



SQL Standard

# SQL Outer Join

7 months ago • by John Otieno

It goes without saying that joins is one of the most identifiable features of relational databases. Joins allow us to combine the data from one or more tables based on a related condition to create a coherent data operation.

There are various types of joins in SQL, each with a unique way of how it handles the data from the participating tables or the resulting set. One of the most common type of join in SQL is an OUTER JOIN.

An OUTER JOIN in SQL retrieves all the matching rows from the involved tables as well as the unmatched rows from one or both tables. It comes in handy when you are dealing with tables that contain NULL values or missing sets.

Let us explore further what these joins do, how they work, and how we can use them in an SQL database.

## Requirements:

For this tutorial, we will work with MySQL 8.0 and use the Sakila sample database. However, feel free to use any other dataset that you deem applicable.

## Types of Outer Joins

There are three main types of OUTER JOINS in SQL. These types of OUTER JOINS include:

### 1. Left OUTER JOINS

In the case of LEFT OUTER JOINS, the join retrieves all the rows from the left table and only the matching rows from the right table. If there are no matching rows from the right table, the join returns the NULL values for the columns on the right table.

### 2. Right OUTER JOINS

This is similar to a RIGHT OUTER JOIN. However, it retrieves all the rows from the right table but only the matching rows from the left table. If there are no matching rows from the left table, the join includes the NULL values for the columns on the left table.

### 3. FULL OUTER JOINS

Lastly, we have the FULL OUTER JOINS. This type of join combines both the RIGHT and LEFT outer joins. As a result, the join retrieves all rows when there is a match in either the left or the right table. If there is no match, the join returns the NULL values for the columns from the table with no match.

## Syntax of the SQL OUTER JOIN

The following expresses the syntax of an SQL OUTER JOIN. It is however good to keep in mind that the syntax may vary slightly depending on the target database engine.

The following is a general structure:

```
SELECT columns  
FROM table1  
[LEFT | RIGHT | FULL] OUTER JOIN table2  
ON table1.column_name = table2.column_name;
```

The syntax of an OUTER JOIN in SQL is pretty self-explanatory.

## Examples:

Let us look at some sample usage on how we can apply the various types of OUTER JOINS in SQL.

As we mentioned, we will use the Sakila sample database for demonstration. In this case, we use the “customer” and “payment” tables.

### Example 1: LEFT OUTER JOIN

Let us start with an OUTER JOIN. Suppose we want to retrieve all the customer information along with their payment information, if available.

This makes a LEFT OUTER JOIN applicable as we want all customer information (on left) and payment information if available (right).

If the customer has not made any payment, the join will show the NULL values for the payment-related columns.

An example is as follows:

```
SELECT  
    c.customer_id,
```

```

        c.first_name,
        c.last_name,
        p.amount,
        p.payment_date
FROM
        customer c
LEFT OUTER JOIN payment p
ON
        c.customer_id = p.customer_id;

```

In the given query, we include the “customer\_id”, “first\_name”, and “last\_name” columns from the “customer” table. We also include the amount and “payment\_date” from the “payment” table.

We then perform a LEFT OUTER JOIN between the “customer” and “payment” tables based on the “customer\_id”.

These are all the customers (whether payment made or not) along with their payment details (if any).

An example output is as follows:

customer_id	first_name	last_name	amount	payment_date
1	MARY	SMITH	2.99	2005-05-25 11:30:37
1	MARY	SMITH	0.99	2005-05-28 10:35:23
1	MARY	SMITH	5.99	2005-06-15 00:54:12
1	MARY	SMITH	0.99	2005-06-15 18:02:53
1	MARY	SMITH	9.99	2005-06-15 21:08:46
1	MARY	SMITH	4.99	2005-06-16 15:18:57
1	MARY	SMITH	4.99	2005-06-18 08:41:48
1	MARY	SMITH	0.99	2005-06-18 13:33:59
1	MARY	SMITH	3.99	2005-06-21 06:24:45
1	MARY	SMITH	5.99	2005-07-08 03:17:05
1	MARY	SMITH	5.99	2005-07-08 07:33:56
1	MARY	SMITH	4.99	2005-07-09 13:24:07

## Example 2: RIGHT OUTER JOIN

Now, let’s move on to the RIGHT OUTER JOIN. Suppose we wish to include all payment information and the associated customer in this case, if any.

In this case, if a payment is made by a customer, the join will display that customer’s details. If there is a payment with no associated customer, it will show the NULL values for customer-related columns.

```

SELECT
        c.customer_id,
        c.first_name,
        c.last_name,
        p.amount,
        p.payment_date
FROM
        customer c

```

```
RIGHT OUTER JOIN payment p
ON
    c.customer_id = p.customer_id;
```

The resulting set is as follows:

8	SUSAN	WILSON	6.99	2005-05-30	03:43:54
8	SUSAN	WILSON	2.99	2005-06-15	09:59:16
8	SUSAN	WILSON	5.99	2005-06-17	18:21:35
8	SUSAN	WILSON	4.99	2005-06-20	17:57:47
8	SUSAN	WILSON	5.99	2005-07-05	23:01:21
8	SUSAN	WILSON	0.99	2005-07-07	00:09:02
8	SUSAN	WILSON	2.99	2005-07-07	10:02:03
8	SUSAN	WILSON	3.99	2005-07-07	21:47:29
8	SUSAN	WILSON	3.99	2005-07-08	02:17:12
8	SUSAN	WILSON	2.99	2005-07-09	15:40:46
8	SUSAN	WILSON	2.99	2005-07-09	17:13:23
8	SUSAN	WILSON	4.99	2005-07-11	21:39:46
8	SUSAN	WILSON	0.99	2005-07-12	10:43:53
8	SUSAN	WILSON	1.99	2005-07-29	21:42:49
8	SUSAN	WILSON	2.99	2005-07-31	04:54:43
8	SUSAN	WILSON	0.99	2005-07-31	22:08:29
8	SUSAN	WILSON	2.99	2005-08-01	13:05:35
8	SUSAN	WILSON	9.99	2005-08-02	13:04:12

### Example 3: FULL OUTER JOIN

A FULL OUTER JOIN, on the other hand, retrieves all customer information and payment. This includes all customers and all payments and shows the NULL values where there is no match between the tables.

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    p.amount,
    p.payment_date
FROM
    customer c
FULL OUTER JOIN payment p
ON
    c.customer_id = p.customer_id;
```

It is good to keep in mind that MySQL does not natively support a FULL OUTER JOIN. You have to do some jujitsu magic with the LEFT JOIN, UNION, and RIGHT JOIN. Quite annoying, we might add.

### Conclusion

In this tutorial, we learned all about OUTER JOINS. We learned what is an OUTER JOIN in SQL, the types of OUTER JOINS, and the examples of how to use these types of OUTER JOINS.