**Module-4 Database**

* **What is RDBMS**

**RDBMS (Relational Database Management System)** is a type of database management system that stores data in the form of related tables (relations). It supports SQL to manage and query data and ensures data integrity through constraints and relationships.

* **What is SQL**

**SQL (Structured Query Language)** is a programming language used to communicate with and manage databases. It is used to perform tasks such as querying data, updating records, creating and modifying schemas, and managing access.

* **Write SQL Commands**

SQL commands are grouped into categories:

* DDL (Data Definition Language): CREATE, ALTER, DROP, TRUNCATE
* DML (Data Manipulation Language): SELECT, INSERT, UPDATE, DELETE
* DCL (Data Control Language): GRANT, REVOKE
* TCL (Transaction Control Language): COMMIT, ROLLBACK, SAVEPOINT
* **What is join?**

A **Join** is an SQL operation used to combine rows from two or more tables based on a related column between them.

* **Write type of joins.**

Types of Joins

1. INNER JOIN – returns records with matching values in both tables.
2. LEFT JOIN (LEFT OUTER JOIN) – returns all records from the left table, and matched records from the right table.
3. RIGHT JOIN (RIGHT OUTER JOIN) – returns all records from the right table, and matched records from the left table.
4. FULL JOIN (FULL OUTER JOIN) – returns all records when there is a match in either left or right table.
5. CROSS JOIN – returns Cartesian product of rows from tables.

* **How Many constraint and describes it self**

Common constraints in SQL

* NOT NULL: Prevents null values in a column.
* UNIQUE: Ensures all values in a column are unique.
* PRIMARY KEY: Uniquely identifies each record in a table.
* FOREIGN KEY: Ensures referential integrity between tables.
* CHECK: Ensures the value in a column meets a specific condition.
* DEFAULT: Sets a default value if none is provided.
* **Difference between RDBMS vs DBMS**

|  |  |  |
| --- | --- | --- |
| Aspect | DBMS | RDBMS |
| Data Storage | Stores data as files or hierarchies | Stores data in tables (relations) |
| Relationships | No relationships between data | Supports relationships between tables |
| Support | Limited support for transactions | Supports ACID transactions |
| Data Integrity | Limited | Enforces data integrity and constraints |

* **What is an SQL alias?**

An SQL alias is a temporary name given to a table or column in a query to make it easier to read or reference.

Example:

SELECT employee\_name AS EmpName FROM employees;

* **Write a query to create the table in Structured Query Language.**

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100), Salary DECIMAL(10,2)

);

* **Write a query to insert data into table.**

INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, Salary)

VALUES (1, 'John', 'Doe', 'john.doe@example.com', 60000);

* **Write a query to update data into table with validations.**

UPDATE Employees

SET Salary = 65000

WHERE EmployeeID = 1 AND Salary < 65000;

* **Write a query to delete data from table with validations.**

DELETE FROM Employees

WHERE EmployeeID = 1 AND Salary < 50000;

* **Write a query to insert new column in existing table.**

ALTER TABLE Employees

ADD DateOfBirth DATE;

* **Write a query to drop table and database**.

DROP TABLE Employees;

-- To drop a database:

DROP DATABASE CompanyDB;

* **Write a query to find max and min value from table**.

SELECT MAX(Salary) AS HighestSalary, MIN(Salary) AS LowestSalary

FROM Employees;

* **Create two tables named Seller and Product apply foreign key in product table Fetch data from both table using**

CREATE TABLE Seller (

SellerID INT PRIMARY KEY,

SellerName VARCHAR(100)

);

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Price DECIMAL(10,2),

SellerID INT,

FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)

);

### Fetch data from both tables using JOIN

SELECT p.ProductID, p.ProductName, p.Price, s.SellerName

FROM Product p

INNER JOIN Seller s ON p.SellerID = s.SellerID;

* **What is API Testing**

API Testing is a type of software testing that focuses on verifying that Application Programming Interfaces (APIs) work as expected. It tests the functionality, reliability, performance, and security of APIs by sending requests and validating responses without using a user interface.

API testing checks:

* Whether the API returns correct responses for given requests
* Data accuracy and integrity
* Error handling and status codes
* Performance under load
* Security aspects like authentication and authorization
* **Types of API Testing**
* Functional Testing: Verifies the API functions correctly as per requirements.
* Load Testing: Checks the API's behavior under heavy load or many simultaneous requests.
* Security Testing: Ensures API security via authentication, authorization, and encryption.
* Validation Testing: Validates API response data and format.
* Reliability Testing: Confirms the API's stability over time.
* Regression Testing: Ensures that new changes don’t break existing API functionalities.
* Negative Testing: Tests how the API handles invalid inputs or unexpected behavior.

* **What is Responsive Testing?**

**Responsive Testing** is the process of checking whether a website or application works well and looks good on different screen sizes, resolutions, and devices (such as desktops, tablets, and mobile phones). It ensures that the UI adapts fluidly and usability remains consistent across platforms.

* **Which types of tools are available for Responsive Testing**

Common tools for responsive testing include:

Browser Developer Tools: Chrome DevTools, Firefox Developer Tools (with device simulation)

Online Responsive Testing Tools: BrowserStack, Sauce Labs, LambdaTest, CrossBrowserTesting

Automated Testing Frameworks: Selenium WebDriver with different viewport sizes

Emulators/Simulators: Android Emulator, iOS Simulator

Open Source Tools: Responsively App, Screenfly

* **What is the full form of .ipa, .apk**

.ipa: iOS Application Archive — the file format for iOS mobile apps (Apple devices)

.apk: Android Package Kit — the file format for Android mobile apps

* **How to create step for to open the developer option mode ON**

To enable Developer Options on Android devices, the usual steps are:

1. Go to Settings on your Android device.
2. Scroll down and tap About phone (or About device).
3. Find Build number.
4. Tap on Build number 7 times consecutively.
5. You may be prompted to enter your PIN/password.
6. After the 7th tap, you’ll see a message saying “You are now a developer!”.
7. Go back to Settings.
8. You will now see Developer options listed, usually under System or Advanced settings.
9. Tap Developer options and toggle it ON.