# TOP SQL INTERVIEW QUESTIONS WITH ANSWERS

Neha Jain

## 1. Basic SQL Concepts

1. **Q: What is SQL? Why is it important?**
   SQL (Structured Query Language) is used to interact with relational databases. It allows users to retrieve, insert, update, and delete data, making it essential for database management in any application.
2. **Q: What are the different types of SQL commands?**
   SQL commands are categorized as:
   - **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`, `SAVEPOINT`
3. **Q: What is a primary key?**
   A primary key is a column or combination of columns that uniquely identifies each row in a table.
   - It must contain unique values.
   - It cannot contain `NULL`.
4. **Q: What is the difference between `WHERE` and `HAVING`?**
   - `WHERE` filters rows before grouping is applied.
   - `HAVING` filters groups after `GROUP BY` is applied.
5. **Q: What are the differences between SQL and NoSQL databases?**
   - **SQL:** Uses structured schema (tables with rows and columns), best for relational data.
   - **NoSQL:** Schema-less, supports unstructured data like JSON, ideal for large-scale distributed systems.

---

## 2. Advanced SQL Concepts

6. **Q: What is a foreign key? How is it different from a primary key?**

   A foreign key is a column in a table that creates a relationship between two tables by referencing the primary key in another table.
   - Primary keys uniquely identify rows in a table.
   - Foreign keys establish relationships between tables.
7. **Q: What is normalization? Why is it important?**

   Normalization organizes data to reduce redundancy and improve data integrity.
   - It divides data into smaller tables and defines relationships using keys.
   - Example: Removing duplicate data ensures efficient storage and consistency.

NEHA JAIN

8. **Q: Explain the concept of indexing in SQL.**
   An index is a database structure that improves the speed of data retrieval.
   - **Clustered Index:** Sorts and stores data rows in the table based on key values.
   - **Non-clustered Index:** Contains pointers to data instead of sorting.
9. **Q: What is a view in SQL?**
   A view is a virtual table based on a result set of a SQL query.
   - It doesn't store data physically.
   - Useful for security and simplifying complex queries.
     ```
     CREATE VIEW view_name AS

     SELECT column1, column2 FROM table_name WHERE condition;
     ```
10. **Q: What is a stored procedure? How is it different from a function?**
    - **Stored Procedure:** A set of SQL statements executed as a single unit. It may or may not return a value.
    - **Function:** Returns a single value or table and is used in SQL queries.

---

## 3. Performance and Optimization

11. **Q: What are some common techniques for optimizing SQL queries?**
    - Use proper indexing.
    - Avoid using `SELECT *`; specify required columns.
    - Use joins instead of subqueries where possible.
    - Analyze execution plans.
    - Limit the number of rows using `LIMIT` or `TOP`.
12. **Q: What is the difference between `UNION` and `UNION ALL`?**
    - **`UNION`:** Removes duplicates and combines results from multiple queries.
    - **`UNION ALL`:** Combines results without removing duplicates.
13. **Q: What is an execution plan? Why is it useful?**
    An execution plan shows how SQL Server executes a query.
    - Helps in identifying performance bottlenecks like missing indexes or expensive operations.
14. **Q: What is the difference between `INNER JOIN` and `OUTER JOIN`?**
    - **`INNER JOIN`:** Returns matching rows from both tables.
    - **`OUTER JOIN`:** Returns matching rows plus unmatched rows from one or both tables.
15. **Q: How do you prevent SQL injection?**
    - Use parameterized queries or prepared statements.
    - Validate user input.
    - Avoid dynamic SQL.

---

NEHA JAIN

## 4. Practical Questions

16. **Q: How do you fetch the nth highest salary from an employee table?**

```
SELECT DISTINCT salary

FROM employees

ORDER BY salary DESC

LIMIT n-1, 1;
```

17. **Q: How do you find duplicate rows in a table?**

```
SELECT column_name, COUNT(*)

FROM table_name

GROUP BY column_name

HAVING COUNT(*) > 1;
```

16. **Q: How do you fetch data between two dates?**

```
SELECT * FROM table_name

WHERE column_date BETWEEN '2024-01-01' AND '2024-12-31';
```

17. **Q: What is the difference between `TRUNCATE` and `DELETE`?**
    - `DELETE:` Removes rows but can be rolled back.
    - `TRUNCATE:` Removes all rows and cannot be rolled back.
18. **Q: How do you fetch records where a column starts with a specific character?**

```
SELECT * FROM table_name

WHERE column_name LIKE 'A%';
```

---

## 5. Data Manipulation and Security

21. **Q: What is the difference between `COMMIT` and `ROLLBACK`?**
    - **`COMMIT:`** Saves all changes made in the transaction permanently.
    - **`ROLLBACK:`** Undoes all changes made in the transaction.
22. **Q: Explain database constraints.**
    Constraints enforce rules on data:
    - `NOT NULL`
    - `UNIQUE`

NEHA JAIN

    - `PRIMARY KEY`
    - `FOREIGN KEY`
    - `CHECK`
    - `DEFAULT`
23. **Q: What is a trigger in SQL?**
    A trigger is a special type of stored procedure that automatically executes in response to certain events (INSERT, UPDATE, DELETE).
24. **Q: What is the difference between `GRANT` and `REVOKE`?**
    - **`GRANT`:** Assigns permissions to users.
    - **`REVOKE`:** Removes permissions from users.
25. **Q: How do you implement row-level security?**
    Use views or `WHERE` clauses with conditions based on user roles or IDs.

---

## 6. Advanced Theoretical Questions

26. **Q: What is denormalization? Why is it used?**
    Denormalization combines tables to improve read performance at the cost of increased redundancy.
27. **Q: What is ACID in databases?**
    - **Atomicity:** Transactions are all-or-nothing.
    - **Consistency:** Ensures data integrity.
    - **Isolation:** Transactions don't interfere with each other.
    - **Durability:** Changes persist even after a crash.

28. **Q: What are window functions? Give an example.**
    Window functions operate on a set of rows related to the current row.

```
SELECT employee_id,

       salary,

       RANK() OVER (PARTITION BY department ORDER BY salary DESC)
AS rank

FROM employees;
```

29. **Q: What is a recursive query in SQL?**

    A query that refers to itself to retrieve hierarchical data.

NEHA JAIN

```
WITH RECURSIVE hierarchy AS (
    SELECT id, parent_id FROM employees WHERE parent_id IS NULL
    UNION ALL
    SELECT e.id, e.parent_id
    FROM employees e
    INNER JOIN hierarchy h ON e.parent_id = h.id)
SELECT * FROM hierarchy;
```

30. **Q: What are transactions in SQL?**
    Transactions ensure a group of operations are executed as a single unit, maintaining data integrity.

## 7. Query Optimization

31. **Q: What are common causes of slow queries?**
    - Lack of proper indexing.
    - Use of `SELECT *` instead of selecting specific columns.
    - Complex joins without optimization.
    - Absence of query limiters like `WHERE` or `LIMIT`.
    - Large table scans due to missing filters or inefficient data organization.
32. **Q: How do you identify and optimize slow queries?**
    - Use `EXPLAIN` or `EXPLAIN PLAN` to analyze query execution steps.
    - Add appropriate indexes.
    - Rewrite queries to minimize subqueries or replace them with joins.
    - Partition large tables for better query performance.
33. **Q: What is database sharding?**
    Database sharding divides large datasets into smaller, more manageable chunks (shards) distributed across multiple databases.
    - It improves performance and scalability.
34. **Q: How does indexing impact write operations?**
    - While indexes speed up read operations, they can slow down write operations like `INSERT`, `UPDATE`, or `DELETE` because the index also needs to be updated.

---

NEHA JAIN

## 8. Joins and Relationships

35. **Q: Explain the different types of joins with examples.**

    **Inner Join:** Returns matching rows from both tables.
    ```
    SELECT * FROM table1 INNER JOIN table2 ON table1.id = table2.id;
    ```

    **Left Join:** Returns all rows from the left table and matching rows from the right.
    ```
    SELECT * FROM table1 LEFT JOIN table2 ON table1.id = table2.id;
    ```

    **Right Join:** Returns all rows from the right table and matching rows from the left.
    ```
    SELECT * FROM table1 RIGHT JOIN table2 ON table1.id = table2.id;
    ```

    **Full Outer Join:** Combines results of left and right joins.
    ```
    SELECT * FROM table1 FULL OUTER JOIN table2 ON table1.id =
    table2.id;
    ```

36. **Q: What is a self-join? Provide an example.**
    A self-join is a join where a table is joined to itself.
    ```
    SELECT A.id, B.id

    FROM employees A, employees B

    WHERE A.manager_id = B.id;
    ```

37. **Q: How do you implement many-to-many relationships in SQL?**
    Create a junction table with foreign keys referencing the primary keys of the related tables.
    ```
    CREATE TABLE student_course (

     student_id INT,

        course_id INT,

        PRIMARY KEY (student_id, course_id),

        FOREIGN KEY (student_id) REFERENCES students(id),

        FOREIGN KEY (course_id) REFERENCES courses(id)

    );
    ```

NEHA JAIN

## 9. Security and Permissions

38. **Q: What is SQL injection? How do you prevent it?**
    SQL injection is a code injection technique where malicious SQL statements are inserted into queries.
    **Prevention Techniques:**
    - Use parameterized queries or prepared statements.
    - Validate and sanitize user inputs.
    - Use ORM frameworks like Hibernate or Entity Framework.
39. **Q: How do you encrypt data in SQL?**
    Use database-level encryption methods like Transparent Data Encryption (TDE) or SQL functions like `ENCRYPTBYKEY` and `DECRYPTBYKEY`.

---

## 10. Theoretical Concepts

40. **Q: What is a cursor in SQL?**
    A cursor is a database object used to retrieve and manipulate rows from a result set one at a time.
    - Useful for row-by-row operations but may reduce performance.
41. **Q: What is the difference between clustered and non-clustered indexes?**
    - **Clustered Index:** Sorts and stores data rows physically in the table. Only one per table.
    - **Non-clustered Index:** Contains pointers to the actual data rows. Can have multiple per table.
42. **Q: What is a materialized view?**
    A materialized view stores the result of a query physically.
    - Unlike regular views, they improve performance but require refreshing to stay updated.
43. **Q: What is the difference between `ISNULL()` and `COALESCE()`?**
    - **`ISNULL(expr1, expr2)`:** Returns expr2 if expr1 is `NULL`.
    - **`COALESCE(expr1, expr2, ...)`:** Returns the first non-NULL value from the list of expressions.

---

## 11. Practical Scenarios

44. **Q: Write a query to get the second highest salary from the employees table.**

```
SELECT MAX(salary)

FROM employees
```

NEHA JAIN

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

45. **Q: How do you fetch employees with the same salary?**

```
SELECT salary, COUNT(*)

FROM employees

GROUP BY salary

HAVING COUNT(*) > 1;
```

46. **Q: Write a query to delete duplicate rows from a table.**

```
DELETE FROM table_name

WHERE id NOT IN (

    SELECT MIN(id)

    FROM table_name

    GROUP BY column1, column2, ...);
```

47. **Q: How do you find the first and last record in a table?**

```
SELECT * FROM table_name ORDER BY id ASC LIMIT 1; -- First record

SELECT * FROM table_name ORDER BY id DESC LIMIT 1; -- Last record
```

---

## 12. Real-World Applications

49. **Q: How do you perform a database backup?**

```
mysqldump -u username -p database_name > backup.sql
```

50. **Q: What is database partitioning, and how does it help?**
    Database partitioning splits a table into smaller, manageable pieces.
    - Types: Range, List, Hash, Composite.
    - Improves performance and query efficiency for large datasets.

NEHA JAIN