Introduction to SQL Server AND operator

The AND is a logical operator that allows you to combine two Boolean expressions. It returns TRUE only when both expressions evaluate to TRUE.

The following illustrates the syntax of the AND operator:

boolean\_expression AND boolean\_expression

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

The boolean\_expression is any valid Boolean expression that evaluates to TRUE, FALSE, and UNKNOWN.

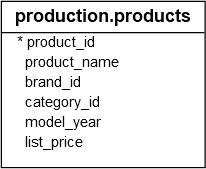
The following table shows the result when you combine TRUE, FALSE, and UNKNOWN values using the AND operator:

|  | **TRUE** | **FALSE** | **UNKNOWN** |
| --- | --- | --- | --- |
| **TRUE** | TRUE | FALSE | UNKNOWN |
| **FALSE** | FALSE | FALSE | FALSE |
| **UNKNOWN** | UNKNOWN | FALSE | UNKNOWN |

When you use more than one logical operator in an expression, SQL Server always evaluates the AND operators first. However, you can change the order of evaluation by using parentheses.

SQL Server AND operator examples

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



A) Using AND operator example

The following example finds the products where the category identification number is one and the list price is greater than 400:

SELECT

\*

FROM

production.products

WHERE

category\_id = 1

AND list\_price > 400

ORDER BY

list\_price DESC;

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

Here is the result:



B) Using multiple AND operators example

The following statement finds the products that meet all the following conditions: category id is 1, the list price is greater than 400, and the brand id is 1:

SELECT

\*

FROM

production.products

WHERE

category\_id = 1

AND list\_price > 400

AND brand\_id = 1

ORDER BY

list\_price DESC;

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

The result is as follows:

SQL Server AND multiple operators example

C) Using the AND operator with other logical operators

See the following query example:

SELECT

\*

FROM

production.products

WHERE

brand\_id = 1

OR brand\_id = 2

AND list\_price > 1000

ORDER BY

brand\_id DESC;

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

The following shows the result:



In this example, we used both [OR](https://www.sqlservertutorial.net/sql-server-basics/sql-server-or/) and AND operators in the condition. As always, SQL Server evaluated the AND operator first. Therefore, the query retrieved the products whose brand id is two and list price is greater than 1,000 or the products whose brand id is one.

To get the product whose brand id is one or two and list price is larger than 1,000, you use parentheses as follows:

SELECT

\*

FROM

production.products

WHERE

(brand\_id = 1 OR brand\_id = 2)

AND list\_price > 1000

ORDER BY

brand\_id;

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

The query returned the following result:



Introduction to SQL Server OR operator

The SQL Server OR is a logical operator that allows you to combine two Boolean expressions. It returns TRUE when either of the conditions evaluates to TRUE.

The following shows the syntax of the OR operator:

boolean\_expression OR boolean\_expression

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

In this syntax, the boolean\_expression is any valid Boolean expression that evaluates to true, false, and unknown.

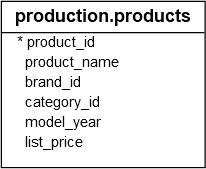
The following table shows the result of the OR operator when you combine TRUE, FALSE, and UNKNOWN:

|  | **TRUE** | **FALSE** | **UNKNOWN** |
| --- | --- | --- | --- |
| **TRUE** | TRUE | TRUE | TRUE |
| **FALSE** | TRUE | FALSE | UNKNOWN |
| **UNKNOWN** | TRUE | UNKNOWN | UNKNOWN |

When you use more than one logical operator in a statement, SQL Server evaluates the OR operators after the [AND](https://www.sqlservertutorial.net/sql-server-basics/sql-server-and/) operator. However, you can use the parentheses to change the order of evaluation.

SQL Server OR operator examples

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



A) Using OR operator example

The following example finds the products whose list price is less than 200 or greater than 6,000:

SELECT

product\_name,

list\_price

FROM

production.products

WHERE

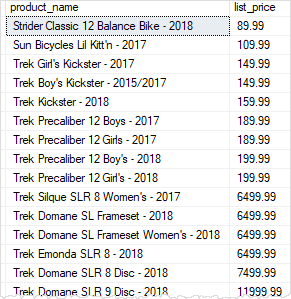
list\_price < 200

OR list\_price > 6000

ORDER BY

list\_price;

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



B) Using multiple OR operators example

The following statement finds the products whose brand id is 1, 2, or 4:

SELECT

product\_name,

brand\_id

FROM

production.products

WHERE

brand\_id = 1

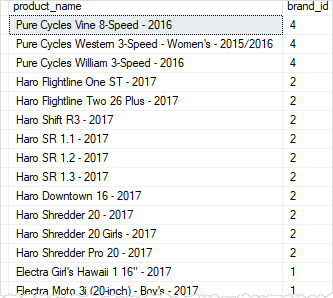
OR brand\_id = 2

OR brand\_id = 4

ORDER BY

brand\_id DESC;

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



You can replace multiple [OR](https://www.sqlservertutorial.net/sql-server-basics/sql-server-or/) operators by the [IN](https://www.sqlservertutorial.net/sql-server-basics/sql-server-in/) operator as shown in the following query:

SELECT

product\_name,

brand\_id

FROM

production.products

WHERE

brand\_id IN (1, 2, 3)

ORDER BY

brand\_id DESC;

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

C) Using OR operator with AND operator example

Consider the following example:

SELECT

product\_name,

brand\_id,

list\_price

FROM

production.products

WHERE

brand\_id = 1

OR brand\_id = 2

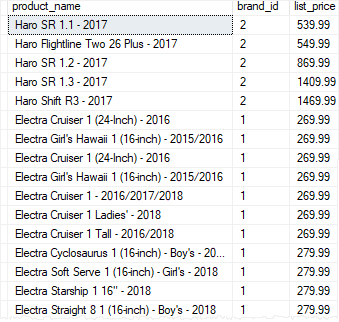
AND list\_price > 500

ORDER BY

brand\_id DESC,

list\_price;

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



In this example, we used both OR and AND operators. As always, SQL Server evaluated the AND operator first. Therefore, the query returned the products whose brand id is 2 and the list price is greater than 500 or the products whose brand id is 1.

To find the products whose brand id is 1 or 2 and list price is greater than 500, you use the parentheses as shown in the following query:

SELECT

product\_name,

brand\_id,

list\_price

FROM

production.products

WHERE

(brand\_id = 1 OR brand\_id = 2)

AND list\_price > 500

ORDER BY

brand\_id;

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

