

Set Operators in MS-SQL Server



Soyeb Ghachi

Technical Trainer | Mentor | SME

Overview



SET OPERATORS

- UNION
- UNION ALL
- INTERSECT
- EXCEPT

SET Operators

What is SET Operator?

Set operators in SQL Server are used to combine the result sets of multiple queries into a single result set based on set operations like union, intersect, and except. These operators allow you to perform operations on sets of rows returned by queries and retrieve the desired results.

The Set operators are UNION, UNION ALL, INTERSECT and EXCEPT

- **UNION**

SQL Server UNION is one of the set operations that allows you to combine results of two SELECT statements into a single result set which includes all the rows that belongs to the SELECT statements in the union.

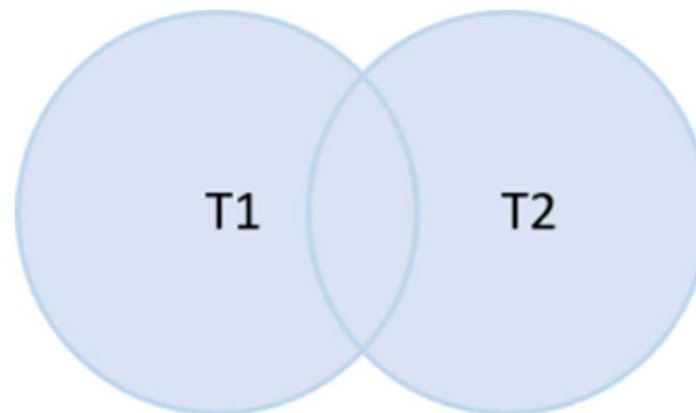
The following illustrates the syntax of the SQL Server UNION:

```
1 query_1  
2 UNION  
3 query_2
```

The following are requirements for the queries in the syntax above:

- The number and the order of the columns must be the same in both queries.
- The data types of the corresponding columns must be the same or compatible.

The following Venn diagram illustrates how the result set of the T1 table unions with the result set of the T2 table:



The following example combines names of staff and customers into a single list:

```
SELECT
    first_name,
    last_name
FROM
    staffs
UNION
SELECT
    first_name,
    last_name
FROM
    customers;
```

first_name	last_name
Aaron	Knapp
Abbey	Pugh
Abby	Gamble
Abram	Copeland
Adam	Henderson
Adam	Thomton
Addie	Hahn
Adelaida	Hancock
Adelle	Larsen
Adena	Blake
Adrien	Hunter
Adriene	Rivera

- **UNION ALL**

By default, the UNION operator removes all duplicate rows from the result sets. However, if you want to retain the duplicate rows, you need to specify the ALL keyword explicitly as shown below:

```
query_1  
UNION ALL  
query_2
```


The following example combines names of staff and customers into a single list:

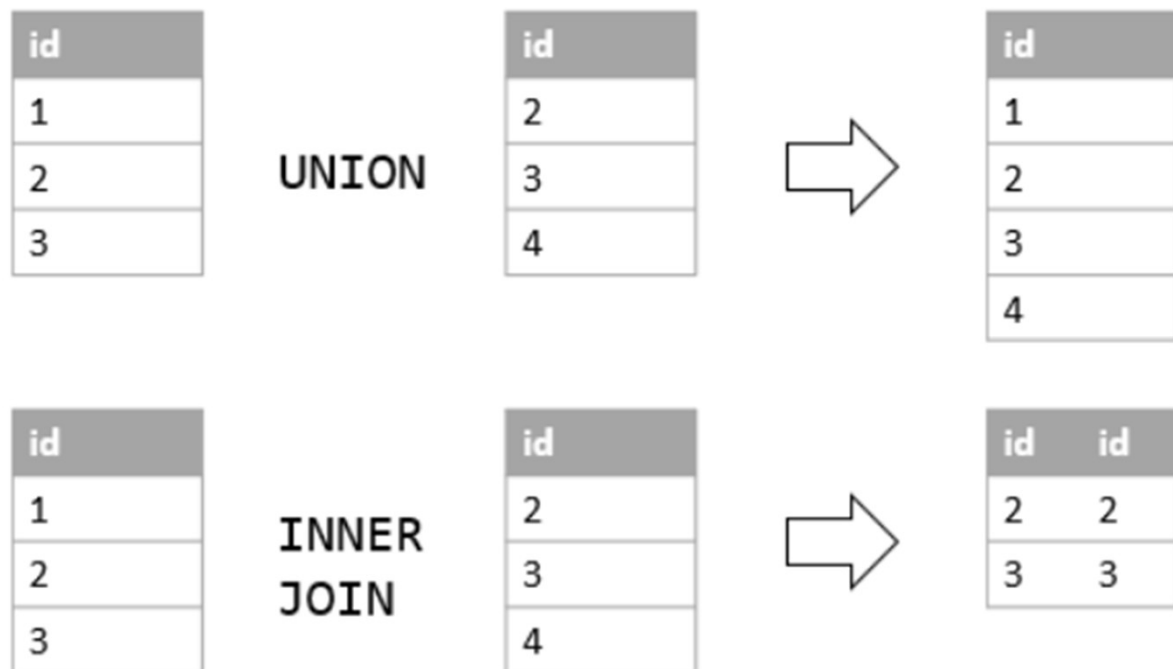
```
SELECT
    first_name,
    last_name
FROM
    staffs
UNION ALL
SELECT
    first_name,
    last_name
FROM
    customers;
```

first_name	last_name
Aaron	Knapp
Abbey	Pugh
Abby	Gamble
Abram	Copeland
Adam	Henderson
Adam	Thomton
Addie	Hahn
Adelaida	Hancock
Adelle	Larsen
Adena	Blake
Adrien	Hunter
Adriene	Rivera

UNION vs. JOIN

The join such as INNER JOIN or LEFT JOIN combines **columns** from two tables while the UNION combines **rows** from two queries.

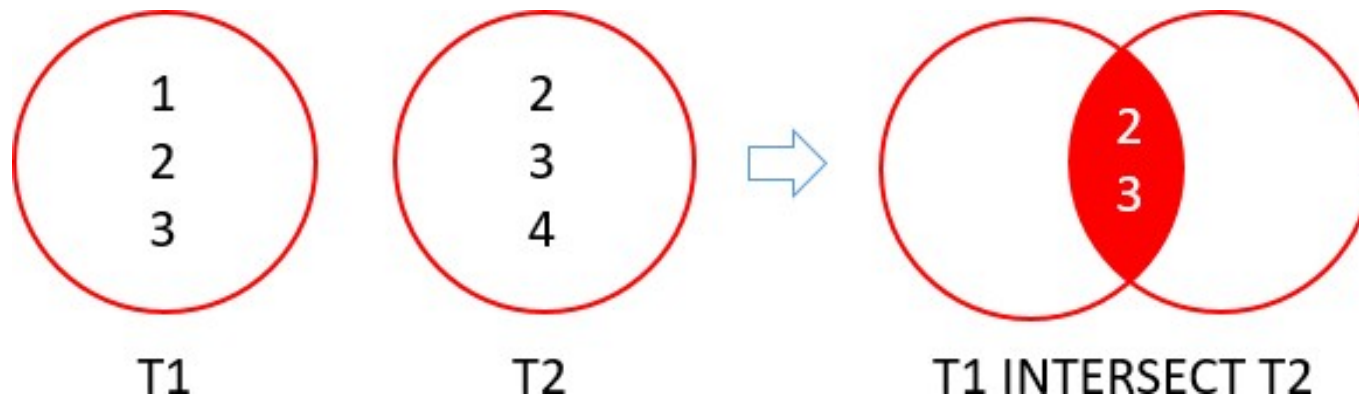
In other words, join appends the result sets horizontally while union appends result set vertically.



- **INTERSECT**

The SQL Server INTERSECT combines result sets of two or more queries and returns distinct rows that are output by both queries.

```
1 query_1  
2 INTERSECT  
3 query_2
```



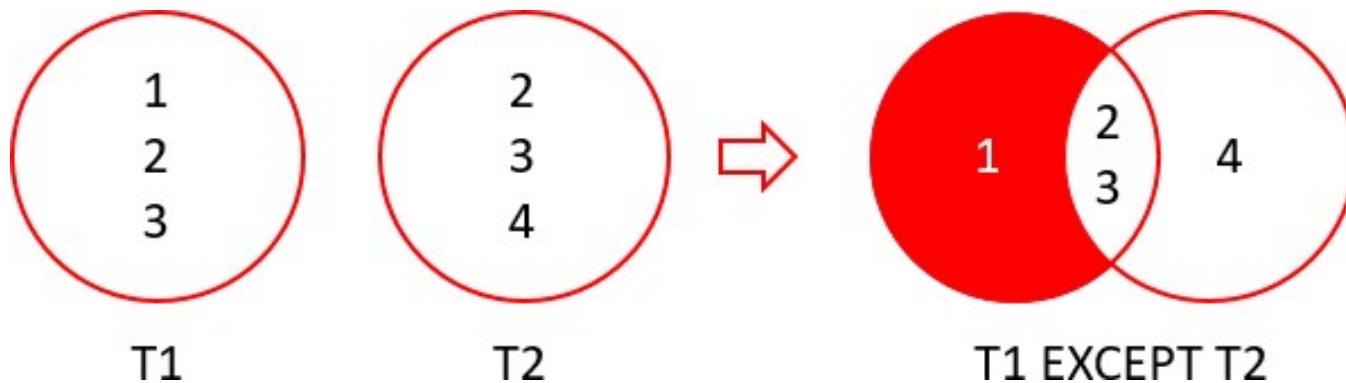
```
SELECT
    city
FROM
    customers
INTERSECT
SELECT
    city
FROM
    stores
ORDER BY
    city;
```

city
Baldwin
Rowlett
Santa Cruz

- **EXCEPT**

The SQL Server EXCEPT compares the result sets of two queries and returns the **distinct rows from the first query that are not output by the second query**. In other words, the EXCEPT subtracts the result set of a query from another.

```
1 query_1  
2 EXCEPT  
3 query_2
```



```
SELECT
    product_id
FROM
    products
EXCEPT
SELECT
    product_id
FROM
    order_items;
```

product_id
284
195
318
321
267
121
319
1
316
125
317
291
154
320

References

<https://learn.microsoft.com/en-us/sql/relational-databases/performance/joins>

https://www.w3schools.com/sql/sql_where.asp

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-where/>

<https://www.javatpoint.com/sql-server-joins>

<https://www.javatpoint.com/sql-server-joins>

<https://www.guru99.com/sql-server-joins.html>

<https://learn.microsoft.com/en-us/sql/relational-databases/performance/subqueries>