Most Recently Asked SQL Interview Questions and Answers

Basic Level

1. **What is the difference between WHERE and HAVING clause?**

   o `WHERE` filters rows before grouping; `HAVING` filters groups after aggregation.

   o Example:

   ```sql
   SELECT department, COUNT(*) FROM employees
   WHERE salary > 30000
   GROUP BY department
   HAVING COUNT(*) > 5;
   ```

2. **What are the different types of JOINs in SQL? Explain with example.**

   o INNER JOIN: Returns matching records from both tables.
   o LEFT JOIN: Returns all records from left table and matched records from right.
   o RIGHT JOIN: Opposite of LEFT JOIN.
   o FULL OUTER JOIN: Returns all records from both tables.
   o CROSS JOIN: Returns Cartesian product.

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

   o `DELETE`: Removes rows, can use WHERE, logged.
   o `TRUNCATE`: Removes all rows, faster, can't use WHERE.
   o `DROP`: Deletes table structure entirely.

4. **What is a Primary Key and Foreign Key?**

   o Primary Key: Uniquely identifies each record in a table.
   o Foreign Key: A field in one table that refers to the Primary Key in another.

5. **What is the difference between IN, EXISTS, and ANY?**

   o `IN`: Matches a value against a list.
   o `EXISTS`: Returns true if a subquery returns any result.
   o `ANY`: Compares a value to any value in a subquery.

Intermediate Level

6. **Write a SQL query to find the second highest salary.**

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

7. **What is the difference between ROW_NUMBER(), RANK(), and DENSE_RANK()?**

   - `ROW_NUMBER()`: Assigns unique sequential number.
   - `RANK()`: Same ranks get same number, next rank is skipped.
   - `DENSE_RANK()`: No ranks skipped.

8. **What are CTEs (Common Table Expressions)?**

   - Temporary named result set, used for recursion or simplifying queries.

   - Example:

   ```sql
   WITH Sales AS (
     SELECT * FROM orders WHERE year = 2024
   )
   SELECT * FROM Sales;
   ```

9. **What is the difference between clustered and non-clustered indexes?**

   - Clustered Index: Alters physical order of data.
   - Non-clustered Index: Separate from actual table, stores pointers.

10. **Explain the use of window functions in SQL.**

- Perform calculations across a set of rows related to current row.
- Example: `ROW_NUMBER()`, `RANK()`, `LEAD()`, `LAG()`.

Advanced Level

11. **Write a query to pivot data in SQL (convert rows to columns).**

```sql
SELECT dept,
  SUM(CASE WHEN year = 2023 THEN sales ELSE 0 END) AS sales_2023,
  SUM(CASE WHEN year = 2024 THEN sales ELSE 0 END) AS sales_2024
FROM revenue
GROUP BY dept;
```

12. **What is normalization? Explain 1NF, 2NF, 3NF.**
- 1NF: Atomic values.
- 2NF: No partial dependency on PK.
- 3NF: No transitive dependency.

13. **How do you optimize a slow SQL query?**
- Use indexes.
- Avoid SELECT *.
- Use joins efficiently.
- Analyze execution plan.

14. **What is the difference between UNION and UNION ALL?**
- `UNION`: Removes duplicates.

- `UNION ALL`: Keeps all records, including duplicates.

15. **Explain transactions in SQL. What are ACID properties?**

- A: Atomicity
- C: Consistency
- I: Isolation
- D: Durability

Real-World Scenarios

16. **How would you detect and remove duplicate records from a table?**

```sql
DELETE FROM employees
WHERE id NOT IN (
  SELECT MIN(id) FROM employees GROUP BY name, salary
);
```

17. **Write a query to fetch the employee with the maximum salary in each department.**

```sql
SELECT * FROM employees e
WHERE salary = (
  SELECT MAX(salary) FROM employees
  WHERE department = e.department
);
```

18. **How do you handle NULL values in SQL queries?**

- Use `IS NULL`, `IS NOT NULL`, or `COALESCE()` to replace nulls.

19. **What is SQL Injection? How can you prevent it?**

- A code injection technique that can destroy your database.
- Prevent using prepared statements, stored procedures, and input validation.

20. **How does indexing affect performance? When should you avoid indexing?**

- Speeds up reads, slows down writes.
- Avoid indexing on frequently updated columns or small tables.