Module 3 (Testing on Live Application)

1) What is RDBMS

* 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 a specific attribute or field of that record.

2) What is SQL

* SQL (Structured Query Language) is a standard programming language used to manage and manipulate relational databases.

3) Write SQL Commands

* **Creating a Table**:
* sqlCopy code
* CREATE TABLE Employees (  
   EmployeeID INT PRIMARY KEY,  
   FirstName VARCHAR(50),  
   LastName VARCHAR(50),  
   Department VARCHAR(50),  
   Salary DECIMAL(10, 2)  
  );
* **Inserting Data**:
* sqlCopy code
* INSERT INTO Employees (EmployeeID, FirstName, LastName, Department, Salary)  
  VALUES (1, 'John', 'Doe', 'IT', 50000.00);
* **Selecting Data**:
* sqlCopy code
* SELECT \* FROM Employees;
* **Updating Data**:
* sqlCopy code
* UPDATE Employees  
  SET Salary = 55000.00  
  WHERE EmployeeID = 1;
* **Deleting Data**:
* sqlCopy code
* DELETE FROM Employees  
  WHERE EmployeeID = 1;
* **Querying with Conditions**:
* sqlCopy code
* SELECT \* FROM Employees  
  WHERE Department = 'IT' AND Salary > 60000.00;
* **Creating Index**:
* sqlCopy code
* CREATE INDEX idx\_department ON Employees (Department);
* **Altering Table (Adding a Column)**:
* sqlCopy code
* ALTER TABLE Employees  
  ADD COLUMN Email VARCHAR(100);
* **Dropping a Table**:
* sqlCopy code
* DROP TABLE Employees;
* **Granting Privileges**:
* sqlCopy code
* GRANT SELECT, INSERT, UPDATE, DELETE ON Employees TO user1;

4) What is join?

* In the context of relational databases, a "join" is an operation that combines rows from two or more tables based on a related column between them. It allows you to retrieve data from multiple tables simultaneously by specifying how the tables are related.

5) Write type of joins

* There are four main types of JOINs in SQL:
* INNER JOIN,
* OUTER JOIN,
* CROSS JOIN, and
* SELF JOIN.

6) How Many constraint and describes it self

* In SQL, there are several types of constraints that you can apply to columns in a table to enforce rules for the data they contain. Here are some common constraints and their descriptions:
* **NOT NULL**: Ensures that a column cannot contain NULL values. Every row must have a value for that column.
* **UNIQUE**: Ensures that all values in a column are unique (no duplicate values). It can be applied to one or more columns.
* **PRIMARY KEY**: A combination of NOT NULL and UNIQUE constraints. It uniquely identifies each row in a table and ensures that the value of the column or columns specified as the primary key is unique for each row.
* **FOREIGN KEY**: Establishes a relationship between two tables. It ensures referential integrity by requiring that the values in a column (or columns) in one table match the values in a column (or columns) in another table's primary key.
* **CHECK**: Allows you to specify a condition that each row must satisfy. If the condition evaluates to false for any row, the constraint prevents the row from being inserted or updated.
* **DEFAULT**: Specifies a default value for a column. If a value is not provided for the column during an INSERT operation, the default value is used.
* **INDEX**: Improves the speed of data retrieval operations (such as SELECT statements) on a table by creating an index on one or more columns. Indexes can be created for columns with unique values or frequently searched columns to optimize query performance.

7) Difference between RDBMS vs DBMS

|  |  |
| --- | --- |
| **DBMS** | **RDBMS** |
| Data is stored in a database management system (DBMS) as a file | Tables are used to store information |
| Data is stored in a database management system (DBMS) in either a navigational or hierarchical format | RDBMS employs a tabular format, with column names as headers and associated data as rows |
| Only a single user is supported by the DBMS | It may be used by numerous people |
| The data in a typical database may not be stored according to the ACID model  This can lead to database discrepancies | Relational databases are more difficult to create, but they are more consistent and organised  They follow the rules of ACID (Atomicity, Consistency, Isolation, Durability) |
| It is an application that is used to manage databases over computer networks as well as the system hard drives | The database systems are used to keep track of the relationships between the tables |
| Software and hardware requirements are minimal | Higher hardware and software requirements are required |

8) What is API Testing

* API testing involves evaluating the functionality, performance, reliability, and security of Application Programming Interfaces (APIs).

9) Types of API Testing

* **Unit Testing**: This involves testing individual units or components of the API in isolation, typically at the code level. Unit tests focus on verifying the behavior of specific functions, methods, or modules within the API to ensure that they produce the expected output for a given input.
* **Functional Testing**: Functional testing validates the functional behavior of the API by testing its endpoints and verifying that they perform as expected according to the API specifications or requirements. This includes testing different types of requests (GET, POST, PUT, DELETE) and ensuring that the API returns the correct responses, handles errors gracefully, and adheres to the defined contract.
* **Integration Testing**: Integration testing evaluates how the API interacts with other components, services, or systems within the application ecosystem. It verifies that the API integrates seamlessly with other software components and that data is exchanged correctly between them. Integration testing can involve testing APIs in conjunction with databases, message queues, authentication services, or third-party APIs.
* **End-to-End Testing**: End-to-end testing involves testing the entire workflow or process that involves the API, from the client request to the backend processing and response. It verifies that the API functions correctly in real-world scenarios and that all components of the system work together harmoniously to achieve the desired outcome.
* **Load Testing**: Load testing assesses the performance and scalability of the API under various load conditions. It involves subjecting the API to simulated loads, such as concurrent user requests or high transaction volumes, to measure its response times, throughput rates, and resource utilization. Load testing helps identify performance bottlenecks and ensures that the API can handle expected levels of traffic without degradation in performance.
* **Security Testing**: Security testing evaluates the security posture of the API by identifying and mitigating potential vulnerabilities and security threats. It includes testing for authentication mechanisms, authorization controls, data encryption, input validation, and protection against common security risks such as injection attacks, cross-site scripting (XSS), and cross-site request forgery (CSRF).
* **Usability Testing**: Usability testing assesses the ease of use and user experience of the API, focusing on aspects such as API documentation, error messages, response formats, and ease of integration. It ensures that the API is intuitive and developer-friendly, making it easy for developers to understand, use, and integrate into their applications.

10) What is Responsive Testing?

* Responsive testing is a type of testing that evaluates how a website or web application behaves and appears across different devices, screen sizes, resolutions, and orientations.

11) Which types of tools are available for Responsive Testing.

* **Browser Developer Tools**: Most modern web browsers, such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge, come with built-in developer tools that include responsive design testing features. These tools allow you to emulate various devices, view the website in different viewport sizes, and inspect the responsive layout and elements. Examples include Chrome DevTools, Firefox Developer Tools, Safari Web Inspector, and Microsoft Edge DevTools.
* **Responsive Design Testing Websites**: There are online tools and websites specifically designed for responsive testing, allowing you to enter a website URL and preview how it looks on different devices and screen sizes. These tools often provide a simple interface for selecting device types, orientations, and resolutions. Examples include Responsinator, BrowserStack Responsive, Am I Responsive?, and Screenfly.
* **Emulators and Simulators**: Emulators and simulators are software tools that simulate the behavior of mobile devices and allow you to test websites and applications in a virtual environment. They provide a way to emulate different device configurations, screen sizes, and operating systems without needing physical devices. Examples include Android Virtual Device (AVD) Manager for Android emulation, iOS Simulator for iOS app testing, and Genymotion for Android emulation.
* **Cross-Browser Testing Platforms**: Cross-browser testing platforms enable testing across multiple browsers and devices to ensure compatibility and consistency. While primarily focused on browser compatibility testing, these platforms often include responsive testing features that allow you to test websites across various devices and screen sizes. Examples include BrowserStack, Sauce Labs, CrossBrowserTesting, and LambdaTest.
* **Responsive Design Testing Plugins and Extensions**: Some browser extensions and plugins provide additional features for responsive testing directly within the browser. These tools often integrate with browser developer tools and provide additional functionalities such as side-by-side comparisons, pixel measurements, and screenshot capturing. Examples include Window Resizer, Responsive Viewer, Responsive Web Design Tester, and Responsive Inspector.
* **Automated Testing Frameworks**: Automated testing frameworks and tools can also be used for responsive testing as part of continuous integration (CI) and continuous delivery (CD) pipelines. These tools allow you to automate the execution of responsive test cases across different devices and screen sizes, helping to ensure consistent behavior and layout across releases. Examples include Selenium WebDriver, Puppeteer, Appium, and Cypress.

12) What is the full form of .ipa, .apk

* Apk **Android Application Package**.
* Ipk iphone apllication pacakage.

13) How to create step for to open the developer option mode ON?

* **Open Settings**: Go to the home screen of your Android device and locate the Settings app. It's usually represented by a gear icon.
* **Navigate to About Phone**: In the Settings menu, scroll down and find the "About phone" or "About device" option. This may be located at the bottom of the Settings menu or within a sub-menu.
* **Find Build Number**: In the "About phone" section, look for an option called "Build number" or "Software information." This option may vary slightly depending on the Android version and manufacturer.
* **Tap Build Number Multiple Times**: Tap on the "Build number" option repeatedly, usually about seven times in quick succession. You may be prompted to enter your device's PIN, pattern, or password for security verification.
* **Enable Developer Options**: After tapping the "Build number" multiple times, you should see a message indicating that Developer Options have been enabled. This message may vary but typically states something like "You are now a developer!" or "Developer mode has been enabled."
* **Access Developer Options**: Now, go back to the main Settings menu, and you should see a new option called "Developer options" or "Developer settings." It is usually located towards the bottom of the Settings menu, above or below the "About phone" section.
* **Open Developer Options**: Tap on the "Developer options" menu to access a wide range of advanced settings and features that are typically hidden from regular users.