SQL

Basic SQL Interview Questions

1. What is SQL?
SQL (Structured Query Language) is used to interact with relational databases, allowing users to insert, query, update, and delete data.

2. What are the different types of SQL commands? (DDL, DML, DCL, TCL, DQL)

- **DDL**: CREATE, ALTER, DROP
- **DML**: INSERT, UPDATE, DELETE
- **DCL**: GRANT, REVOKE
- **TCL**: COMMIT, ROLLBACK, SAVEPOINT
- **DQL**: SELECT

3. What is the difference between `DELETE`, `TRUNCATE`, and `DROP`?

- **DELETE**: Removes rows, can use WHERE, can be rolled back.
- **TRUNCATE**: Removes all rows, faster, no WHERE, minimal logging.
- **DROP**: Removes the entire table structure and data.

4. What are Primary Key, Foreign Key, Unique Key, and Composite Key?

- **Primary Key**: Uniquely identifies a row, no NULLs.
- **Foreign Key**: Enforces relationship between tables.
- **Unique Key**: Ensures uniqueness, allows one NULL.
- **Composite Key**: Key made of multiple columns.

5. What is the difference between `WHERE` and `HAVING`?

- **WHERE**: Filters rows before aggregation.
- **HAVING**: Filters after aggregation.

6. What are `INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN`, and `FULL OUTER JOIN`?

- **INNER JOIN**: Common rows between tables.
- **LEFT JOIN**: All rows from left + matches from right.
- **RIGHT JOIN**: All rows from right + matches from left.
- **FULL OUTER JOIN**: All rows from both tables.

7. What is the difference between `UNION` and `UNION ALL`?

- **UNION**: Combines results and removes duplicates.
- **UNION ALL**: Combines results and keeps duplicates.

8. What are Views? Advantages and Disadvantages.

- **View**: A virtual table based on a query.
- **Advantages**: Security, reusability, simplification.
- **Disadvantages**: Can be slower, sometimes not updatable.

9. What is the difference between `CHAR` and `VARCHAR`?

- **CHAR(n)**: Fixed length, pads spaces.
- **VARCHAR(n)**: Variable length, saves space.

10. What are Indexes? What is the difference between Clustered and Non-clustered indexes?

- **Index**: Improves search performance.
- **Clustered**: Sorts + stores actual data rows (1 per table).
- **Non-clustered**: Separate structure with pointers (many allowed).

Intermediate SQL Interview Questions

11. Explain Normalization and its types.
Normalization reduces redundancy and improves data integrity.

- 1NF: Atomic values.
- 2NF: No partial dependency.
- 3NF: No transitive dependency.
- BCNF: Stronger 3NF.

12. What is Denormalization? When is it used?
Denormalization adds redundancy for faster queries, often used in data warehousing/OLAP.

13. What is a Subquery? Difference between Correlated and Non-Correlated Subquery?

- **Subquery**: Query inside another query.
- **Correlated**: Executes per row of outer query.
- **Non-correlated**: Independent, runs once.

14. What is the difference between `ROW_NUMBER()`, `RANK()`, and `DENSE_RANK()`?

- **ROW_NUMBER**: Sequential, no gaps.
- **RANK**: Gives rank, leaves gaps for ties.
- **DENSE_RANK**: Ranks without gaps.

15. Explain Window Functions with an example.
Window functions operate on a set of rows without collapsing them.

```
SELECT name, salary,
RANK() OVER (ORDER BY salary DESC) AS rnk
FROM employees;
```

16. What are Common Table Expressions (CTEs)?
A temporary result set defined using `WITH`, improves readability, supports recursion.

17. What is the difference between `EXISTS` and `IN`?

- **EXISTS**: Checks if subquery returns rows (efficient for large datasets).
- **IN**: Checks value against a list/result set.

18. What is a Stored Procedure? How is it different from Functions?

- **Procedure**: Can return multiple values, allows DML.

- **Function**: Returns a single value, usually used in SELECT, no DML allowed.

19. What are Triggers? When would you use them?
Triggers are automatic procedures that execute on events like INSERT/UPDATE/DELETE. Used for auditing, constraints, and logging.

20. What is the difference between OLTP and OLAP databases?

- **OLTP**: Transaction-oriented, normalized, real-time operations.

- **OLAP**: Analytical, denormalized, used for reporting.

Advanced SQL Interview Questions

21. What is the difference between a `SELF JOIN` and a `CROSS JOIN`?

- **SELF JOIN**: Joins a table with itself.

- **CROSS JOIN**: Cartesian product of rows (m × n).

22. What is a Materialized View? How is it different from a Normal View?

- **Materialized View**: Stores physical data, can be refreshed.

- **Normal View**: Virtual table, doesn't store data.

23. How does Indexing improve performance? What are the downsides?

- Improves query performance by faster lookups.

- Downsides: Slower inserts/updates, more storage.

24. What are ACID properties in SQL?

- **Atomicity**: All or nothing.
- **Consistency**: Valid state transitions.
- **Isolation**: Transactions don't interfere.
- **Durability**: Changes persist after commit.

25. What is Deadlock? How do you resolve it in SQL?
Deadlock happens when two transactions wait on each other's locked resources.
Solution: Kill one transaction, use proper indexing, shorter transactions.

26. What is Query Optimization? Techniques to optimize SQL queries.
Process of improving query performance.
Techniques: Indexing, avoid SELECT *, proper joins, partitioning, limit subqueries.

27. What is Partitioning? Types of Partitioning in SQL.
Partitioning divides a large table into smaller parts for performance.
Types: Range, List, Hash, Composite.

28. What are Windowing Functions (`OVER` clause)? Example.
They perform calculations across rows without collapsing.
Example: `SUM(salary) OVER (PARTITION BY dept_id)`.

29. What is the difference between `DELETE CASCADE` and `ON DELETE SET NULL`?

- **CASCADE**: Deletes child rows when parent is deleted.
- **SET NULL**: Sets child foreign key to NULL on parent delete.

30. What are Transactions? What are COMMIT and ROLLBACK?
Transaction: Unit of work with ACID properties.

- **COMMIT**: Saves changes permanently.
- **ROLLBACK**: Undoes changes.

Scenario-Based SQL Interview Questions

31. Find the 2nd highest salary from an Employee table.

```
SELECT MAX(salary)
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

32. Find duplicate records in a table.

```
SELECT name, COUNT(*)
FROM employees
GROUP BY name
HAVING COUNT(*) > 1;
```

33. Find employees who do not have a manager.

```
SELECT *
FROM employees
WHERE manager_id IS NULL;
```

34. Write a query to get the department with the maximum number of employees.

```
SELECT department_id, COUNT(*) AS emp_count
FROM employees
GROUP BY department_id
ORDER BY emp_count DESC
LIMIT 1;
```

35. Find the nth highest salary (without `TOP` or `LIMIT`).

```
SELECT salary
FROM (
  SELECT salary, DENSE_RANK() OVER (ORDER BY salary DESC) AS rnk
  FROM employees
) t
WHERE rnk = 3;  -- nth highest
```
