Module-4

1. **What is RDBMS?**
2. RDBMS stands for R elational D atabase M anagement S ystem. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

➢ A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

➢ Most of today's databases are relational: database contains 1 or more tables table contains 1 or more records record contains 1 or more fields fields contain the data

➢ So why is it called "relational"?

➢ tables are related (joined) based on common fields.

1. **What is SQL?**
2. SQL tutorial gives unique learning on Structured Query Language and it helps to make practice on SQL commands which provides immediate results.

➢ SQL is a language of database, it includes database creation, deletion, fetching rows and modifying rows etc.

➢ SQL is an ANSI (American National Standards Institute) standard but there are many different versions of the SQL language.

➢ SQL is the standard programming language of relational DBs ➢ SQL is a standard computer language for accessing and manipulating databases.

1. **Write SQL Commands.**

DDL – Data Definition Language

Command: CREATE: Creates a new table, a view of a table, or other object in database

ALTER: Modifies an existing database object, such as a table

DROP: Deletes an entire table, a view of a table or other object in the database.

DQL – Data Query Language

Command: SELECT: Retrieves certain records from one or more tables

DML – Data Mnipulation language

Command: INSERT: Creates a record

UPDATE: Modifies records

DELETE: Deletes records

DCL – Data Control Language

Commands: GRANT: Gives a privilege to user

REVOKE: Takes back privileges granted from user

1. **What is join?**
2. In SQL, a JOIN clause is used to combine rows from two or more tables based on a related column between them. This operation allows you to retrieve data that is spread across multiple tables in a relational database, presenting it as a single, unified result set.
3. **Write type of joins**.

INNER JOIN: returns rows when there is a match in both tables.

LEFT JOIN: returns all rows from the left table, even if there are no matches in the right table.

RIGHT JOIN: returns all rows from the right table, even if there are no matches in the left table.

FULL JOIN: returns rows when there is a match in one of the tables.

1. **How Many constraint and describes it self.**
2. There are varying numbers of constraints depending on the field (e.g., six for project management, six for RESTful APIs, or three basic types in coaching), but they all represent limiting factors or restrictions that define what can or cannot be done, shaping objectives, solutions, or system behavior. They define boundaries for a system, task, or solution by limiting available resources, influencing choices, and restricting possibilities.
3. **Difference between RDBMS vs DBMS**
4. DBMS

Data is stored as files in a hierarchical structure.

Data elements need to access individually.

It does not support client-server architecture.

Normalization is not present.

DBMS does not support distributed database.

No relationship between data.

Data redundancy is common in this model.

Examples are Windows Registry, Microsoft Access, etc.

Data fetching is slower for the large amount of data.

RDBMS

Data is stored in tabular form where columns are attributes and rows contain corresponding data. Multiple data elements can be accessed at the same time.

It supports client-server architecture.

Normalization is present.

RDBMS supports distributed database.

Data is stored in the form of tables which are related to each other.

Keys and indexes do not allow Data redundancy.

Examples are SQL Server, MySQL, Oracle, PostgreSQL, etc.

Data fetching is fast because of relational approach.

1. **What is an SQL alias?**
2. In SQL, an alias is a temporary, alternative name assigned to a table or a column within a query. This temporary name exists only for the duration of that specific query and does not alter the actual names in the database schema.
3. **Write a query to create the table in Structured Query Language**.
4. CREATE TABLE table\_name( column1 datatype, column2 datatype, column3 datatype, ..... , columnN datatype, PRIMARY KEY( one or more columns ) );
5. **Write a query to insert data into table.**
6. INSERT INTO table\_name( column1, column2....columnN) VALUES ( value1, value2....valueN);
7. **Write a query to update data into table with validations.**
8. UPDATE table\_name SET column1 = value1, column2 = value2....columnN=valueN WHERE CONDITION ;
9. **Write a query to delete data from table with validations.**
10. DELETE FROM table\_name WHERE (CONDITION);
11. **Write a query to insert new column in existing table**.
12. ALTER TABLE table\_name{ADD|DROP|MODIFY}column\_name{data\_ype};
13. **Write a query to drop table and database**.
14. DROP TABLE table\_name;

ALTER TABLE table\_name DROP INDEX index\_name;

1. **Write a query to find max and min value from table.**
2. Select max(column name) from table name group by column name;

Select min(column name) from table name group by column name;

1. **What is API Testing?**
2. Application Programming Interface (API) is a software interface that allows two applications to interact with each other without any user intervention

➢ another definition , API (Application Programming Interface) is a computing interface which enables communication and data exchange between two separate software systems.

➢ The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces.

➢ In API Testing, instead of using standard user inputs(keyboard) and outputs, you use software to send calls to the API, get output, and note down the system’s response.

➢ API tests are very different from GUI Tests and won’t concentrate on the look and feel of an application.

1. **Types of API Testing**
2. ➢ Open APIs: These types of APIs are publicly available to use like OAuth APIs from Google. It has also not given any restriction to use them. So, they are also known as Public APIs.

➢ Partner APIs: Specific rights or licenses to access this type of API because they are not available to the public.

➢ Internal APIs: Internal or private. These APIs are developed by companies to use in their internal systems. It helps you to enhance the productivity of your teams

1. **What is Responsive Testing?**
2. A responsive web design involves creating a flexible web page that is accessible from any device, starting from a mobile phone to a tablet.

➢ Furthermore, a responsive web design improves users’ browsing experience. Considering this from a quality assurance perspective, a responsive web design requires thorough evaluation using a variety of devices before it is ready to go live.

➢ Software testers may find it challenging to perform responsive design testing as a variety of factors are to be looked into during the testing phase.

1. **Which types of tools are available for Responsive Testing.**
2. LT Browser

➢ Lembda Testing

➢ Google Resizer

➢ I am responsive

➢ Pixel tuner

1. **What is the full form of .ipa, .apk**
2. ipa is an iOS App Package

apk is an android package or android application package kit.

1. **How to create step for to open the developer option mode ON?**
2. To open the Developer Options mode on an Android device, go to Settings, then tap About phone or About device. Find and repeatedly tap the Build number (usually seven times) until you see a message confirming you are a developer. Developer options will then appear in your main Settings menu or under the System section.
3. **Create two tables named Seller and Product apply foreign key in product table Fetch data from both table using different joins.**
4. create table seller(seller\_id int,seller\_name varchar(50),email varchar(100),mobile bigint(100));
5. create table product(product\_id int,product\_name varchar(100),seller\_id int,price bigint);
6. insert into seller(seller\_id,seller\_name,email,mobile) values(1,"abhi","abhi@gmail.com",7425369885);
7. insert into seller(seller\_id,seller\_name,email,mobile) values(3,"parth","parth@gmail.com",9987452365);
8. insert into seller(seller\_id,seller\_name,email,mobile) values(4,"griva","griva@gmail.com",9987456213);
9. insert into product(product\_id,product\_name,seller\_id,price) values(10,"sonu",1,500);
10. insert into product(product\_id,product\_name,seller\_id,price) values(20,"pinky",2,300);
11. insert into product(product\_id,product\_name,seller\_id,price) values(30,"pooja",3,800);
12. insert into product(product\_id,product\_name,seller\_id,price) values(40,"pihu",4,5000);
13. create table product(product\_id int,product\_name varchar(100),seller\_id int,price varchar(100)) FOREIGN KEY (seller\_id) REFERENCES seller(seller\_id));
14. select seller.seller\_name,seller.email,product.product\_name,product.price from seller inner join product on seller.seller\_id=product.seller\_id;
15. select seller.seller\_name,seller.email,product.product\_name,product.price from seller left join product on seller.seller\_id=product.seller\_id;
16. select seller.seller\_name,seller.email,product.product\_name,product.price from seller right join product on seller.seller\_id=product.seller\_id;