**1**> **What is RDBMS**

**:-** A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

**A relational database example**

Here’s a simple example of two tables a small business might use to process orders for its products. The first table is a customer info table, so each record includes a customer’s name, address, shipping and billing information, phone number, and other contact information. Each bit of information (each attribute) is in its own column, and the database assigns a unique ID (a key) to each row. In the second table—a customer order table—each record includes the ID of the customer that placed the order, the product ordered, the quantity, the selected size and colour, and so on—but not the customer’s name or contact information.

These two tables have only one thing in common: the ID column (the key). But because of that common column, the relational database can create a relationship between the two tables. Then, when the company’s order processing application submits an order to the database, the database can go to the customer order table, pull the correct information about the product order, and use the customer ID from that table to look up the customer’s billing and shipping information in the customer info table. The warehouse can then pull the correct product, the customer can receive timely delivery of the order, and the company can get paid.

**2**> **What is SQL**

**:-** Structured query language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. You can use SQL statements to store, update, remove, search, and retrieve information from the database. You can also use SQL to maintain and optimize database performance.

**3**> **Write SQL Commands**

**:-** SQL stands for Structured Query Language. SQL commands are the instructions used to communicate with a database to perform tasks, functions, and queries with data.

SQL commands can be used to search the database and to do other functions like creating tables, adding data to tables, modifying data, and dropping tables.

Here is a list of basic SQL commands (sometimes called clauses) you should know if you are going to work with SQL.

### **1.SELECT and FROM**

The SELECT part of a query determines which columns of the data to show in the results. There are also options you can apply to show data that is not a table column.

The example below shows three columns SELECTED FROM the “student” table and one calculated column. The database stores the student ID, First Name, and Last Name of the student. We can combine the First and the Last name columns to create the Full Name calculated column.

Example:

SELECT studentID, FirstName, LastName, FirstName + ' ' + LastName AS FullName

FROM student;

### **2.CREATE TABLE**

CREATE TABLE does just what it sounds like: it creates a table in the database. You can specify the name of the table and the columns that should be in the table.

Example:

CREATE TABLE table\_name (

column\_1 datatype,

column\_2 datatype,

column\_3 datatype

);

### **3.ALTER TABLE**

ALTER TABLE changes the structure of a table. Here is how you would add a column to a database:

Example:

ALTER TABLE table\_name

ADD column\_name datatype;

### **4.CHECK**

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a single column it allows only certain values for this column. If you define a CHECK constraint on a table, it can limit the values in certain columns based on values in other columns in the row.

Example:

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

CHECK (Age>=18)

);

### **5.WHERE**

**(AND, OR, IN, BETWEEN, and LIKE)**

The WHERE clause is used to limit the number of rows returned.

As an example, first we will show you a SELECT statement and results without a WHERE statement. Then we will add a WHERE statement that uses all five qualifiers above.

Example:

SELECT studentID, FullName, sat\_score, rcd\_updated FROM student;

### **6.UPDATE**

To update a record in a table you use the UPDATE statement.

Use the WHERE condition to specify which records you want to update. It is possible to update one or more columns at a time. The syntax is:

Example:

UPDATE table\_name

SET column1 = value1,

column2 = value2, ...

WHERE condition;

### **7.GROUP BY**

GROUP BY allows you to combine rows and aggregate data.

Here is the syntax of GROUP BY:

Example:

SELECT column\_name, COUNT(\*)

FROM table\_name

GROUP BY column\_name;

### **8.HAVING**

HAVING allows you to filter the data aggregated by the GROUP BY clause so that the user gets a limited set of records to view.

Here is the syntax of HAVING:

**Example:**

SELECT column\_name, COUNT(\*)

FROM table\_name

GROUP BY column\_name

HAVING COUNT(\*) > value;

**4**> **What is join?**

:- A join is a way to combine data from two or more tables based on a common column. For example, let’s say we have two tables: Customers and Orders. The Customers table contains information about each customer, including their name, address, and email address. The Orders table contains information about each order, including the order date, product name, and quantity.

To combine data from these two tables, we can join them on the customer ID column, which is common to both tables. By doing so, we can retrieve information about each customer’s orders in a single query.

**5**> **Write type of joins**

## :- Types of Joins

There are several types of joins in DBMS, each with its own syntax and use case. The following are the different types of Joins in DBMS.

1.Inner Join

2.Natural Join

3.Outer Join

- Left Outer Join

- Right Outer Join

- Full Outer Join

### **1.Inner Join**

An Inner Join returns only the rows in both tables that match the join condition.

### **2.Natural Join**

A Natural Join is a type of Join that matches columns with the same name in both tables.

### **3.Outer Join**

An Outer Join in DBMS returns all the rows from one table and the matching rows from the other table. If there is no match, NULL values are returned for the missing rows.

* **Left Outer Join**

A Left Outer Join in DBMS returns all the rows from the left table and the matching rows from the right table. If there is no match, NULL values are returned for the missing rows.

**Syntax of Left Outer Join**

SELECT table1.column1, table2.column2

FROM table1

LEFT JOIN table2

ON table1.columnX = table2.columnY;

* **Right Outer Join**

A Right Outer Join returns all the rows from the right table and the matching rows from the left table. If there is no match, NULL values are returned for the missing rows.

**Syntax of Right Outer Join**

SELECT table1.column1, table2.column2

FROM table1

RIGHT JOIN table2

ON table1.columnX = table2.columnY;

* **Full Outer Join**

A Full Outer Join returns all the rows from both tables and NULL values for the missing rows.

**Syntax of Full Outer Join**

SELECT table1.column1, table2.column2

FROM table1

FULL OUTER JOIN table2

ON table1.columnX = table2.columnY;

**6**> **How Many constraint and describes it self**

# **:- Types of constraints**

* [**NOT NULL constraints**](https://www.ibm.com/docs/en/SSEPGG_11.5.0/com.ibm.db2.luw.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/SSEPGG_11.5.0/com.ibm.db2.luw.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/SSEPGG_11.5.0/com.ibm.db2.luw.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/SSEPGG_11.5.0/com.ibm.db2.luw.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/SSEPGG_11.5.0/com.ibm.db2.luw.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/SSEPGG_11.5.0/com.ibm.db2.luw.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.

**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 |
| The integrity constraints are not supported by DBMS  At the file level, the integrity constraints are not imposed | At the schema level, RDBMS provides integrity restrictions  Values outside of a certain range cannot be stored in the RDBMS column |
| Normalization is not supported by DBMS. | A relational database management system (RDBMS) can be normalised. |
| Distributed databases are not supported by DBMS | Distributed databases are supported by RBMS |

**8**> **What is API Testing?**

**:-** API testing is a type of software testing that analyses an application program interface (API) to verify that it fulfils its expected functionality, security, performance and reliability. The tests are performed either directly on the API or as part of integration testing.

An API is code that enables the communication exchange of data between two software programs. An application typically consists of multiple layers, including an API layer. API layers focus on the business logic in applications, defining requests such as how to make them and the data formats used.

As opposed to user interface (UI) testing, which focuses on validating the application's look and feel, API testing focuses on analysing the application's business logic as well as security and data responses. An API test is generally performed by making requests to one or more API endpoints and comparing the responses with expected results.

API testing is frequently automated and used by DevOps, quality assurance and development teams for continuous testing practices. API testing is generally performed by using software to send calls to API endpoints to validate the system's response.

**9**> **Types of API Testing**

## :-

## 9 Types of API Testing

### **1. Validation Testing**

This type of testing ensures that the API is returning the expected results and in the correct format. Validation testing involves checking that the input parameters, output format, response code, and data type are correct.

### **2. UI Testing**

UI testing validates that the API works correctly within the application’s user interface. This type of testing ensures that the UI is accurately reflecting the API’s results and that the API is handling the UI’s inputs correctly.

### **3. Functional Testing**

Functional testing verifies that the API functions correctly and meets the required specifications. This type of testing can include testing the API’s business logic, input validation, output validation, and error handling.

### **4. Load Testing**

Load testing involves testing the API’s performance and stability under stressful conditions. This type of testing simulates high traffic and heavy usage scenarios to ensure that the API can handle a large number of concurrent users and requests.

### **5. Runtime and Error Detection**

This type of testing ensures that the API can handle runtime errors and exceptions. This includes testing for network timeouts, memory leaks, incorrect input parameters, and other errors that can occur during runtime.

### **6. Penetration Testing**

Penetration testing is a type of security testing that involves simulating attacks from hackers to detect vulnerabilities and weaknesses in the API. This type of testing can include network scanning, vulnerability scanning, and manual penetration testing.

### **7. API Hacking**

API hacking is security testing techniques that exploits vulnerabilities in an API. Attackers (and testers) can target API endpoints to gain access to data, disrupt services, or hijack the entire system. Ethical hackers can train by attacking intentionally vulnerable APIs, which can be downloaded from the Internet. Then, they can turn to the organization’s own APIs to test their resilience and find weaknesses.

### **8. Security Testing**

Security testing aims to identify security-related vulnerabilities and flaws in the API and ensure that the API meets the required security standards. This type of testing includes testing for vulnerabilities such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and others.

### **9. Fuzz Testing**

Fuzz testing involves feeding unexpected and invalid inputs into the API to test its ability to handle unexpected input and recover from errors. This type of testing can uncover security vulnerabilities or unexpected behaviour in the API.

**10**> **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.

**11**> **Which types of tools are available for Responsive Testing**

## :-

## ****7 Responsive Web Design Testing Tools****

### **1. Test sigma:**

Test sigma, an automated cross browser and responsiveness testing tool, provides you with more than 1000+ Browser-OS and 2000+ iOS and Android devices on-demand to perform extensive automated responsiveness testing of your websites and applications. With features such as script less automation testing, parallel testing of test suites, drill-down test reports, and many more, Test sigma should be the go-to tool for all your testing needs.

### **2. Responsinator:**

Responsinator is the simplest tool used widely for testing the responsiveness of websites. With a simple and sleek interface, you just need to enter the URL of your website. Without any hassles, you can easily see how the pages are rendered on some generic screens.

### **3. Screenfly:**

Screenfly is a free, in-browser tool that allows you to perform responsiveness testing on different screen sizes and devices. It allows you to add any custom screen size you want. Its featured devices include laptops, tablets, smartphones, desktops, and televisions. It allows you to rotate, scroll, and perform more such actions using simple click buttons at the top.

### **4. LambdaTest:**

The LTBrowser application by LambdaTest provides all the sets of tools needed to perform responsive website testing, responsive mobile test and make the website mobile-friendly. It contains around 27+ devices and it also allows you to create custom devices. It also allows you to perform complete website auditing and share bug reports with your team.

### **5. Am I Responsive?:**

It provides its users with 4 different apple devices with resolutions – desktop (1600 x 992 Pixel), laptop (1280 x 802 Pixel), tablet (768 x 1024 Pixel), and mobile (320 x 480 Pixel). It allows all these different sizes to be compared directly. It is quick, simple, and easy to use.

### **6. CrossBrowserTesting:**

It offers 2050+ real desktop and mobile browsers to test website for responsiveness. You can easily validate public or locally hosted pages across multiple mobile browsers for complete compatibility. Similar to Testsigma, it is a one-stop solution to perform parallel automated tests, compare screenshots visually, remote debug the code, and interact with the website on real-world devices.

### **7. Browserstack:**

It gives your users a seamless experience by allowing you to perform responsiveness testing on 2000+ real browsers and iOS and Android devices. It provides large testing coverage and the breadth of testing options is invaluable. Its ease-of-use tools allow you to perform responsiveness testing quickly and efficiently.

**12**> **What is the full form of .ipa, .apk**

**:-** The full form of APK is an**Android Application Package**. APK is the application file type used in the Android operating system, as well as a wide range in many other Android-based operating applications in mobile phones, video games & middleware for distribution and installation. APK is similar to other Microsoft Windows software packages such as Delian package or APPX in Delian-based software applications.

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

**:-** Your Android phone is more than just an everyday tech device, it’s a lifestyle. And now that [modern smartphones](https://www.digitaltrends.com/mobile/best-smartphones/) are more feature-packed than ever, it’s hard to imagine a world where the [Samsung Galaxy line up](https://www.digitaltrends.com/mobile/samsung-galaxy-s24-review/) and [Google Pixel family](https://www.digitaltrends.com/mobile/google-pixel-8-review/) can do even more. But screens and menus of untapped settings and hidden features are cleverly tucked away in your phone’s developer options. Fortunately, unlocking these additional customizations isn’t rocket science, and we’ve put together this guide to help you along the way.

If you’ve ever wanted to enable USB debugging, or reduce the speed of certain phone automations, you’ll want to unlock developer options. Here’s how.

Be aware before we start: Developer options are hidden for a reason. The entire menu has been designed for use by developers who want to test functionalities and make changes that may impact your phone in one way or another. It may improve your performance, but it may also have the opposite effect. It's not a good idea to go in and change things without an understanding of each option and a reason for doing so. So make sure you've done your homework before you start.

## How to access developer options on your Android phone

You can unlock the developer options on any Android [smartphone](https://www.digitaltrends.com/mobile/best-smartphones/) or [tablet](https://www.digitaltrends.com/mobile/best-android-tablets/) by locating the build number in your Settings menu and tapping it multiple times. However, the exact location of the aforementioned build number may differ depending on your phone’s manufacturer. The following example uses a [Google Pixel 7 Pro](https://www.digitaltrends.com/mobile/google-pixel-7-pro-review/) running [Android 13](https://www.digitaltrends.com/mobile/android-13-phones-list/).

**Step 1:** Go to Settings > About phone.

**Step 2:** Scroll down to Build number.

**Step 3:** Tap Build number seven times. After the first few taps, you should see the steps counting down until you unlock the developer options. You may also have to tap in your PIN for verification.

**Step 4:** Once developer options are activated, you will see a message that reads, you are now a developer.

**Step 5:** Go back to the Settings pane and head to System, where you will now find Developer options as an entry.

**Step 6:** Tap it and toggle the switch on if it is not already, and from there, you can proceed to make adjustments to your phone.