SQL Interview Questions and Answers

1) What is the difference between INNER JOIN and LEFT JOIN?

- **INNER JOIN:** Returns only the rows where there is a match in both tables.
- **LEFT JOIN:** Returns all rows from the left table, and matching rows from the right table. If there is no match, NULL values are returned for columns from the right table.

2) What is a subquery in SQL?

A subquery is a query within another query. It is used to perform operations that need the results of another query. Subqueries can be used in SELECT, INSERT, UPDATE, and DELETE statements.

3) What is normalization?

Normalization is the process of organizing data in a database to avoid redundancy and dependency by dividing large tables into smaller ones and defining relationships between them.

4) What is a primary key and a foreign key?

Primary Key: A column or a set of columns that uniquely identify each row in a table.

Foreign Key: A column that creates a relationship between two tables by referring to the primary key of another table.

5) What is an aggregate function in SQL? Give examples.

Aggregate functions operate on a set of values and return a single value. Examples include:

- COUNT(): Counts the number of rows.
- SUM(): Returns the sum of values.
- AVG(): Calculates the average of values.
- MAX() and MIN(): Return the highest and lowest values.

6) What is the difference between WHERE and HAVING clause?

- WHERE: Filters records before any grouping is applied.

- HAVING: Filters records after the GROUP BY clause has been applied.

7) What is the GROUP BY clause in SQL?

The GROUP BY clause is used to arrange identical data into groups. It is often used with aggregate functions to perform operations like summing, counting, or averaging grouped data.

8) Explain the DISTINCT keyword?

The DISTINCT keyword is used to remove duplicate rows from the result set and return only unique records.

9) What is the difference between UNION and UNION ALL?

- UNION: Combines the results of two queries and removes duplicate records.

- UNION ALL: Combines the results of two queries, including duplicates.

10) What is the difference between DELETE and TRUNCATE?

- DELETE: Removes rows from a table based on a condition, and it can be rolled back.

- TRUNCATE: Removes all rows from a table without logging individual row deletions, and it cannot be rolled back.

11) Explain the concept of ACID properties in database management?

ACID stands for:

- **Atomicity:** Ensures all operations within a transaction are completed successfully.
- **Consistency:** Guarantees that a transaction brings the database from one valid state to another.
- **Isolation:** Ensures that transactions are executed independently.
- **Durability:** Ensures that once a transaction is committed, it remains so, even in the event of a system crash.

12) Explain Constraints in SQL?

Constraints in SQL are rules that are applied to columns in a table to ensure the accuracy and integrity of the data within that table. They enforce certain conditions on the data, ensuring that it adheres to business rules or other requirements. Here's an overview of the main types of constraints in SQL:

- **Primary Key**: Uniquely identifies each row in a table, cannot be NULL.
- **Foreign Key**: Links columns between tables, ensuring referential integrity.
- **Unique**: Ensures all values in a column are distinct.
- **Check**: Validates values in a column based on a condition.
- **Not Null**: Ensures a column cannot have NULL values.
- **Default**: Sets default values for a column when none are provided.
- **Index**: Optimizes query performance, often related to primary/foreign keys.

13) What are subsets in SQL?

In SQL, the term "subsets" is not commonly used in a formal context, but it can refer to the different parts or components of SQL that work together to manipulate and retrieve data. These can include:

- **Data Query Language (DQL)**: Used for querying data. The primary command is SELECT.
- **Data Definition Language (DDL)**: Used to define and modify database structures. Key commands include CREATE, ALTER, DROP, and TRUNCATE.
- **Data Manipulation Language (DML)**: Used for managing data within tables. Key commands are INSERT, UPDATE, and DELETE.
- **Data Control Language (DCL)**: Deals with permissions and access control. Includes commands like GRANT and REVOKE.
- **Transaction Control Language (TCL)**: Used to manage transactions in a database. Includes commands like COMMIT, ROLLBACK, and SAVEPOINT.

14) what are SET operators in SQL?

In SQL, set operators are used to combine the results of two or more SELECT queries into a single result set. The most common set operators are:

- UNION: Combines and removes duplicates.
- UNION ALL: Combines and includes duplicates.
- INTERSECT: Returns common rows.
- EXCEPT: Returns rows that are only in the first query, not the second.

15) What are the differences between IN and BETWEEN operator?

- IN checks for specific values in a list or subquery.
- BETWEEN checks for a range of values (numeric, date, or other comparable data types.

16) What is the difference between RANK and DENSE RANK function?

In SQL, both RANK() and DENSE_RANK() assign ranks to rows based on a specific order, but differ in handling ties. RANK() assigns the same rank to tied rows but leaves gaps in subsequent ranks (e.g., if two rows are ranked 1, the next rank will be 3). In contrast, DENSE_RANK() also assigns the same rank to tied rows but does not leave gaps, so after rank 1, the next rank will be 2, even if there were ties at rank 1.

17) What is the difference between views and tables?

The key difference between views and tables in SQL is that tables are physical database objects that store data directly, while views are virtual tables that do not store data but present a query's result set dynamically. A view is essentially a saved SQL query that retrieves data from one or more tables. Views are useful for simplifying complex queries, enhancing security by limiting access to specific columns or rows, and providing abstraction. However, since views are not physical, updates on them depend on certain conditions and may not always be allowed.

18) How would you optimize a slow query?

- Add Indexes: Create indexes on frequently filtered or joined columns.
- Analyze Query Plan: Use EXPLAIN or EXPLAIN ANALYZE.
- Optimize Joins: Avoid nested loops, prefer indexed joins.
- Reduce Data Scanned: Use LIMIT, avoid SELECT *, and filter early with WHERE.
- Partitioning/Sharding: Split large tables for faster access.
- Caching: Cache frequent results.