1) What is RDBMS?

Relational Database Management System

It is a type of database management system that stores and manages data in a structured format using tables.

Examples of RDBMS:

- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle Database
- SQLite

2) What is SQL?

Structured Query Language is a proggraming language design for managing relational databases.

Uses of SQL

Retrieve data for using select query

Add data for insert new records

Update data for modify existing records

Delete data for remove records

SQL is essential for anyone working with data, databases, or applications that store and retrieve information.

4) what is join?

A join is an operation used to combine data from two or more tables based on a related column.

Common types of joins in SQL:

INNER JOIN: Returns rows with matching values in both tables.

LEFT JOIN : Returns all rows from the left table, and matching rows from the right table. If no match is found, NULLs are returned for the right table's columns.

RIGHT JOIN : Returns all rows from the right table, and matching rows from the left table.

FULL JOIN : Returns rows when there is a match in either table, filling in NULLs where no match exists.

5) Difference between RDBMS vs DBMS

The main differences between a Database Management System (DBMS) and a Relational Database Management System (RDBMS) are related to the way data is organized, managed, and queried.

DBMS is a broader category of systems that manage databases without strict adherence to a relational model.

RDBMS is a specialized type of DBMS that organizes data into relations (tables) and emphasizes relationships and data integrity.

6) Types of API testing

Functional testing:

Verifies the response to request and uses Get, Put, Post and Delete

Load testing:

Make sure that how API handle high traffic without crashing down

Validation testing:

Verifies that the input/output formats, data types, and error handling are correct.

Stress testing:

Determines the API’s behavior under extreme conditions like massive spikes in traffic or high user loads.

Security testing:

Ensures the API is secure from vulnerabilities.

Usability testing:

Ensure that it is useful for a non IT person and user friendly requirement.

Regression testing:

Ensures that new changes or updates to the API do not negatively affect existing functionality.

End to end testing:

Tests the API in the context of the entire application.