Module 4

Database

1. **What is RDBMS?** 
   * RDBMS means Relational Database Management System. It's a type of database system that stores and retrieves data in a tabular format organized in the form of rows and columns. This tabular structure makes querying and processing data very efficient.
   * Popular RDBMS Example

1. mySQL

2. Oracle database

3. Microsoft SQL server

1. **What is SQL?**
   * SQL means Structured Query Language. SQL is a standard programming language used for managing and manipulating relational databases. It allows users to perform various operations on databases, such as creating, inserting, updating, and deleting data. SQL is essential for working with databases because it provides a set of commands for interacting with the data stored in a structured format.
   * SQL is widely used in RDBMS like MySQL, oracle, SQL server site.
2. **Write SQL Commands.**
   * SQL commands for create, edit, delete for database(db) automation whaich have table name customer which have column like cid, cname, cemail cpassword.
3. **Create db**

Create database automation

1. **create db table**

Create table customer(

cid int NOT null AUTO\_INCREMENT,

cname varchar(30) NOT null,

cemail varchar(50) NOT null,

cpassword varchar(25) NOT null,

PRIMARY KEY(cid)

)

1. **Insert Query**

INSERT INTO customer (cname, cemail cpassword) values (“riya”, [riya@gmail.com](mailto:riya@gmail.com)”, “riya123”)

1. **select query**

SELECT \* from customer(to fetch data of whole table)

SELECT cname, cid from customer(to fetch data of particular column)

SELECT \* from customer where id = 2(to fetch data of particular row)

1. **Delete value**

DELETE from customer where id=3(delete row)

DROP table customer(drop whole table)

DROP database automation(drop whole database)

1. **What is join?**
   * In the relational databases, a JOIN is a powerful SQL clause that allows you to combine data from two or more tables based on a related column. This is crucial when you need to retrieve information that's spread across multiple tables.
   * If we 2 table 1) customer 2) order, now we perform join command
   * SELECT Customers.CName, Orders.OrderDate FROM Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
   * This query would return a table with two columns: "Name" from the Customers table and "OrderDate" from the Orders table. It would only include customers who have placed at least one order.
2. **Write type of joins.**
3. **INNER JOIN:**
   * Returns rows where there is a match in both tables.
   * Most commonly used join.
4. **LEFT JOIN:**
   * Returns all rows from the left table, and the matching rows from the right table.
   * If there's no match in the right table, NULL values are used.
5. **RIGHT JOIN:**
   * Returns all rows from the right table, and the matching rows from the left table.
   * If there's no match in the left table, NULL values are used.
6. **FULL OUTER JOIN:**
   * Returns all rows when there is a match in either of the tables.
   * If there's no match in one of the tables, NULL values are used.
7. **How Many constraint and describes it self**

 **NOT NULL:**

* Ensures that a column cannot contain NULL values.
* Guarantees that a specific piece of information is always provided for each record.

 **UNIQUE:**

* Ensures that all values in a column or a set of columns are unique.
* Allows NULL values.

 **PRIMARY KEY:**

* A special type of UNIQUE constraint.
* Uniquely identifies each row in a table.
* Cannot contain NULL values.

 **FOREIGN KEY:**

* Defines a link between two tables.
* References the PRIMARY KEY of another table.
* Ensures referential integrity between tables.

 **CHECK:**

* Defines a condition that all rows in a table must satisfy.
* Allows you to enforce specific business rules or limitations on data.

 **DEFAULT:**

* Specifies a default value for a column if no value is provided during data insertion.

1. **Difference between RDBMS vs DBMS**

|  |  |
| --- | --- |
| RDBMS | DBMS |
| Relational Database Management system | Database Management system |
| Stores data in table form with rows and columns | Stores data in files |
| Higher security and faster data retrieval | Lower security and slower retrieval |
| Defines relationships between tables using keys. | No clear relationship between data |
| Ex.-MySQL, Oracle, SQLite | Ex.-XML, Windows registry |

1. **What is an SQL alias?**
   * In SQL, an alias is a temporary name given to a table or a column. It's like a nickname you use within a specific query to make the query more readable or to avoid naming conflicts. The alias only exists for the duration of that query; it doesn't change the actual name of the table or column in the database.
2. **Write a query to create the table in Structured Query Language.**

SQL

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(25),

LastName VARCHAR(25),

Email VARCHAR(70)

);

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

SQL

INSERT INTO Customer (CustomerID, FirstName, LastName, Email) VALUES ('Riya', 'Bhatt', 'Riyabhatt44@example.com');

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

UPDATE Customer SET Email = ‘nextemail@gmail.com' WHERE CustomerID = 1;

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

DELETE FROM Customer WHERE CustomerID = 1;

1. **Write a query to insert new column in existing table.**
   * SQL
   * ALTER TABLE Customer ADD COLUMN PhoneNumber VARCHAR(20);
2. **Write a query to drop table and database.** 
   * SQL
   * DROP TABLE Customer;
   * DROP DATABASE MyDatabase;
3. **Write a query to find max and min value from table.**
   * SQL

SELECT MAX(Age), MIN(Age)

FROM Employees;

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

CREATE TABLE Seller (

SellerID INT PRIMARY KEY,

SellerName VARCHAR(255)

);

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(255),

SellerID INT,

FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)

);

-- INNER JOIN

SELECT Product.ProductName, Seller.SellerName

FROM Product

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

-- LEFT JOIN

SELECT Product.ProductName, Seller.SellerName

FROM Product

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

-- RIGHT JOIN

SELECT Product.ProductName, Seller.SellerName

FROM Product

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

-- FULL OUTER JOIN

SELECT Product.ProductName, Seller.SellerName

FROM Product

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

1. **What is API Testing?**
   * API testing is a way to check if an application programming interface (API) works correctly. An API is like a messenger that allows different software systems to talk to each other. Since APIs don’t have a user interface (like buttons or screens), testing them is done by sending requests and checking the responses.
2. **Types of API testing**

**i) Unit Testing**:

* + Testing individual endpoints or functions within the API.
  + Ensures each part of the API works correctly in isolation.

**ii) Integration Testing**:

* + Testing the interaction between different APIs or between the API and other systems.
  + Ensures that integrated components work together as expected.

**iii) End-to-End Testing**:

* + Testing the entire flow of the API from start to finish.
  + Ensures the API works correctly in a real-world scenario.

**iv) Load Testing**:

* + Testing the API under expected and peak load conditions.
  + Ensures the API can handle high traffic without performance degradation.

**v) Security Testing**:

* + Testing the API for vulnerabilities and ensuring data protection.
  + Ensures the API is secure from unauthorized access and attacks

1. **What is Responsive Testing?**
   * **Responsive Testing** is a type of testing used to ensure that a website or application works well and looks good on different devices, screen sizes, and resolutions. With so many people using smartphones, tablets, laptops, and desktops, it’s important that a website or app adapts to all these devices seamlessly. This is called **responsiveness**, and testing for it is called **responsive testing**.
2. **Which types of tools are available for Responsive testing?**

 **Browser-Based Tools:**

* Developer Tools (Chrome, Firefox, Safari)
* Browser Extensions (e.g., Window Resizer, Responsive Design Mode)

 **Online Responsive Design Checkers:**

* Responsinator
* Am I Responsive?
* Screenfly

 **Cross-Browser Testing Platforms:**

* BrowserStack
* LambdaTest

 **Emulators and Simulators:**

* Android Emulators
* iOS Simulators

 **Responsive Design Frameworks (with testing utilities):**

* Bootstrap
* Foundation

1. **What is the full form of .ipa, .apk?**
   * **.ipa**- iOS app Store Pack
   * **.apk-** Android package kit