Day 5-Interview questions

1. What is the purpose of the ORDER BY clause in SQL?

The ORDER BY clause is used to sort the result set of a query based on one or more columns in ascending or descending order.

2. Can you use multiple columns in the ORDER BY clause? If yes, how do you specify the order for each column?

Yes, you can use multiple columns in the ORDER BY clause. To specify the order for each column, you list them separated by commas. For example: ORDER BY column1 ASC, column2 DESC.

3. Explain the difference between ORDER BY and GROUP BY.

ORDER BY is used to sort the result set, while GROUP BY is used to group rows that have the same values in specified columns into summary rows.

4. What is the default sort order in the ORDER BY clause if you don't specify ASC or DESC?

The default sort order is ascending (ASC) if you don't specify ASC or DESC.

5. Can you use column aliases in the ORDER BY clause?

No, you cannot use column aliases in the ORDER BY clause. You must use the original column name or column expression.

6. What is the purpose of the GROUP BY clause in SQL?

The GROUP BY clause is used to group rows that have the same values in specified columns into summary rows, like using aggregate functions.

7. Can you use aggregate functions without using the GROUP BY clause?

Yes, you can use aggregate functions without the GROUP BY clause, but they will operate on the entire result set, not on grouped subsets.

8. Explain the HAVING clause in conjunction with the GROUP BY clause.

The HAVING clause is used in conjunction with GROUP BY to filter the results of a grouped query. It is similar to the WHERE clause but is applied after the grouping.

9. What is the difference between WHERE and HAVING when used with the GROUP BY clause?

The WHERE clause filters rows before grouping, while the HAVING clause filters grouped rows after the grouping.

10. Can you use the GROUP BY clause without using any aggregate functions?

No, when using GROUP BY, every selected column must be either an aggregate function or included in the GROUP BY clause.

11. What is the purpose of the UNION operator in SQL?

The UNION operator is used to combine the result sets of two or more SELECT statements. It removes duplicate rows from the combined result set.

12. Explain the key characteristics of the UNION operator.

The number and order of columns must be the same in all SELECT statements.
Column data types must be compatible.
The result set is sorted and duplicates are removed by default.
Use UNION ALL if you want to include duplicate rows.

13. How does the INTERSECT operator work in SQL?

The INTERSECT operator returns the common rows between the result sets of two SELECT statements. It is used to find the intersection of sets, meaning it retrieves rows that appear in both result sets.

14. Can you use the INTERSECT operator with multiple SELECT statements?

In standard SQL, the INTERSECT operator is binary and can be used with only two SELECT statements. However, some database systems may support additional syntax for more than two sets.

15. Explain the common use case for the SET operators (UNION, INTERSECT, and EXCEPT).

SET operators are useful when dealing with different tables or conditions where you want to perform operations similar to set operations, such as combining, finding intersections, or finding differences between sets of data.

16. Can you use the WHERE clause with the UNION operator?

Yes, you can use the WHERE clause with each SELECT statement within the UNION to filter the rows before combining them.

17. How does the EXCEPT operator differ from the INTERSECT operator?

The EXCEPT operator returns the rows that are unique to the first SELECT statement but not present in the second SELECT statement. In contrast, the INTERSECT operator returns common rows between two SELECT statements.