SQL Server EXISTS operator overview

The EXISTS operator is a logical operator that allows you to check whether a [subquery](https://www.sqlservertutorial.net/sql-server-basics/sql-server-subquery/) returns any row. The EXISTS operator returns TRUE if the [subquery](https://www.sqlservertutorial.net/sql-server-basics/sql-server-subquery/) returns one or more rows.

The following shows the syntax of the SQL Server EXISTS operator:

EXISTS ( subquery)

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

In this syntax, the subquery is a SELECT statement only. As soon as the subquery returns rows, the EXISTS operator returns TRUE and stop processing immediately.

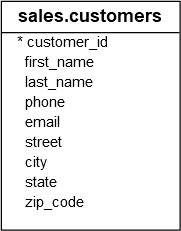
Note that even though the subquery returns a NULL value, the EXISTS operator is still evaluated to TRUE.

SQL Server EXISTS operator examples

Let’s take some examples to understand how EXISTS operator works.

A) Using EXISTS with a subquery returns NULL example

See the following customers table from the [sample database](https://www.sqlservertutorial.net/sql-server-sample-database/).



The following example returns all rows from the  customers table:

SELECT

customer\_id,

first\_name,

last\_name

FROM

sales.customers

WHERE

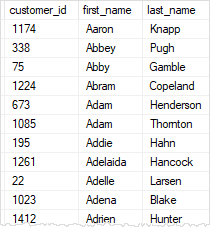
EXISTS (SELECT NULL)

ORDER BY

first\_name,

last\_name;

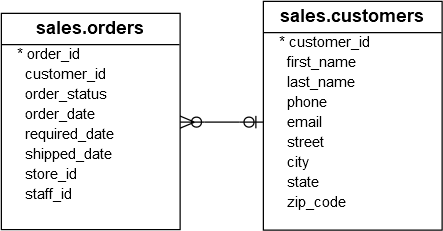
Code language: SQL (Structured Query Language) (sql)



In this example, the subquery returned a result set that contains NULL which causes the EXISTS operator to evaluate to TRUE. Therefore, the whole query returns all rows from the customers table.

B) Using EXISTS with a correlated subquery example

Consider the following customers and orders tables:



The following example finds all customers who have placed more than two orders:

SELECT

customer\_id,

first\_name,

last\_name

FROM

sales.customers c

WHERE

EXISTS (

SELECT

COUNT (\*)

FROM

sales.orders o

WHERE

customer\_id = c.customer\_id

GROUP BY

customer\_id

HAVING

COUNT (\*) > 2

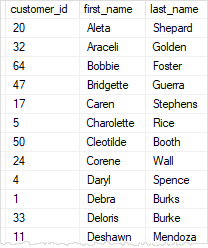
)

ORDER BY

first\_name,

last\_name;

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



In this example, we had a [correlated subquery](https://www.sqlservertutorial.net/sql-server-basics/sql-server-correlated-subquery/) that returns customers who place more than two orders.

If the number of orders placed by the customer is less than or equal to two, the subquery returns an empty result set that causes the EXISTS operator to evaluate to FALSE.

Based on the result of the EXISTS operator, the customer will be included in the result set.

C) EXISTS vs. IN example

The following statement uses the [IN](https://www.sqlservertutorial.net/sql-server-basics/sql-server-in/) operator to find the orders of the customers from San Jose:

SELECT

\*

FROM

sales.orders

WHERE

customer\_id IN (

SELECT

customer\_id

FROM

sales.customers

WHERE

city = 'San Jose'

)

ORDER BY

customer\_id,

order\_date;

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

The following statement uses the EXISTS operator that returns the same result:

SELECT

\*

FROM

sales.orders o

WHERE

EXISTS (

SELECT

customer\_id

FROM

sales.customers c

WHERE

o.customer\_id = c.customer\_id

AND city = 'San Jose'

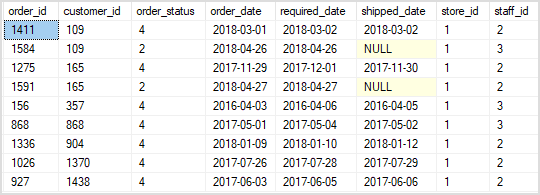
)

ORDER BY

o.customer\_id,

order\_date;

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



EXISTS vs. JOIN

The EXISTS operator returns TRUE or FALSE while the [JOIN](https://www.sqlservertutorial.net/sql-server-basics/sql-server-joins/) clause returns rows from another table.

You use the EXISTS operator to test if a subquery returns any row and short circuits as soon as it does. On the other hand, you use JOIN to extend the result set by combining it with the columns from related tables.

In practice, you use the EXISTS when you just need to check the existence of rows from related tables without returnning data from them.