PG-DAC AUGUST 25
Assignment No-1
Database & SQL Interview Questions

Answers

Explore & Read about MYSQL & RDBMS .

1. What is a database, tuple & Records?

A database is an organized collection of data stored electronically. A tuple is a single row in a table, representing one record. A record is another name for a tuple, containing related data fields.

2. What is the difference between a database and a DBMS?

A database stores the data, while a DBMS (Database Management System) is the software that manages, retrieves, and manipulates that data.

3. What are the different types of database models?

Common database models include the hierarchical model, network model, relational model, object-oriented model, and document-based model (used in NoSQL).

4. What is normalization? Why is it important?

Normalization is the process of organizing data to reduce redundancy and improve integrity. It ensures efficient storage and consistent data.

5. Explain the different normal forms (1NF, 2NF, 3NF, BCNF).

1NF: No repeating groups; each field holds atomic values.
2NF: 1NF + no partial dependency on a composite key.
3NF: 2NF + no transitive dependencies.
BCNF: Stricter 3NF; every determinant is a candidate key.

6. What is denormalization and when should we use it?

Denormalization combines normalized tables to improve read performance. It's used in reporting or OLAP systems where speed is more critical than storage efficiency.

7. What is a primary key and a foreign key?

A primary key uniquely identifies each record. A foreign key links one table's column to another table's primary key, enforcing relationships.

8. What are constraints in SQL?

Constraints are rules that enforce data integrity (e.g., PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT).

9. What is an index? How does it improve performance?

An index speeds up query performance by allowing the database to find rows quickly without scanning the entire table.

10. What is a view in SQL?

A view is a virtual table based on a query. It doesn't store data but displays results dynamically from underlying tables.

11. What are DDL, DML, DCL, and TCL in SQL?
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).

12. Write a SQL query to create a table.
CREATE TABLE Employees (EmpID INT PRIMARY KEY, Name VARCHAR(50), Salary DECIMAL(10,2));

13. How do you insert data into a table?
INSERT INTO Employees (EmpID, Name, Salary) VALUES (1, 'John', 50000);

14. How do you update data in a table?
UPDATE Employees SET Salary = 60000 WHERE EmpID = 1;

15. How do you delete data from a table?
DELETE FROM Employees WHERE EmpID = 1;

16. What is the difference between DELETE and TRUNCATE & DROP?
DELETE removes specific rows. TRUNCATE removes all rows but keeps structure. DROP deletes the table structure entirely.

17. Explain the difference between WHERE and HAVING.
WHERE filters rows before aggregation; HAVING filters after aggregation (used with GROUP BY).

18. How do you select distinct records from a table?
SELECT DISTINCT column_name FROM table_name;

19. Write a query to find the maximum salary from an employee table.
SELECT MAX(Salary) FROM Employees;

20. What is the Types of Joins & difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?
INNER JOIN: Returns matching records.
LEFT JOIN: All from left + matching right.
RIGHT JOIN: All from right + matching left.
FULL OUTER JOIN: All records from both.

21. What is a subquery?
A subquery is a query inside another query, used to fetch intermediate results.

22. Explain correlated vs. non-correlated subqueries.
Non-correlated subquery runs independently; correlated depends on the outer query for each row.

23. How does the GROUP BY clause work?
GROUP BY groups rows with the same values into summary rows, often used with aggregate functions (SUM, AVG).

24. How can you prevent duplicate records from being inserted into a table?
Use a UNIQUE constraint or check before insertion using WHERE NOT EXISTS.

25. What is a transaction in SQL?
A transaction is a group of SQL operations that execute as a single unit of work.

26. Explain ACID properties in the context of a transaction.
ACID: Atomicity, Consistency, Isolation, Durability — ensures reliable transactions.

27. What is the difference between UNION and UNION ALL?
UNION removes duplicates; UNION ALL keeps them.

28. How do you perform a self-join? Provide an example use case.
A self-join joins a table with itself. Example: finding employees and their managers in the same table.

29. What is the use of the CASE statement in SQL?
CASE allows conditional logic in SQL queries, similar to if-else.

30. How do you handle NULL values in queries?
Handle NULL using IS NULL, IS NOT NULL, or functions like COALESCE() and IFNULL().

31. What is a stored procedure?
A stored procedure is a precompiled set of SQL statements stored in the database for reuse.

32. What is a trigger? Give an example use case.
A trigger automatically executes in response to events like INSERT, UPDATE, or DELETE (e.g., logging changes).

33. What is indexing? What are the different types of indexes?
Indexing improves search speed. Types: clustered, non-clustered, unique, composite, and full-text indexes.

34. What is the difference between clustered and non-clustered index?
Clustered index defines physical data order; non-clustered creates a separate structure pointing to data.

35. Explain the concept of views.
A view presents query results as a virtual table, simplifying complex queries and enhancing security.

36. What is a materialized view? How is it different from a regular view?
A materialized view stores query results physically; a regular view does not.

37. What is the difference between OLAP and OLTP databases?
OLTP (Online Transaction Processing) handles real-time transactions. OLAP (Online Analytical Processing) is for analysis and reporting.

38. How do you optimize a slow-running SQL query?
Optimize queries by indexing, avoiding SELECT *, using joins efficiently, and analyzing execution plans.

39. What is a deadlock in database systems? How do you prevent it?
A deadlock occurs when two transactions wait on each other's locks. Prevent it using consistent locking order and shorter transactions.

40. What are foreign key constraints, and how do they help maintain referential integrity?
Foreign key constraints ensure that related records exist, maintaining referential integrity between tables.

41. How would you design a database for an online bookstore?
For an online bookstore, design tables for Books, Authors, Customers, Orders, and OrderDetails with foreign keys for relationships.

42. How do you enforce uniqueness in a column?
Enforce uniqueness with a UNIQUE constraint or index.

43. What are the implications of using a composite primary key?
A composite primary key uses multiple columns; it ensures uniqueness across combined values but increases join complexity.

44. Explain how you would back up and restore a database.
Use BACKUP DATABASE and RESTORE DATABASE commands (or export/import tools).

45. How do you handle concurrent database access?
Handle concurrency with transactions, locking, and isolation levels.

46. What is a database trigger, and when would you use one?
A trigger runs automatically on data changes — used for auditing or enforcing rules.

47. How do you manage user permissions in a database?
Manage permissions with GRANT and REVOKE statements.

48. What is sharding in database systems?
Sharding splits a large database horizontally into smaller, faster pieces (shards) across servers.

49. What is the difference between horizontal scaling and vertical scaling in databases?
Horizontal scaling adds more machines; vertical scaling adds more resources (CPU/RAM) to one machine.

50. How would you approach migrating a database from MySQL to PostgreSQL?
To migrate from MySQL to PostgreSQL, export schema/data (using tools like pgloader), adjust data types, and test compatibility.