

A. Database Basic & SQL Overview

1. What is a database? What are different types of databases?

A database is an organized collection of data stored and accessed electronically.

- Relational → Hierarchical → Object-Oriented.
- Non-relational → Network

2. What is a relational database? How it is different from non-relational databases?

→ Relational database stores data in tables with rows and columns and uses keys for relationships.

- * Relational - Structured schema, SQL, ACID compliance
- * Non Relational - flexible schema, JSON like etc.,, high scalability

3. What is SQL? What are its major components?

SQL is used to interact with relational databases.

- * DDL, DML, DCL, TCL are major components

4. What are different types of SQL commands? Explain each briefly.

- DDL (Data definition language) : CREATE, ALTER, DROP
- DML (Data manipulation language) : SELECT, INSERT, UPDATE, DELETE
- DCL (Data control language) : GRANT, REVOKE
- TCL (Transaction control language) : COMMIT, ROLLBACK, SAVEPOINT

5. What is the role of SQL in a DBMS?

SQL is the standard language to create, manage, manipulate and secure data in DBMS.

6. How do you install and setup MySQL on a system?

- Download MySQL installer from mysql.com
- Run installer → choose server type → Configure root user → Start server
- use mysql -u root -p to connect

7. What is the syntax of creating a database and table in SQL?

CREATE DATABASE School;

USE School;

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR (50),

Age INT

);

8. How do you insert a single and multiple records into a table?

INSERT INTO Students (StudentID, Name, Age)

VALUES (1, 'Alice', 20); → Single

(2, 'Bob', 22); → Multiple.

9. What are the key differences between DDL, DML, DCL and TCL commands?

→ DDL - Defines structure

→ DML - Manipulates data

→ DCL - Controls access

→ TCL - Manages transactions.

10. What are some common SQL data types?

INT, VARCHAR, CHAR, DATE, DECIMAL, FLOAT, BOOLEAN, TEXT

B. Data Retrieval and Filtering

11. What is the purpose of the SELECT statement in SQL?

To retrieve data from one or more tables.

12. How can you retrieve only specific columns from a table?

SELECT Name, Age FROM Students;

13. What does the DISTINCT keyword do in a SQL query?

SELECT DISTINCT Department FROM Employees;

14. How would you filter records using the WHERE clause?

SELECT * FROM Employees WHERE Salary > 50000;

15. What are the comparison operators in SQL? Give examples.

=, <, >, <, >, =, <=

e.g., WHERE Age >= 18.

16. What are logical operators (AND, OR, NOT) and how are they used in SQL?

SELECT * FROM Employees

WHERE Department = 'HR' AND Salary > 50000;

17. How does the LIKE operator work? What do % and _ mean?

- → pattern matching for any number of characters.
- → single character.

WHERE Name LIKE 'A%' → starts with A.

18. How do BETWEEN and IN work for filtering values?

WHERE Age BETWEEN 20 AND 30

WHERE Department IN ('HR', 'Finance')

19. What is the difference between ISNULL and = NULL?

→ ISNULL → checks for NULL's

WHERE Manager ISNULL

→ = NULL → doesn't work (returns False)

20. How do you sort query results using ORDER BY? How do you sort by multiple columns?

ORDER BY Salary DESC, Name ASC;

c. Aggregation, grouping and filtering groups.

21. What are aggregate functions in SQL? List and explain 5 with ex.

→ COUNT() → Number of rows

→ SUM() → Sum of values

→ AVG() → Average value

→ MAX() → maximum

→ MIN() → minimum

e.g.: SELECT AVG(Salary) FROM Employees;

eg: SELECT COUNT(*) FROM Employees;

22. How does the COUNT() function behave with NULL values?

COUNT(*) counts all rows.

COUNT(column) ignores NULL's.

23. What is the difference between SUM() and COUNT()?

SUM() → adds numeric values.

COUNT() → counts rows

24. How do you group data in SQL using the GROUP BY clause?

SELECT Department, AVG(Salary)

FROM Employees

GROUP BY Department;

25. What is the purpose of HAVING clause? How is it different from WHERE?

HAVING filters groups after aggregation

WHERE filters rows before grouping

26. Can you use aggregate functions in WHERE clause? Why or why not?

No, aggregate functions require GROUP BY; use HAVING instead.

27. Write a query to find departments with more than 5 employees using GROUP BY and HAVING.

SELECT Department, COUNT(*)

FROM Employees

GROUP BY Department

HAVING COUNT(*) > 5;

28. How do you find the maximum and minimum salary from employee table?
 SELECT MAX(Salary), MIN(Salary) FROM Employees;
29. Write a query to calculate the average salary by department.
 SELECT Department, AVG(Salary)
 FROM Employees
 GROUP BY Department;
30. What happens if you use GROUP BY without any aggregate functions?
 It groups rows but only unique combinations are returned.
- #### D. Joins and Relationships.
31. What are joins in SQL? Why are they used?
 To combine rows from two or more tables based on related columns.
32. What is the difference between INNER JOIN and LEFT JOIN?
 → INNER JOIN → matching rows only.
 → LEFT JOIN → All rows from the left table + matching right.
33. Explain RIGHT JOIN and give a use case.
 → RIGHT JOIN → All rows from the right table + matching left.
 use find unmatched rows in the left table.
34. What is a FULL OUTER JOIN? Is it supported in MySQL?
 Returns all rows from both tables, with NULL's where no match. NOT directly supported in MySQL. (can simulate using UNION)
35. What is a SELF JOIN? How and why would you use it?
 A join where a table is joined with itself.
 ex: Find employees with the same manager.

36. write a query using a self join to find employees who report to the same manager.

```
SELECT e1.Name, e2.Name  
FROM Employees e1  
JOIN Employees e2 ON e1.ManagerID = e2.ManagerID  
WHERE e1.EmployeeID <> e2.EmployeeID;
```

37. What is the purpose of using aliases in joins especially in self joins?

Aliases make queries readable and distinguish between table instances.

38. What is the difference between an equi join and non-equi join?

→ Equi join → uses = for matching.

→ Non-equijoin → uses operators like <, > (e.g. range joins)

E. Subqueries, UNIONS and Advance queries.

39. What is a subquery? what are the types of subqueries in SQL?

→ A query inside another query.

Type → single-row → correlated

 → multiple-row → non-correlated.

40. How is a correlated subquery different from non-correlated subquery?

→ correlated → References outer query for each row

→ non-correlated → Executed independently.

41. Give an example of a subquery in a SELECT clause.

```
SELECT Name,  
(SELECT DepartmentName FROM Departments WHERE  
Departments.ID = Employees.DeptID)
```

AS Department

FROM Employees;

42. write a query to find employees who earn more than the average salary using a subquery.

SELECT * FROM Employees;

WHERE salary > (SELECT AVG(Salary) FROM Employees);

43. what is the difference between UNION and UNION ALL?

→ UNION → Removes duplicates.

→ UNION ALL → Keeps duplicates.

44. What is the INTERSECT operator used for? It is available in MySQL?

→ Returns rows common to both queries.

→ Not available in MySQL, simulate with JOIN.

45. How can you combine JOIN, GROUP BY and HAVING in a single query?

SELECT d.Department, COUNT(e.EmployeeID)

FROM Departments d,

JOIN Employees e ON d.ID = e.DeptID

GROUP BY d.Department

HAVING COUNT(e.EmployeeID) > 5;

46. CASE Statement, Date and conditional logic.

What is a CASE WHEN statement in SQL? provide an example.

46. What is a CASE WHEN statement in SQL? provide an example.

→ Conditional logic in SELECT

SELECT Name,

CASE

WHEN salary > 50000 THEN 'High'

ELSE 'Low'

END AS Salary Grade

FROM Employees;

47. How would you retrieve all employees who joined in last 6 months using date functions?

SELECT * FROM Employees

WHERE JoinDate >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH);

48. Data modification and constraint

→ Explain the purpose and syntax of INSERT, UPDATE, DELETE.

→ INSERT - Add data

INSERT INTO Students VALUES ('');

→ UPDATE - modify data.

UPDATE Students SET Age = 21 WHERE ID = 1;

→ DELETE - Remove data.

DELETE FROM Students WHERE ID = 1;

49. What are the diff types of constraints? How they ensure integrity

→ PRIMARY KEY - Uniquely identifies rows

→ FOREIGN KEY - Enforces referential integrity

→ UNIQUE - Ensures unique values

→ NOT NULL - No NULL's allowed

→ CHECK - Validates values.

50. Transactions and ACID properties

50. What are transactions in SQL. Explain COMMIT, ROLLBACK, SAVEPOINT and the ACID properties.

A transaction is a unit of work performed as a single logical operation

→ COMMIT - Save changes

→ ROLLBACK - Undoes changes

→ SAVEPOINT - Sets a point to roll back to
ACID

→ Atomicity - All or nothing

→ Consistency - maintains valid state

→ Isolation - concurrent transactions don't interfere

→ Durability - changes persist after commit.