iNeuron

SQL Query

Interview Questions

1. What is a cross join?
Ans. A cross join is a join that gives the Cartesian product of all the rows from both the tables in the join. In other words, each row from the first table and each row from the second table are combined. For example - If table 1 is -
Letters
A
B
C
And table 2 is -
Numbers
1
2
3
The cross join of both these tables will give us -
Letters Numbers
A 1
A 2
A 3
B 1
B 2
B 3
C 1
C 2
C 3

2. Which command is used to remove duplicates from the result-set obtained by the second SELECT query from the result-set obtained by the first SELECT query and then return the filtered results from the first?
Ans. The distinct command is used to remove duplicates.

3. What is a cursor?
Ans. The rows (one or more) that a SQL statement returns are stored in a cursor. You can give a cursor a name so that a program can use it to retrieve and handle the rows returned by the SQL statement one at a time. Two different types of cursors exist - Implicit cursors and Explicit cursors.

4. How do the clustered and non-clustered indexes differ in SQL?
Ans. An index aids in enhancing both overall performance and speed of data retrieval. If the primary key and unique constraint are defined on the table, the index is automatically constructed. Clustered and non-clustered indexes are two different types of indexes.
A clustered index on a column is automatically created in SQL Server by the primary key constraint. There can only be one clustered index per table, as specified by the protocol. A clustered index is used, much like a dictionary, to define the order, sort the table, or organize the data in alphabetical order.
A non-clustered index records data in a different location than where it is collected. Pointers to the location of that data are contained in the index. There can be multiple non-clustered indexes on a table, as specified by the protocol. It does not impact the order of the data stored in the table.
Further, no extra space is required to store the logical structure of a clustered index whereas extra space is required to store the logical structure of an unclustered index.
If data retrieval is your priority then clustered index works best whereas if updating data is your priority then non-clustered indexes work best.

5. What is a view? How is it different from a table?
Ans. A view is nothing but a virtual table. The rows and columns in a view are similar to those in a table. A view is a database object that is built on top of another table (or view), as opposed to a table, which holds data on its own. If data in the underlying table changes, the display will also reflect those changes. A view may be constructed on top of one or more tables. A view can also be defined on top of a different view.

6. Explain the different forms of normalization. What is the use of normalization?
Ans. Redundancy from a relation or group of relations is minimized through the process of normalization. Insertion, deletion, and update abnormalities could result from relational redundancy. Redundancy in database tables can be reduced with the help of normal forms. The different forms of normalization are explained as follows -
a. First Normal Form – A relation violates the first normal form if it has composite or multi-valued attributes, or it is in the first normal form if neither of these attributes is present. If all of the attributes in a relation are singled valued attributes, the connection is said to be in the first normal form.
b. Second Normal Form – A relation must be in the first normal form and be devoid of any partial dependencies in order to be in the second normal form. If a relation has No Partial Dependency, which means that no non-prime attribute—i.e., an attribute that is not included in any candidate key—is dependent on any suitable subset of any candidate key in the table, then the relation is in 2NF.
c. Third Normal Form – A relation is in the third normal form if it is both in the second normal form and there is no transitive dependency for non-prime characteristics. If at least one of the

following applies to every non-trivial function dependence X -> Y, a relation is in 3NF - Super key X is used or the prime attribute is Y.
d. Boyce-Codd Normal Form (BCNF) – If a relation R is in Third Normal Form for every FD, it is said to be in BCNF. The LHS acts as the super key. A relation is in the BCNF if X is a super key in every non-trivial functional dependency X -> Y

7. Explain how TRUNCATE, DELETE and DROP statements differ from one another
Ans.
● TRUNCATE in SQL is a DDL command. It is used to eliminate every record from a table. An existing table's records are deleted, but the table itself is left intact. The table's schema or structure is maintained. As a DDL command, the TRUNCATE TABLE statement cannot be undone i.e. it cannot be rolled back.
● DELETE in SQL is a DML command. It is used to remove current records from a table that already exists. Depending on the query's criterion, we can delete a single record or a number of records. Since DELETE is a DML command, it can be undone i.e. it can be rolled back.
● DROP is a DDL Command. Existing database objects can be deleted using the DROP statement. You can use it to remove databases, tables, views, triggers, and other objects. The deletion of an object with the DROP command cannot be undone i.e. rolled back and is irreversible.
● Prior to deletion, the DELETE statement checks every row. As a result, it takes longer than the TRUNCATE command. Using TRUNCATE instead of DELETE when deleting all the records from a table is recommended because it is quicker. The DROP command eliminates the entire schema/structure of the table from the

database, in contrast, to TRUNCATE which simply deletes the data of the tables.

8. What are Scalar functions?
Ans. There are some built-in functions in SQL that are known as scalar functions, and no matter what input is passed to a scalar function, it will always return a single value as its output. In SQL, the scalar functions treat each record separately. Scalar functions are often used and include the following: UCASE(), LCASE(), MID(), LENGTH(), ROUND(), NOW(), FORMAT()

9. What are aggregate functions in SQL?
Ans. A column's multiple values are calculated by an aggregate function in SQL, which in turn produces a single value. The GROUP BY and HAVING clauses in a SELECT statement frequently go hand in hand with aggregate functions. Avg, count, sum, min, max, and many other aggregate methods are available in SQL. With the exception of the count function, an aggregate function does not take into account NULL values while calculating results.

10. What are OLAP and OLTP?
Ans. Data analysis for business choices is done using a class of software tools known as online analytical processing (OLAP). OLAP offers a setting where users can simultaneously access insights from the database gathered from several database systems. Examples: An OLAP system is any type of data warehousing system. In a three-tier design, online transaction processing (OLTP) offers transaction-oriented applications. The daily operations of an organization are managed by OLTP. The use of OLTP is possible for online banking, sending text messages, and adding clothes to shopping carts.

11. What do we use for pattern matching in SQL?
Ans. We use wildcard characters for pattern matching in SQL.

12. Explain triggers in SQL
Ans. A trigger is a specific kind of stored procedure that launches automatically whenever a database server event takes place. When a user attempts to edit data using a data manipulation language (DML) event, DML triggers are activated. DML operations are statements that INSERT, UPDATE, or DELETE data from a table or view. Whether or whether table rows are affected, these triggers are triggered whenever a legitimate event occurs.

13. Is a NULL value equivalent to a blank space or zero?
Ans. No, a NULL value is not equivalent to a black space or zero. Any value that is "unavailable, unassigned, unknown, or not applicable" is referred to as a NULL value. Whereas, a blank space is a character and zero is a number.

14. What is the use of the COALESCE function?
Ans. The COALESCE function is used to return the very first value from a series that is NOT NULL. It evaluates the expressions in the order and returns the first value which is not null.

15. Explain the difference between where and having clauses?
Ans. Similar to a WHERE clause, a HAVING clause only applies to groups as a whole (i.e., to the rows in the result set that represent groups), but a WHERE clause applies to specific rows. Both a WHERE clause and a HAVING clause may be present in a query. If so, then -
● Individual rows in the tables are the first to get the WHERE clause's application. The rows that meet the WHERE clause's criteria are grouped together as a result set.

● The rows in the result set are then subjected to the HAVING clause. The query output only contains the groups that satisfy the HAVING requirements.

16. Explain character manipulation functions in SQL
Ans. A function that accepts one or more characters or numbers as input and outputs a character value is known as a character manipulation function. A string value is returned as a result set from basic string functions, which have a number of features. Here are the SQL character functions:
● To lowercase all the characters in a string, use the SQL LOWER() method.
● Using the SQL UPPER() method, all characters in a string are changed to uppercase.
● The SQL TRIM() function eliminates leading and trailing characters from character strings, or both.
● The SQL TRANSLATE() method swaps out one string's set of characters for another string. A single character is replaced at a time using the function.

17. How can we display the current date in SQL?
Query- SELECT GETDATE();

18. How can we display alternate records from a table?
Query- SELECT * FROM table_name WHERE column_name%2 = 1;

19. How can you find the second highest salary from the given employee table?
Query- SELECT MAX(emp_salary) FROM Employee WHERE SALARY < (SELECT MAX(emp_salary) FROM Employee);

20. How can we copy the entire data from one table to another in SQL?
Query- INSERT INTO new_table SELECT * FROM old_table;