

PG-DAC AUGUST 25

Assignment No- 1

Database & SQL Interview Questions

Explore & Read about MYSQL & RDBMS .

1. What is a database ,tuple & Records ?
Database = An organized collection of data, usually stored and accessed electronically.
Tuple = A single row in a table (represents a data item).
Record = Another word for a tuple/row (a collection of related fields).
2. What is the difference between a database and a DBMS?
Database = The actual collection of data.
DBMS = Software used to create, manage, and interact with the database.
3. What are the different types of database models?
Hierarchical model
Network model
Relational model
cloud
4. What is normalization? Why is it important?
Normalization = Process of organizing data to reduce redundancy and improve integrity.
Importance = Avoids data anomalies, saves storage, ensures consistency.
5. Explain the different normal forms (1NF, 2NF, 3NF, BCNF).
1NF = Atomic values (no repeating groups).
2NF = 1NF + no partial dependency (non-key attributes depend on the whole primary key).
3NF = 2NF + no transitive dependency.
BCNF = Stronger version of 3NF; every determinant must be a candidate key.
6. What is denormalization and when should we use it?
Opposite of normalization It allows redundancy for performance.
Used in reporting/analytics where speed > storage efficiency.
7. What is a primary key and a foreign key?
Primary Key = Unique identifier for a record.
Foreign Key = Attribute in one table that references the primary key of another.
8. What are constraints in SQL?
NOT NULL = Column cannot have NULL.
UNIQUE = No duplicate values.
PRIMARY KEY = Unique + Not Null.
FOREIGN KEY = Referential integrity.
CHECK = Restricts values (e.g., salary > 0).
DEFAULT = Default value if not provided.

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

Data structure that speeds up queries.

Improves read performance but slows write operations (insert/update/delete).

10. What is a view in SQL?

Virtual table created from a query. Doesn't store data, just a saved query.

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

DDL = Data Definition Language (CREATE, ALTER, DROP, RENAME, TRUNCATE).

DML → Data Manipulation Language (INSERT, UPDATE, DELETE, SELECT).

DCL = Data Control Language (GRANT, REVOKE).

TCL = Transaction Control Language (COMMIT, ROLLBACK, SAVEPOINT).

12. Write a SQL query to create a table.

```
CREATE TABLE (Tablename){  
ID INT,  
Name VARCHAR(10)};
```

13. How do you insert data into a table?

```
INSERT INTO (Tablename) (ID, Name) VALUES (1, 'Madhuri');
```

14. How do you update data in a table?

```
UPDATE (Tablename) SET ID=1 WHERE NewID = 101;
```

15. How do you delete data from a table?

```
DELETE FROM (Tablename) WHERE NewID = 101;
```

16. What is the difference between DELETE and TRUNCATE & DROP ?

DELETE = Removes rows, can use WHERE. Logs each row deletion.

TRUNCATE = Removes all rows, faster, no WHERE.

DROP = Deletes the entire table structure.

17. Explain the difference between WHERE and HAVING.

WHERE =Filters rows before grouping.

HAVING =Filters groups after aggregation.

18. How do you select distinct records from a table?

```
SELECT DISTINCT ID FROM (Tablename);
```

19. Write a query to find the maximum salary from an employee table.

```
SELECT MAX (salary) FROM (Tablename);
```

20. What are the Types of Joins & difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?

INNER JOIN =Matching records only.

LEFT JOIN = All left + matching right.

RIGHT JOIN = All right + matching left.

FULL OUTER JOIN = All rows from both sides.

21. What is a subquery?

Query inside another query.

22. Explain correlated vs. non-correlated subqueries.

Independent of outer query and Depends on outer query (runs row by row).

23. How does the GROUP BY clause work?
Groups rows by column to apply aggregate functions like SUM, AVG, COUNT.
24. How can you prevent duplicate records from being inserted into a table?
By UNIQUE constraint or PRIMARY KEY.
25. What is a transaction in SQL?
A unit of work (e.g., money transfer).
26. Explain ACID properties in the context of a transaction.
Atomicity =All or nothing.
Consistency =Valid state after transaction.
Isolation =Transactions don't affect each other.
Durability =Changes persist after commit.
27. What is the difference between UNION and UNION ALL?
Removes duplicates and Keeps duplicates.
28. How do you perform a self-join? Provide an example use case.
SELECT A.EmpID, B.Name AS Manager
FROM Employee A
JOIN Employee B ON A.ManagerID = B.EmpID;
29. What is the use of the CASE statement in SQL?
SELECT Name,
CASE
WHEN Salary > 50000 THEN 'High'
ELSE 'Low'
END AS SalaryLevel
FROM Employee;
30. How do you handle NULL values in queries?
Use IS NULL / IS NOT NULL. COALESCE() or IFNULL() to replace NULL.
31. What is a stored procedure?
A saved block of SQL logic. Improves reusability & performance.
32. What is a trigger? Give an example use case.
Code that runs automatically on an event (INSERT, UPDATE, DELETE). Eg. Audit logs.
33. What is indexing? What are the different types of indexes?
creation of special lookup tables or data structures that significantly accelerate data retrieval operations within a database. Similar to an index in a book
34. What is the difference between clustered and non-clustered index?
Clustered Index = Sorts data physically (Rearrange).
Non-Clustered Index = Creates a separate structure pointing to data (not Rearrange).
35. Explain the concept of views.
Virtual tables created from queries. Don't store saves it.
36. What is a materialized view? How is it different from a regular view?
Stores query result physically. Faster but needs refresh.

37. What is the difference between OLAP and OLTP databases?
OLTP = Transactional, frequent updates (banking).
OLAP = Analytical, large queries (data warehouse).
38. How do you optimize a slow-running SQL query?
Use indexes, avoid SELECT *, optimize joins, partitioning, proper constraints.
39. What is a deadlock in database systems? How do you prevent it?
Two transactions waiting for each other's locks. And Prevent by proper locking order, timeout, smaller transactions.
40. What are foreign key constraints, and how do they help maintain referential integrity?
Maintain referential integrity (child must have valid parent key).
41. How would you design a database for an online bookstore?
High-level entities & relationships
Users — customers (and optionally staff/admin).
Books — product catalog.
Authors — authors (many-to-many with books).
Orders — one per checkout; contains multiple order items.
OrderItems — each book + quantity in an order (many orderItems → one Order).
Payments — payment records for orders (one-to-one or one-to-many if partial payments).
Inventory — tracks stock per book (could be part of Books for simple cases).
Categories/Tags (optional) — classification.
Reviews (optional) — user reviews/ratings.
Relationships:
Book = Author : many-to-many → BookAuthors join table.
User = Order : one-to-many.
Order = OrderItems : one-to-many.
Order = Payment : one-to-one or one-to-many (support refunds/part-pay).
Book = Inventory : one-to-one (or included in Books).
Table schemas (normalized)
Key choices: Use surrogate integer IDs (INT/BIGINT) as PKs for simplicity and performance.
Use appropriate constraints: NOT NULL, UNIQUE, CHECK where useful. Add indexes on columns frequently used in WHERE/join clauses (e.g., isbn, title, user_id, order_id). Keep monetary values in DECIMAL(precision, scale).
42. How do you enforce uniqueness in a column?
Use UNIQUE constraint or PRIMARY KEY.
43. What are the implications of using a composite primary key?
A primary key with multiple columns. And Ensures uniqueness across combined fields.
44. Explain how you would back up and restore a database.
Use BACKUP DATABASE (SQL Server) or mysqldump. Restore using RESTORE DATABASE or import file.
45. How do you handle concurrent database access?
Use locks, isolation levels, transactions.
46. What is a database trigger, and when would you use one?
Runs automatically on data change and used for logging, auditing, cascades.

47. How do you manage user permissions in a database?
Use GRANT, REVOKE, CREATE ROLE.
48. What is sharding in database systems?
Splitting large databases into smaller parts across servers for scalability.
49. What is the difference between horizontal scaling and vertical scaling in databases?
Horizontal = Add more servers (distributed).
Vertical = Add more resources (CPU/RAM) to one server.
50. How would you approach migrating a database from MySQL to PostgreSQL?
Export schema & data (mysqldump). Convert SQL syntax differences. Use tools like pgloader or manual ETL.