LAB - EXCEPT

In this lab, you will learn how to use SQL Server `EXCEPT` to subtract a result set of a query from another result set of another query.

The SQL Server `EXCEPT` compares the result sets of two queries and returns the distinct rows from the first query that are not output by the second query. In other words, the `EXCEPT` subtracts the result set of a query from another.

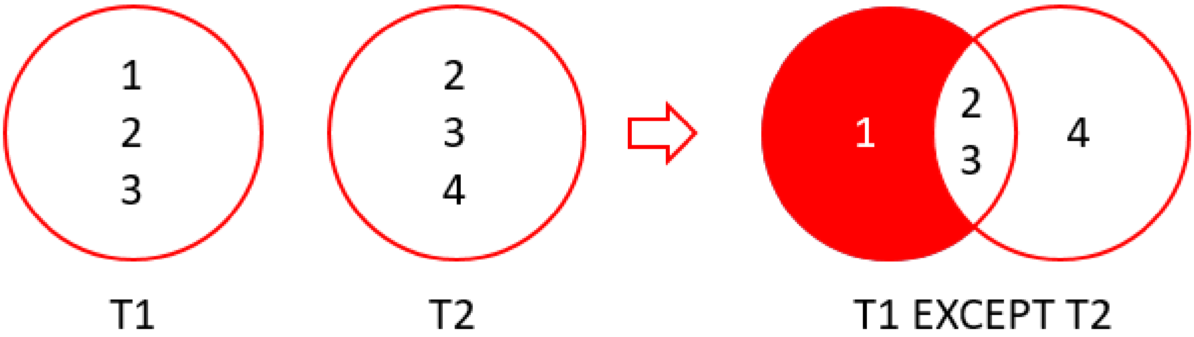
The following shows the syntax of the SQL Server `EXCEPT`:

```
query_1
EXCEPT
query_2
```

The following are the rules for combining the result sets of two queries in the above syntax:

- The number and order of columns must be the same in both queries.
- The data types of the corresponding columns must be the same or compatible.

The following picture shows the `EXCEPT` operation of the two result sets T1 and T2:



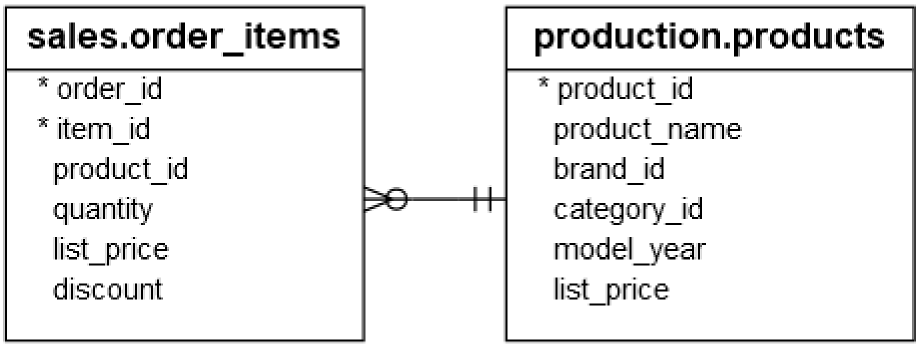
In this illustration:

- T1 result set includes 1, 2, and 3.
- T2 result set includes 2, 3, and 4.

The `except` of the T1 and T2 returns 1 which is the distinct row from the T1 result set that does not appear in the T2 result set.

SQL Server `EXCEPT` example

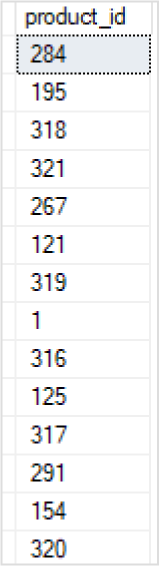
See the following `products` and `order_items` tables from the sample database:



A) Simple `EXCEPT` example

The following example uses the `EXCEPT` operator to find the products that have no sales:

```
SELECT
    product_id
FROM
    production.products
EXCEPT
SELECT
    product_id
FROM
    sales.order_items;
```



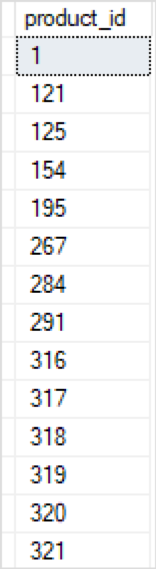
product_id
284
195
318
321
267
121
319
1
316
125
317
291
154
320

In this example, the first query returns all the products. The second query returns the products that have sales. Therefore, the result set includes only the products that have no sales.

B) `EXCEPT` with `ORDER BY` example

To sort the result set created by the `EXCEPT` operator, you add the `ORDER BY` clause in the last query. For example, the following example finds the products that had no sales and sorts the products by their id in ascending order:

```
SELECT
    product_id
FROM
    production.products
EXCEPT
SELECT
    product_id
FROM
    sales.order_items
ORDER BY
    product_id;
```



product_id
1
121
125
154
195
267
284
291
316
317
318
319
320
321

In this lab, you have learned how to use the SQL Server `EXCEPT` to combine result sets of two queries.