

Right Joins

by Sophia

WHAT'S COVERED

This tutorial explores using the RIGHT JOIN and RIGHT OUTER JOIN to query data from tables in two parts:

1. Introduction
2. Examples

1. Introduction

Similar to the left join and left outer join, there is also a right join and a right outer join. They work in exactly the same way, except they start by getting data from the right table and compare it to the left table. If the values are equal, the right join creates a new row that contains columns from both tables. If they are not equal, the right join will fill the column from the left table with NULL.

The structure of the query looks like this:

```
SELECT <columnlist>
FROM <table1>
RIGHT JOIN <table2>
ON <table1>.<table1column1> = <table2>.<table2column1>
```

2. Examples

Let's again revisit our data set with the representatives and departments:

```
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);
```

Let's see what they would look like with the RIGHT JOIN:

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

Query Results

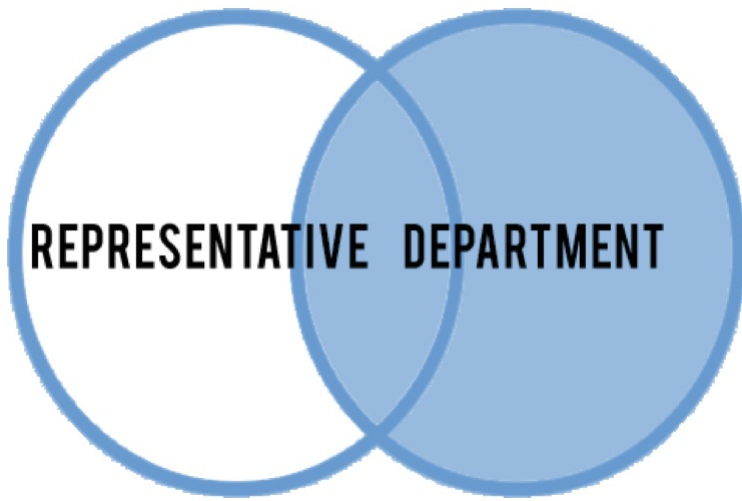
Row count: 5

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	

The query itself isn't that different than the left join, other than specifying RIGHT instead of LEFT. Notice that the last two rows are from the department table, where there is not a match with the representative table.

Recall that the table listed in the FROM clause (representative) is considered to be the left table and table after the JOIN clause (department) is considered to be the right table. The right join starts by selecting data from the right table (department). It compares the manager_id from the department table to the representative_id in the representative table. If those values are equal, the right join creates a row that contains the columns of both tables and adds the new row in the result set. You can see that in the first three rows in the table. If the values are not equal, the right join also creates a row containing the columns from both of the tables, but fills the columns from the left table (representative) with null values.

The Venn diagram for the right join looks like the following:



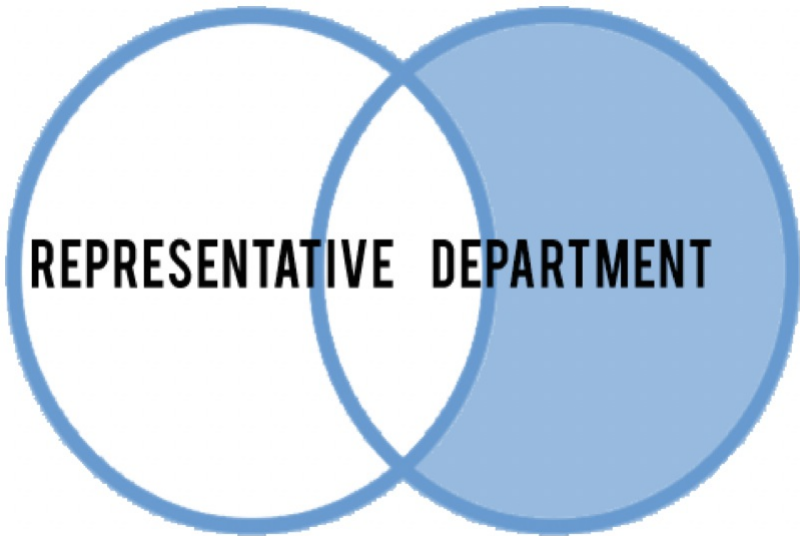
The right outer join just retrieves the rows from the right table that do not match any of the rows in the left table.

```
SELECT *
FROM representative
RIGHT JOIN department ON representative.representative_id = department.manager_id
WHERE manager_id IS NULL;
```

Query Results
 Row count: 2

representative_id	first_name	last_name	department_id	department_name	manager_id
			5	Support	
			4	Finance	

The following Venn diagram illustrates the right outer join:



Video Transcription

[MUSIC PLAYING] With the left outer join, we looked at data that's in the left table, but not in the right table. However, we can also do the opposite approach to be able to set up a right join, by simply changing left to right in this case here, with the type of join. And once we run this, we'll be able to display the data that matches between the two tables, as well as data that's in the right table, but doesn't exist in the left table.

You might also think that you can easily changes to left join by simply replacing the two way tables to have department first and representative. And you'd be correct. They'd be an easy way to make that switch as well. In this case here, we can also do a right outer join by adding on a WHERE clause, by identifying where that representative ID is now, meaning that there is no value that's associated with it. So in this case here, we have those two departments that doesn't have any manager ID.

[MUSIC PLAYING]



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.



SUMMARY

The right join and right outer join will query data from two tables and return the matching data as well as the non-matching data from the right table.

Source: Authored by Vincent Tran