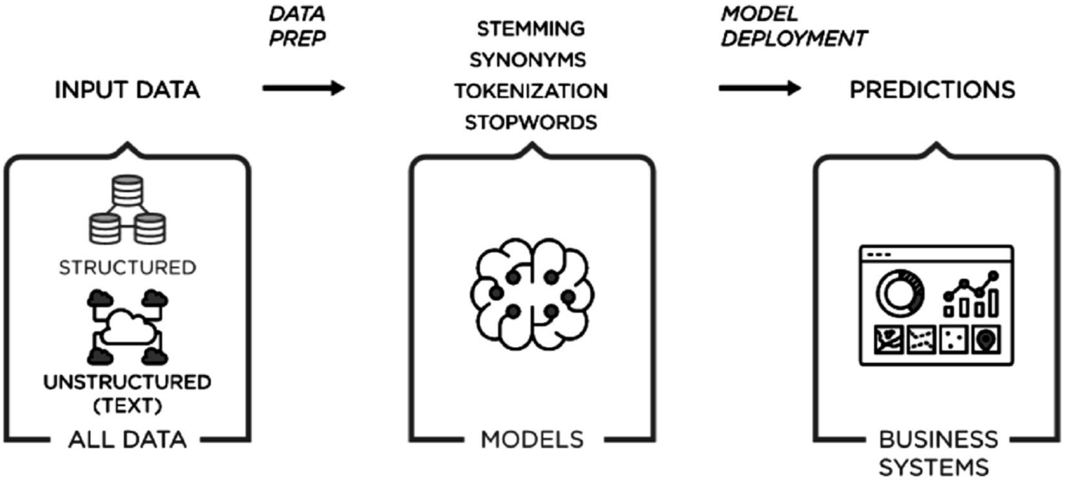
Since decision maker seeks to minimize the expected opportunity loss, he must select course of action S_1 to produce the smallest expected opportunity loss.

4.5 TEXT ANALYTICS

Q15. What is Text Analytics. State the Benefits of Text Analytics.

Ans :

Text analytics combines a set of machine learning, statistical and linguistic techniques to process large volumes of unstructured text or text that does not have a predefined format, to derive insights and patterns. It enables businesses, governments, researchers, and media to exploit the enormous content at their disposal for making crucial decisions. Text analytics uses a variety of techniques – sentiment analysis, topic modelling, named entity recognition, term frequency, and event extraction.



Benefits of Text Analytics

There are a range of ways that text analytics can help businesses, organizations, and event social movements:

- Help businesses to understand customer trends, product performance, and service quality. This results in quick decision making, enhancing business intelligence, increased productivity, and cost savings.
- Helps researchers to explore a great deal of pre-existing literature in a short time, extracting what is relevant to their study. This helps in quicker scientific breakthroughs.
- Assists in understanding general trends and opinions in the society, that enable governments and political bodies in decision making.
- Text analytic techniques help search engines and information retrieval systems to improve their performance, thereby providing fast user experiences.
- Refine user content recommendation systems by categorizing related content.

Q16. Explain various Techniques of Text Analytics.

Ans : **(Imp.)**

1. Text Analytics Techniques and Use Cases

There are several techniques related to analyzing the unstructured text. Each of these techniques is used for different use case scenarios.

Sentiment analysis

Sentiment analysis is used to identify the emotions conveyed by the unstructured text. The input text includes product reviews, customer interactions, social media posts, forum discussions, or blogs. There are different types of sentiment analysis. Polarity analysis is used to identify if the text expresses positive or negative sentiment. The categorization technique is used for a more fine-grained analysis of emotions - confused, disappointed, or angry.

Use cases of sentiment analysis

- Measure customer response to a product or a service
- Understand audience trends towards a brand

- Understand new trends in consumer space
- Prioritize customer service issues based on the severity
- Track how customer sentiment evolves over time

2. Topic Modelling

This technique is used to find the major themes or topics in a massive volume of text or a set of documents. Topic modeling identifies the keywords used in text to identify the subject of the article.

Use cases of topic modeling

- Large law firms use topic modeling to examine hundreds of documents during large litigations.
- Online media uses topic modeling to pick up trending topics across the web.
- Researchers use topic modeling for exploratory literature review.
- Businesses can determine which of their products are successful.
- Topic modeling helps anthropologists to determine the emergent issues and trends in a society based on the content people share on the web.

3. Named Entity Recognition (NER)

NER is a text analytics technique used for identifying named entities like people, places, organizations, and events in unstructured text. NER extracts nouns from the text and determines the values of these nouns.

Use cases of named entity recognition:

- NER is used to classify news content based on people, places, and organizations featured in them.
- Search and recommendation engines use NER for information retrieval.
- For large chain companies, NER is used to sort customer service requests and assign them to a specific city, or outlet.
- Hospitals can use NER to automate the analysis of lab reports.

4. Term frequency – inverse document frequency

TF-IDF is used to determine how often a term appears in a large text or group of documents and therefore that term's importance to the document. This technique uses an inverse document frequency factor to filter out frequently occurring yet non-insightful words, articles, propositions, and conjunctions.

5. Event extraction

This is a text analytics technique that is an advancement over the named entity extraction. Event extraction recognizes events mentioned in text content, for example, mergers, acquisitions, political moves, or important meetings. Event extraction requires an advanced understanding of the semantics of text content. Advanced algorithms strive to recognize not only events but the venue, participants, date, and time wherever applicable. Event extraction is a beneficial technique that has multiple uses across fields.

Use cases of event extraction

1. **Link analysis**

   This is a technique to understand "who met whom and when" through event extraction from communication over social media. This is used by law enforcement agencies to predict possible threats to national security.

2. **Geospatial analysis**

   When events are extracted along with their locations, the insights can be used to overlay them on a map. This is helpful in the geospatial analysis of the events.

3. **Business risk monitoring**

   Large organizations deal with multiple partner companies and suppliers. Event extraction techniques allow businesses to monitor the web to find out if any of their partners, like suppliers or vendors, are dealing with adverse events like lawsuits or bankruptcy.

Q17. Explain Steps involved in Text Analytics

Ans : **(Imp.)**

Text analytics is a sophisticated technique that involves several pre-steps to gather and cleanse the unstructured text. There are different ways in which text analytics can be performed. This is an example of a model workflow.

1. **Data gathering**

   Text data is often scattered around the internal databases of an organization, including in customer chats, emails, product reviews, service tickets and Net Promoter Score surveys. Users also generate external data in the form of blog posts, news, reviews, social media posts and web forum discussions. While the internal data is readily available for analytics, the external data needs to be gathered.

2. **Preparation of data**

   Once the unstructured text data is available, it needs to go through several preparatory steps before machine learning algorithms can analyze it. In most of the text analytics software, this step happens automatically. Text preparation includes several techniques using natural language processing as follows:

   - **Tokenization**

     In this step, the text analysis algorithms break the continuous string of text data into tokens or smaller units that make up entire words or phrases. For instance, character tokens could be each individual letter in this word: F-I-S-H. Or, you can break up by subword tokens: Fishing. Tokens represent the basis of all natural language processing. This step also discards all the unwanted contents of the text, including white spaces.

   - Part-of-speech-tagging: In this step, each token in the data is assigned a grammatical category like noun, verb, adjective, and adverb.

   - **Parsing**

     Parsing is the process of understanding the syntactical structure of the text. Dependency parsing and constituency parsing are two popular techniques used to derive syntactical structure.

   - Lemmatization and stemming: These are two processes used in data preparation to remove the suffixes and affixes associated with the tokens and retain its dictionary form or lemma.

   - Stopword removal: This is the phase when all the tokens that have frequent occurrence but bear no value in the text analytics. This includes words such as 'and', 'the' and 'a'.

3. **Text analytics**

   After the preparation of unstructured text data, text analytics techniques can now be performed to derive insights. There are several techniques used for text analytics. Prominent among them are text classification and text extraction.

   (i) Text classification

   This technique is also known as text categorization or tagging. In this step, certain tags are assigned to the text based on its meaning. For example, while analyzing customer reviews, tags like "positive" or "negative" are assigned. Text classification often is done using rule-based systems or machine learning-based systems. In rule-based systems, humans define the association between language pattern and a tag. "Good" may indicate positive review; "bad" may identify a negative review.

Machine learning systems use past examples or training data to assign tags to a new set of data. The training data and its volume are crucial, as larger sets of data helps the machine learning algorithms to give accurate tagging results. The main algorithms used in text classification are Support Vector Machines (SVM), Naive Bayes family of algorithms (NB), and deep learning algorithms.

(ii) Text extraction

This is the process of extracting recognizable and structured information from the unstructured input text. This information includes keywords, names of people, places and events. One of the simple methods for text extraction is regular expressions. However, this is a complicated method to maintain when the complexity of input data increases. Conditional Random Fields (CRF) is a statistical method used in text extraction. CRF is a sophisticated but effective way of extracting vital information from the unstructured text.

4.6 Web analytics

Q18. Explain briefly about Web Analytics.

Ans :

Web analytics is the process of analyzing the behavior of visitors to a website. This involves tracking, reviewing and reporting data to measure web activity, including the use of a website and its components, such as webpages, images and videos.

Data collected through web analytics may include traffic sources, referring sites, page views, paths taken and conversion rates. The compiled data often forms a part of customer relationship management analytics (CRM analytics) to facilitate and streamline better business decisions.

Web analytics enables a business to retain customers, attract more visitors and increase the dollar volume each customer spends.

Analytics can help in the following ways

➢ Determine the likelihood that a given customer will repurchase a product after purchasing it in the past.

➢ Personalize the site to customers who visit it repeatedly.

➢ Monitor the amount of money individual customers or specific groups of customers spend.

➢ Observe the geographic regions from which the most and the least customers visit the site and purchase specific products.

➢ Predict which products customers are most and least likely to buy in the future.

The objective of web analytics is to serve as a business metric for promoting specific products to the customers who are most likely to buy them and to determine which products a specific customer is most likely to purchase. This can help improve the ratio of revenue to marketing costs.

In addition to these features, web analytics may track the clickthrough and drilldown behavior of customers within a website, determine the sites from which customers most often arrive, and communicate with browsers to track and analyze online behavior. The results of web analytics are provided in the form of tables, charts and graphs.

Q19. Explain the Process of Web Analytics.

Ans : **(Imp.)**

The web analytics process involves the following steps:

1. Setting goals

The first step in the web analytics process is for businesses to determine goals and the end results they are trying to achieve. These goals can include increased sales, customer satisfaction and brand awareness. Business goals can be both quantitative and qualitative.

2. Collecting data

The second step in web analytics is the collection and storage of data. Businesses can collect data directly from a website or web analytics tool, such as Google Analytics. The data mainly comes from Hypertext Transfer Protocol requests -- including data at the network and application levels -- and can be combined with external data to interpret web usage. For example, a user's Internet Protocol address is typically associated with many factors, including geographic location and clickthrough rates.

3. **Processing data**

The next stage of the web analytics funnel involves businesses processing the collected data into actionable information.

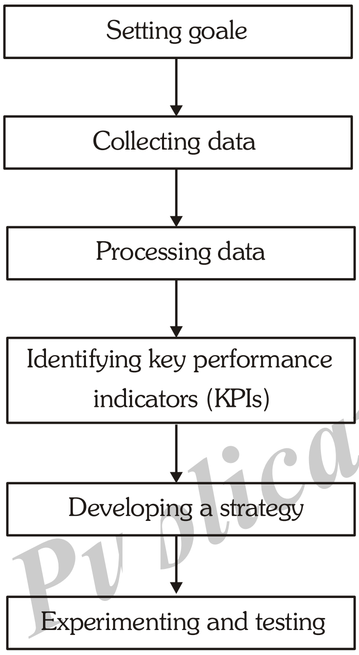

Setting goale
Collecting data
Processing data
Identifying key performance indicators (KPIs)
Developing a strategy
Experimenting and testing

Fig.: Process of Web Analytics

4. **Identifying key performance indicators (KPIs)**

In web analytics, a KPI is a quantifiable measure to monitor and analyze user behavior on a website. Examples include bounce rates, unique users, user sessions and on-site search queries.

5. **Developing a strategy**

This stage involves implementing insights to formulate strategies that align with an organization's goals. For example, search queries conducted on-site can help an organization develop a content strategy based on what users are searching for on its website.

6. **Experimenting and testing**

Businesses need to experiment with different strategies in order to find the one that yields the best results. For example, A/B testing is a simple strategy to help learn how an audience responds to different content. The process involves creating two or more versions of content and then displaying it to different audience segments to reveal which version of the content performs better.

Q20. What are the main categories of web analytics.

Ans :

The two main categories of web analytics are off-site web analytics and on-site web analytics.

1. **Off-site web analytics**

   The term off-site web analytics refers to the practice of monitoring visitor activity outside of an organization's website to measure potential audience. Off-site web analytics provides an industrywide analysis that gives insight into how a business is performing in comparison to competitors. It refers to the type of analytics that focuses on data collected from across the web, such as social media, search engines and forums.

2. **On-site web analytics**

   On-site web analytics refers to a narrower focus that uses analytics to track the activity of visitors to a specific site to see how the site is performing. The data gathered is usually more relevant to a site's owner and can include details on site engagement, such as what content is most popular. Two technological approaches to on-site web analytics include log file analysis and page tagging.

   Log file analysis, also known as log management, is the process of analyzing data gathered from log files to monitor, troubleshoot and report on the performance of a website. Log files hold records of virtually every action taken on a network server, such as a web server, email server, database server or file server.

   Page tagging is the process of adding snippets of code into a website's HyperText Markup Language code using a tag management system to track website visitors and their interactions across the website. These snippets of code are called tags. When businesses add these tags to a website, they can be used to track any number of metrics, such as the number of pages viewed, the number of unique visitors and the number of specific products viewed.

Q21. Discuss the various of tools of web analytics.

Ans : **(Imp.)**

Web analytics tools report important statistics on a website, such as where visitors came from, how long they stayed, how they found the site and their online activity while on the site. In addition to web analytics, these tools are commonly used for product analytics, social media analytics and marketing analytics.

Some examples of web analytics tools include the following:

(i) **Google Analytics**

Google Analytics is a web analytics platform that monitors website traffic, behaviors and conversions. The platform tracks page views, unique visitors, bounce rates, referral Uniform Resource Locators, average time on-site, page abandonment, new vs. returning visitors and demographic data.

(ii) **Optimizely**

Optimizely is a customer experience and A/B testing platform that helps businesses test and optimize their online experiences and marketing efforts, including conversion rate optimization.

(iii) **Kissmetrics**

Kissmetrics is a customer analytics platform that gathers website data and presents it in an easy-to-read format. The platform also serves as a customer intelligence tool, as it enables businesses to dive deeper into customer behavior and use this information to enhance their website and marketing campaigns.

(iv) **Crazy Egg**

Crazy Egg is a tool that tracks where customers click on a page. This information can help organizations understand how visitors interact with content and why they leave the site. The tool tracks visitors, heat maps and user session recordings.

Short Question & Answers

1. What is linear programming problem.

Ans :

In 1947, George dantzig and his associates, while working in the U.S. department of Air Force, observed that a large of military programming and planning problems could be formulated as maximizing/minimizing a linear form of profit/cost function whose variables were restricted to values satisfying a system of linear constraints.

A linear form is meant a thematical expression of the type $a_1x_1 + a_2x_2 + ... + a_nx_n$, where $a_1, a_2, ..., a_n$ are constants and $x_1, x_2, ..., x_n$ are variables. The term 'Programming' refers to the process of determining a particular programme or plan of action. So Linear Programming (L.P.) is one of the most important optimization (maximization/ minimization) techniques developed in the field of Operations Research.

The geneal LPP calls for optimizing (maximizing/minimizing) a linear function of variables called the '*Objective Function*' subject to a set of linear equations and / or inequalities called the '*constraints*' or *restrictions*.

2. Limitations of Linear Programming

Ans :

i) In some problems objective functions and constraints are not linear. Generally, in real life situations concerning business and industrial problems constraints are not linearly created to variables.

ii) There is no guarantee of getting integer valued solutions, for example, in finding out how may men and machines would be required to perform a particular job, rounding off the solution to the nearest integer will not give an optimal solution. Integer programming deals with such problems.

iii) Linear programming model does not take into consideration the effect of time and uncertainty. Thus the model should be defined in such a way that any change due to internal as well as external factors can be incorporated.

iv) Sometimes large-scale problems cannot be solved with linear programming techniques even when the computer facility is available. Such difficulty may be removed by decomposing the main problem into several small problems and then solving them separately.

v) Parameters appearing in the model are assumed to be constant. But, in real life situations they are neither constant not deterministic.

vi) Linear programming deals with only single objective, whereas in real life situations problems come across with multiobjectives.

3. Advantages of linear programming problem.

Ans :

i) It helps in organisation and study of the information in the same way that the scientific approach to the problem requires.

ii) With LP the execute builds into his planning a true reflection of the limitations and restrictions under which he must operate.

iii) Once a basic plan is arrived at through LP, it can be re-evaluated for changing conditions.

iv) Highlighting of bottlenecks in the production process is the striking advanages of this technique.

v) It provides flexibility in analysing a variety of multi-dimensional problems.

4. Explain Non Linear Programming?

Ans :

➢ Non Linear Programming (NLP). If an LP problem is feasible then, at least in theory, it can always be solved because. We know the solution is a "corner point": a point where lines or planes intersect.

- There are a finite number of possible solution points. The simplex algorithm will find that point. Also, a very informative sensitivity analysis is relatively easy to obtain for LP problems. But in many interesting, real-world problems, the objective function may not be a linear function, or some of the constraints may not be linear constraints
- Optimization problems that involve nonlinearities are called nonlinear programming (NLP) problems. Many NLPs do not have any constraints. They are called unconstrained NLPs. Solutions to NLPs are found using search procedures.
- Solutions are more difficult to determine, compared to LPs. One problem is difficulty in distinguishing between a local and global minimum or maximum point.
- Nonlinear programming is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear.

5. Cutting Plane Method

Ans :

- In mathematical optimization, the cutting-plane method is any of a variety of optimization
- Methods that iteratively refine a feasible set or objective function by means of linear inequalities, termed cuts. Such procedures are commonly used to find integer solutions to mixed integer linear programming (MILP) problems, as well as to solve general, not necessarily differentiable convex optimization problems. The use of cutting planes to solve MILP was introduced by Ralph E. Gomory.
- Cutting plane methods for MILP work by solving a non-integer linear program, the linear relaxation of the given integer program.
- The theory of Linear Programming dictates that under mild assumptions (if the linear program has an optimal solution, and if the feasible region does not contain a line), one can always find an extreme point or a corner point that is optimal.

6. Decision analysis

Ans :

The term decision analysis was coined in 1964 by Ronald A. Howard, professor of management science and engineering at Stanford University. Decision analysis refers to a systematic, quantitative and interactive approach to addressing and evaluating important choices confronted by organisations in the private and public sector. Decision analysis is interdisciplinary and draws on theories from the fields of psychology, economics, and management science. It utilises a variety of tools which include models for decision-making under conditions of uncertainty or multiple objectives; techniques of risk analysis and risk assessment; experimental and descriptive studies of decision-making behaviour; economic analysis of competitive and strategic decisions; techniques for facilitating decision-making by groups; and computer modeling software and expert systems for decision support.

7. What is Text Analytics

Ans :

Text analytics combines a set of machine learning, statistical and linguistic techniques to process large volumes of unstructured text or text that does not have a predefined format, to derive insights and patterns. It enables businesses, governments, researchers, and media to exploit the enormous content at their disposal for making crucial decisions. Text analytics uses a variety of techniques – sentiment analysis, topic modelling, named entity recognition, term frequency, and event extraction.

Choose the Correct Answer

1. For a linear programming equations, convex set of equations is included in region of ________. [a]

   (a) feasible solutions (b) disposed solutions

   (c) profit solutions (d) loss solutions

2. In linear programming, objective function and objective constraints are ________. [b]

   (a) solved (b) linear

   (c) quadratic (d) adjacent

3. In graphical solutions of linear inequalities, solution can be divided into ________. [b]

   (a) one subset (b) two subsets

   (c) three subsets (d) four subsets

4. Objective of linear programming for an objective function is to ________. [a]

   (a) maximize or minimize (b) subset or proper set modeling

   (c) row or column modeling (d) adjacent modeling

5. Linear programming is used to optimize mathematical procedure and is ________. [a]

   (a) Subset of mathematical programming

   (b) Dimension of mathematical programming

   (c) Linear mathematical programming

   (d) All of above

6. One of the models of operation research includes [c]

   (a) LPP (b) Networking

   (c) Game theory (d) Transportation

7. The graphical method of LP problem uses ________. [d]

   (a) Constraint equations (b) Objective function equations

   (c) Linear equations (d) All of the above

8. Alternative solutions exist in an LP model when ________. [a]

   (a) Objective function equation is parallel to one of the constraints

   (b) One of the constraints is redundant

   (c) Two constraints are parallel

   (d) Adding another constraint/variable.

9. One of the following is an assumption of linear programming model [c]

   (a) $\geq$ or $\leq$ constraints (b) Maximize profits

   (c) Divisibility (d) Minimize cost

10. If a non-redundant constraint is removed from a LP problem, then [a]

(a) Feasible region will become larger

(b) Feasible region will become smaller

(c) Solution will become infeasible

(d) The solution is unbounded

11. ________ is the process of transforming unstructured text into a structured format to identify meaningful patterns and new insights. [b]

(a) Data mining (b) Text mining

(c) File mining (d) Deep mining

12. In which database, data is a blend between structured and unstructured data formats? [c]

(a) Full-structured data (b) Partial-structured data

(c) Semi-structured data (d) Uni-structured data

13. The process of breaking out long-form text into sentences and words called? [d]

(a) Stem (b) Cluster

(c) Bag (d) Tokens

14. Text mining is being used by large media companies, to clarify information and to provide readers with greater search experiences. [a]

(a) TRUE (b) FALSE

(c) Can be true or false (d) Can not say

15. Typical text mining tasks include? [d]

(a) Text categorization (b) Text clustering

(c) Entity relation modeling (d) All of the above

16. Which of the following technique is not a part of flexible text matching? [c]

(a) Soundex (b) Metaphone

(c) Keyword Hashing (d) Edit Distance

Fill in the Blanks

1. The general LPP calls for optimizing (maximizing/minimizing) a linear function of variables called the __________.
2. LPP stands for __________.
3. Simplex algorithm was originally proposed by __________.
4. The best use of linear programming technique is to find an optimal use of ________.
5. __________ programming is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear.
6. MILP stands for __________.
7. __________ methods for general convex continuous optimization.
8. The term decision analysis was coined in __________.
9. __________ is interdisciplinary and draws on theories from the fields of psychology, economics, and management science.
10. EMV stands for __________.

Answers

1. Objective Function
2. Linear Programming Problem
3. G.B Dantzig in 1948
4. Money, manpower and machine
5. Nonlinear
6. Mixed integer linear programming
7. Cutting-plane
8. 1964
9. Decision analysis
10. Expected Monetary Value

UNIT V

PROGRAMMING USING R

R Environment, R packages, Reading and Writing data in R, R functions, Control Statements, Frames and Subsets, Managing and Manipulating data in R.

5.1 Programming Using R

Q1. What is R? Explain the features of R?

Ans :

R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalog of statistical and graphical methods. It includes machine learning algorithm, linear regression, time series, statistical inference to name a few. Most of the R libraries are written in R, but for heavy computational task, C, C++ and Fortran codes are preferred.

R is not only entrusted by academic, but many large companies also use R programming language, including Uber, Google, Airbnb, Facebook and so on.

Data analysis with R is done in a series of steps; programming, transforming, discovering, modeling and communicate the results

- **Program**: R is a clear and accessible programming tool
- **Transform**: R is made up of a collection of libraries designed specifically for data science
- **Discover**: Investigate the data, refine your hypothesis and analyze them
- **Model**: R provides a wide array of tools to capture the right model for your data
- **Communicate**: Integrate codes, graphs, and outputs to a report with R Markdown or build Shiny apps to share with the world

Features of R

As R is a leading programming language. There are so many features of **R programming** which makes it important to learn. Let's discuss them one by one.

Statistical Features of R

1. **R has some topical relevance**
   - It is free, open source software.
   - R is available under free software Foundation.
2. **R has some statistical features**
   - **Basic Statistics :** Mean, variance, median.
   - **Static graphics :** Basic plots, graphic maps.
   - **Probability distributions :** Beta, Binomial.

   Any Doubt yet in Why Learn R programming? Please Comment.

Programming Features of R

1. **R has some topical relevance**
   - Data inputs such as data type, **importing data**, keyboard typing.
   - Data Management such as data variables, operators.

2. **R has some programming features**

   - **Distributed Computing** – Distributed computing is an open source, high-performance platform for the R language. It splits tasks between multiple processing nodes to reduce execution time and analyze large datasets.

   - **R packages** – **R packages** are a collection of **R functions**, compiled code and sample data. By default, **R installs** a set of packages during installation.

Q2. Explain the basic tips for using R?

Ans :

- R is command-line driven. It requires you to type or copy-and-paste commands after a command prompt (>) that appears when you open R. After typing a command in the R console and pressing **Enter** on your keyboard, the command will run. If your command is not complete, R issues a continuation prompt (signified by a plus sign: +). Alternatively you can write a script in the script window, and select a command, and click the **Run** button.

- R is case sensitive. Make sure your spelling and capitalization are correct.

- Commands in R are also called functions. The basic format of a function in R is: function. name(argument, options).

- The up arrow (^) on your keyboard can be used to bring up previous commands that you've typed in the R console.

- The $ symbol is used to select a particular column within the table (e.g., table$column).

- Any text that you do not want R to act on (such as comments, notes, or instructions) needs to be preceded by the # symbol (a.k.a. hash-tag, comment, pound, or number symbol). R ignores the remainder of the script line following

  For example: Plot(x, y) # This text will not affect the plot function because of the comment.

Q3. What is R environment.

Ans :

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- An effective data handling and storage facility,
- A suite of operators for calculations on arrays, in particular matrices,
- A large, coherent, integrated collection of intermediate tools for data analysis,
- Graphical facilities for data analysis and display either on-screen or on hardcopy, and
- A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

Q4. Explain the various types of operators in R program.

Ans :

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Miscellaneous Operators

1. Arithmetic Operators

Following table shows the arithmetic operators supported by R language. The operators act on each element of the vector.

Operator	Description	Example

+	Adds two vectors	v <- c(2, 5.5, 6)
		t <- c(8, 3, 4)
		print(v+t)
		It produces the following result:
		[1] 10.0 8.5 10.0

–	Subtracts second vector from the first	v <- c(2, 5.5, 6)
		t <- c(8, 3, 4)
		print(v-t)
		It produces the following result:
		[1]-6.0 2.5 2.0

*	Multiplies both vectors	v <- c(2, 5.5, 6)
		t <- c(8, 3, 4)
		print(v*t)
		It produces the following result:
		[1] 16.0 16.5 24.0

/	Divides the first vector with the second	v <- c(2,5.5, 6)
		t<- c(8, 3,4) print(v/t)
		When we execute the above code, it produces the following result:
		[1] 0.250000 1.833333 1.500000

%%	Gives the remainder of the first vector with the second	v <- c(2,5.5,6)
		t <- c(8, 3, 4)
		print(v%%t)
		it produces the following result -
		[1] 2.0 2.5 2.0

% / %	The result of division of first vector with second (quotient)	v <- c(2,5.5,6)
		t <- c(8, 3,4)
		print(v%/%t)
		It produces the following result:
		[1] 0 1 1

^	The first vector raised to the exponent of, second vector	v <- c(2,5.5,6)
		t<- c(8, 3,4)
		print(v ^ t)
		It produces the following result:
		[1] 256.000 166.375 1296.000

2. Relational Operators

Following table shows the relational operators supported by R language. Each element of the first vector is compared with the corresponding element of the second vector. The result of comparison is a Boolean value.

Operator	Description	Example

>	Checks if each element of the first vector is greater than the corresponding element of the second vector.
	It produces the following result:
		v <- c(2,5.5,6,9)
		t <_ $_c$(8,2.5,14,9)
		print(v>t)
		[1] FALSE TRUE FALSE FALSE

<	Checks if each element of the first vector is less than the corresponding element of the second vector.
	It produces the following result:
		v <- c(2,5.5,6,9)
		t <- c(8,2.5,14,9)
		print(v < t)
		[1] TRUE FALSE TRUE FALSE

= =	Checks if each element of the first vector is equal to the corresponding element of the second vector.
	It produces the following result:
		v <- c(2,5.5,6,9)
		t <- c(8,2.5,14,9)
		print(v = = t)
		[1]FALSE FALSE FALSE TRUE

<=	Checks if each element of the first vector is less than or equal to the corresponding element of the second vector.	v <- c(2,5.5,6,9)
		t <- c(8,2.5,14,9)
		print(v<=t)
		It produces the following result:
		[1] TRUE FALSE TRUE TRUE

>=	Checks if each element of the first vector is greater than or equal to the corresponding element of the second	v <- c(2,5.5,6,9)
		t <- c(8,2.5,14,9)
		print(v>=t)
		It produces the following result:
		[1] FALSE TRUE FALSE TRUE

!=	Checks if each element of the first vector is unequal to the corresponding element of the second vector.	v<-c(2,5.5,6,9)
		t <- c(8,2.5,14,9)
		print(v!=t)
		It produces the following result:
		[1] TRUE TRUE TRUE FALSE

3. Logical Operators

Following table shows the logical operators supported by R language. It is applicable only to vectors of type logical, numeric or complex. All numbers greater than 1 are considered as logical value TRUE.

Each element of the first vector is compared with the corresponding element of the second vector. The result of comparison is a Boolean value.

Operator	Description	Example

&	It is called Element-wise Logical AND operator. It combines each element of the first vector with the corresponding element of the second vector and gives a output TRUE if both the elements are TRUE.	v <- c(3, l, TRUE, 2 + 3i)
		t <- c(4, l, FALSE, 2 +3i)
		print(v&t)
		It produces the following result:
		[1] TRUE TRUE FALSE TRUE

|	It is called Element-wise Logical OR operator. It combines each element of the first vector with the corresponding element of the second vector and gives a output TRUE if one the elements is TRUE.	v <- c(3,0,TRUE,2+2i)
		t <- c(4,0,FALSE,2+3i)
		print(v|t)
		It produces the following result:
		[1] TRUE FALSE TRUE TRUE

!	It is called Logical NOT operator. It takes each element of the vector and gives the opposite logical value.	v <- c(3,0,TRUE,2+2i)
		print(!v)
		It produces the following result:
		[1] FALSE TRUE FALSE FALSE

The logical operator && and || considers only the first element of the vectors and gives a vector of single element as output.

Operator	Description	Example

&&	It is called Logical AND operator. It takes first element of both the vectors and gives the TRUE only if both are TRUE.	v <- c(3,0,TRUE,2+2i)
		t <-c(1,3,TRUE,2+3i)
		print(v&&t)
		It produces the following result:
		[1] TRUE

||	It is called Logical OR operator. It takes first element of both the vectors and gives the TRUE if one of them is TRUE.	v <- c(0,0,TRUE,2+2i)
		t <-c(0,3,TRUE,2+3i)
		print(v||t)
		It produces the following result:
		[1] FALSE

Assignment Operators

These operators are used to assign values to vectors.

Operator	Description	Example

<–
or
=
or
<<–
	It is called Left Assignment.	v1 <- c(3,l,TRUE,2+3i)
		v2 <<-c(3,l,TRUE,2+3i)
		v3 = c(3,l,TRUE,2+3i)
		print(vl)
		print(v2)
		print(v3)
		It produces the following result:
		[1] 3+0i 1+Oi 1+Oi 2+3i
		[1] 3+0i 1+Oi 1+Oi 2+3i
		[1] 3+0i 1+Oi 1+Oi 2+3i

–> or –>>	It is called Right Assignment.	c(3, l, TRUE, 2+3i) -> v1 c(3,l,TRUE,2+3i) –>> v2 print(vl) print(v2) It produces the following result: [1] 3+0i 1+Oi 1+Oi 2+3 i [1] 3+0i 1+Oi 1+Oi 2+3i

Q5. What Is A Package?

Ans :

A package is a suitable way to organize your own work and, if you want to, share it with others. Typically, a package will include code (not only R code!), documentation for the package and for the functions inside, some tests to check everything works as it should, and data sets.

The basic information about a package is provided in the DESCRIPTION file, Which means a file contains basic information about the package where you can find out what the package does, who the author is, what version the documentation belongs to, the date, the type of license its use, and the package dependencies.

Besides finding the DESCRIPTION files such as cran.r-project.org or stat.ethz.ch, you can also access the description file inside R with the command package Description (“package”), via the documentation of the package help(package = “package”), or online in the repository of the package.

5.2 R Package

Q6. What is R package?

Ans :

R packages are a collection of R functions, complied code and sample data. They are stored under a directory called **”library”** in the R environment. By default, R installs a set of packages during installation.

More packages are added later, when they are needed for some specific purpose. When we start the R console, only the default packages are available by default. Other packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them.

All the packages available in R language are listed at R Packages.

Below is a list of commands to be used to check, verify and use the R packages.

Check Available R Packages

Get library locations containing R packages

.libPaths()

When we execute the above code, it produces the following result. It may vary depending on the local settings of your pc.

```
[2] "C:/Program Files/R/R-3.2.2/library"
```

Get the list of all the packages installed

library()

When we execute the above code, it produces the following result. It may vary depending on the local settings of your pc.

```
Packages in library 'C:/Program Files/R/R-3.2.2/library':
   base           The R Base Package
   boot           Bootstrap Functions (Originally by Angelo Canty for S)
   class          Functions for Classification
   cluster        "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.
   codetools      Code Analysis Tools for R
   compiler       The R Compiler Package
   datasets       The R Datasets Package
   foreign        Read Data Stored by 'Minitab', 'S', 'SAS',
                  'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...
   graphics       The R Graphics Package
   grDevices      The R Graphics Devices and Support for Colours and Fonts
   grid           The Grid Graphics Package
   KernSmooth     Functions for Kernel Smoothing Supporting Wand & Jones (1995)
   lattice        Trellis Graphics for R
   MASS           Support Functions and Datasets for Venables and Ripley's MASS
   Matrix         Sparse and Dense Matrix Classes and Methods
   methods        Formal Methods and Classes
```

mgcv	Mixed GAM Computation Vehicle with GCV/AIC/REML Smoothness Estimation
nlme	Linear and Nonlinear Mixed Effects Models
nnet	Feed-Forward Neural Networks and Multinomial Log-Linear Models
parallel	Support for Parallel computation in R
rpart	Recursive Partitioning and Regression Trees
spatial	Functions for Kriging and Point Pattern Analysis
splines	Regression Spline Functions and Classes
stats	The R Stats Package
stats4	Statistical Functions using S4 Classes
survival	Survival Analysis
tcltk	Tcl/Tk Interface
tools	Tools for Package Development
utils	The R Utils Package

Get all packages currently loaded in the R environment

```
search()
```

When we execute the above code, it produces the following result. It may vary depending on the local settings of your pc.

```
[1]  ".GlobalEnv"          "package:stats"     "package:graphics"
[4] "package:grDevices"   "package:utils"     "package:datasets"
[7]  "package:methods"     "Autoloads"          "package:base"
```

Q7. How to Install a R New Package.

Ans :

There are two ways to add new R packages. One is installing directly from the CRAN directory and another is downloading the package to your local system and installing it manually.

(i) Install directly from CRAN

The following command gets the packages directly from CRAN webpage and installs the package in the R environment. You may be prompted to choose a nearest mirror. Choose the one appropriate to your location.

```
install.packages("Package Name")

# Install the package named "XML".

install.packages("XML")
```

ii) Install package manually

Go to the link R Packages to download the package needed. Save the package as a **.zip** file in a suitable location in the local system.

Now you can run the following command to install this package in the R environment.

install.packages(file_name_with_path, repos = NULL, type = "source")

# Install the package named "XML"

install.packages("E:/XML_3.98-1.3.zip", repos = NULL, type = "source")

iii) Load Package to Library

Before a package can be used in the code, it must be loaded to the current R environment. You also need to load a package that is already installed previously but not available in the current environment.

A package is loaded using the following command " library("package Name", lib.loc = "path to library")

# Load the package named "XML"

install.packages("E:/XML_3.98-1.3.zip", repos = NULL, type = "source")

5.3 Reading and Writing Data in R

Q8. How the data can be read and write in R?

Ans :

Reading Data in R

For reading, (importing) data into R following are some functions.

- read.table(), and read.csv(), for reading tabular data
- readLines() for reading lines of a text file
- source() for reading in R code files (inverse of dump)
- dget() for reading in R code files (inverse of dput)
- load() for reading in saved workspaces.

Writing Data in R

Following are few functions for writing (exporting) data to files.

- write.table(), and write.csv() exports data to wider range of file format including csv and tab-delimited.
- writeLines() write text lines to a text-mode connection.
- dump() takes a vector of names of R objects and produces text representations of the objects on a file (or connection). A dump file can usually be sourced into another R session.
- dput() writes an ASCII text representation of an R object to a file (or connection) or uses one to recreate the object.
- save() writes an external representation of R objects to the specified file.

Reading data files with read.table()

The read.table() function is one of the most commonly used functions for reading data into R. It has a few important arguments.

- file, the name of a file, or a connection
- header, logical indicating if the file has a header line
- sep, a string indicating how the columns are separated
- colClasses, a character vector indicating the class of each column in the data set
- nrows, the number of rows in the dataset
- comment.char, a character string indicating the comment character
- skip, the number of lines to skip from the beginning
- stringsAsFactors, should character variables be coded as factors?

read.table() and read.csv() Examples

```
> data<-read.table("foo.txt")
> data<-read.table("D:\\datafiles\\mydata.txt")
> data<-read.csv("D:\\datafiles\\mydata.csv")
```

R will automatically skip lines that begin with a #, figure out how many rows there are (and how much memory needs to be allocated). R also figure out what type of variable is in each column of the table.

Writing data files with write.table()

Following are few important arguments usually used in write.table() function.

- x, the object to be written, typically a data frame
- file, the name of the file which the data are to be written to
- sep, the field separator string
- col.names, a logical value indicating whether the column names of x are to be written along with x, or a character vector of column names to be written
- row.names, a logical value indicating whether the row names of x are to be written along with x, or a character vector of row names to be written
- na, the string to use for missing values in the data

write.table() and write.csv() Examples

```
> x <- data.frame(a = 5, b = 10, c = pi)
> write.table(x, file = "data.csv", sep = ",")
> write.table(x, "c:\\mydata.txt", sep = "\t")
> write.csv(x, file = "data.csv").
```

5.4 R Functions

Q9. What is R Function? Explain the components of R Functions.

Ans :

A function is a set of statements organized together to perform a specific task. R has a large number of in-built functions and the user can create their own functions. In R, a function is an object so the R interpreter is able to pass control to the function, along with arguments that may be necessary for the function to accomplish the actions.

The function in turn performs its task and returns control to the interpreter as well as any result which may be stored in other objects.

Definition

An R function is created by using the keyword **function**. The basic syntax of an R function definition is as follows :

```
function_name <- function(arg_1, arg_2, ...) {
      Function body
}
```

Components of R

The different parts of a function are:

- **Function Name:** This is the actual name of the function. It is stored in R environment as an object with this name.
- **Arguments:** An argument is a placeholder. When a function is invoked, you pass a value to the argument. Arguments are optional; that is, a function may contain no arguments. Also arguments can have default values.
- **Function Body:** The function body contains a collection of statements that defines what the function does.
- **Return Value:** The return value of a function is the last expression in the function body to be evaluated.

R has many **in-built** functions which can be directly called in the program without defining them first. We can also create and use our own functions referred as **user defined** functions.

Q10. Explain different types of functions?

Ans :

i) Built-in Function

Simple examples of in-built functions are **seq()**, **mean()**, **max()**, **sum(x)** and **paste(...)** etc. They are directly called by user written programs. You can refer most widely used R functions.

```
# Create a sequence of numbers from 32          to 44.
    print(seq(32,44))
# Find mean of numbers from 25 to 82.
    print(mean(25:82))
# Find sum of numbers frm 41 to 68.
    print(sum(41:68))
```

When we execute the above code, it produces the following result “

```
[1] 32 33 34 35 36 37 38 39 40 41 42 43 44
[1] 53.5
[1] 1526
```

ii) User-defined Function

We can create user-defined functions in R. They are specific to what a user wants and once created they can be used like the built-in functions. Below is an example of how a function is created and used.

```
# Create a function to print squares of numbers in sequence.
```

```
new.function <- function(a) {
for(i in 1:a) {
    b <- i^2
    print(b)
}
}
```

iii) Calling a Function

```
# Create a function to print squares of numbers in sequence.
new.function <- function(a) {
    for(i in 1:a) {
        b <- i^2
    print(b)
}
}
# Call the function new.function supplying 6 as an argument.
new.function(6)
```

When we execute the above code, it produces the following result “

```
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
```

iv) Calling a Function without an Argument

```
# Create a function without an argument.
new.function <- function() {
    for(i in 1:5) {
        print(i^2)
    }
}
# Call the function without supplying an argument.
new.function()
```

When we execute the above code, it produces the following result “

```
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
```

v) Calling a Function with Argument Values (by position and by name)

The arguments to a function call can be supplied in the same sequence as defined in the function or they can be supplied in a different sequence but assigned to the names of the arguments.

```
# Create a function with arguments.
    new.function <- function(a,b,c) {
        result <- a * b + c
    print(result)
}
# Call the function by position of arguments.
new.function(5,3,11)
# Call the function by names of the arguments.
new.function(a = 11, b = 5, c = 3)
```

When we execute the above code, it produces the following result “

```
[1] 26
[1] 58
```

vi) Calling a Function with Default Argument

We can define the value of the arguments in the function definition and call the function without supplying any argument to get the default result. But we can also call such functions by supplying new values of the argument and get non default result.

```
# Create a function with arguments.
new.function <- function(a = 3, b = 6) {
result <- a * b
print(result)
}
# Call the function without giving any argument.
new.function()
# Call the function with giving new values of the argument.
new.function(9,5)
```

When we execute the above code, it produces the following result “

```
[1] 18
[1] 45
```

vii) Lazy Evaluation of Function

Arguments to functions are evaluated lazily, which means so they are evaluated only when needed by the function body.

```
# Create a function with arguments.
     new.function <- function(a, b) {
     print(a^2)
     print(a)
     print(b)
}
# Evaluate the function without supplying one of the arguments.
new.function(6)
```

When we execute the above code, it produces the following result “

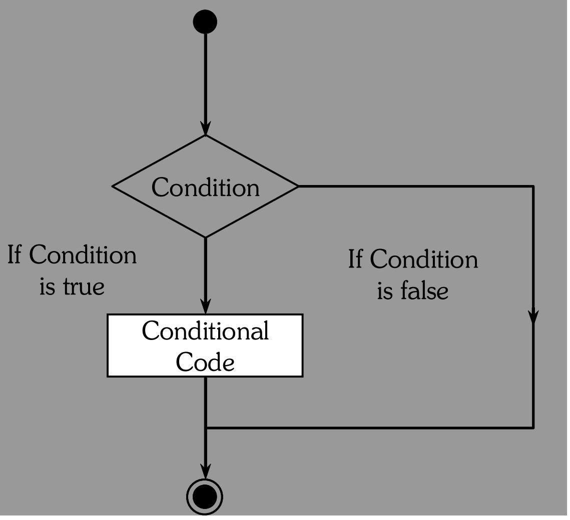
```
[1] 36
[1] 6
Error in print(b) : argument “b” is missing, with no default.
```

5.5 Control Statements

Q11. Explain briefly about control statements.

Ans :

Looping is similiar to creating functions in that they are merely a means to automate a certain multi step process by organizing sequences of R expressions. R consists of several loop control statements which allow you to perform repetititve code processes with different intentions and allow these automated expressions to naturally respond to features of your data. Consequently, learning these loop control statements will go a long ways in reducing code redundancy and becoming a more efficient data wrangler.



- **if** statement for conditional programming
- **if...else** statement for conditional programming
- for loop to iterate over a fixed number of iterations
- while loop to iterate until a logical statement returns FALSE
- repeat loop to execute until told to break
- break/next arguments to exit and skip interations in a loop.

Q12. Explain briefly about if statement.

Ans :

The conditional if statement is used to test an expression. If the test_expression is TRUE, the statement gets executed. But if it’s FALSE, nothing happens.

```
# syntax of if statement
if (test_expression) {
      statement
}
```

The following is an example that tests if any values in a vector are negative. Notice there are two ways to write this if statement; since the body of the statement is only one line you can write it with or without curly braces. I recommend getting in the habit of using curly braces, that way if you build onto if statements with additional functions in the body or add an else statement later you will not run into issues with unexpected code procedures.

```
x <- c(8, 3, -2, 5)
# without curly braces
if(any(x < 0)) print(“x contains negative numbers”)
## [1] “x contains negative numbers”
# with curly braces produces same result
if(any(x < 0)){
      print(“x contains negative numbers”)
}
## [1] “x contains negative numbers”
# an if statement in which the test expression is FALSE
# does not produce any output
y <- c(8, 3, 2, 5)
if(any(y < 0)){
print(“y contains negative numbers”)
}.
```

Q13. Explain briefly about if.... else statement?

Ans :

The conditional if...else statement is used to test an expression similar to the if statement. However, rather than nothing happening if the test_expression is FALSE, the else part of the function will be evaluated.

```
# syntax of if...else statement
if (test_expression) {
      statement 1
} else {
      statement 2
}
```

The following extends the previous example illustrated for the if statement in which the if statement tests if any values in a vector are negative; if TRUE it produces one output and if FALSE it produces the else output.

```
# this test results in statement 1 being executed
x <- c(8, 3, -2, 5)
if(any(x < 0)){
      print(“x contains negative numbers”)
} else{
      print(“x contains all positive numbers”)
}
## [1] “x contains negative numbers”
# this test results in statement 2 (or the else statement) being executed
y <- c(8, 3, 2, 5)
if(any(y < 0)){
      print(“y contains negative numbers”)
} else{
      print(“y contains all positive numbers”)
}
## [1] “y contains all positive numbers”
```

Simple if...else statements, as above, in which only one line of code is being executed in the statements can be written in a simplified alternative manner. These alternatives are only recommended for very short if...else code:

```
x <- c(8, 3, 2, 5)

# alternative 1

if(any(x < 0)) print("x contains negative numbers")
else print("x contains all positive numbers")

## [1] "x contains all positive numbers"

# alternative 2 using the ifelse function

ifelse(any(x < 0), "x contains negative numbers",
"x contains all positive numbers")

## [1] "x contains all positive numbers"
```

We can also nest as many if...else statements as required (or desired). For example:

```
# this test results in statement 1 being executed

x <- 7

if(x >= 10){
     print("x exceeds acceptable tolerance levels")
} else if(x >= 0 & x < 10){
     print("x is within acceptable tolerance levels")
} else {
      print("x is negative")
}

## [1] "x is within acceptable tolerance levels"
```

Q14. Define loop? Explain different kinds of loops in R programming.

Ans :

A loop statement allows us to execute a statement or group of statements multiple times. The following is the general form of a loop statement in most of the programming languages:

- **repeat loop:** Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
- **while loop:** Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
- **for loop:** Like a while statement, except that it tests the condition at the end of the loop body.

Q15. Explain briefly about loop statement?

Ans :

The for loop is used to execute repetitive code statements for a particular number of times. The general syntax is provided below where i is the counter and as i assumes each sequential value defined (1 through 100 in this example) the code in the body will be performed for that ith value.

```
# syntax of for loop
for(i in 1:100) {
      <do stuff here with i>
}
```

For example, the following for loop iterates through each value (2010, 2011, ..., 2016) and performs the paste and print functions inside the curly brackets.

```
for (i in 2010:2016){
      output <- paste("The year is", i)
       print(output)
}
       ## [1] "The year is 2010"
       ## [1] "The year is 2011"
       ## [1] "The year is 2012"
       ## [1] "The year is 2013"
       ## [1] "The year is 2014"
       ## [1] "The year is 2015"
       ## [1] "The year is 2016"
```

If you want to perform the for loop but have the outputs combined into a vector or other data structure than you can initiate the output data structure prior to the for loop. For instance, if we want to have the previous outputs combined into a single vector x we can initiate x first and then append the for loop output to x.

```
x <- NULL
for (i in 2010:2016){
      output <- paste("The year is", i)
       x <- append(x, output)
}
x
## [1] "The year is 2010" "The year is 2011" "The year is 2012" "The year is 2013"
## [5] "The year is 2014" "The year is 2015" "The year is 2016"
```

However, an important lesson to learn is that R is not efficient at *growing* data objects. As a result, it is more efficient to create an empty data object and *fill* it with the for loop outputs. In the previous example we *grew* x by appending new values to it. A more efficient practice is to initiate a vector (or other data structure) of the right size and fill the elements. In the example that follows, we create the vector x of the right size and then fill in each element within the for loop. Although this inefficiency is not noticed in this small example, when you perform larger repetitions it will become noticable so you might as well get in the habit of *filling* rather than *growing*.

```
x <- vector(mode = "numeric", length = 7)
counter <- 1
for (i in 2010:2016){
      output <- paste("The year is", i)
       x[counter] <- output
       counter <- counter + 1
}
x
## [1] "The year is 2010" "The year is 2011" "The year is 2012" "The year is 2013"
## [5] "The year is 2014" "The year is 2015" "The year is 2016"
```

Another example in which we create an empty matrix with 5 rows and 5 columns. The for loop then iterates over each column (note how *i* takes on the values 1 through the number of columns in the my.mat matrix) and takes a random draw of 5 values from a poisson distribution with mean *i* in column *i*:

```
my.mat <- matrix(NA, nrow = 5, ncol = 5)
for(i in 1:ncol(my.mat)){
      my.mat[, i] <- rpois(5, lambda = i)
}
my.mat
##       [,1] [,2] [,3] [,4] [,5]
## [1,]    0    2    1    7    1
## [2,]    1    2    2    3    9
## [3,]    2    1    5    6    6
## [4,]    2    1    5    2   10
## [5,]    0    2    2    2    4
```

Q16. Explain briefly about while loop?

Ans :

While loops begin by testing a condition. If it is true, then they execute the statement. Once the statement is executed, the condition is tested again, and so forth, until the condition is false, after which the loop exits. It's considered a best practice to include a counter object to keep track of total iterations

```
# syntax of while loop
counter <- 1
while(test_expression) {
      statement
       counter <- counter + 1
}
```

while loops can potentially result in infinite loops if not written properly; therefore, you must use them with care. To provide a simple example to illustrate how similiar for and while loops are:

```
counter <- 1
while(counter <= 10) {
        print(counter)
        counter <- counter + 1
}
# this for loop provides the same output
counter <- vector(mode = "numeric", length =
10)
for(i in 1:length(counter)) {
        print(i)
}
```

The primary difference between a for loop and a while loop is: a for loop is used when the number of iterations a code should be run is known where a while loop is used when the number of iterations is not known. For instance, the following takes value x and adds or subtracts 1 from the value randomly until x exceeds the values in the test expression. The output illustrates that the code runs 14 times until x exceeded the threshold with the value 9.

```
counter <- 1
x <- 5
set.seed(3)
while(x >= 3 && x <= 8 ) {
        coin <- rbinom(1, 1, 0.5)
        if(coin == 1) { ## random walk
                x <- x + 1
        } else {
                x <- x - 1
        }
        cat("On iteration", counter, ", x =", x, '\n')
        counter <- counter + 1
}
## On iteration 1 , x = 4
## On iteration 2 , x = 5
## On iteration 3 , x = 4
## On iteration 4 , x = 3
## On iteration 5 , x = 4
## On iteration 6 , x = 5
## On iteration 7 , x = 4
## On iteration 8 , x = 3
## On iteration 9 , x = 4
## On iteration 10 , x = 5
## On iteration 11 , x = 6
## On iteration 12 , x = 7
## On iteration 13 , x = 8
## On iteration 14 , x = 9
```

Q17. Explain briefly about repeat loop?

Ans :

A repeat loop is used to iterate over a block of code multiple number of times. There is test expression in a repeat loop to end or exit the loop. Rather, we must put a condition statement explicitly inside the body of the loop and use the break function to exit the loop. Failing to do so will result into an infinite loop.

```
# syntax of repeat loop
counter <- 1
repeat {
        statement

        if(test_expression){
                break
        }
        counter <- counter + 1
}
```

For example ,say we want to randomly draw values from a uniform distribution between 1 and 25. Furthermore, we want to continue to draw values randomly until our sample contains at least each integer value between 1 and 25; however, we do not care if we've drawn a particular value multiple times. The following code repeats the random draws of values between 1 and 25 (in which we round). We then include an if statement to check if all values between 1 and 25 are present in our sample. If so, we use the break statement to exit the loop. If not, we add to our counter and let the loop repeat until the conditional ifstatement is found to be true. We can then check the counter object to assess how many iterations were required to reach our conditional requirement.

```
counter <- 1
x <- NULL
repeat {
    x <- c(x, round(runif(1, min = 1, max =
25)))

    if(all(1:25 %in% x)){
        break
    }

    counter <- counter + 1
}
counter
## [1] 75
```

break/next arguments

The break argument is used to exit a loop immediately, regardless of what iteration the loop may be on. break arguments are typically embedded in an if statement in which a condition is assessed, if TRUE break out of the loop, if FALSE continue on with the loop. In a nested looping situation, where there is a loop inside another loop, this statement exits from the innermost loop that is being evaluated.

In this example, the for loop will iterate for each element in x; however, when it gets to the element that equals 3 it will break out and end the for loop process.

```
x <- 1:5
for (i in x) {
    if (i == 3){
        break
        }
    print(i)
}
## [1] 1
## [1] 2
```

The next argument is useful when we want to skip the current iteration of a loop without terminating it. On encountering next, the R parser skips further evaluation and starts the next iteration of the loop. In this example, the forloop will iterate for each element in x; however, when it gets to the element that equals 3 it will skip the for loop execution of printing the element and simply jump to the next iteration.

```
x <- 1:5
for (i in x) {
    if (i == 3){
        next
        }
    print(i)
}
## [1] 1
## [1] 2
## [1] 4
## [1] 5
```

5.6 Frames and Subsets

Q18. Explain briefly about data frame in R?

Ans :

A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

Following are the characteristics of a data frame.

- The column names should be non-empty.
- The row names should be unique.
- The data stored in a data frame can be of numeric, factor or character type.
- Each column should contain same number of data items.

i) Create Data Frame

```
# Create the data frame.
     emp.data <- data.frame(
       emp_id = c (1:5),
       emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
        salary = c(623.3,515.2,611.0,729.0,843.25),
        Join_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15",
                          "2014-05-11", "2015-03-27")),
     stringsAsFactors = FALSE
)
# Print the data frame.
print(emp.data)
```

When we execute the above code, it produces the following result:

S.No.	emp_id	emp_name	salary	Join_date
1	1	Rick	623.30	2012-01-01
2	2	Dan	515.20	2013-09-23
3	3	Michelle	611.00	2014-11-15
4	4	Ryan	729.00	2014-05-11
5	5	Gary	843.25	2015-03-27

Get the Structure of the Data Frame

The structure of the data frame can be seen by using **str()** function.

```
# Create the data frame.
emp.data <- data.frame(
     emp_id = c (1:5),
emp_name = c(“Rick”,”Dan”,”Michelle”,”Ryan”,”Gary”),
salary  =  c(623.3,515.2,611.0,729.0,843.25),
Join_date  =  as.Date(c(“2012-01-01”,  “2013-09-23”,  “2014-11-15”,
                        “2014-05-11”,  “2015-03-27”)),
stringsAsFactors = FALSE
)
# Get the structure of the data frame.
str(emp.data)
```

When we execute the above code, it produces the following result “

```
‘data.frame’:   5 obs. of  4 variables:
$ emp_id    : int  1 2 3 4 5
$ emp_name  : chr  “Rick” “Dan” “Michelle” “Ryan” ...
$ salary    : num  623 515 611 729 843
$ Join_date: Date, format: “2012-01-01” “2013-09-23” “2014-11-15” “2014-05-11” ...
```

ii) Summary of Data in Data Frame

The statistical summary and nature of the data can be obtained by applying **summary()** function.

```
# Create the data frame.
emp.data <- data.frame(
emp_id = c (1:5),
emp_name = c(“Rick”,”Dan”,”Michelle”,”Ryan”,”Gary”),
salary  =  c(623.3,515.2,611.0,729.0,843.25),
Join_date  =  as.Date(c(“2012-01-01”,  “2013-09-23”,  “2014-11-15”,  “2014-05-11”,
                      “2015-03- 27”)),
stringsAsFactors = FALSE
)
# Print the summary.
print(summary(emp.data))
```

When we execute the above code, it produces the following result “

emp_id	emp_name	salary	Join_date
Min. :1	Length:5	Min. :515.2	Min. :2012-01-01
1st Qu.:2	Class :character	1st Qu.:611.0	1st Qu.:2013-09-23
Median :3	Mode :character	Median :623.3	Median :2014-05-11
Mean :3		Mean :664.4	Mean :2014-01-14
3rd Qu.:4		3rd Qu.:729.0	3rd Qu.:2014-11-15
Max. :5		Max. :843.2	Max. :2015-03-27

iii) Extract Data from Data Frame

Extract specific column from a data frame using column name.

```
# Create the data frame.
emp.data <- data.frame(
    emp_id = c (1:5),
    emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
    salary = c(623.3,515.2,611.0,729.0,843.25),
Join_date = as.Date(c("2012-01-01","2013-09-23","2014-11-15",
                    "2014-05-11", "2015-03-27")),
stringsAsFactors = FALSE
)
# Extract Specific columns.
result <- data.frame(emp.data$emp_name,emp.data$salary)
print(result)
```

When we execute the above code, it produces the following result:

emp.data.	emp_name	emp.data.salary
1	Rick	623.30
2	Dan	515.20
3	Michelle	611.00
4	Ryan	729.00
5	Gary	843.25

Extract the first two rows and then all columns

# Create the data frame.

```
emp.data <- data.frame(
   emp_id = c (1:5),
   emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
    salary = c(623.3,515.2,611.0,729.0,843.25),
     Join_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15",
                              "2014-05-11", "2015-03-27")),
   stringsAsFactors = FALSE
)
# Extract first two rows.
result <- emp.data[1:2,]
print(result)
```

When we execute the above code, it produces the following result"

emp_id	emp_name	salary	start_date
1 1	Rick	623.3	2012-01-01
2 2	Dan	515.2	2013-09-23

Extract 3rd and 5th row with 2nd and 4th column

```
# Create the data frame.
emp.data <- data.frame(
     emp_id = c (1:5),
     emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
     salary = c(623.3,515.2,611.0,729.0,843.25),
     Join_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",
                          "2015-03-27")),
     stringsAsFactors = FALSE
)
# Extract 3rd and 5th row with 2nd and 4th column.
result <- emp.data[c(3,5),c(2,4)]
print(result)
```

When we execute the above code, it produces the following result

emp_name	Join_date
3 Michelle	2014-11-15
5 Gary	2015-03-27

iv) Expand Data Frame

A data frame can be expanded by adding columns and rows.

Add Column

Just add the column vector using a new column name.

```
# Create the data frame.
emp.data <- data.frame(
emp_id = c (1:5),
emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
salary = c(623.3,515.2,611.0,729.0,843.25),
Join_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",
                      "2015-03-27")),
stringsAsFactors = FALSE
)
# Add the "dept" coulmn.
emp.data$dept <- c("IT","Operations","IT","HR","Finance")
v <- emp.data
print(v)
```

When we execute the above code, it produces the following result:

S.No.	emp_id	emp_name	salary	Join_date	dept
1	1	Rick	623.30	2012-01-01	IT
2	2	Dan	515.20	2013-09-23	Operations
3	3	Michelle	611.00	2014-11-15	IT
4	4	Ryan	729.00	2014-05-11	HR
5	5	Gary	843.25	2015-03-27	Finance

Add Row

To add more rows permanently to an existing data frame, we need to bring in the new rows in the same structure as the existing data frame and use the **rbind()** function.

In the example below we create a data frame with new rows and merge it with the existing data frame to create the final data frame.

```
# Create the first data frame.
emp.data <- data.frame(
  emp_id = c (1:5),
  emp_name = c(“Rick”,”Dan”,”Michelle”,”Ryan”,”Gary”),
   salary = c(623.3,515.2,611.0,729.0,843.25),
   Join_date = as.Date(c(“2012-01-01”, “2013-09-23”, “2014-11-15”, “2014-05-11”, “2015-03-27”)),
  dept = c(“IT”,”Operations”,”IT”,”HR”,”Finance”),
  stringsAsFactors = FALSE
)
# Create the second data frame
emp.newdata <- data.frame (
  emp_id = c (6:8),
  emp_name = c(“Rasmi”,”Pranab”,”Tusar”),
   salary = c(578.0,722.5,632.8),
   Join_date = as.Date(c(“2013-05-21”,”2013-07-30",”2014-06-17")),
  dept = c(“IT”,”Operations”,”Fianance”),
  stringsAsFactors = FALSE
)
# Bind the two data frames.
emp.finaldata <- rbind(emp.data,emp.newdata)
print(emp.finaldata)
```

When we execute the above code, it produces the following result “

S.No.	emp_id	emp_name	salary	Join_date	dept
1	1	Rick	623.30	2012-01-01	IT
2	2	Dan	515.20	2013-09-23	Operations
3	3	Michelle	611.00	2014-11-15	IT
4	4	Ryan	729.00	2014-05-11	HR
5	5	Gary	843.25	2015-03-27	Finance
6	6	Rasmi	578.00	2013-05-21	IT
7	7	Pranab	722.50	2013-07-30	Operations
8	8	Tusar	632.80	2014-06-17	Fianance

Q19. What is mean by subsets in R ? Explain in detail?

Ans :

Subsetting Data

R has powerful indexing features for accessing object elements. These features can be used to select and exclude variables and observations. The following code snippets demonstrate ways to keep or delete variables and observations and to take random samples from a dataset.

Selecting (Keeping) Variables

```
# select variables v1, v2, v3
myvars <- c(“v1”, “v2”, “v3”)
newdata <- mydata[myvars]
# another method
myvars <- paste(“v”, 1:3, sep=””)
newdata <- mydata[myvars]
# select 1st and 5th thru 10th variables
newdata <- mydata[c(1,5:10)]
```

To practice this interactively, try the selection of data frame elements exercises in the Data frames chapter of this introduction to R course.

Excluding (DROPPING) Variables

```
# exclude variables v1, v2, v3
myvars <- names(mydata) %in% c(“v1”, “v2”, “v3”)
newdata <- mydata[!myvars]
# exclude 3rd and 5th variable
newdata <- mydata[c(-3,-5)]
# delete variables v3 and v5
mydata$v3 <- mydata$v5 <- NULL
```

Selecting Observations

```
# first 5 observations
newdata <- mydata[1:5,]
# based on variable values
newdata <- mydata[ which(mydata$gender==’F’
& mydata$age > 65), ]
# or
attach(mydata)
newdata <- mydata[ which(gender==’F’ & age > 65),]
detach(mydata)
```

Selection using the Subset Function

The **subset( )** function is the easiest way to select variables and observations. In the following example, we select all rows that have a value of age greater than or equal to 20 or age less then 10. We keep the ID and Weight columns.

```
# using subset function
newdata <- subset(mydata, age >= 20 | age < 10,
select=c(ID, Weight))
```

In the next example, we select all men over the age of 25 and we keep variables weight *through* income (weight, income and all columns between them).

```
# using subset function (part 2)
newdata <- subset(mydata, sex=="m" & age > 25,
select=weight:income)
```

To practice the **subset()** function, try this this interactive exercise. on subsetting data.tables.

Random Samples

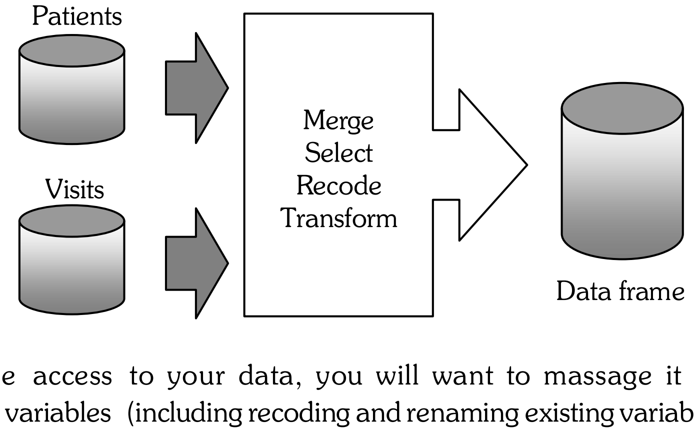
Use the **sample( )** function to take a **random sample of size n** from a dataset.

```
# take a random sample of size 50 from a dataset mydata
# sample without replacement
mysample <- mydata[sample(1:nrow(mydata), 50,
replace=FALSE),]
```

5.7 Managing and Manipulating Data in R

Q20. How data can be managed in R?

Ans :



Once you have access to your data, you will want to massage it into useful form. This includes creating new variables (including recoding and renaming existing variables), sorting and merging datasets, aggregating data, reshaping data, and subsetting datasets (including selecting observations that meet criteria, randomly sampling observeration, and dropping or keeping variables).

Each of these activities usually involve the use of R's built-in operators (arithmetic and logical) and functions (numeric, character, and statistical). Additionally, you may need to use control structures (if-then, for, while, switch) in your programs and/or create your own functions. Finally you may need to convert variables or datasets from one type to another (e.g. numeric to character or matrix to data frame).

This section describes each task from an R perspective.

Q21. How to manipulate data in R?

Ans :

Data Manipulation In R

Data structures provide the way to represent data in data analytics. We can manipulate data in R for analysis and visualization.

Before we start playing with data in R, let us see how to import data in R and ways to export data from R to different external sources like SAS, SPSS, text file or CSV file.

One of the most important aspects of computing with data Data Manipulation in R and enable its subsequent analysis and visualization. Let us see few basic data structures in R:

(a) Vectors in R

These are ordered a container of primitive elements and are used for 1-dimensional data.

Types – integer, numeric, logical, character, complex

(b) Matrices in R

These are Rectangular collections of elements and are useful when all data is of a single class that is numeric or characters.

Dimensions – two, three, etc.

(c) Lists in R

These are ordered a container for arbitrary elements and are used for higher dimension data, like customer data information of an organization. When data cannot be represented as an array or a data frame, list is the best choice. This is so because lists can contain all kinds of other objects, including other lists or data frames, and in that sense, they are very flexible.

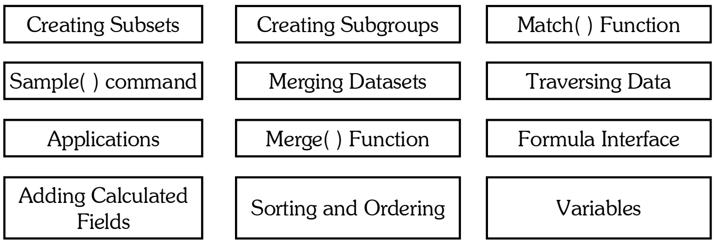
Q22. What is Data Manipulation and Data Processing?

Ans :

In this R Programming we can learn **Data manipulation in R** and data processing with R. Moreover, we will see three subset operators in R and how to perform R data manipulation like subsetting in R, sorting and merging of data in R programming language. Also, we will learn data structures in R, how to create subsets in R and usage of R sample() command, ways to create R data subgroups or bins of data in R. Along with this, we will look at different ways to combine data in R, how to merge data in R, sorting and ordering data in R, ways to traverse data in R and formula interface in R. At last, this R Data Manipulation topics will provide you complete tutorial on ways for manipulating and processing data in R.

So, let's start Data Manipulation in R.

Data Manipulation in R



Creating Subsets

Creating Subgroups

Match( ) Function

Sample( ) command

Merging Datasets

Traversing Data

Applications

Merge( ) Function

Formula Interface

Adding Calculated Fields

Sorting and Ordering

Variables

Data Manipulation In R Tutorial | R Processing Data

Short Question and Answers

1. What is R? Explain the features of R?

Ans :

R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalog of statistical and graphical methods. It includes machine learning algorithm, linear regression, time series, statistical inference to name a few. Most of the R libraries are written in R, but for heavy computational task, C, C++ and Fortran codes are preferred.

R is not only entrusted by academic, but many large companies also use R programming language, including Uber, Google, Airbnb, Facebook and so on.

Data analysis with R is done in a series of steps; programming, transforming, discovering, modeling and communicate the results

- **Program**: R is a clear and accessible programming tool
- **Transform**: R is made up of a collection of libraries designed specifically for data science
- **Discover**: Investigate the data, refine your hypothesis and analyze them
- **Model**: R provides a wide array of tools to capture the right model for your data
- **Communicate**: Integrate codes, graphs, and outputs to a report with R Markdown or build Shiny apps to share with the world

Features of R

As R is a leading programming language. There are so many features of **R program-ming** which makes it important to learn. Let's discuss them one by one.

Statistical Features of R

i) R has some topical relevance

- It is free, open source software.
- R is available under free software Foundation.

ii) R has some statistical features

- **Basic Statistics :** Mean, variance, median.
- **Static graphics :** Basic plots, graphic maps.
- **Probability distributions :** Beta, Binomial.

Any Doubt yet in Why Learn R programming? Please Comment.

Programming Features of R

i) R has some topical relevance

- Data inputs such as data type, **importing data**, keyboard typing.
- Data Management such as data variables, operators.

ii) R has some programming features

- **Distributed Computing** – Distributed computing is an open source, high-performance platform for the R language. It splits tasks between multiple processing nodes to reduce execution time and analyze large datasets.
- **R packages** – **R packages** are a collection of **R functions**, compiled code and sample data. By default, **R installs** a set of packages during installation.

2. R environment

Ans :

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- An effective data handling and storage facility,
- A suite of operators for calculations on arrays, in particular matrices,
- A large, coherent, integrated collection of intermediate tools for data analysis,

- Graphical facilities for data analysis and display either on-screen or on hardcopy, and
- A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

3. Reading Data in R

Ans :

For reading, (importing) data into R following are some functions.

- read.table(), and read.csv(), for reading tabular data
- readLines() for reading lines of a text file
- source() for reading in R code files (inverse of dump)
- dget() for reading in R code files (inverse of dput)
- load() for reading in saved workspaces.

4. Writing Data in R

Ans :

Following are few functions for writing (exporting) data to files.

- write.table(), and write.csv() exports data to wider range of file format including csv and tab-delimited.
- writeLines() write text lines to a text-mode connection.
- dump() takes a vector of names of R objects and produces text representations of the objects on a file (or connection). A dump file can usually be sourced into another R session.
- dput() writes an ASCII text representation of an R object to a file (or connection) or uses one to recreate the object.
- save() writes an external representation of R objects to the specified file.

5. R Function

Ans :

A function is a set of statements organized together to perform a specific task. R has a large number of in-built functions and the user can create their own functions. In R, a function is an object so the R interpreter is able to pass control to the function, along with arguments that may be necessary for the function to accomplish the actions.

The function in turn performs its task and returns control to the interpreter as well as any result which may be stored in other objects.

Definition

An R function is created by using the keyword **function**. The basic syntax of an R function definition is as follows :

```
      function_name <- function(arg_1, arg_2, ...)
{
            Function body
      }
```

6. Components of R

Ans :

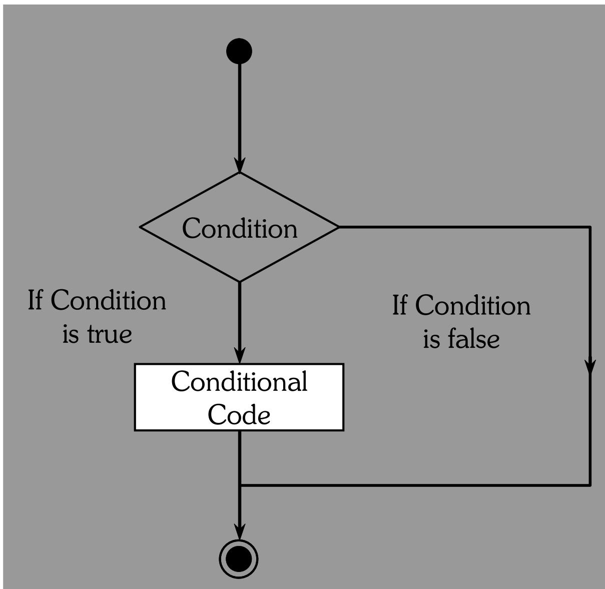
The different parts of a function are:

- **Function Name:** This is the actual name of the function. It is stored in R environment as an object with this name.
- **Arguments:** An argument is a placeholder. When a function is invoked, you pass a value to the argument. Arguments are optional; that is, a function may contain no arguments. Also arguments can have default values.
- **Function Body:** The function body contains a collection of statements that defines what the function does.
- **Return Value:** The return value of a function is the last expression in the function body to be evaluated.

7. Control statements.

Ans :

Looping is similiar to creating functions in that they are merely a means to automate a certain multi step process by organizing sequences of R expressions. R consists of several loop control statements which allow you to perform repetititve code processes with different intentions and allow these automated expressions to naturally respond to features of your data. Consequently, learning these loop control statements will go a long ways in reducing code redundancy and becoming a more efficient data wrangler.



- **if** statement for conditional programming
- **if...else** statement for conditional programming
- for loop to iterate over a fixed number of iterations
- while loop to iterate until a logical statement returns FALSE
- repeat loop to execute until told to break
- break/next arguments to exit and skip interations in a loop.

8. Data frame in R

Ans :

A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

Following are the characteristics of a data frame.

- The column names should be non-empty.
- The row names should be unique.
- The data stored in a data frame can be of numeric, factor or character type.
- Each column should contain same number of data items.

Q9. What is Data Manipulation and Data Processing?

Ans :

In this R Programming we can learn Data manipulation in R and data processing with R. Moreover, we will see three subset operators in R and how to perform R data manipulation like subsetting in R, sorting and merging of data in R programming language. Also, we will learn data structures in R, how to create subsets in R and usage of R sample() command, ways to create R data subgroups or bins of data in R. Along with this, we will look at different ways to combine data in R, how to merge data in R, sorting and ordering data in R, ways to traverse data in R and formula interface in R. At last, this R Data Manipulation topics will provide you complete tutorial on ways for manipulating and processing data in R.

Choose the Correct Answer

1. In 2004, ________ purchased the S language from Lucent for $2 million. [a]

   (a) Insightful (b) Amazon

   (c) IBM (d) All of the mentioned

2. Which will be the output of following code? [c]

```
x - 3
Switch (6, 2+2, mean (1:10), morm(5))
```

   (a) 10 (b) 1

   (c) NULL (d) All of the mentioned

3. ________ programming language is a dialect of S. [c]

   (a) B (b) C

   (c) R (d) K

4. Which of the following is primary tool for debugging? [a]

   (a) debug 0 (b) trace 0

   (c) browser 0 (d) All of the mentioned

5. Point out the wrong statement: [d]

   (a) R is a language for data analysis and graphics

   (b) K is language for statistical modelling and graphics

   (c) One key limitation of the S language was that it was only available in a commercial package, S-PLUS

   (d) None of the mentioned

6. ________ is used to skip an iteration of a loop. [a]

   (a) next (b) skip

   (c) group (d) All of the mentioned

7. Which of the following may be used for linear regression? [c]

   (a) X %*% Y (b) solve (A)

   (c) solve (A,B) (d) All of the mentioned

8. Point out the WRONG statement: [a]

   (a) Early versions of the S language contain functions for statistical modeling.

   (b) The book "Programming with Data" by John Chambers documents S version of the language.

   (c) In 1993, Bell Labs gave StatSci (later Insightful Corp.) an exclusive license to develop and sell the S language.

   (d) All of the mentioned

9. Functions are defined using the __________ directive and are stored as R objects [a]

   (a) function 0 (b) funct 0

   (c) Functions 0 (d) All of the mentioned

10. In 1991, R was created by Ross Ihaka and Robert Gentleman in the Department of Statistics at the University of __________. [d]

   (a) John Hopkins (b) California

   (c) Harvard (d) Auckland

11. Which of the following adds marginal sums to an existing table ? [b]

   (a) par() (b) prop.table()

   (c) addmargins() (d) All of the mentioned

12. Which of the following lists names of variables in a data.frame ? [a]

   (a) quantile() (b) names()

   (c) barchart() (d) All of the mentioned

Fill in the blanks

1. ________ is a programming language developed by Ross Ihaka and Robert Gentleman in 1993.
2. The term ________ is intended to characterize it as a fully planned and coherent system.
3. R packages are a collection of R functions, complied code and sample data. They are stored under a directory called ________.
4. There are ________ ways to add new R packages.
5. A ________ is a set of statements organized together to perform a specific task.
6. ________ is similar to creating functions in that they are merely a means to automate a certain multi step process by organizing sequences of R expressions.
7. ________ begin by testing a condition.
8. A ________ loop is used to iterate over a block of code multiple number of times.
9. ________ structures provide the way to represent data in data analytics.
10. ________ language is made up of a collection of libraries designed specifically for data science.

Answers

1. R
2. Environment
3. Library
4. Two
5. Function
6. Looping
7. While loops
8. Repeat
9. Data
10. R

FACULTY OF MANAGEMENT

M.B.A III - Semester Examination

Model Paper - I

BUSINESS ANALYTICS

Time : 3 Hours] **[Max. Marks : 80**

Note : Answer **all** the questions

PART - A (5 × 4 = 20 Marks)
[Short Answer Type]

ANSWERS

1. Different business analytical methods **(Unit-I, SQA-2)**
2. Benefits of cross tabulation **(Unit-II, SQA-5)**
3. Business Intelligence **(Unit-III, SQA-12)**
4. Advantages of linear programming problem **(Unit-IV, SQA-3)**
5. Reading Data in R **(Unit-V, SQA-3)**

PART - B (5 × 12 = 60 Marks)
[Essay Answer type]

Answer all the questions using the internal choice

6. (a) Explain the different business analytical methods. **(Unit-I, Q.No. 3)**

(OR)

(b) What is Data and Explain the various types of data ? **(Unit-I, Q.No. 16)**

7. (a) What are the techniques of Data Visualization? **(Unit-II, Q.No. 8)**

(OR)

(b) Explain briefly about Gantt Chart. **(Unit-II, Q.No. 13)**

8. (a) Explain the assumptions of simple linear regressions ? **(Unit-III, Q.No. 7)**

(OR)

(b) Explain the various techniques are used in data mining. **(Unit-III, Q.No. 17)**

9. (a) Explain briefly about Non Linear Programming? **(Unit-IV, Q.No. 7)**

(OR)

(b) Explain the Process of Web Analytics. **(Unit-IV, Q.No. 19)**

10. (a) Explain the various types of operators in R program. **(Unit-V, Q.No. 4)**

(OR)

(b) Explain briefly about data frame in R? **(Unit-V, Q.No. 18)**

FACULTY OF MANAGEMENT

M.B.A III - Semester Examination

Model Paper - II

BUSINESS ANALYTICS

Time : 3 Hours] **[Max. Marks : 80**

Note : Answer **all** the questions

PART - A (5 × 4 = 20 Marks)
[Short Answer Type]

Answers

1. Dimensions of big data **(Unit-I, SQA-4)**
2. What is Data visualization? **(Unit-II, SQA-3)**
3. Data Reduction **(Unit-III, SQA-10)**
4. Cutting Plane Method **(Unit-IV, SQA- 5)**
5. Writing Data in R **(Unit-V, SQA-4)**

PART - B (5 × 12 = 60 Marks)
[Essay Answer type]
Answer all the questions using the internal choice

6. (a) Explain the different models in Business Analytics? **(Unit-I, Q.No. 4)**

(OR)

(b) Explain the relationship of big data with other areas? **(Unit-I, Q.No. 11)**

7. (a) How "data visualization technique –Tables" can display data analysis reports using Ms.Excel? **(Unit-II, Q.No. 9)**

(OR)

(b) Explain briefly about pivot charts. **(Unit-II, Q.No. 14)**

8. (a) Explain briefly about data mining. **(Unit-III, Q.No. 14)**

(OR)

(b) Explain the History and Evolution of Business Intelligence. **(Unit-III, Q.No. 26)**

9. (a) Explain Steps involved in Text Analytics **(Unit-IV, Q.No. 17)**

(OR)

(b) Explain about decision making under uncertainty. **(Unit-IV, Q.No. 13)**

10. (a) How the data can be read and write in R? **(Unit-V, Q.No. 8)**

(OR)

(b) Explain briefly about control statements. **(Unit-V, Q.No. 11)**

FACULTY OF MANAGEMENT

M.B.A III - Semester Examination

Model Paper - III

BUSINESS ANALYTICS

Time : 3 Hours] **[Max. Marks : 80**

Note : Answer **all** the questions

PART - A (5 × 4 = 20 Marks)
[Short Answer Type]

		Answers
1.	Digital Marketing	(Unit-I, SQA-6)
2.	Descriptive Statistics.	(Unit-II, SQA-2)
3.	Moving Averages	(Unit-III, SQA-4)
4.	Decision analysis	(Unit-IV, SQA- 6)
5.	R environment	(Unit-V, SQA-2)

PART - B (5 × 12 = 60 Marks)
[Essay Answer type]
Answer all the questions using the internal choice

6. (a) Discuss briefly about role of business analytics in current business environment. **(Unit-I, Q.No. 5)**

(OR)

(b) Explain the evolution of Big Data. **(Unit-I, Q.No. 9)**

7. (a) Explain briefly about “Cross tabulations charts” by using Ms. Excel? **(Unit-II, Q.No. 10)**

(OR)

(b) What are the steps to create interactive Excel Dash Board? **(Unit-II, Q.No. 16)**

8. (a) What is predictive modeling? Explain different types of Predictive Modeling. **(Unit-III, Q.No. 11)**

(OR)

(b) What is data exploration? Explain the Steps of Data Exploration and Preparation. **(Unit-III, Q.No. 20)**

9. (a) Explain briefly about the Cutting Plane Method. **(Unit-IV, Q.No. 8)**

(OR)

(b) Explain about decision making under risk. **(Unit-IV, Q.No. 14)**

10. (a) Explain different types of functions? **(Unit-V, Q.No. 10)**

(OR)

(b) Explain briefly about loop statement? **(Unit-V, Q.No. 15)**