(1) What is RDBMS?

- In software testing, Relational Database Management Systems (RDBMS) like MySQL, Oracle, or SQL Server play a crucial role in verifying the data integrity, functionality, and performance of applications that interact with databases.
- RDBMS provides tools and methods for testing various aspects of database systems, including functional, relationship, data quality, structural, and performance aspects.
- Testers use SQL to interact with RDBMS and write test cases to verify that the database behaves as expected, and that data is stored and retrieved accurately.
- Example:
  (i) Here's a simple database schema for tracking sales
  (ii) 3 tables, related by primary keys (CID called as Customer, OID called as Order, PID called as Product)
  (iii)primary keys (in boldface) are unique record identifiers
  (iv)customer may place order for one product at a time

(2) What is SQL?

- To generate SQL, type a comment in the query editor starting with -- followed by a single-line comment, and then press Return.
- For example, if you enter the prompt -- add a row to table singers and press Return, then Gemini generates SQL that's similar to the following: See more code actions.
- SQL tutorial gives unique learning on Structured Query Language and it helps to make practice on SQL commands which provides immediate results.
- SQL is a language of database, it includes database creation, deletion, fetching rows and modifying rows etc.
- SQL is an ANSI (American National Standards Institute) standard but there are many different versions of the SQL language.
- SQL is the standard programming language of relational DBs
- SQL is a standard computer language for accessing and manipulating databases.
- 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.

☑ SQL important for following reason:

- Allows users to access data in relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures, and views

(3) Write SQL Commands

☑ SQL Commands are following

- DDL (Data Definition Language)_Create, Alter, Rename, Drop, Truncate
  - SQL CREATE DATABASE STATEMENT: CREATE DATABASE database_name;
  - SQL DROP DATABASE Statement: DROP DATABASE database_name;
  - SQL USE STATEMENT USE DATABASE database_name;
  - SQL CREATE TABLE STATEMENT CREATE TABLE table_name(column1 datatype, column2 datatype, column3 datatype, ... , columnN datatype, PRIMARY KEY(one or more columns));
  - SQL DROP TABLE STATEMENT DROP TABLE table_name;
  - SQL TRUNCATE TABLE STATEMENT TRUNCATE TABLE table_name;
  - SQL ALTER TABLE STATEMENT ALTER TABLE table_name{ADD|DROP|MODIFY}column_name{data_type};
  - SQL ALTER TABLE STATEMENT (RENAME) ALTER TABLE table_name RENAME TO new_table_name;
- DML (Data Manipulation Language)_Insert, Update, Delete
  - SQL INSERT INTO STATEMENT INSERT INTO table_name(column1, column2....columnN) VALUES (value1, value2....valueN);
  - SQL UPDATE STATEMENT UPDATE table_name SET column1 = value1, column2 = value2....columnN=valueN [WHERE CONDITION];
  - SQL DELETE STATEMENT DELETE FROM table_name WHERE {CONDITION};
- DCL (Data Control Language)_Grant, Revoke
- DQL (Data Query Language)_Select
  - SQL SELECT STATEMENT SELECT column1, column2....columnN FROM table_name;
  - SQL DISTINCT CLAUSE SELECT DISTINCT column1, column2....columnN FROM table_name;
  - SQL WHERE CLAUSE SELECT column1, column2....columnN FROMtable_name WHERE CONDITION;
  - SQL AND/OR CLAUSE SELECT column1, column2....columnN FROMtable_name WHERE CONDITION-1 {AND|OR} CONDITION-2;
  - SQL IN CLAUSE SELECT column1, column2....columnN FROMtable_name WHERE column_name IN (val-1, val-2,...val-N);
  - SQL BETWEEN CLAUSE SELECT column1, column2....columnN FROMtable_name WHERE column_name BETWEEN val-1 AND val-2;
  - SQL LIKE CLAUSE SELECT column1, column2....columnN FROMtable_name WHERE column_name LIKE { PATTERN };
  - SQL ORDER BY CLAUSE SELECT column1, column2....columnN FROMtable_name WHERE CONDITION ORDER BY column_name {ASC|DESC};
  - SQL GROUP BY CLAUSE SELECT SUM(column_name) FROMtable_name WHERE CONDITION GROUP BY column_name;
  - SQL COUNT CLAUSE SELECT COUNT(column_name)FROM table_name WHERE CONDITION;

- SQL HAVING CLAUSE SELECT SUM(column_name) FROMtable_name WHERE CONDITION GROUP BY column_name HAVING (arithematicfunction condition);
- TCL (Transaction Control Language)_Commit, Save Point, Rollback
- SQL COMMIT STATEMENT COMMIT;
- SQL ROLLBACK STATEMENT ROLLBACK;

(4) Write type of joins.

- SQL Joint are as following Type
- INNER JOIN: returns rows when there is a match in both tables.
- LEFT JOIN: returns all rows from the left table, even if there are no matches in the right table.
- RIGHT JOIN: returns all rows from the right table, even if there are no matches in the left table.
- FULL JOIN: returns rows when there is a match in one of the tables.

(5) What is join?

- In software testing, especially when dealing with databases, "join" refers to a method for combining data from multiple tables based on a related column.
- This allows testers to retrieve information that is distributed across different tables and verify the integrity of data relationships.
- ☑ Here's a more detailed explanation:
- **Relational Databases:**

Software often interacts with relational databases (like MySQL, PostgreSQL, SQL Server) where data is stored in organized tables.

- **Tables and Relationships:**

These tables are designed to represent different entities (e.g., users, products, orders) with related information stored in separate tables.

- **Joining Tables:**

A "join" operation combines data from two or more tables based on a common column (or key) that links the tables. This allows testers to:

- Verify data relationships: Ensure that related data in different tables is consistent and accurate.
- Test data integration: Confirm that data from different tables can be combined correctly to produce meaningful results.
- Create complex queries: Retrieve data from multiple sources to test specific scenarios or requirements.

(6) How Many constraint and describes itself.

- Constraints in SQL are a set of rules or restrictions defined on columns of a table in a relational database to control the data or data type that can be input or stored in a column. These rules or restrictions ensure the accuracy and reliability of the input data. After the constraint is defined, the particular operation is aborted if any operation in the database violates the specified rule.
- In SQL, constraints are broadly divided into two types:
- **Column Level Constraints: It refers to a single column in the table and does not specify the column's name except the CHECK constraints.**

- **Table Level Constraints:** It refers to one or more columns in the table and specify the column name in which they apply.

- **Syntax**
- In SQL, constraints are applied during the creation of the table using the CREATE TABLE command or after the creation of the table using the ALTER TABLE command.

☑ **Types of SQL Constraints:**

- There are mainly six different constraints that are used in SQL:

☑ **NOT NULL**

- Applied only on the column level
- By default, any column can have a NULL values
- It restricts the columns in a table from having NULL values

☑ **UNIQUE**

- This applies on both table and column level
- It ensures that the column has only unique values

☑ **DEFAULT**

- This applies on both table and column level
- It provides a default value for a column if no value is assigned

☑ **CHECK**

- This applies on both table and column level
- It is used to restrict the value of a column between a range
- It is similar to data validation in Excel

☑ **PRIMARY KEY**

- This applies on both table and column level
- The primary Key constraint is a combination of both UNIQUE and NOT NULL constraints.
- It helps to retrieve query results from the table.
- Only one PRIMARY KEY can be created per table

☑ **FOREIGN KEY**

- This applies on both table and column level
- It is used to relate two or more tables and prevents the operation that destroys the link between the tables.
- A foreign key can be a primary key if the table is connected by a one-to-one relationship, not a one-to-many relationship.
- Multiple foreign keys can be created per table

(7) 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 doesn't 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, Window Registry, Forxpro, dbaseIIIplus etc.	• Examples: MySQL, PostgreSQL, SQL Server, Oracle, Microsoft Access etc.

(8) What is an SQL alias?

☑ SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name.

Aliases are often used to make column names more readable.

An alias only exists for the duration of that query.

An alias is created with the AS keyword.

Example

```
SELECT CustomerID AS ID
FROM Customers;
```

☑ AS is Optional

Actually, in most database languages, you can skip the AS keyword and get the same result:

Example

```
SELECT CustomerID ID
FROM Customers;
```

☑ Syntax

When alias is used on column:

```
SELECT column_name AS alias_name
FROM table_name;
```

When alias is used on table:

```
SELECT column_name(s)
FROM table_name AS alias_name;
```

☑ Demo Database

Below is a selection from the **Customers** and **Orders** tables used in the examples:

☑ Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

☑ Orders

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	7/4/1996	3
10249	81	6	7/5/1996	1
10250	34	4	7/8/1996	2

☑ Alias for Columns

The following SQL statement creates two aliases, one for the CustomerID column and one for the CustomerName column:

Example

```
SELECT CustomerID AS ID, CustomerName AS Customer
FROM Customers;
```

☑ Using Aliases With a Space Character

If you want your alias to contain one or more spaces, like "My Great Products", surround your alias with square brackets or double quotes.

Example

Using [square brackets] for aliases with space characters:

```
SELECT ProductName AS [My Great Products]
FROM Products;
```

Example

Using "double quotes" for aliases with space characters:

```
SELECT ProductName AS "My Great Products"
FROM Products;
```

Note: Some database systems allows both [] and "", and some only allows one of them.

☑ Concatenate Columns

The following SQL statement creates an alias named "Address" that combine four columns (Address, PostalCode, City and Country):

Example

SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address
FROM Customers;

Note: To get the SQL statement above to work in MySQL use the following

☑ MySQL Example

SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address
FROM Customers;

Note: To get the SQL statement above to work in Oracle use the following:

☑ Oracle Example

SELECT CustomerName, (Address || ', ' || PostalCode || ' ' || City || ', ' || Country) AS Address
FROM Customers;

☑ Alias for Tables

The same rules applies when you want to use an alias for a table.

Example

Refer to the Customers table as Persons instead:

SELECT * FROM Customers AS Persons;

It might seem useless to use aliases on tables, but when you are using more than one table in your queries, it can make the SQL statements shorter.

The following SQL statement selects all the orders from the customer with CustomerID=4 (Around the Horn). We use the "Customers" and "Orders" tables, and give them the table aliases of "c" and "o" respectively (Here we use aliases to make the SQL shorter):

Example

SELECT o.OrderID, o.OrderDate, c.CustomerName
FROM Customers AS c, Orders AS o
WHERE c.CustomerName='Around the Horn' AND c.CustomerID=o.CustomerID;

The following SQL statement is the same as above, but without aliases:

Example

SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName
FROM Customers, Orders

WHERE Customers.CustomerName='Around the
Horn' AND Customers.CustomerID=Orders.CustomerID;

☑ Aliases can be useful when:

- There are more than one table involved in a query
- Functions are used in the query
- Column names are big or not very readable
- Two or more columns are combined together

(9) Write a query to create the table in Structured Query Language.

- Query 1
  Create database_name;
  Use users;

  Create table students(
  Id int(11) Primary key auto_increment,
  Name varchar(255),
  Age bigint(03),
  Contact bigint(11),
  Address varchar(255),
  Email varchar(255),
  Password varchar(255),Registration_date (date);
- Student
  SELECT * from users_students;

  Insert into students(name.contact,address,email,password,registration_date);
  Value('Jay',028,8469416729,'Ahmedabad','Jaydalal@gmail.com','Jay@123',2025-05-08);

☑ Output of SQL Coding are as following

Name	Age	Contact	Address	Email	Password	Registration date

(10) Write a query to insert data into table.

- As per following Que (9) for creating table
  Then,
- INSERT DATA
  SELECT * from users_students;

  Insert into students(name.contact,address,email,password,registration_date);
  Value('Jay',028,8469416729,'Ahmedabad','Jaydalal@gmail.com','Jayd@123',2025-05-08);

☑ Output of SQL Coding are as following

Name	Ag e	Contact	Address	Email	Password	Registration date
Jay	028	8469416729	Ahmedabad	Jaydalal@gmail.co m	Jayd@123	2025-05-08

(11) Write a query to update data into table with validations.

- As per following Que (9) for creating table and following Que (10) for inserting data Then,

☑ UPDATE DATA
Update students set address='delhi',password='jay@123',where id=1;

☑ Output of SQL Coding are as following

Name	Age	Contact	Address	Email	Password	Registration date
Jay	028	8469416729	delhi	Jaydalal@gmail.com	Jay@123	2025-05-08

(12) Write a query to delete data from table with validations.

- As per following Que (9) for creating table and following Que (10) for inserting data Then,

☑ DELETE DATA
Delete from student where id=1;

☑ Output of SQL Coding are as following

Name	Age	Contact	Address	Email	Password	Registration date

(13) Write a query to insert new column in existing table.

- As per following Que (9) for creating table and following Que (10) for inserting data Then,

☑ fetching Data
select * from student;

☑ Output of SQL Coding are as following

Name	Age	Contact	Address	Email	Password	Registrati on Date
Jay	028	8469416729	Delhi	Jaydalal@gmail.com	Jay@123	2025-05-08

(14) Write a query to drop table and database.

- As per following Que (9) for creating table and following Que (10) for inserting data Then,

☑ Drop Table

```
DROP TABLE categories;
```

☑ Output of SQL Coding are as following

07:15:25 PM	Started executing query at line 1 Command Completed Successfully. Total execution time: 00:00:00.025

(15) Write a query to find max and min value from table.

- As per following Que (9) for creating table and following Que (10) for inserting data Then,

☑ Find Max & Min Value

```
SELECTMAX(Age)
        FROM Table;
```

☑ Output of SQL Coding are as following

Max(Age)
028

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

☑ Create Tables: Seller and Product

```
SQL
-- Create the Seller table
CREATE TABLE Seller (
    seller_id INT PRIMARY KEY,
    seller_name VARCHAR(100) NOT NULL,
    seller_location VARCHAR(100)
);

-- Create the Product table with a foreign key
CREATE TABLE Product (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    seller_id INT,
    -- Define the foreign key constraint
    FOREIGN KEY (seller_id) REFERENCES Seller(seller_id)
);
```

☑ FOREIGN KEY constraint

- Insert Sample Data into Tables

```
INSERT INTO Seller (seller_id, seller_name, seller_location) VALUES
(101, 'Electronics Hub', 'New York'),
 (102, 'Fashionista Store', 'Los Angeles'),
(103, 'Bookworm Corner', 'Chicago'),

(104, 'Home Essentials', 'Houston');
```

Comment [a1]: -- Insert data into the Seller table

```
INSERT INTO Product (product_id, product_name, price, seller_id) VALUES
(1, 'Laptop', 1200.00, 101),
(2, 'Smartphone', 800.00, 101),
(3, 'T-Shirt', 25.00, 102),
(4, 'Jeans', 60.00, 102),
(5, 'Novel', 15.00, 103),
(6, 'Cookbook', 30.00, 103),
(7, 'Coffee Maker', 75.00, 104),
(8, 'Bluetooth Speaker', 90.00, 101),
(9, 'Desk Lamp', 40.00, NULL);
```

Comment [a2]: -- Insert data into the Product table

☑ Fetch Data Using Different Joins

- ➢ INNER JOIN:

```
SELECT P.product_name,
P.price,
 S.seller_name,
S.seller_location
FROM
 Product AS P
INNER JOIN
Seller AS S ON P.seller_id = S.seller_id;
```

- OUTPUT

  All products that have an associated seller, along with their seller's details. The 'Desk Lamp' (product_id 9) will not appear as it has NULL for seller_id.

- ➢ LEFT JOIN (or LEFT OUTER JOIN):

```
SELECT P.product_name,
P.price,
S.seller_name,
S.seller_location
FROM
 Product AS P
LEFT JOIN
 Seller AS S ON P.seller_id = S.seller_id;
```

- OUTPUT

  All products will be listed. For products without a matching seller (like 'Desk Lamp'), the seller_name and seller_location columns will show NULL.

➢ RIGHT JOIN (or RIGHT OUTER JOIN):

```
SELECT
   P.product_name,
   P.price,
   S.seller_name,
   S.seller_location
FROM
   Product AS P
RIGHT JOIN
   Seller AS S ON P.seller_id = S.seller_id;
```

▪ OUTPUT

All sellers will be listed. For sellers who don't have any products associated with them (e.g., if we had a seller 105 with no products yet), the product_name and price columns would show NULL.

➢ FULL OUTER JOIN (Note: Not supported by MySQL directly, but supported by PostgreSQL, SQL Server, Oracle):

```
  SELECT P.product_name,
   P.price,
  S.seller_name,
   S.seller_location
  FROM
   Product AS P FULL OUTER JOIN Seller AS S ON P.seller_id = S.seller_id;
```

▪ OUTPUT

This would show all products (even those without sellers) and all sellers (even those without products), with NULLs where there's no corresponding match in the other table.

➢ CROSS JOIN:

```
SELECT P.product_name,
S.seller_name
FROM
 Product AS P CROSS JOIN Seller AS S;
```

▪ OUTPUT

If you have 9 products and 4 sellers, this will result in 9 * 4 = 36 rows, showing every product paired with every seller, regardless of their actual relationship.

(17) What is API Testing?

- Application Programming Interface (API) is a software interface that allows two applications to interact with each other without any user intervention
- another definition , API (Application Programming Interface) is a computing interface which enables communication and data exchange between two separate software systems.
- 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.

(18) Types of API Testing

- There are mainly 3 types of API Testing, which are describe in following:

(i) Open APIs: These types of APIs are publicly available to use like OAuth APIs from Google. It has also not given any restriction to use them. So, they are also known as Public APIs.

(ii) Partner APIs: Specific rights or licenses to access this type of API because they are not available to the public.

(iii) Internal APIs: Internal or private. These APIs are developed by companies to use in their internal systems. It helps you to enhance the productivity of your teams.

(19) What is Responsive Testing?

- A responsive web design involves creating a flexible web page that is accessible from any device, starting from a mobile phone to a tablet.
- Furthermore, a responsive web design improves users' browsing experience.
- Considering this from a quality assurance perspective, a responsive web design requires thorough evaluation using a variety of devices before it is ready to go live.
- Software testers may find it challenging to perform responsive design testing as a variety of factors are to be looked into during the testing phase.
- Some points to be understand for Responsive Testing.
  - The challenges involved in testing a responsive website
  - How website testing differs from a mobile device to a computer
  - Rules and guidelines to be followed during responsive design testing and
  - Lastly, various tools available to perform responsive testing

(20) Which types of tools are available for Responsive Testing.

- tools are available for Responsive Testing are following

Browser Developer Tools:

- **Google Chrome DevTools:**

  This built-in tool allows developers to simulate different devices and screen sizes, inspect elements, and debug responsive designs in real-time.
- **Mozilla Firefox Web Developer Toolbar:**
  Firefox offers a similar "Responsive Design View" for testing website responsiveness.
  Online Responsive Testing Tools:
- **Responsive Design Checker:**

  This tool checks websites for responsiveness across various screen sizes and provides screenshots for mockups.
- **Screenfly:**

This free tool lets you preview websites on different device sizes and resolutions.

- **Responsinator:**

  This tool simulates different device sizes and orientations to help test responsive designs.

- **BrowserStack:**

  BrowserStack offers real-device testing, allowing users to test websites on a wide range of devices and browsers.

- **LambdaTest:**

  LambdaTest is another platform that provides real-device testing, screenshot testing, and responsive testing capabilities.

- **Viewport Resizer:**

  This bookmarklet or browser extension allows users to resize their browser to different screen sizes and viewports.

- **Polypane:**

  Polypane allows for visual and interactive testing of responsive designs, with the ability to modify CSS and HTML directly in the browser.

- **Ghostlab:**

  Ghostlab is a platform that enables remote collaborative testing of web apps and websites.

- **Am I Responsive:**

  This tool checks if a website is responsive by simply entering the URL.

- **Blisk:**
  Blisk is a browser extension that allows you to open multiple browser windows and test responsiveness across different devices.

(21) What is the full form of .ipa, .apk

- The full form of .ipa is iOS App Store Package, and the full form of .apk is Android Package Kit. .IPA files are used to distribute and install applications on iOS devices, while .APK files are used for the same purpose on Android devices.

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

- To enable developer options on an Android device, navigate to Settings, then About phone or About device. Locate the Build number and tap it seven times. After tapping, a message will indicate you've unlocked developer options, and you can access them in your settings menu.

☑ **Detailed Steps:**

1. **Open Settings:** Navigate to the settings menu on your Android device.
2. **Go to About Phone/Device:** Scroll down and select "About phone" or "About device".
3. **Find Build Number:** Scroll down until you find the "Build number" entry.
4. **Tap Seven Times:** Rapidly tap the "Build number" entry seven times.
5. **Enable Developer Mode:** After tapping, you should see a message confirming that developer options are now unlocked.

6. **Access Developer Options:** Return to the main settings menu. Scroll down to find the "Developer options" or "Developer mode" option, depending on your device.