SQL QUESTIONS

1. **Types of commands and examples:**
    - DDL: Data Definition Language (CREATE, ALTER, DROP)
    - DML: Data Manipulation Language (INSERT, UPDATE, DELETE)
    - DQL: Data Query Language (SELECT)
    - DCL: Data Control Language (GRANT, REVOKE)
    - TCL: Transaction Control Language (COMMIT, ROLLBACK)
2. **Normalization and Denormalization:**
    - Normalization: Organizing data to reduce redundancy.
    - Denormalization: Adding redundancy to improve read performance.
3. **1NF, 2NF, 3NF:**
    - 1NF: No repeating groups, each column has atomic values.
    - 2NF: 1NF + no partial dependency.
    - 3NF: 2NF + no transitive dependency.
4. **Use case for denormalization:**
    - Used in reporting systems to avoid joins and make queries faster.
5. **Primary Key and Foreign Key:**
    - Primary Key: Unique ID for each row.
    - Foreign Key: Links to primary key in another table.
6. **Alternate and Candidate Key:**
    - Candidate Key: Possible keys to uniquely identify a row.
    - Alternate Key: Candidate key not chosen as primary key.
7. **Window Functions:**
    - Perform calculation across rows related to the current row.
8. **Ranking Functions:**
    - RANK(), DENSE_RANK(), ROW_NUMBER()

- Example: Ranking employees by salary in descending order.

9. **Types of Joins:**
    - INNER JOIN: Common data.
    - LEFT JOIN: All left + matched right.
    - RIGHT JOIN: All right + matched left.
    - FULL JOIN: All from both sides.
    - CROSS JOIN: All combinations.

10. **Self Join Use Case:**

- To find manager of each employee in the same table.

11. **Subquery:**

- A query inside another query.

12. **Correlated Subquery:**

- Subquery uses data from the outer query.

13. **CTE (Common Table Expression):**

- Temporary result set for complex queries.

14. **Derived Table:**

- A subquery used in the FROM clause.

15. **Third highest salary:**

```
SELECT DISTINCT salary FROM employees ORDER BY salary DESC LIMIT 1 OFFSET 2;
```

16. **Third highest salary per department:**

- Use ROW_NUMBER() PARTITION BY department.

17. **Find duplicates in one column:**

```
SELECT column, COUNT(*) FROM table GROUP BY column HAVING COUNT(*) > 1;
```

18. **Find duplicates in multiple columns:**

```
SELECT col1, col2, COUNT(*) FROM table GROUP BY col1, col2 HAVING COUNT(*) > 1;
```

19. **ACID Properties:**

- Atomicity, Consistency, Isolation, Durability

20. **UNION vs UNION ALL:**

- UNION removes duplicates.
- UNION ALL keeps duplicates.

21. **Primary vs Unique Key:**

- Primary: One per table, no null.
- Unique: Can have many, allows one null.

22. **TRUNCATE vs DELETE:**

- TRUNCATE: Fast, removes all rows.
- DELETE: Can remove specific rows.

23. **HAVING vs WHERE:**

- WHERE: Before grouping.
- HAVING: After grouping.

24. **SQL Execution Order:**
    FROM > WHERE > GROUP BY > HAVING > SELECT > ORDER BY > LIMIT

25. **Indexes:**

- Help find data fast.
- Types: Clustered, Non-clustered, Unique, Full-text

26. **SQL Optimization:**

- Use indexes, avoid *, use joins properly, limit rows.