Introduction to SQL Server 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:

query\_1

UNION

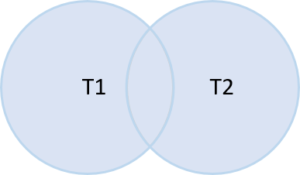
query\_2

Code language: SQL (Structured Query Language) (sql)

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

Code language: SQL (Structured Query Language) (sql)

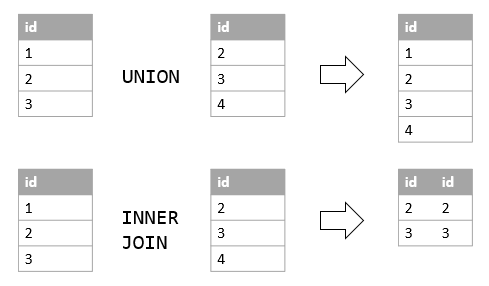
In other words, the UNION  operator removes the duplicate rows while the UNION ALL operator includes the duplicate rows in the final result set.

UNION vs. JOIN

The join such as [INNER JOIN](https://www.sqlservertutorial.net/sql-server-basics/sql-server-inner-join/) or [LEFT JOIN](https://www.sqlservertutorial.net/sql-server-basics/sql-server-left-join/) combines **columns** from two tables while the UNION combines **rows** from two queries.

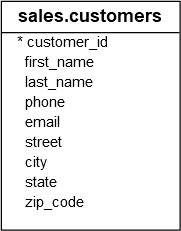
In other words, join appends the result sets horizontally while union appends result set vertically.

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



SQL Server UNION examples

See the following staffs and customers tables from the [sample database](https://www.sqlservertutorial.net/sql-server-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;

Code language: SQL (Structured Query Language) (sql)



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

Code language: SQL (Structured Query Language) (sql)

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;

Code language: SQL (Structured Query Language) (sql)

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](https://www.sqlservertutorial.net/sql-server-basics/sql-server-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;

Code language: SQL (Structured Query Language) (sql)

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;

Code language: SQL (Structured Query Language) (sql)



## Introduction to SQL Server EXCEPT

The SQL Server EXCEPT compares the result sets of two queries and returns the [distinct](https://www.sqlservertutorial.net/sql-server-basics/sql-server-select-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.

The following shows the syntax of the SQL Server EXCEPT:

query\_1

EXCEPT

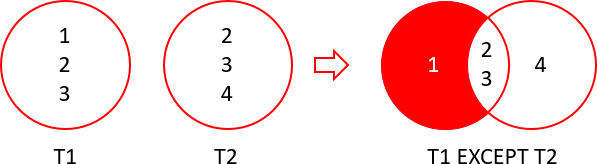
query\_2

Code language: SQL (Structured Query Language) (sql)

The following are the rules for combining the result sets of two queries in the above syntax:

* The number and order of columns must be the same in both queries.
* The [data types](https://www.sqlservertutorial.net/sql-server-basics/sql-server-data-types/) of the corresponding columns must be the same or compatible.

The following picture shows the EXCEPT operation of the two result sets T1 and T2:



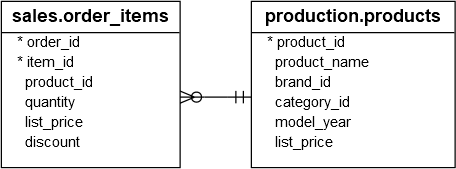
In this illustration:

* T1 result set includes 1, 2, and 3.
* T2 result set includes 2, 3, and 4.

The except of  the T1 and T2 returns 1 which is the distinct row from the T1 result set that does not appear in the T2 result set.

## SQL Server EXCEPT example

See the following products and order\_items tables from the [sample database](https://www.sqlservertutorial.net/sql-server-sample-database/):



### A) Simple EXCEPT example

The following example uses the EXCEPT operator to find the products that have no sales:

SELECT

product\_id

FROM

production.products

EXCEPT

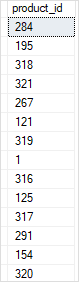
SELECT

product\_id

FROM

sales.order\_items;

Code language: SQL (Structured Query Language) (sql)



In this example, the first query returns all the products. The second query returns the products that have sales. Therefore, the result set includes only the products that have no sales.

### B) EXCEPT with ORDER BY example

To sort the result set created by the EXCEPT operator, you add the [ORDER BY](https://www.sqlservertutorial.net/sql-server-basics/sql-server-order-by/) clause in the last query. For example, the following example finds the products that had no sales and sorts the products by their id in ascending order:

SELECT

product\_id

FROM

production.products

EXCEPT

SELECT

product\_id

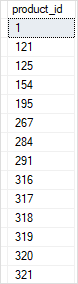
FROM

sales.order\_items

ORDER BY

product\_id;

Code language: SQL (Structured Query Language) (sql)



Introduction to SQL Server INTERSECT

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

The following illustrates the syntax of the SQL Server INTERSECT:

query\_1

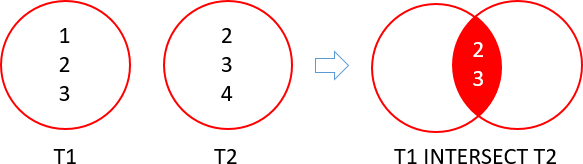
INTERSECT

query\_2

Code language: SQL (Structured Query Language) (sql)

Similar to the [UNION](https://www.sqlservertutorial.net/sql-server-basics/sql-server-union/) operator, the queries in the syntax above must conform to the following rules:

* Both queries must have the same number and order of columns.
* The data type of the corresponding columns must be the same or compatible.



The following picture illustrates the INTERSECT operation:

In this illustration, we had two result sets T1 and T2:

* T1 result set includes 1, 2, and 3.
* T2 result set includes 2, 3, and 4.

The intersection of T1 and T2 result sets returns the distinct rows which are 2 and 3.

SQL Server INTERSECT example

Consider the following query:

SELECT

city

FROM

sales.customers

INTERSECT

SELECT

city

FROM

sales.stores

ORDER BY

city;

Code language: SQL (Structured Query Language) (sql)



The first query finds all cities of the customers and the second query finds the cities of the stores. The whole query, which uses INTERSECT, returns the common cities of customers and stores, which are the cities output by both input queries.

Notice that we added the [ORDER BY](https://www.sqlservertutorial.net/sql-server-basics/sql-server-order-by/) clause to the last query to sort the result set.