1. What is RDBMS?

* RDBMS stands for Relational Database Management System.
* RDBMS is a program used to maintain a relational database.
* RDBMS is the basis for all modern database systems such as MySQL, Microsoft SQL Server, Oracle, and Microsoft Access.
* RDBMS uses [SQL queries](https://www.w3schools.com/sql/default.asp) to access the data in the database.
* A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codded in 1970.

1. What is SQL?

* SQL stands for Structured Query Language
* SQL lets you access and manipulate databases
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

3. Write SQL Commands

* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index

4. what is Join? Write types of join

* A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

* (INNER) JOIN: Returns records that have matching values in both tables
* LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
* RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
* FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table



5. how many constraints and describe itself

A constraint is a rule that is used for optimization purposes.

Constraints can be categorized into five types:

* A NOT NULL constraint is a rule that prevents null values from being entered into one or more columns within a table.
* A unique constraint (also referred to as a unique key constraint) is a rule that forbids duplicate values in one or more columns within a table. Unique and primary keys are the supported unique constraints. For example, a unique constraint can be defined on the supplier identifier in the supplier table to ensure that the same supplier identifier is not given to two suppliers.
* A primary key constraint is a column or combination of columns that has the same properties as a unique constraint. You can use a primary key and foreign key constraints to define relationships between tables.
* A foreign key constraint (also referred to as a referential constraint or a referential integrity constraint) is a logical rule about values in one or more columns in one or more tables. For example, a set of tables shares information about a corporation's suppliers. Occasionally, a supplier's name changes. You can define a referential constraint that states the ID of the supplier in a table must match a supplier ID in the supplier information. This constraint prevents insert, update, or delete operations that would otherwise result in missing supplier information.
* A table check constraint (also called a check constraint) sets restrictions on data that is added to a specific table. For example, you can use a table check constraint whenever salary data is added or updated in a table that contains personnel information. For such operations, the table check constraint can ensure that the salary level for an employee is at least $20 000.

An informational constraint is an attribute of a certain type of constraint, but the attribute is not enforced by the database manager.

* [**NOT NULL constraints**](https://www.ibm.com/docs/en/SSHRBY/com.ibm.swg.im.dashdb.admin.dbobj.doc/doc/c0052354.html)  
  NOT NULL constraints prevent null values from being entered into a column.
* [**Unique constraints**](https://www.ibm.com/docs/en/SSHRBY/com.ibm.swg.im.dashdb.admin.dbobj.doc/doc/c0020151.html)  
  Unique constraints ensure that the values in a set of columns are unique and not null for all rows in the table. The columns specified in a unique constraint must be defined as NOT NULL. The database manager uses a unique index to enforce the uniqueness of the key during changes to the columns of the unique constraint.
* [**Primary key constraints**](https://www.ibm.com/docs/en/SSHRBY/com.ibm.swg.im.dashdb.admin.dbobj.doc/doc/c0020150.html)  
  You can use primary key and foreign key constraints to define relationships between tables.
* [**(Table) Check constraints**](https://www.ibm.com/docs/en/SSHRBY/com.ibm.swg.im.dashdb.admin.dbobj.doc/doc/c0020152.html)  
  A check constraint (also referred to as a table check constraint) is a database rule that specifies the values allowed in one or more columns of every row of a table. Specifying check constraints is done through a restricted form of a search condition.
* [**Foreign key (referential) constraints**](https://www.ibm.com/docs/en/SSHRBY/com.ibm.swg.im.dashdb.admin.dbobj.doc/doc/c0020153.html)  
  Foreign key constraints (also known as referential constraints or referential integrity constraints) enable definition of required relationships between and within tables.
* [**Informational constraints**](https://www.ibm.com/docs/en/SSHRBY/com.ibm.swg.im.dashdb.admin.dbobj.doc/doc/c0023324.html)  
  An informational constraint is a constraint attribute that can be used by the SQL compiler to improve the access to data. Informational constraints are not enforced by the database manager, and are not used for additional verification of data; rather, they are used to improve query performance.

6. What is responsive testing?

* Responsive website testing is a process that ensures your website works well on multiple devices by using CSS media queries based on the user's device where the website is accessed.
* In simpler terms, responsive testing is a process that enables you to check how well a website works on various types of devices, including desktops and smartphones. A website that responds well to all screen sizes and resolutions gives your business a competitive edge over other companies.
* Responsive design incorporates many elements, including media queries, flexible grids, and responsive typography. It makes it easy to build websites that adjust automatically to any screen size. While a responsive design may seem simple, incorporating it into ongoing projects is tricky; it's best to follow its principles before starting a new project.
* Website responsive testing is part of the final stage of responsive web design testing. It can be performed using the same toolset as cross-browser testing, which is responsible for improving a website's UI/UX. Responsive testing ensures that your website is not only cross-browser compatible but also adjusts to screen resolution changes.

Types of Responsive Testing:

* **Visual Regression Testing** Visual regression testing is a part of regression testing that involves taking screenshots of a website on different devices and comparing them to ensure that the layout and design are consistent across all screens.
* **Visual Layout Testing:** Visual layout testing tools allow users to check that the website's layout adjusts correctly to different screen sizes and orientations and that all content is displayed correctly and is easily readable and navigable.
* **Cross browser testing:** Cross browser compatibility testing is the most significant kind of front end testing. Testers can determine if a website functions as intended when viewed using various browsers/devices/OS combinations. In addition, cross browser testing makes it possible for people to experience the same thing across multiple browsers.
* **Functional Testing:** This involves testing the website's functionality on different devices to ensure that all features and interactions work as expected. Functional testing evaluates the various functions of the application. It checks the user interface, database, APIs, client/server communication, security, and other components.
* **Performance Testing:** Performance testing assesses a product's quality and capability under varying workloads. Performance testing ensures that the system performs adequately, reliably, and with stability. This involves testing the website's performance on different devices and networks to ensure that it loads quickly and runs smoothly.
* **Usability Testing:** Usability testing is a technique for evaluating the user experience of a web product or service by testing it with users. This involves testing the website's usability on different devices to ensure it is easy for users to navigate and use.

7. Which types of tools are available for responsive testing?

1. Testsigma:

2. Responsinator:

3. Screenfly:

4. LambdaTest:

5. Am I Responsive?

6. CrossBrowserTesting:

7. Browserstack:

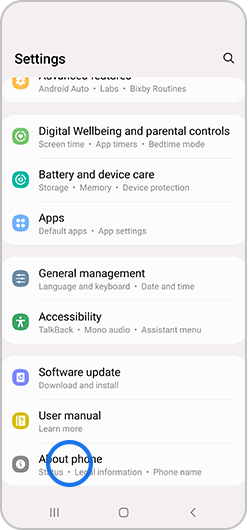
8. What is the full form of .apk and .ipa?

.APK: Android Application Package

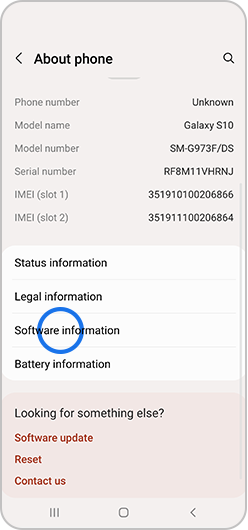
.IPA: IOS Package App store

9. How to create a step for enable developer mode in android

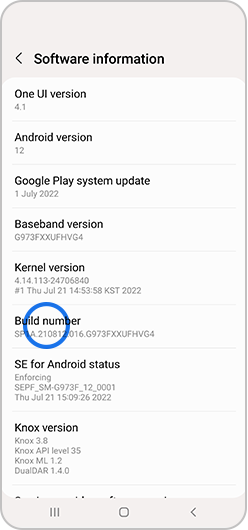
1. Go to "Settings"



2. Tap "About device" or "About phone"



3. Tap “Software information”

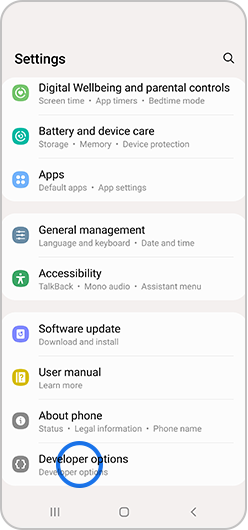


4. Tap “Build number” seven times

Depending on your device and operating system,

you may not need to follow step number 3.

5. Enter your pattern, PIN or password to enable the Developer options menu

6. The "Developer options" menu will now appear in your Settings menu

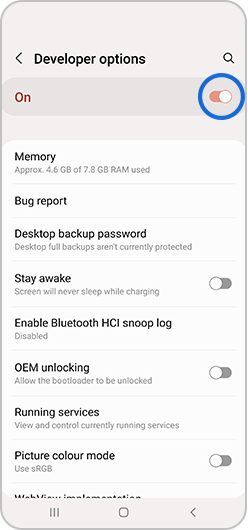
Depending on your device,

it may appear under

Settings > General > Developer options.

7. To disable the Developer

options menu, tap the switch

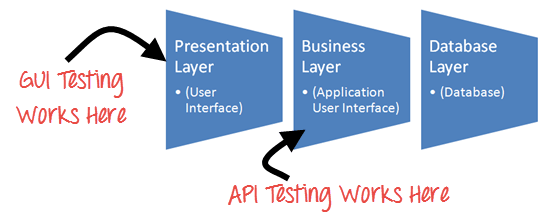


10. Difference Between DBMS vs RDBMS

| **DBMS** | **RDBMS** |
| --- | --- |
| [DBMS](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/) stores data as file. | [RDBMS](https://www.geeksforgeeks.org/rdbms-architecture/) 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, PostgreSQL, SQL Server, Oracle, Microsoft Access etc. |

11. what is API testing.

* **API Testing** is a software testing type that validates Application Programming Interfaces (APIs). The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces.
* In API Testing, instead of using standard user inputs(keyboard) and outputs, you use software to send calls to the API, get output, and note down the system’s response.
* API tests are very different from GUI Tests and won’t concentrate on the look and feel of an application.
* It mainly concentrates on the business logic layer of the software.



Set-up of API Test Automation Environment

API automation Testing requires an application that can be interacted via an API. In order to test an API, you will need to

1. Use Testing Tool to drive the API
2. Write your own code to test the API

**Few Points to note**

* API Testing is different than other software testing types as GUI is not available, and yet you are required to set up initial environment that invokes API with a required set of parameters and then finally examine the test result.
* Hence, Setting up a testing environment for API automation testing seems a little complex.
* Database and server should be configured as per the application requirements.
* Once the installation is done, the API Function should be called to check whether that API is working.

Types of Output of an API

An output of API could be

1. Any type of data
2. Status (say Pass or Fail)
3. Call another API function.

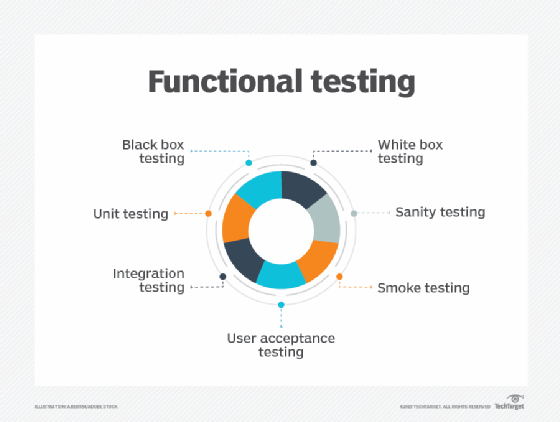
12. Types of API testing

1. Validation testing

* Validation testing analyzes API projects based on three distinct sets of criteria: the API's usability as a product, its transactional behavior and its operational efficiency. Here are some typical questions asked during validation testing:
* Is the API designed in a way that meets its product goals or solves the problem it's supposed to?
* Were there any major coding missteps that would push the API in an unsustainable direction?
* Is the API accessing data in accordance with predefined policies?
* Is the API storing data in accordance with security or compliance rules?
* Would any code alterations improve the API's overall functionality?

2. Functional testing

* [Functional testing](https://www.techtarget.com/searchsoftwarequality/definition/functional-testing) ensures the API performs exactly as it is supposed to. This test analyzes specific functions within the codebase to guarantee the API functions within its expected parameters and can handle errors when the results are outside the designated parameters.



3. Load testing

* Load testing is used to see how many calls an API can handle. This test is often performed after a specific unit or codebase is completed to determine whether the theoretical solution can also work as a practical solution when acting under a given load.

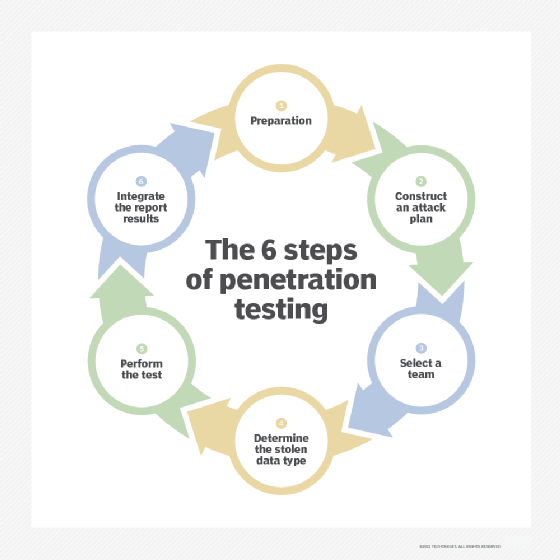
4. Reliability testing

* Reliability testing ensures the API can produce consistent results and the connection between platforms is reliable.

5. Security testing

* Security testing attempts to validate the encryption methods the API uses as well as the access control design. It includes the validation of authorization checks for resource access and user rights management.

6. Penetration testing

* [Penetration testing](https://www.techtarget.com/searchsecurity/definition/penetration-testing) builds on security testing. In this type of test, the API is attacked by a person with limited or no knowledge of the software. This enables testers to analyze the [attack vector](https://www.techtarget.com/searchsecurity/definition/attack-vector) from an outside perspective. The attacks used in penetration testing can be limited to specific elements of the API or can target the API in its entirety.

7. Fuzz testing

* [Fuzz testing](https://www.techtarget.com/searchsecurity/definition/fuzz-testing) forcibly inputs huge amounts of random data -- also called noise or fuzz -- into the system, attempting to create negative behavior, such as a forced crash or overflow.

8. Unit testing

* [Unit testing](https://www.techtarget.com/searchsoftwarequality/definition/unit-testing) is a testing process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. The process of unit testing an API includes testing single endpoints with a single request.

9. Integration Testing

* Integration tests are a type of software testing in which the different units, modules or components of an application are tested as a combined entity. Because APIs are used in integrations between two or more pieces of software, an integration test analyzes how the API integrates the software.