LAB - Where Clause

In this lab, you will learn how to use the SQL Server `WHERE` clause to filter rows returned by a query.

When you use the `SELECT` statement to query data against a table, you get all the rows of that table, which is unnecessary because the application may only process a set of rows at the time.

To get the rows from the table that satisfy one or more conditions, you use the `WHERE` clause as follows:

```
SELECT
    select_list
FROM
    table_name
WHERE
    search_condition;
```

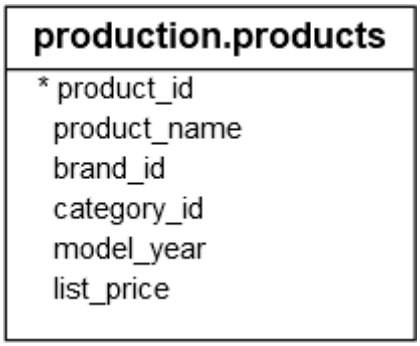
In the `WHERE` clause, you specify a search condition to filter rows returned by the `FROM` clause. The `WHERE` clause only returns the rows that cause the search condition to evaluate to `TRUE`.

The search condition is a logical expression or a combination of multiple logical expressions. In SQL, a logical expression is often called a *predicate*.

Note that SQL Server uses three-valued predicate logic where a logical expression can evaluate to `TRUE`, `FALSE`, or `UNKNOWN`. The `WHERE` clause will not return any row that causes the predicate evaluates to `FALSE` or `UNKNOWN`.

Examples

We will use the `production.products` table for the demonstration



A) Finding rows by using a simple equality

The following statement retrieves all products with the category id 1:

```sql
SELECT
    product_id,
    product_name,
    category_id,
    model_year,
    list_price
FROM
    production.products
WHERE
    category_id = 1
ORDER BY
    list_price DESC;
```

product_id	product_name	category_id	model_year	list_price
100	Electra Townie 3i EQ (20-inch) - Boys' - 2017	1	2017	489.99
98	Electra Straight 8 3i (20-inch) - Boy's - 2017	1	2017	489.99
280	Trek Superfly 24 - 2017/2018	1	2018	489.99
266	Trek Superfly 20 - 2018	1	2018	399.99
288	Electra Straight 8 1 (20-inch) - Boy's - 2018	1	2018	389.99
290	Electra Superbolt 3i 20" - 2018	1	2018	369.99
292	Electra Sweet Ride 3i (20-inch) - Girls' - 2018	1	2018	369.99
277	Trek Precaliber 24 21-speed Boy's - 2018	1	2018	369.99

B) Finding rows that meet two conditions

The following example returns products that meet two conditions: category id is 1 and the model is 2018. It uses the logical operator `AND` to combine the two conditions.

```sql
SELECT
    product_id,
    product_name,
    category_id,
    model_year,
    list_price
FROM
    production.products
WHERE
    category_id = 1 AND model_year = 2018
ORDER BY
    list_price DESC;
```

product_id	product_name	category_id	model_year	list_price
280	Trek Superfly 24 - 2017/2018	1	2018	489.99
266	Trek Superfly 20 - 2018	1	2018	399.99
288	Electra Straight 8 1 (20-inch) - Boy's - 2018	1	2018	389.99
290	Electra Superbolt 3i 20" - 2018	1	2018	369.99
292	Electra Sweet Ride 3i (20-inch) - Girls' - 2018	1	2018	369.99
294	Electra Tiger Shark 3i (20-inch) - Boys' - 2018	1	2018	369.99
296	Electra Treasure 3i 20" - 2018	1	2018	369.99
277	Trek Precaliber 24 21-speed Boy's - 2018	1	2018	369.99

C) Finding rows by using a comparison operator

The following statement finds the products whose list price is greater than 300 and model is 2018.

```
SELECT
    product_id,
    product_name,
    category_id,
    model_year,
    list_price
FROM
    production.products
WHERE
    list_price > 300 AND model_year = 2018
ORDER BY
    list_price DESC;
```

product_id	product_name	category_id	model_year	list_price
155	Trek Domane SLR 9 Disc - 2018	7	2018	11999.99
149	Trek Domane SLR 8 Disc - 2018	7	2018	7499.99
156	Trek Domane SL Frameset - 2018	7	2018	6499.99
157	Trek Domane SL Frameset Wo...	7	2018	6499.99
169	Trek Emonda SLR 8 - 2018	7	2018	6499.99
177	Trek Domane SLR 6 Disc - 2018	7	2018	5499.99
148	Trek Domane SL 8 Disc - 2018	7	2018	5499.99
154	Trek Domane SLR 6 Disc Wom...	7	2018	5499.99

D) Finding rows that meet any of two conditions

The following query finds products whose list price is greater than 3,000 or model is 2018. Any product that meets one of these conditions is included in the result set.

```sql
SELECT
    product_id,
    product_name,
    category_id,
    model_year,
    list_price
FROM
    production.products
WHERE
    list_price > 3000 OR model_year = 2018
ORDER BY
    list_price DESC;
```

product_id	product_name	category_id	model_year	list_price
155	Trek Domane SLR 9 Disc - 2018	7	2018	11999.99
149	Trek Domane SLR 8 Disc - 2018	7	2018	7499.99
156	Trek Domane SL Frameset - 2...	7	2018	6499.99
157	Trek Domane SL Frameset Wo...	7	2018	6499.99
169	Trek Emonda SLR 8 - 2018	7	2018	6499.99
51	Trek Silque SLR 8 Women's - ...	7	2017	6499.99
50	Trek Silque SLR 7 Women's - ...	7	2017	5999.99
56	Trek Domane SLR 6 Disc - 2017	7	2017	5499.99

Note that the `OR` operator was used to combine the predicates.

E) Finding rows with the value between two values

The following statement finds the products whose list prices are between 1,899 and 1,999.99:

```sql
SELECT
    product_id,
    product_name,
    category_id,
    model_year,
    list_price
FROM
    production.products
WHERE
    list_price BETWEEN 1899.00 AND 1999.99
ORDER BY
    list_price DESC;
```

product_id	product_name	category_id	model_year	list_price
57	Trek Emonda S 5 - 2017	7	2017	1999.99
317	Trek Checkpoint ALR 5 - 2019	7	2019	1999.99
318	Trek Checkpoint ALR 5 Wo...	7	2019	1999.99
128	Surly ECR 27.5 - 2018	6	2018	1899.00
161	Surly ECR - 2018	7	2018	1899.00

F) Finding rows that have a value in a list of values

The following example uses the `IN` operator to find products whose list price is 299.99 or 466.99 or 489.99.

```
SELECT
    product_id,
    product_name,
    category_id,
    model_year,
    list_price
FROM
    production.products
WHERE
    list_price IN (299.99, 369.99, 489.99)
ORDER BY
    list_price DESC;
```

product_id	product_name	category_id	model_year	list_price
64	Electra Townie Original 7D - 2017	3	2017	489.99
98	Electra Straight 8 3i (20-inch) - Boy's - 2017	1	2017	489.99
100	Electra Townie 3i EQ (20-inch) - Boys' - 2...	1	2017	489.99
102	Electra Townie Original 7D - 2017	2	2017	489.99
113	Trek Marlin 5 - 2018	6	2018	489.99
280	Trek Superfly 24 - 2017/2018	1	2018	489.99
290	Electra Superbolt 3i 20" - 2018	1	2018	369.99
292	Electra Sweet Ride 3i (20-inch) - Girls' - 2...	1	2018	369.99
294	Electra Tiger Shark 3i (20-inch) - Boys' - 2...	1	2018	369.99
296	Electra Treasure 3i 20" - 2018	1	2018	369.99
277	Trek Precaliber 24 21-speed Boy's - 2018	1	2018	369.99
278	Trek Precaliber 24 21-speed Girl's - 2018	1	2018	369.99
99	Electra Sugar Skulls 1 (20-inch) - Girl's - 2...	1	2017	299.99

G) Finding rows whose values contain a string

The following example uses the `LIKE` operator to find products whose name contains the string `Cruiser`:

```sql
SELECT
    product_id,
    product_name,
    category_id,
    model_year,
    list_price
FROM
    production.products
WHERE
    product_name LIKE '%Cruiser%'
ORDER BY
    list_price;
```

product_id	product_name	category_id	model_year	list_price
13	Electra Cruiser 1 (24-Inch) - 2016	3	2016	269.99
21	Electra Cruiser 1 (24-Inch) - 2016	1	2016	269.99
213	Electra Cruiser 1 - 2016/2017/2018	3	2018	269.99
220	Electra Cruiser 1 Ladies' - 2018	3	2018	269.99
222	Electra Cruiser 1 Tall - 2016/2018	3	2018	269.99
227	Electra Cruiser 7D (24-Inch) Ladies' - 2016/2018	3	2018	319.99
228	Electra Cruiser 7D Tall - 2016/2018	3	2018	319.99

In this lab, you have learned how to use the SQL Server `WHERE` clause to filter rows based on one or more conditions.