### What is a result set?

* 1. A result set is a set of records, including not only the data itself, but also metadata like column names, types and sizes.

### What is the difference between Union and Union All? (Union vs. Union all)

1. Union will automatically clean up duplications, while Union All will not.

2. Union will sort the result by the order of first column by default, while Union all will remain the original sequence by default.

3. Union all is faster.

4. Union cannot be used inside a recursive CTE.

### What are the other Set Operators SQL Server has?

1. INTERSECT, EXCEPT, MINUS

What is the difference between Union and Join?

Union is used to combine multiple queries into a single query with all the records of queries and the same column forms. (Vertically)

Join is used to combine data from multiple queries with the column names of all queries. The number of rows depends on the kinds of joins and the queries. (Horizontally)

### What is the difference between INNER JOIN and FULL JOIN?

> INNER JOIN will return the rows that has matched elements in both left and right query.

> FULL JOIN will return all the rows of both queries, even if there is no match.

### What is difference between left join and outer join?

> Left join is a kind of outer join. Left outer join will return all rows from left query, even if there are unmatched data.

> There are another two kinds of outer joins, which are right join and full join.

### What is cross join?

> Cross join returns the product of query multiplication, which represents all the combination of queries elements.

### Join:

> 1. \*\*inner join\*\* : will fetch the data from both right and left table which will satisfy the join condition.

> 2. \*\*left outer join\*\* : it will bring all the records from the left table but only those records from the right table which will satisfy the join condition. for non-matching records right table will return null value.

> 3. \*\*right join\*\* : it will bring all the records from the right table but only those records from the left.

> 4. \*\*Self Join\*\* : Self Join is used to join a table to itself after temporally renaming. There's no key word like SELF JOIN. self join is achieved by key word like WHERE.

### What is the difference between WHERE clause and HAVING clause?

> WHERE could only work on columns that already exist in original queries, after FROM and before key words like GROUP BY.

> Having could be used on columns that don't exist in original queries, but created during the former processes. Having shall be written after GROUP BY (all in specific sequence).

> When the query has a GROUP BY clause, WHERE can only be used before GROUP BY, while HAVING can only be used after GROUP BY.

> We cannot use HAVING without GROUP BY. However, we can use WHERE without GROUP BY.

### Can there be multiple group by columns?

> Yes, there could be. If there are multiple columns following GROUP BY, SQL will put the rows with the same values in all those columns in the same group.

### CTE benefits

> 1. CTE can be used to create a recursive query.

> 2. CTE does not store the definition in metadata.

> 3. CTE improves readability and manageability of complex SQL statements.

### CTE vs. View

> 1. CTE is the substitute for a View when the general use of a view is not required.

> 2. CTE does not store the definition in metadata, while a view stores the definition in metadata.

### Unique constraint vs. Primary key

> 1. Unique constraint allows one null value, but primary key does not.

> 2. In a table only one primary key is allowed but multiple unique constraints

> 3. Primary key will sort the data in asc order by default, but unique key does not do that.

> 4. Primary key by default creates the clustered index but unique create non clustered index.

### Transactions

> 1. When transaction1 allows transaction2 to read the uncommitted data and after that transaction1 rollbacks then dirty reads happens

> 2. when transaction1 and transaction2 read and modify the same data but transaction2 finishes it work before transaction1 then lost update happens

>> 1. read uncommitted and read committed isolation level.

> 3. non-repeatable read concurrency problem

>> 1. transaction1 is reading the same data twice but in between transaction2 updates the data so in both reads by transaction1 we will get different results.

> 4. Phantom read:

>> 1. when transaction1 reads the same data twice but transaction2 inserts new data

### View

> 1. \*\*benefits\*\*

>> 1. it can make complex queries easy.

>> 2. it can give different result set of the same table.

> 2. \*\*Disadvantages\*\*

>> 1. it cannot accept the parameters (which can cause SQL injection)

>> 2. it cannot be recursive.

>> 3. modifying data using view does not give the desired results always if there are multiple base tables.

### Trigger

> 1. Special Tables: inserted, deleted.

### Index

> 1. Clustered index is created automatically when a primary key is created, non-clustered index is created when a unique constraint is applied.

> 2. A table can have only one clustered index, but it can have multiple non-clustered-255.

> 3. A clustered index will by default sort the data in a physical order, but non-clustered index cannot sort the data.

> 4. \*\*When to use index?\*\*

>> 1. when you have multiple rows (millions) and you need to fetch up to 5%-10%

>> 2. Create index on a column which is frequently used in the where clause.

>> 3. Create index on a column which can contain multiple null values.

>> 4. Create index on column with foreign key relationship (those columns which participates the join condition)

### Value type vs. Reference type

1. Value type holds the data within its own memory allocation, but a reference type contains a pointer to another memory location that holds the data.

2. Value type is stored in the stack while reference type is stored in the heap.

### Difference between overriding and hiding

1. Overriding only redefines the implementation of the method, while hiding can completely redefine the method.
2. In overriding, when base class reference variable pointing to the object of the derived class, then it will call the overridden method in the derived class, while in hiding, when base class reference variable pointing to the object of the derived class, then it will call the hidden method in the base class.
3. Overriding is an object type, while hiding is a reference type.

### Abstract class vs. Interface

1. A class can only extend one abstract class but can implement multiple interfaces.
2. Abstract Class can have an access modifier, while interface does not have access modifiers.
3. Abstract class can have class members like private or protected, while interfaces are public by default.
4. Abstract Class can have data fields, while interface cannot contain data fields.
5. An abstract class can give complete, default code which should be overridden, while interface cannot provide any implemented code.
6. An abstract class defines the identity of a class, while Interfaces help to define the peripheral abilities of a class.
7. An abstract class can declare constructors and destructors, while an interface cannot declare constructors or destructors.
8. Abstract class can have abstract and non-abstract methods, while Interface can have only abstract methods.