LAB - BETWEEN Operator

In this lab, you will learn how to use the SQL Server `BETWEEN` operator to specify a range to test.

The `BETWEEN` operator is a logical operator that allows you to specify a range to test.

The following illustrates the syntax of the `BETWEEN` operator:

```
column | expression BETWEEN start_expression AND end_expression
```

In this syntax:

- First, specify the column or expression to test.
- Second, place the `start_expression` and `end_expression` between the `BETWEEN` and the `AND` keywords. The `start_expression`, `end_expression` and the `expression` to test must have the same data type.

The `BETWEEN` operator returns `TRUE` if the expression to test is greater than or equal to the value of the `start_expression` and less than or equal to the value of the `end_expression`.

You can use the greater than or equal to (>=) and less than or equal to (<=) to substitute the `BETWEEN` operator as follows:

```
column | expression <= end_expression AND column | expression >= start_expression
```

The condition that uses the `BETWEEN` operator is much more readable the one that uses the comparison operators >=, <= and the logical operator `AND`.

To negate the result of the `BETWEEN` operator, you use `NOT BETWEEN` operator as follows:

```
column | expression NOT BETWEEN start_expression AND end_expresion
```

The `NOT BETWEEN` returns `TRUE` if the value in the column or expression is less than the value of the `start_expression` and greater than the value of the `end_expression`. It is equivalent to the following condition:

```
column | expression < start_expression AND column | expression > end_expression
```

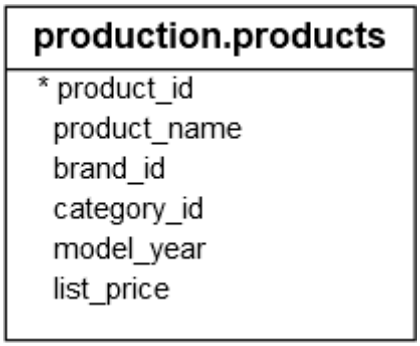
Note that if any input to the `BETWEEN` or `NOT BETWEEN` is `NULL`, then the result is `UNKNOWN`.

Examples

Let's take some examples of using the `BETWEEN` operator to understand how it works.

A) `BETWEEN` with numbers example

See the following `products` table:



The following query finds the products whose list prices are between 149.99 and 199.99:

```
SELECT
    product_id,
    product_name,
    list_price
FROM
    production.products
WHERE
    list_price BETWEEN 149.99 AND 199.99
ORDER BY
    list_price;
```

product_id	product_name	list_price
83	Trek Boy's Kickster - 2015/2017	149.99
86	Trek Girl's Kickster - 2017	149.99
268	Trek Kickster - 2018	159.99
87	Trek Precaliber 12 Boys - 2017	189.99
88	Trek Precaliber 12 Girls - 2017	189.99
267	Trek Precaliber 12 Girl's - 2018	199.99
269	Trek Precaliber 12 Boy's - 2018	199.99

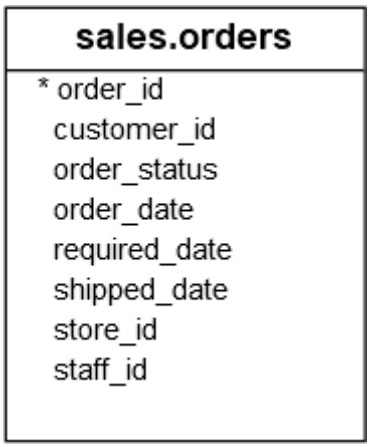
To get the products whose list prices are not in the range 149.99 and 199.99, you use the `NOT BETWEEN` operator as follows:

```
SELECT
    product_id,
    product_name,
    list_price
FROM
    production.products
WHERE
    list_price NOT BETWEEN 149.99 AND 199.99
ORDER BY
    list_price;
```

product_id	product_name	list_price
263	Strider Classic 12 Balance Bike - 2018	89.99
84	Sun Bicycles Lil Kitt'n - 2017	109.99
89	Trek Precaliber 16 Boys - 2017	209.99
90	Trek Precaliber 16 Girls - 2017	209.99
92	Haro Shredder 20 - 2017	209.99
93	Haro Shredder 20 Girls - 2017	209.99
270	Trek Precaliber 16 Boy's - 2018	209.99
271	Trek Precaliber 16 Girl's - 2018	209.99
274	Trek Precaliber 20 Boy's - 2018	229.99
275	Trek Precaliber 20 Girl's - 2018	229.99
264	Strider Sport 16 - 2018	249.99

B) `BETWEEN` with dates example

Consider the following `orders` table:



The following query finds the orders that customers placed between `January 15, 2017` and `January 17, 2017`:

```
SELECT
    order_id,
    customer_id,
    order_date,
    order_status
FROM
    sales.orders
WHERE
    order_date BETWEEN '20170115' AND '20170117'
ORDER BY
    order_date;
```

order_id	customer_id	order_date	order_status
655	347	2017-01-16	4
656	949	2017-01-16	4
657	349	2017-01-17	4
658	1051	2017-01-17	4
659	1391	2017-01-17	4

Notice that to specify a date constant, you use the format ' `YYYYMMDD` ' where `YYYY` is 4-digits year e.g., 2017, `MM` is 2-digits month e.g., 01 and `DD` is 2-digits day e.g., 15.

In this lab, you have learned how to use the SQL Server `BETWEEN` operator to form a condition that tests against a range of values.