**🎯 MODULE 4: DATABSE 🎯**

**Que.1) What is RDBMS?**

**Ans.** An RDBMS is a type of database management system that stores data in a structured format, using rows and columns. The data is organized into tables (also called relations), and these tables are related to each other through keys (like primary and foreign keys).

**Que.2) What is SQL?**

**Ans.** SQL (Structured Query Language) is a computer language for storing, manipulating and retrieving data stored in database.

**Que.3) Write a SQL Commands.**

**Ans.** 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. |

➢ DML – Data Manipulation Language

|  |  |
| --- | --- |
| Command | Description |
| **INSERT** | Creates a record |
| **UPDATE** | Modifies Record |
| **DELETE** | Delete a Record |

➢ DCL – Data Control Language

|  |  |
| --- | --- |
| Command | Description |
| **GRANT** | Gives a privilege to user |
| **REVOKE** | Takes back privileges granted from user |

DQL – Data Query Language

|  |  |
| --- | --- |
| Command | Description |
| **SELECT** | Retrieves certain records from one or more tables |

**Que.4) Write types of Joins.**

|  |  |
| --- | --- |
| **Type** | **Description** |
| **INNER JOIN** | Returns **only the matching rows** from both tables. |
| **LEFT JOIN** (LEFT OUTER JOIN) | Returns **all rows from the left table**, and matched rows from the right table. Unmatched right rows are NULL. |
| **RIGHT JOIN** (RIGHT OUTER JOIN) | Returns **all rows from the right table**, and matched rows from the left table. Unmatched left rows are NULL. |
| **FULL JOIN** (FULL OUTER JOIN) | Returns **all rows when there is a match in one of the tables**. Missing values are filled with NULL. |
| **CROSS JOIN** | Returns the **Cartesian product** (all combinations) of the two tables. |
| **SELF JOIN** | A table is joined with **itself**. |

**Que.5) How many constraint and describes it self.**

|  |  |
| --- | --- |
| **Constraint** | **Description** |
| NOT NULL | Disallows NULL values |
| UNIQUE | Requires all values to be different |
| PRIMARY KEY | Uniquely identifies each row (NOT NULL + UNIQUE) |
| FOREIGN KEY | Links to a primary key in another table |
| CHECK | Enforces specific conditions on values |
| DEFAULT | Sets a default value if none is provided |

**Que.6) What is Join?**

**Ans.** In SQL and in general database terminology, a JOIN is used to combine rows from two or more tables based on a related column between them.

**Que.7) Difference** **between RDBMS VS DBMS.**

**Ans.**

|  |  |  |
| --- | --- | --- |
| **Feature** | **DBMS** | **RDBMS** |
| **Full Form** | Database Management System | Relational Database Management System |
| **Data Storage Format** | Data is stored in files or hierarchical form | Data is stored in tabular (rows and columns) form |
| **Relation Between Data** | No relationships among data | Relationships among tables using keys |
| **Normalization Support** | Does not support normalization | Supports normalization to reduce redundancy |
| **Data Integrity** | No integrity constraints | Supports integrity constraints (like primary key, foreign key) |
| **Security** | Low-level security | High-level security (with roles and permissions) |
| **Examples** | File System, XML, Microsoft Access (non-relational use) | MySQL, Oracle, SQL Server, PostgreSQL |
| **Complex Querying** | Limited querying capability | Advanced querying using SQL |
| **Data Redundancy** | High | Low due to normalization |
| **Multi-user Environment** | Typically single-user | Designed for multi-user |

**Que.8) What is Alias?**

**Ans.** An SQL alias is a temporary name given to a table or column in a SQL query. It makes the query more readable, and can simplify complex expressions or rename output columns.

**Que.9) Write a query to create the table in Structured Query Language.**

**Ans.** CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100) UNIQUE,

phone\_number VARCHAR(15),

hire\_date DATE NOT NULL,

salary DECIMAL(10, 2),

department\_id INT );

**Que.10) Write a query to INSERT data into table with validations.**

**Ans.** CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100) UNIQUE,

phone\_number VARCHAR(15),

hire\_date DATE NOT NULL,

salary DECIMAL(10, 2),

department\_id INT );

**Que.11) Write a query to UPDATE data into table with validations.**

**Ans.** UPDATE employees

SET salary = 65000

WHERE emp id = 101

AND 65000 > 0

AND department != ‘Terminated’;

**Que.12) Write a query to DELETE data into table with validations.**

**Ans.** DELETE FROM customers

WHERE customer\_id = 205

AND status = 'Inactive'

AND created\_at < NOW() - INTERVAL 1 YEAR

AND customer\_id NOT IN (

SELECT customer\_id FROM transactions );

**Que.13) Write a query to insert New Column in existing table.**

**Ans.** ALTER TABLE employees

ADD date of joining DATE;

**Que.14) Write a query to drop table and database.**

**Ans.** DROP TABLE employees;

DROP DATABASE company\_db;

**Que.15) Write a query to find Max and Min value from table.**

**Ans.** SELECT

MAX(salary) AS max salary,

MIN(salary) AS min salary,

FROM employees;

**Que.16) Create two tables named Seller and Product apply foreign key in product table.**

**Ans.** CREATE TABLE Seller (

seller\_id INT PRIMARY KEY,

seller\_name VARCHAR(100) NOT NULL,

city VARCHAR(50) );

-- Create Product table with foreign key referencing Seller table

CREATE TABLE Product (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(100) NOT NULL,

price DECIMAL(10, 2),

seller\_id INT,

FOREIGN KEY (seller\_id) REFERENCES Seller(seller\_id)

);

**Que.17) Fetch data from both table using different joins.**

**Ans.** 1. **INNER JOIN**

SELECT

p.product\_id,

p.product\_name,

p.price,

s.seller\_name,

s.city

FROM Product p

INNER JOIN Seller s ON p.seller\_id = s.seller\_id;

2. **LEFT JOIN**

SELECT

p.product\_id,

p.product\_name,

p.price,

s.seller\_name,

s.city

FROM Product p

LEFT JOIN Seller s ON p.seller\_id = s.seller\_id;

3. **RIGHT JOIN**

SELECT

p.product\_name,

p.price,

s.seller\_name,

s.city

FROM Product p

RIGHT JOIN Seller s ON p.seller\_id = s.seller\_id;

4. **FULL OUTER JOIN**

SELECT

p.product\_name,

p.price,

s.seller\_name,

s.city

FROM Product p

FULL OUTER JOIN Seller s ON p.seller\_id = s.seller\_id;

**Que.18) What is API testing?**

**Ans.** API Testing is a type of software testing that focuses on verifying that Application Programming Interfaces (APIs) work as expected. It checks if the API returns the correct response, handles errors properly, and meets performance, security, and functional requirements.

**Que.19) What is the full form of IPA & .APK?**

**Ans.** IPK Stands for : IOS App Store Package

.APK Stands for : **A**ndroid **P**ackage **K**it (Or Android Application Package)

**Que.20) Types of API Testing?**

**Ans.**

|  |  |
| --- | --- |
| **Type** | **Purpose** |
| **1. Functional Testing** | Validate that the API functions as expected. |
| **2. Validation Testing** | Ensure API returns correct status, format, and data structure. |
| **3. Load Testing** | Test API behavior under normal expected load. |
| **4. Stress Testing** | Check how API behaves under extreme load or stress. |
| **5. Security Testing** | Check for vulnerabilities and unauthorized access. |
| **6. Reliability Testing** | Ensure API gives consistent responses over repeated requests. |
| **7. Integration Testing** | Verify API interaction with other systems or modules. |
| **8. Regression Testing** | Ensure new changes haven't broken existing functionality. |
| **9. Negative Testing** | Test API behavior with invalid input or incorrect usage. |
| **10. Compliance Testing** | Verify API follows standards/protocols (REST, JSON, OAuth, OpenAPI, etc.) |

**Que.21) What is Responsive Testing?**

**Ans.** Responsive Testing is the process of verifying that a website or web application looks and functions correctly across a variety of devices, screen sizes and orientations – including desktops, tablets and mobile phones.

**Que.22) Which types of tools are available for Responsive Testing?**

**Ans.**

|  |  |  |
| --- | --- | --- |
| **Category** | **Tool Name** | **Purpose / Description** |
| 🔹 **Browser DevTools** | Chrome DevTools, Firefox DevTools | Built-in tools to simulate mobile, tablet, and desktop views by changing screen sizes. |
| 🔹 **Cloud-Based Testing** | BrowserStack, Sauce Labs, LambdaTest | Test websites on real devices and browsers online — no local setup needed. |
| 🔹 **Responsive Design Preview** | Responsinator, Screenfly, Am I Responsive? | Preview how your website looks on popular screen sizes. |
| 🔹 **Visual Regression Tools** | Percy, Applitools, Visual Studio Code extensions | Detect visual differences or layout shifts across screen sizes and releases. |
| 🔹 **Automation Testing Tools** | Selenium, Cypress (with viewport commands), Puppeteer | Automate responsive testing by resizing viewports and validating layout changes. |
| 🔹 **CSS/Framework Tools** | Bootstrap, Tailwind CSS DevTools | Help ensure responsive breakpoints are respected during development. |
| 🔹 **Cross-Browser Testing Tools** | CrossBrowserTesting, TestGrid | Test site rendering and responsiveness across multiple browsers and devices. |

**Que.23) How to create step for to open the developer option mode ON?**

**Ans. Steps : -**

1. Unlock your Android phone and go to the Home screen.
2. Open the Settings app  
   (Usually labeled as “Settings”).
3. Scroll down and tap “About phone”.  
   (In some devices, you may need to go to “System” → “About phone”).
4. Find and tap on “Build number”

* On some phones, it’s inside Software Information or Version Info.
* Example path: Settings > About phone > Software Information > Build number.

1. Tap on “Build number” 7 times
2. You may be asked to enter your phone’s PIN/password.
3. You’ll see a message: “You are now a developer!”
4. Go back to the main Settings menu.
5. You’ll now see a new menu: “Developer options”  
   (Usually under System or near the bottom of Settings).
6. Tap on Developer options and toggle it ON at the top.