

LAB - UNION

In this lab, you will learn how to use SQL Server `UNION` to combine the results of two or more queries into a single result set.

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.

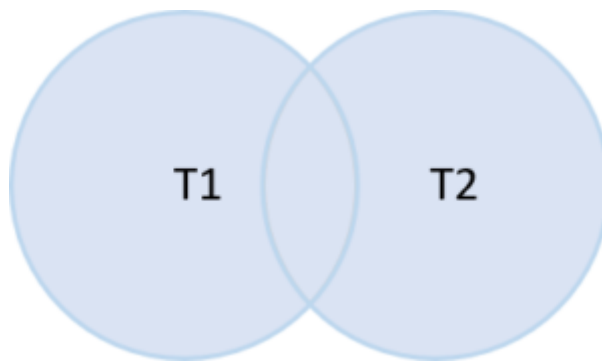
The following illustrates the syntax of the SQL Server `UNION` :

```
query_1
UNION
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:



`UNION` vs. `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
```

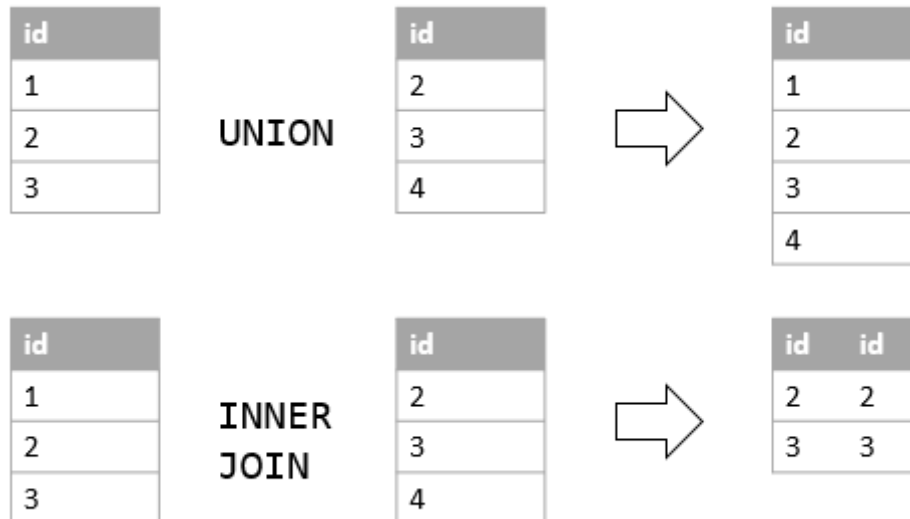
In other words, the `UNION` operator removes the duplicate rows while the `UNION ALL` operator includes the duplicate rows in the final result set.

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.

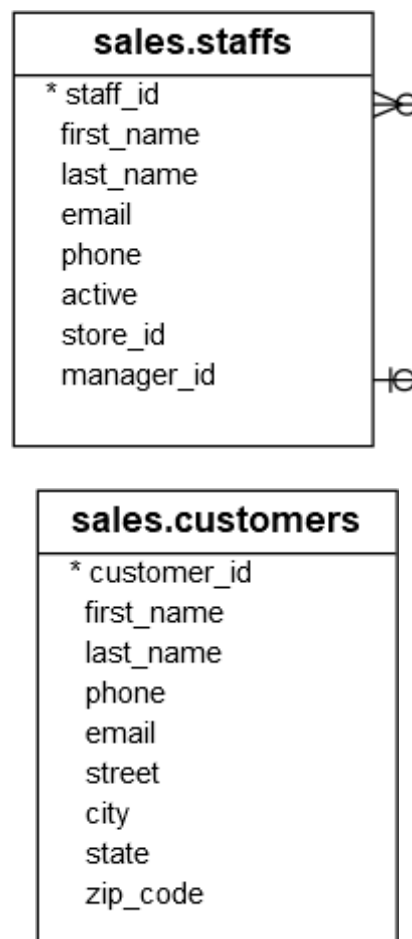
In other words, join appends the result sets horizontally while union appends result set vertically.

The following picture illustrates the main difference between `UNION` and `JOIN` :



Examples

See the following `staffs` and `customers` tables from the sample database:



UNION and UNION ALL examples

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

```
SELECT
    first_name,
    last_name
FROM
    sales.staffs
UNION
SELECT
    first_name,
    last_name
FROM
    sales.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

It returns 1,454 rows.

The `staffs` table has 10 rows and the customers table has 1,445 rows as shown in the following queries:

```
SELECT
    COUNT (*)
FROM
    sales.staffs;
-- 10

SELECT
    COUNT (*)
FROM
    sales.customers;
-- 1454
```

Because the result set of the union returns only 1,454 rows, it means that one duplicate row was removed.

To include the duplicate row, you use the `UNION ALL` as shown in the following query:

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

The query returns 1,455 rows as expected.

UNION and ORDER BY example

To sort the result set returned by the `UNION` operator, you place the `ORDER BY` clause in the last query as follows:

```
SELECT
    select_list
FROM
    table_1
UNION
SELECT
    select_list
FROM
    table_2
ORDER BY
    order_list;
```

For example, to sort the first names and last names of customers and staffs, you use the following query:

```
SELECT
    first_name,
    last_name
FROM
    sales.staffs
UNION ALL
SELECT
    first_name,
    last_name
FROM
    sales.customers
ORDER BY
    first_name,
    last_name;
```

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

In this lab, you have learned how to use the SQL Server `UNION` to combines rows from multiple queries into a single result set.