**ASSIGNMENT-3**

**MODULE-3 (Testing on Live Application)**

**Q-1 What is RDBMS**

**ANS,,** RDBMS stands for Relational Database Management System. It's a type of database management system that organizes data into tables with rows and columns, where each row represents a record and each column represents an attribute of the data.

RDBMSs are based on the relational model of data, proposed by E.F. Codd in the 1970s, which emphasizes the relationships between data elements. These relationships are established through keys, such as primary keys and foreign keys.

Key features of RDBMS include:

Data integrity: RDBMS ensures that data remains accurate and consistent by enforcing constraints and rules on the data.

ACID properties: RDBMS ensures data consistency through transactions that adhere to the principles of Atomicity, Consistency, Isolation, and Durability.

SQL support: RDBMSs typically use Structured Query Language (SQL) for querying and managing the database.

Scalability: RDBMSs can scale vertically (by adding more powerful hardware) or horizontally (by adding more servers) to accommodate increasing data volumes and user loads.

Data normalization: RDBMS supports normalization techniques to reduce redundancy and improve data integrity.

**Q-2 What is SQL**

**ANS..** SQL stands for Structured Query Language. It is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS) or for stream processing in a relational data stream management system (RDSMS). SQL is widely used for tasks such as querying, updating, inserting, and modifying data within a database.

SQL allows users to perform various operations on data stored in relational databases, including:

**Querying Data:** Retrieving specific information from a database using SELECT statements.

**Updating Data:** Modifying existing data in a database using UPDATE statements.

**Inserting Data:** Adding new records to a database using INSERT statements.

**Deleting Data:** Removing records from a database using DELETE statements.

**Creating and Modifying Database Schema:** Defining the structure of a database, including tables, relationships, constraints, indexes, etc., using CREATE, ALTER, and DROP statements.

**Managing Permissions:** Controlling access to the database objects and data by granting or revoking permissions using GRANT and REVOKE statements.

**Transactions Control:** Managing transactions to ensure data integrity and consistency using BEGIN TRANSACTION, COMMIT, and ROLLBACK statements.

**Aggregate Functions:** Computing summary values, such as counts, sums, averages, etc., over sets of data using functions like SUM(), COUNT(), AVG(), etc.

**Q-3 Write SQL Commands**

**ANS..** DDL – Data Definition Language

DML – Data Manipulation Language

DCL – Data Control Language

DQL – Data Query Language

**DDL - Data Definition Language**

Command Description:

**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 Description :

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

**DML – Data Manipulation Language**

Command Description :

**INSERT** Creates a record

**UPDATE** Modifies records

**DELETE** Deletes records

**DCL – Data Control Language**

Command Description :

**GRANT** Gives a privilege to user

**REVOKE** Takes back privileges granted from user

**Create, Drop, Use Database Syntax**

**SQL CREATE DATABASE STATEMENT**

CREATE DATABASE database\_name;

**SQL DROP DATABASE Statement**:

DROP DATABASE database\_name;

**SQL USE STATEMENT**

USE DATABASE database\_name;

**Create, Drop, Alter Table Syntax**

**SQL CREATE TABLE STATEMENT**

CREATE TABLE table\_name( column1 datatype, column2 datatype, column3 datatype, ..... , columnN datatype, PRIMARY KEY( one or more columns ) );

**SQL DROP TABLE STATEMENT**

DROP TABLE table\_name;

**SQL TRUNCATE TABLE STATEMENT**

TRUNCATE TABLE table\_name;

**SQL ALTER TABLE STATEMENT**

ALTER TABLE table\_name{ADD|DROP|MODIFY}column\_name{data\_ype};

**SQL ALTER TABLE STATEMENT (RENAME)**

ALTER TABLE table\_name RENAME TO new\_table\_name;

**Insert, Update, Delete Syntax**

**SQL INSERT INTO STATEMENT**

INSERT INTO table\_name( column1, column2....columnN) VALUES ( value1,value2....valueN);

**SQL UPDATE STATEMENT**

UPDATE table\_name SET column1 = value1, column2=value2....columnN=valueN

**SQL DELETE STATEMENT**

DELETE FROM table\_name WHERE {CONDITION};

**Select Statement Syntax**

**SQL SELECT STATEMENT**

SELECT column1, column2....columnN FROM table\_name;

**SQL DISTINCT CLAUSE**

SELECT DISTINCT column1, column2....columnN FROM table\_name;

**SQL WHERE CLAUSE**

SELECT column1, column2....columnN FROM table\_name WHERE CONDITION;

**SQL AND/OR CLAUSE**

SELECT column1, column2....columnN FROM table\_name WHERE CONDITION-1 {AND|OR} CONDITION-2;

**SQL IN CLAUSE**

SELECT column1, column2....columnN FROM table\_name WHEREcolumn\_name IN (val-1, val-2,...val-N);

**SQL BETWEEN CLAUSE**

SELECT column1, column2....columnN FROM table\_name WHEREcolumn\_name BETWEEN val-1 AND val-2;

**SQL LIKE CLAUSE**

SELECT column1, column2....columnN FROM table\_name WHEREcolumn\_name LIKE { PATTERN };

**SQL ORDER BY CLAUSE**

SELECT column1, column2....columnN FROM table\_name WHERECONDITION ORDER BY column\_name {ASC|DESC};

**SQL GROUP BY CLAUSE**

SELECT SUM(column\_name) FROMtable\_name WHERE CONDITIONGROUP BY column\_name;

**SQL COUNT CLAUSE**

SELECT COUNT(column\_name)FROM table\_name WHERE CONDITION;

**SQL HAVING CLAUSE**

SELECT SUM(column\_name) FROMtable\_name WHERE CONDITION GROUP BY column\_name HAVING (arithematicfunction condition);

**Create and Drop Index Syntax**

**SQL CREATE INDEX Statement :**

CREATE UNIQUE INDEX index\_name ON table\_name( column1,column2,...columnN);

**SQL DROP INDEX STATEMENT**

ALTER TABLE table\_name DROP INDEX index\_name;

**SQL DESC Statement :**

DESC table\_name;

**Commit and Rollback Syntax**

**SQL COMMIT STATEMENT**

COMMIT;

**SQL ROLLBACK STATEMENT**

ROLLBACK;

**Q-4 What is join?**

**ANS..** In simple terms, a "join" in the context of databases refers to combining information from two or more tables based on a related column. Imagine you have two tables of data, such as one containing information about customers and another containing their orders. A join operation allows you to merge these tables together to see, for example, which orders belong to which customers.

There are different types of joins, such as inner join, left join, right join, and outer join, each determining how the rows from the combined tables are matched and displayed. But at its core, a join is about bringing together related data from different tables so you can analyze or work with it more effectively.

**Q-5 Write type of joins.**

**ANS.. SQL Join Types :**

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 nomatches 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.

**Inner Join Syntax**

The most frequently used and important of the joins is the INNER JOIN. They are also referred to as an EQUIJOIN.

The INNER JOIN creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. The query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.

**SYNTAX:**

The basic syntax of INNER JOIN is as follows:

SELECT table1.column1, table2.column2...FROM table1INNER JOIN table2ON table1.common\_filed = table2.common\_field;

**Left Join Syntax**

The SQL LEFT JOIN returns all rows from the left table, even if there are no matches in the right table. This means that if the ON clause matches 0 (zero) records in right table, the join will still return a row in the result, but with NULL in each column from right table.

This means that a left join returns all the values from the left table, plus matched values from the right table or NULL in case of no matching join predicate.

**SYNTAX:**

The basic syntax of LEFT JOIN is as follows:

SELECT table1.column1, table2.column2...FROM table1LEFT JOIN table2ON table1.common\_filed = table2.common\_field;

**Right Join Syntax**

The SQL RIGHT JOIN returns all rows from the right table, even if there are no matches in the left table. This means that if the ON clause matches 0 (zero) records in left table, the join will still return a row in the result, but with NULL in each column from left table.This means that a right join returns all the values from the right table, plus matched values from the left table or NULL in case of no matching join predicate.

**SYNTAX:**

The basic syntax of RIGHT JOIN is as follows:

SELECT table1.column1, table2.column2...FROM table1RIGHT JOIN table2ON table1.common\_filed = table2.common\_field;

**Full Join Syntax**

The SQL FULL JOIN combines the results of both left and right outer joins.The joined table will contain all records from both tables, and fill in NULLs for missing matches on either side.

SYNTAX:

The basic syntax of FULL JOIN is as follows:

SELECT table1.column1, table2.column2...FROM table1FULL JOIN table2ON table1.common\_filed = table2.common\_field;

**Q-6 How many constraint and describes it self**

**ANS..** it would depend on various factors such as the complexity of the application, the type of testing being conducted (e.g., functional testing, performance testing, security testing), and the specific requirements of the application.

Constraints in testing typically refer to limitations or conditions that must be adhered to during the testing process. These constraints can include factors such as time, budget, resources, and technical limitations.

In terms of self-descriptive nature, some testing frameworks or methodologies may incorporate descriptive elements within their constraints to provide clarity and guidance to testers. For example, in Behavior-Driven Development (BDD), tests are often written in a self-descriptive format using natural language constructs to make them more understandable to stakeholders

**Q-7 Difference between RDBMS vs DBMS**

**ANS..** RDBMS (Relational Database Management System) and DBMS (Database Management System) are both systems for managing databases, but they have some key differences, particularly when it comes to testing on live application.

* RDBMS: Organizes data into structured tables with rows and columns. Data is stored in a structured format and follows a relational model, with relationships defined between tables.
* DBMS: Can organize data in various ways, including hierarchical, network, or object-oriented models. It doesn't necessarily enforce the relational model as strictly as RDBMS.
* RDBMS: Enforces integrity constraints, such as primary key, foreign key, unique key, and check constraints to maintain data integrity.
* DBMS: May or may not enforce integrity constraints, depending on the specific implementation. It provides basic data management functionalities without strict enforcement of relational integrity.
* RDBMS: Generally more complex than DBMS due to its adherence to relational principles and support for complex queries and transactions.
* DBMS: Can be simpler in terms of data organization and operations compared to RDBMS.
* Testing on an RDBMS requires thorough validation of relational integrity constraints to ensure data consistency across tables.
* DBMS may not have as strict constraints, so testing for data consistency might involve different approaches depending on how the data is organized and managed.
* RDBMS typically offers optimized query execution strategies for complex SQL queries involving joins, subqueries, and aggregations.
* DBMS may have limitations in query optimization, so testing should include performance testing to ensure that database operations perform adequately under various loads.
* RDBMS provides transaction management features to ensure ACID (Atomicity, Consistency, Isolation, Durability) properties.
* DBMS may have limited support for transactions or may not provide guarantees for ACID properties, which can impact the testing approach for transactional applications.
* RDBMS often includes tools and utilities for data migration, backup, and recovery.
* DBMS may have fewer built-in tools for data migration and backup, requiring additional testing and validation of these processes.

**Q-8 What is API Testing**

**ANS..** 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.

**Q-9 Types of API Testing**

**ANS..** There are mainly 3 types of API Testing :

**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.

**Tools for API Testing :**

PostMan

SoapUI

Jmeter

VRest

**Q-10 What is Responsive Testing?**

**ANS..** 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.

Some points to be understand for Responsive Testing.

The challenges involved in testing a responsive website

How website testing differs from a mobile device to a computer

Rules and guidelines to be followed during responsive design testing and Lastly, various tools available to perform responsive testing.

**Q-11 Which types of tools are available for Responsive Testing**

**ANS..** LT Browser

Lembda Testing

Google Resizer

I am responsive

Pixel tuner

Responsive testing tools help developers and designers ensure that websites or applications look and function well across different devices and screen sizes. Here are some common types of tools used for responsive testing

Viewport Resizers: These tools allow you to see how a website looks on various device screen sizes by resizing the browser window. They simulate different screen resolutions like those of smartphones, tablets, and desktops.

Browser Developer Tools: Most modern web browsers come with built-in developer tools that include features for responsive testing. These tools allow you to toggle between different device views and inspect elements to see how they behave on various screen sizes.

Emulators: Emulators mimic the behavior of different devices, such as iPhones or Android tablets, allowing you to test your website or app as if it were running on those devices. They often provide a more accurate representation of how a website will look and function on specific devices.

Device Labs: Some companies provide physical or virtual device labs where you can test your website or app on a wide range of real devices. This ensures accurate testing across various platforms and screen sizes.

Automated Testing Tools: There are tools available that can automatically test your website's responsiveness across different devices and screen sizes. These tools can save time by quickly identifying any layout or functionality issues.

CSS Frameworks and Libraries: While not traditional testing tools, using responsive CSS frameworks like Bootstrap or Foundation can help ensure that your website is mobile-friendly and adapts well to different screen sizes without requiring extensive testing

**Q-12 What is the full form of .ipa, .apk**

**ANS.. IPA stands for** - "iOS App Store Package". It is a file format used for iOS applications, which can only be installed on devices running Apple's iOS operating system.

**APK stands for -**  "Android Package Kit". It is the file format used for distributing and installing applications on devices running the Android operating system.

**Q-13 How to create step for to open the developer option mode ON?**

**ANS.. Go to Settings:** Open the Settings app on your Android device. You can usually find it in the app drawer or by swiping down from the top of the screen and tapping the gear icon.

**Scroll Down:** Scroll down the settings menu until you find an option called "About phone" or "About device." This option may be located under different headings depending on your device, but it typically contains information about your device, such as model number and software version.

**Tap on "About Phone":** Once you find "About phone" or a similar option, tap on it to open.

**Find "Build Number" or "Software Information":** In the "About phone" section, look for an option called "Build number" or "Software Information." This option might be located at the bottom of the list.

**Tap "Build Number" Repeatedly:** Tap on "Build number" repeatedly (usually about 7 times). You'll see a message saying something like "You are now X steps away from being a developer" with a countdown.

**Enter Your PIN or Password:** If prompted, enter your device's PIN, password, or pattern to confirm your action.

**Developer Options Enabled:** After tapping on "Build number" enough times, you'll see a message saying "You are now a developer!" or similar. This means that Developer Options has been enabled on your device.

**Access Developer Options:** Now, go back to the main settings menu. You'll see a new option called "Developer options" or "System" with Developer Options underneath it. Tap on it to access the Developer Options menu.