Interview Questions

1. What is SQL? How is it different from MySQL or PostgreSQL?
SQL (Structured Query Language) is a language for managing databases.
MySQL and PostgreSQL are database systems that use SQL.

2. What are the different types of SQL statements?
DDL (CREATE, ALTER), DML (INSERT, UPDATE), DCL (GRANT, REVOKE), TCL (COMMIT, ROLLBACK).

3. Difference between WHERE and HAVING?
WHERE filters rows before grouping.
HAVING filters groups after GROUP BY.

4. Explain PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK.

- PRIMARY KEY: Unique ID for rows.
- FOREIGN KEY: Links two tables.
- UNIQUE: No duplicate values.
- CHECK: Restricts column values.

5. Difference between DELETE, TRUNCATE, DROP?

- DELETE: Removes rows, keeps structure.
- TRUNCATE: Removes all rows, resets auto-increment.
- DROP: Deletes table completely.

6. What is normalization? Explain normal forms.
Breaking data into tables to reduce redundancy.
1NF: No repeating groups.
2NF: No partial dependency.
3NF: No transitive dependency.

7. What is denormalization?
Combining tables for faster reads, increases redundancy.

8. Difference between CHAR and VARCHAR?
CHAR: Fixed size, faster.
VARCHAR: Variable size, saves space.

9. What are ACID properties?
Atomicity, Consistency, Isolation, Durability – ensures reliable transactions.

10. Difference between INNER, LEFT, RIGHT, FULL JOIN?

- INNER: Common rows only.
- LEFT: All left + matching right.
- RIGHT: All right + matching left.
- FULL: All rows from both tables.

11. Second highest salary query?

```
select max(salary) from employee where salary < (select max(salary) from employee);
```

12. Department-wise average salary?

```
select department, avg(salary) from employee group by department;
```

13. Retrieve duplicate records?

```
select column_name, count(*) from table group by column_name having count(*) > 1;
```

14. Update column with calculation?

```
update products set price = price * 1.1;
```

15. Delete only duplicate rows?

```
delete from table where id not in (select min(id) from table group by column);
```

16. Customers with more than 5 orders?

```
select customer_id from orders group by customer_id having count(*) > 5;
```

17. Join three tables?

```
select * from a join b on a.id=b.a_id join c on b.id=c.b_id;
```

18. What is a subquery? Difference from JOIN?
A query inside another query.
JOIN combines tables directly; subquery runs separately.

19. What is a correlated subquery?
A subquery that depends on outer query.

Example: select name from employee e where salary > (select avg(salary) from employee where department = e.department);

20. Filter data by date range?

```
select * from orders where order_date between '2024-01-01' and '2024-06-30';
```

21. What are window functions?
Functions like ROW_NUMBER(), RANK() applied over a set of rows.

22. Use of RANK(), DENSE_RANK(), ROW_NUMBER()?

- RANK(): skips ranks on ties.
- DENSE_RANK(): no gaps.
- ROW_NUMBER(): unique row numbers.

23. What is a CTE? Difference from subquery?
Temporary result set using WITH. Easier to read than nested subqueries.

24. What are stored procedures?
Reusable SQL blocks with logic. Used for modular coding.

25. What is a trigger? Example?
Auto-executed SQL on events. Eg: Update seat availability after booking.

26. What is a view? Pros & cons?
Saved SQL query.
Pros: Simplifies access. Cons: Slower on large data.

27. What are indexes?
Speeds up searches. Acts like a book index.

28. What is a materialized view?
Stored view with data. Faster than normal view.

29. Transactions? COMMIT, ROLLBACK?
Group of SQL steps.

- COMMIT: Save changes.
- ROLLBACK: Undo changes.

30. Aggregate functions? Examples?
SUM(), AVG(), COUNT(), MAX(), MIN().

31. Optimize slow queries?
Use indexes, avoid SELECT *, rewrite joins.

32. What is EXPLAIN used for?
Shows query execution plan.

33. Indexing effect on DML?
Speeds SELECT, slightly slows INSERT/UPDATE.

34. What is composite index?
Index on multiple columns. Used when queries filter by those columns.

35. What is normalization overhead?
Too many joins slow queries. Solution: selective denormalization.

36. Avoid Cartesian product?
Always use proper JOIN conditions.

37. What is partitioning?
Splitting tables into parts for better performance.

38. What causes deadlock? Prevention?
Two transactions waiting on each other. Avoid by consistent lock order.

39. Clustered vs non-clustered index?
Clustered: rearranges table rows.
Non-clustered: separate index structure.

40. Tools to monitor SQL performance?
EXPLAIN, MySQL Workbench, pgAdmin, slow query logs.

41. Student-course grading system design?
Tables: Students, Courses, Enrollments, Grades.

42. Attendance tracking?
Table: Attendance (employee_id, date, status).

43. Library system overdue books?
Track due_date, fine = days_late * rate.

44. Missing records fix?
Check logs, restore from backups.

45. Role-based access?
Use GRANT, REVOKE for user roles.

46. Load & clean CSV?
LOAD DATA INFILE, then update NULLs and remove duplicates.

47. Monthly retention?
Count users who logged in again after one month.

48. Secure sensitive data?
Encryption, access control, strong passwords.

49. Daily backup plan?
Use mysqldump, automate with cron jobs.

50. Best practices for SQL in production?

- Use parameterized queries
- Index wisely
- Avoid SELECT *
- Backup regularly