1) What is the difference between inner and outer join? Explain with example.

- Inner join returns rows when there is at least one match in both tables

- Outer join will return matching rows from both tables as well as any unmatched rows from one or both the tables

2) What is the difference between JOIN and UNION?

- SQL JOIN allows us to “lookup” records on other table based on the given conditions between two tables.

- UNION operation allows us to add 2 similar data sets to create resulting data set that contains all the data from the source data sets. Union does not require any condition for joining.

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

- UNION operation returns only the unique records from the resulting data set

- UNION ALL will return all the rows, even if one or more rows are duplicated to each other.

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

- WHERE clause can only be applied on a static non-aggregated column

- we will need to use HAVING for aggregated columns.

5) What is the difference among UNION, MINUS and INTERSECT?

- UNION combines the results from 2 tables and eliminates duplicate records from the result set.

- MINUS operator when used between 2 tables, gives us all the rows from the first table except the rows which are present in the second table.

- INTERSECT operator returns us only the matching or common rows between 2 result sets.

6) How can we transpose a table using SQL (changing rows to column or vice-versa) ?

- The usual way to do it in SQL is to use CASE statement or DECODE statement

7) How to generate row number in SQL Without ROWNUM

- SELECT name, sal, (SELECT COUNT(\*) FROM EMPLOYEE i WHERE o.name >= i.name) row\_num

FROM EMPLOYEE o

order by row\_num

8) How to select first 5 records from a table?

sql server -> SELECT TOP 5 \* FROM EMP;

Oracle -> SELECT \* FROM EMP WHERE ROWNUM <= 5;

Generic -> SELECT name FROM EMPLOYEE o

WHERE (SELECT count(\*) FROM EMPLOYEE i WHERE i.name < o.name) < 5

======================= SQL server questions ==========================================

9) What are DMVs? -

Dynamic Management Views (DMVs), are functions that give you information on the state of the server. DMVs, for the most part, are used to monitor the health of a server. They really just give you a snapshot of what’s going on inside the server. They let you monitor the health of a server instance, troubleshoot major problems and tune the server to increase performance

10) Define a temp table

In a nutshell, a temp table is a temporary storage structure. What does that mean? Basically, you can use a temp table to store data temporarily so you can manipulate and change it before it reaches its destination format.

11) What’s the difference between a local temp table and a global temp table?

Local tables are accessible to a current user connected to the server. These tables disappear once the user has disconnected from the server. Global temp tables, on the other hand, are available to all users regardless of the connection. These tables stay active until all the global connections are closed.

12) Describe the difference between truncate and delete

Delete command removes the rows from a table based on the condition that we provide with a WHERE clause.

Truncate will actually remove all the rows from a table and there will be no data in the table after we run the truncate command.

13) What is a view?

A view is simply a virtual table that is made up of elements of multiple physical or “real” tables. Views are most commonly used to join multiple tables together, or control access to any tables existing in background server processes.

14) What is the default port number for SQL Server? -

Basically, when SQL Server is enabled the server instant listens to the TCP port 1433. It can be changed from the Network Utility TCP/IP properties.

15) What are the difference between clustered and a non-clustered index?

A clustered index is a special type of index that reorders the way records in the table are physically stored. Therefore table can have only one clustered index. The leaf nodes of a clustered index contain the data pages.

A non clustered index is a special type of index in which the logical order of the index does not match the physical stored order of the rows on disk. The leaf node of a non clustered index does not consist of the data pages. Instead, the leaf nodes contain index rows.

16) What is PRIMARY KEY?

A PRIMARY KEY constraint is a unique identifier for a row within a database table. Every table should have a primary key constraint to uniquely identify each row and only one primary key constraint can be created for each table. The primary key constraints are used to enforce entity integrity.

17) What is FOREIGN KEY?

A FOREIGN KEY constraint prevents any actions that would destroy links between tables with the corresponding data values. A foreign key in one table points to a primary key in another table. Foreign keys prevent actions that would leave rows with foreign key values when there are no primary keys with that value. The foreign key constraints are used to enforce referential integrity.

16) What's the difference between a primary key and a unique key?

Both primary key and unique key enforces uniqueness of the column on which they are defined.

But by default primary key creates a clustered index on the column, where are unique creates a non-clustered index by default.

Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

18) What are the advantages of using Stored Procedures?

Stored procedure can reduced network traffic and latency, boosting application performance.

Stored procedure execution plans can be reused, staying cached in SQL Server's memory, reducing server overhead.

Stored procedures help promote code reuse.

Stored procedures can encapsulate logic. You can change stored procedure code without affecting clients.

Stored procedures provide better security to your data.

19) Using a with statement, how would you show who ordered cat food, and the total amount of money spent?

with cat\_food as (

Select Customer\_ID, SUM(Price) as TOTAL\_PRICE

From Orders

Where Item like "%Cat Food%"

Group by Customer\_ID

)

Select Customer\_name, TOTAL\_PRICE

From Customers c

Inner JOIN cat\_food f

ON c.Customer\_ID = f.Customer\_ID

where c.Customer\_ID in (Select Customer\_ID from cat\_food);

Although this was a simple statement, the "with" clause really shines when a complex query needs to be run on a table before joining to another. With statements are nice, because you create a pseudo temp when running your query, instead of creating a whole new table.

The Sum of all the purchases of cat food weren't readily available, so we used a with statement to create the pseudo table to retrieve the sum of the prices spent by each customer, then join the table normally.