Module-4 Database

1. **What is RDBMS ?**

RDBMS stands for Relational Database Management Systems. A database is an organized collection of data stored in a computer system and usually controlled by a database management system (DBMS). The data in common databases is modeled in tables, making querying and processing efficient

1. **What is SQL ?**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database.

SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server use SQL as standard database language.

Also, they are using different dialects, such as:

MS SQL Server using T-SQL, ANSI SQL Oracle using PL/SQL,

MS Access version of SQL is called JET SQL etc.

1. **Write SQL Commands ? Write type of joins ?**

**DDL (Data Definition Language):**

**CREATE**: Creates a new table or database.

**ALTER**: Modifies an existing database object.

**DROP:** Deletes an entire table, database, or other objects.

**TRUNCATE:** Removes all records from a table, deleting the space allocated for the records.

**DML (Data Manipulation Language):**

**SELECT:** Retrieves data from the database.

**INSERT:** Adds new data to a table.

**UPDATE:** Modifies existing data within a table.

**DELETE:** Removes data from a table.

**DCL (Data Control Language):**

**GRANT:** Gives users access privileges to the database.

**REVOKE:** Removes access privileges given with the GRANT command.

**TCL (Transaction Control Language):**

**COMMIT:** Saves all changes made in the current transaction.

**ROLLBACK:** Restores the database to the last committed state.

**SAVEPOINT:** Sets a savepoint within a transaction.

**SET TRANSACTION:** Places a name on a transaction.

1. **What is join ?**

In the SQL three join inner join, right join and left join.

Inner Join :

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 :

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

Left Join :

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:

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

Right Join :

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:

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

1. **How Many constraint and describes it self ?**

NOT NULL : Ensures that a column cannot have a NULL value.

UNIQUE : Ensures that all the values in a column are different.

PRIMARY KEY: A combination of NOT NULL and UNIQUE. Uniquely identifies each record in a table.

FOREIGN KEY: Ensures the referential integrity of the data in one table to match values in another table.

CHECK: Ensures that all values in a column satisfy a specific **condition**.

DEFAULT : Provides a default value for a column when no value is specified

1. **Difference between RDBMS vs DBMS ?**

|  |  |
| --- | --- |
| **DBMS** | **RDBMS** |
| Dbms stores data as file | RDBMS stores data in tabular form |
| Data elements need to access individually | Multiple data elements can be accessed at the same time. |
| No relationship between data. | Data is stored in the form of tables which are related to each other. |
| Normalization is not present. | Normalization is present. |
| DBMS does not support distributed database. | RDBMS supports distributed database. |
| It stores data in either a navigational or hierarchical form. | It uses a tabular structure where the headers are the column names, and the rows contain corresponding values. |
| It deals with small quantity of data. | It deals with large amount of data. |
| Data redundancy is common in this model. | Keys and indexes do not allow Data redundancy. |
| It is used for small organization and deal with small data. | It is used to handle large amount of data |
| Not all Codd rules are satisfied. | All 12 Codd rules are satisfied. |
| Security is less | More security measures provided. |
| It supports single user. | It supports multiple users. |
| Data fetching is slower for the large amount of data. | Data fetching is fast because of relational approach. |
| The data in a DBMS is subject to low security levels with regards to data manipulation. | There exists multiple levels of data security in a RDBMS. |
| Low software and hardware necessities. | Higher software and hardware necessities. |
| Examples:[XML](https://www.geeksforgeeks.org/xml-basics/), Window Registry, Forxpro, dbaseIIIplus etc. | Examples: [MySQL](https://www.geeksforgeeks.org/architecture-of-mysql/), [PostgreSQL](https://www.geeksforgeeks.org/what-is-postgresql-introduction/), [SQL](https://www.geeksforgeeks.org/what-is-sql/) Server, Oracle, Microsoft Access etc. |

1. **What is an SQL alias?**

Aliases are often used to:

* Make column or table names **easier to read**
* Give columns or tables **shorter names**
* Improve the **clarity** of complex queries

Exmpel : For **Column Alias**:

SELECT column\_name AS alias\_name

FROM table\_name;

For **Table Alias**: SELECT column\_name

FROM table\_name AS alias\_name;

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

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Employees1 ( EmployeeID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100), HireDate DATE, Salary DECIMAL(10, 2) );

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

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

VALUES (1, 'Yatrik', 'Prajapati', 'yatrik@example.com', '2024-05-01', 50000.00);

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

VALUES

(2, 'Amit', 'Shah', 'amit@example.com', '2023-07-15', 60000.00),

(3, 'Nisha', 'Patel', 'nisha@example.com', '2022-09-10', 55000.00);

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

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) Employees [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) Salary = 55000.00 WHERE EmployeeID = 1 [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) Salary < 55000.00;

11. Write a query to delete data from table with validations?

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) FROM Employees WHERE EmployeeID = 4 [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) Salary < 30000.00;

12. Write a query to insert new column in existing table?

[ALTER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) Employees ADD PhoneNumber VARCHAR(15);

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

[DROP](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/drop-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/drop-table.html) Employees;

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

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [MAX](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_max)(Salary) AS MaxSalary, [MIN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_min)(Salary) AS MinSalary FROM Employees;

15. Create two tables named Seller and Product apply foreign key in product table Fetch data from both table using different joins?

1. **Create Seller Table:**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Seller ( SellerID INT PRIMARY KEY, SellerName VARCHAR(100), City VARCHAR(50) );

2. **Create Product Table with Foreign Key to Seller:**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Product ( ProductID INT PRIMARY KEY, ProductName VARCHAR(100), Price DECIMAL(10, 2), SellerID INT, FOREIGN KEY (SellerID) REFERENCES Seller(SellerID) );

3. **Fetch Data Using Different Joins:**

**a) INNER JOIN – Fetch only matching records:**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) Product.ProductID, Product.ProductName, Product.Price, Seller.SellerName, Seller.City FROM Product INNER JOIN Seller ON Product.SellerID = Seller.SellerID;

**LEFT JOIN** – All products, even if they don't have a seller:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) Product.ProductID, Product.ProductName, Product.Price, Seller.SellerName, Seller.City FROM Product [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_left) JOIN Seller ON Product.SellerID = Seller.SellerID;

**RIGHT JOIN** – All sellers, even if they have no products:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) Product.ProductID, Product.ProductName, Product.Price, Seller.SellerName, Seller.City FROM Product [RIGHT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_right) JOIN Seller ON Product.SellerID = Seller.SellerID;

**FULL OUTER JOIN** – All products and sellers, matched or not (if supported):

SELECT

Product.ProductID, Product.ProductName, Product.Price,

Seller.SellerName, Seller.City

FROM Product

OUTER JOIN Seller ON Product.SellerID = Seller.SellerID;

16. What is API Testing ?

An **API** allows different software systems to communicate with each other — like a waiter taking your order to the kitchen and bringing food back.

Key Goals of API Testing:

* Check if the **API returns correct data**.
* Ensure **error handling** works properly.
* Validate **response time** and **performance**.
* Confirm the **security** of the API.
* Make sure it behaves correctly with **edge cases or invalid inputs**.

Common Tools Used:

* Postman
* SoapUI
* REST Assured
* JMeter
* Swagger

17. Types of API Testing ?

* Functional Testing :

Checks whether the API performs as expected.

Focus: input → output validation.

Example: Does /login return a token when valid credentials are given?

* **Validation Testing :**

Done after development is complete.

Ensures APIs follow correct behavior, structure, and data formats.

Checks: Status codes, headers, responses, schema.

* **Load Testing**

Tests how the API handles **high volume** of requests.

Goal: Ensure performance under stress (e.g., 1000 users at once).

* **Security Testing:**

Ensures the API is protected against threats.

Checks: Authorization, Authentication, Data encryption.

* **Reliability Testing:**

Confirms that the API gives **consistent results** over time.

Useful for production-level APIs.

* **Penetration Testing:**

A type of security testing done by simulating attacks.

Checks for vulnerabilities hackers could exploit.

* **Negative Testing:**

Tests how the API responds to invalid input or bad requests.

Example: What happens if required parameters are missing?

* **UI Integration Testing :**

Ensures that API works properly with the front-end or other services.

Checks: Data flow between UI and API.

* **Regression Testing :**

Performed after code changes to ensure nothing broke.

Performed after code changes to ensure nothing broke.

18. **What is Responsive Testing?**

* Purpose of Responsive Testing:

To ensure the user interface (UI) looks and works properly across all devices, without layout issues, text overflow, or broken elements.

* What It Tests:
  + Layout and design on various screen sizes
  + Font readability and button usability
  + Image scaling and alignment
  + Navigation menus and clickable elements
  + Orientation change (portrait vs. landscape)
* Common Devices Used:
* Mobile phones (Android, iPhone)
* Tablets (iPad, Android tablets)
* Laptops and desktops (different resolutions)
* Tools for Responsive Testing:
  + Chrome Dev Tools (Device Toolbar)
  + Browser Stack
  + Lambda Test
  + Responsive Design Checker
* Example:
* Text should be readable
* Buttons should be clickable
* No horizontal scrolling should be required

19. Which types of tools are available for Responsive Testing ?

**Chrome DevTools (Device Mode)**

* **Description:** A built-in tool in Chrome that lets you simulate various devices and screen resolutions directly in your browser.
* **Features:**
  + Switch between mobile and tablet devices.
  + Adjust screen orientation (portrait/landscape).
  + Test touch interactions.

**BrowserStack**

* **Description:** A cloud-based testing platform that allows you to test websites on real devices and browsers.
* **Features:**
  + Real-time testing on different OS (Android, iOS, Windows, Mac).
  + Access to hundreds of real mobile and desktop devices.
  + Automated responsive tests.

**LambdaTest**

* **Description:** A cross-browser testing platform for web apps, supporting real-time, automated responsive testing.
* **Features:**
  + Test on various real browsers and devices.
  + Visual UI testing for different screen sizes.
  + Parallel test execution.

**Responsive Design Checker**

* **Description:** A simple online tool that lets you check how your website looks on different screen sizes without the need for coding.
* **Features:**
  + Quick device emulation (desktops, tablets, mobile).
  + Simulate different resolutions and orientations.

**CrossBrowserTesting**

* **Description:** A tool to test websites on real browsers and devices for responsive design and functionality.
* **Features:**
  + Access to over 2,000 real browsers and devices.
  + Screenshot testing for different screen sizes.
  + Manual and automated testing.

**Responsinator**

* **Description:** A simple tool for testing websites on popular mobile devices.
* **Features:**
  + Visualize how a website looks on different mobile devices.
  + Responsive design preview with real-time adjustments.

**MobiReady**

* **Description:** A tool that tests mobile responsiveness by analyzing various design aspects and mobile-friendliness.
* **Features:**
  + Mobile performance and compatibility checks.
  + Reports on usability and design issues for mobile screens.

**Viewport Resizer**

* **Description:** A browser bookmarklet that lets you check your website's responsiveness across different screen sizes directly from the browser.
* **Features:**
  + Adjustable screen sizes.
  + Instant preview on any webpage.

**Figma (with Plugins)**

* **Description:** Figma is a design tool, but with responsive design plugins, you can simulate how your design will look on various screen sizes.
* **Features:**
  + Create responsive web layouts and check designs at various screen sizes.
  + Test grids, typography, and breakpoints.

**Screenfly (by QuirkTools)**

* **Description:** An online tool that allows you to test how websites appear on different screen sizes and devices.
* **Features:**
  + Simulate various screen types (mobile, tablet, desktop).
  + Test design layouts across different viewports.

20. • What is the full form of .ipa, .apk ?

Ipa :- iOS App Store Package.

Apk:- Android Package Kit.

21. How to create step for to open the developer option mode ON?

To enable Developer Options on an Android device, navigate to Settings > About phone > Build number and tap the "Build number" field seven times. You'll see a message indicating Developer Options are now enabled. You can then find the Developer Options menu in your Settings.

Here's a step-by-step guide:

1. **Open Settings:** Access the Settings app on your Android device.
2. **Go to About Phone:** Navigate to the "About phone" or "About device" section within the Settings menu.
3. **Locate Build Number:** Find the "Build number" entry, which usually appears within the "Software information" section.
4. **Tap Build Number Seven Times:** Quickly tap the "Build number" field seven times in a row.
5. **Enable Developer Mode:** You should see a message indicating that Developer Options are now enabled. You may need to enter a PIN or password if you have one set up.
6. **Access Developer Options:** The Developer Options menu will now appear in your main Settings menu, usually under "System" or "About phone".