

Outer Joins

by Sophia

WHAT'S COVERED

This tutorial explores full outer joins to query data from two or more tables in two parts:

1. Introduction
2. Example

1. Introduction

There are instances when we want to look at data that matches between two or more tables. There are also instances when we want to return data that does not match. We can look for data that does not exist in the left table, or for data that does not exist in the right table. In this tutorial, we will focus on data that does not match in either table.

2. Example

Let's come back to our initial dataset with the representative and department tables to help illustrate the outer joins:

```
CREATE TABLE representative ( representative_id INT PRIMARY KEY, first_name VARCHAR (30) NOT NULL, last_name VARCHAR (30) NOT NULL );
```

```
CREATE TABLE department ( department_id INT PRIMARY KEY, department_name VARCHAR (100) NOT NULL, manager_id INT, constraint fk_manager FOREIGN KEY (manager_id)
```

```
INSERT INTO representative (representative_id, first_name, last_name)
VALUES (1, 'Bob','Evans'), (2, 'Tango','Rushmore'), (3, 'Danika','Arkane'), (4, 'Mac','Anderson');
```

```
INSERT INTO department (department_id, department_name, manager_id)
VALUES (1, 'Sales', 1), (2, 'Marketing', 3), (3, 'IT', 4), (4, 'Finance', null), (5, 'Support', null);
```

There are multiple types of outer joins, including the full outer join, as well as the left and right outer joins, which we will discuss in forthcoming tutorials.

The full outer join, or full join, returns the data from all rows in both the left and right tables. If they match, it will return data from both sides. If they don't match, the columns of the table will be filled with NULL.

Let's first look at the structure of the statement:

```
SELECT <columnlist>
FROM <table1>
FULL OUTER JOIN <table2> ON <table1>.<table1column1> = <table2>.<table2column1>;
```

Notice that this is very similar to a regular JOIN ON statement, with only the FULL OUTER added to the JOIN. Using our two tables, we can run the full outer join like this:

```
SELECT *
FROM representative
FULL OUTER JOIN department ON representative.representative_id = department.manager_id;
```

This will return the three records that match the representative_id from the representative table and the department_id from the department table. It will also return the rows from the department table that do not have a matching row in the representative table (4th and 5th row in the result set). It will also return the rows from the representative table that do not have a match in the department table (6th row):

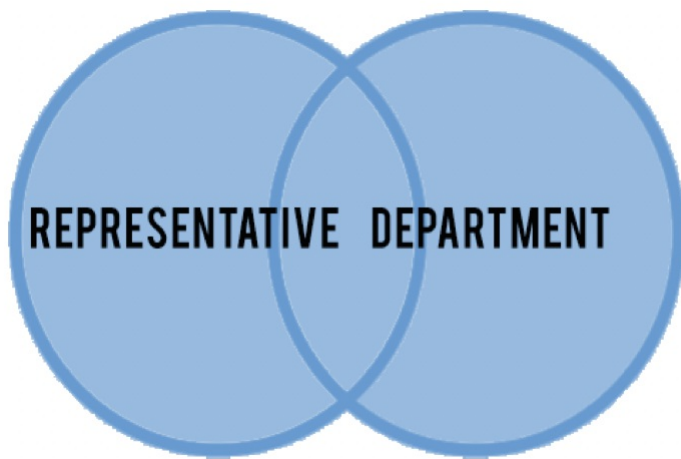
Query Results

Row count: 6

representative_id	first_name	last_name	department_id	department_name	manager_id
1	Bob	Evans	1	Sales	1
3	Danika	Arkane	2	Marketing	3
4	Mac	Anderson	3	IT	4
			4	Finance	
			5	Support	
2	Tango	Rushmore			

This is an important point. If every row in both tables matches one another, the full outer join would have the same result set.

The following Venn diagram shows what the data would return:



Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.



The full outer join will return the rows that match between two tables, as well as the rows from each individual table that do not have matches in the other table.

Source: Authored by Vincent Tran