LAB - IN Operator

In this lab, you will learn how to use the SQL Server `IN` operator to check whether a value matches any value in a list.

The `IN` operator is a logical operator that allows you to test whether a specified value matches any value in a list.

The following shows the syntax of the SQL Server `IN` operator:

```
column | expression IN ( v1, v2, v3, ...)
```

In this syntax:

- First, specify the column or expression to test.
- Second, specify a list of values to test. All the values must have the same type as the type of the column or expression.

If a value in the column or the expression is equal to any value in the list, the result of the `IN` operator is `TRUE`.

The `IN` operator is equivalent to multiple `OR` operators, therefore, the following predicates are equivalent:

```
column IN (v1, v2, v3)


column = v1 OR column = v2 OR column = v3
```

To negate the `IN` operator, you use the `NOT IN` operator as follows:

```
column | expression NOT IN ( v1, v2, v3, ...)
```

The result the `NOT IN` operator is `TRUE` if the column or expression does not equal to any value in the list.

In addition to a list of values, you can use a subquery that returns a list of values with the `IN` operator as shown below:

```
column | expression IN (subquery)
```

In this syntax, the subquery is a `SELECT` statement that returns a list of values of a single column.

Note that if a list contains `NULL` , the result of `IN` or `NOT IN` will be `UNKNOWN` .

SQL Server `IN` operator examples

See the following `production.products` table

A) Using SQL Server `IN` with a list of values example

The following statement finds the products whose list price is one of the following values: 89.99, 109.99, and 159.99:

```sql
SELECT
    product_name,
    list_price
FROM
    production.products
WHERE
    list_price IN (89.99, 109.99, 159.99)
ORDER BY
    list_price;
```

product_name	list_price
Strider Classic 12 Balance Bike - 2018	89.99
Sun Bicycles Lil Kitt'n - 2017	109.99
Trek Kickster - 2018	159.99

The query above is equivalent to the following query that uses the `OR` operator instead:

```sql
SELECT
    product_name,
    list_price
FROM
    production.products
WHERE
    list_price = 89.99 OR list_price = 109.99 OR list_price = 159.99
ORDER BY
    list_price;
```

To find the products whose list prices are not one of the prices above, you use the `NOT IN` operator as shown in the following query:

```sql
SELECT
    product_name,
    list_price
FROM
    production.products
WHERE
    list_price NOT IN (89.99, 109.99, 159.99)
ORDER BY
    list_price;
```

product_name	list_price
Trek Girl's Kickster - 2017	149.99
Trek Boy's Kickster - 2015/2017	149.99
Trek Precaliber 12 Boys - 2017	189.99
Trek Precaliber 12 Girls - 2017	189.99
Trek Precaliber 12 Girl's - 2018	199.99
Trek Precaliber 12 Boy's - 2018	199.99
Trek Precaliber 16 Boy's - 2018	209.99
Trek Precaliber 16 Girl's - 2018	209.99
Trek Precaliber 16 Boys - 2017	209.99
Trek Precaliber 16 Girls - 2017	209.99
Haro Shredder 20 - 2017	209.99
Haro Shredder 20 Girls - 2017	209.99

B) Using SQL Server `IN` operator with a subquery example

The following query returns a list of product identification numbers of the products located in the store id one and has the quantity greater than or equal to 30:

```sql
SELECT
    product_id
FROM
    production.stocks
WHERE
    store_id = 1 AND quantity >= 30;
```

product_id
30
61
64
68
106
109
188
193
219
292

You can use the query above as a subquery in as shown in the following query:

```sql
SELECT
    product_name,
    list_price
FROM
    production.products
WHERE
    product_id IN (
        SELECT
            product_id
        FROM
            production.stocks
        WHERE
            store_id = 1 AND quantity >= 30
    )
ORDER BY
    product_name;
```

product_id	product_name	list_price
30	Surly Ice Cream Truck Frameset - 2017	999.99
61	Trek Powerfly 8 FS Plus - 2017	4999.99
64	Electra Townie Original 7D - 2017	489.99
68	Sun Bicycles Cruz 3 - 2017	449.99
106	Sun Bicycles Cruz 3 - 2017	449.99
109	Sun Bicycles Cruz 7 - Women's - 2017	416.99
188	Trek XM700+ Lowstep - 2018	3499.99
193	Trek Lift+ - 2018	2799.99
219	Electra Moto 3i - 2018	639.99
292	Electra Sweet Ride 3i (20-inch) - Girls' - 2018	369.99

In this example:

- First, the subquery returned a list of product id.
- Second, the outer query retrieved the product names and list prices of the products whose product id matches any value returned by the subquery.

We will learn more about Sub Queries in a later lab.

In this lab, you have learned how to use the SQL Server `IN` operator to check whether a value matches any value in a list or returned by a subquery.