LEFT JOINs

by Sophia

WHAT'S COVERED

In this lesson, you will explore a LEFT JOIN, in two parts. Specifically, this lesson will cover:

1. Understanding LEFT JOINs
2. Examples

1. Understanding LEFT JOINs

A **LEFT JOIN** combines data from two tables based on a specified condition. The result set includes all the records from the left (or first) table, plus matching records from the right (or second table. Any unmatched rows from the left table will have NULL values. Using a LEFT JOIN, you can keep the context of all the left-hand records while incorporating relevant data from the right-hand table.

The structure of the query looks like this:

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

TERM TO KNOW

LEFT JOIN
A clause that combines data from two tables based on a specified condition. It retrieves all records from the left (or first) table and matching records from the right (or second) table. In the result set, any unmatched rows from the left table will have NULL values.

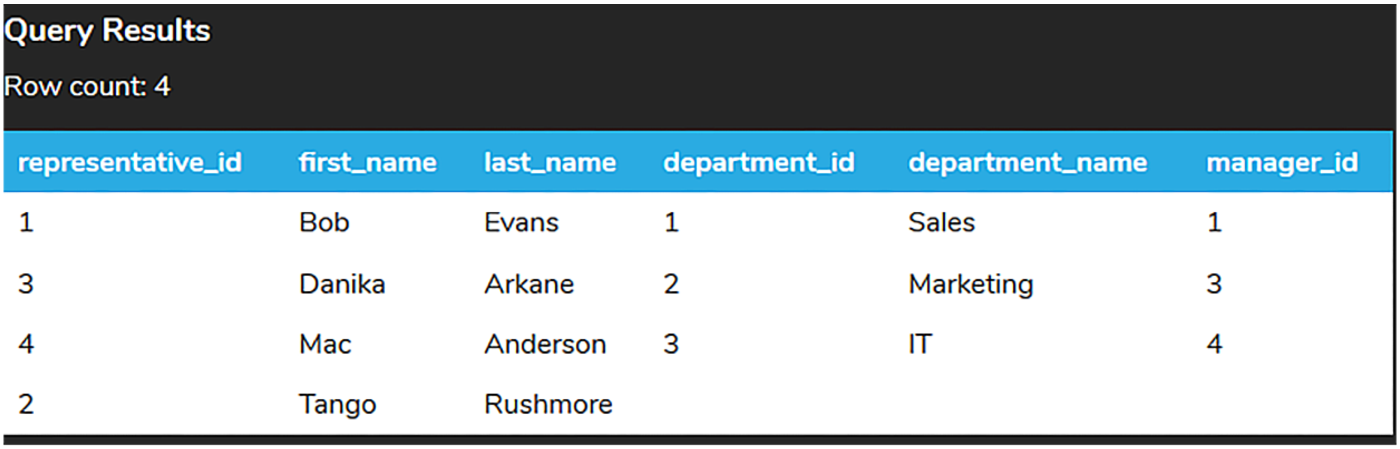
2. Examples

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

```
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) REFERENCES representative(representative_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);
```

The LEFT JOIN clause is used to join the representative table with the department table. The first table listed in the FROM clause is considered to be the left table, and the second table is considered to be the right table.

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

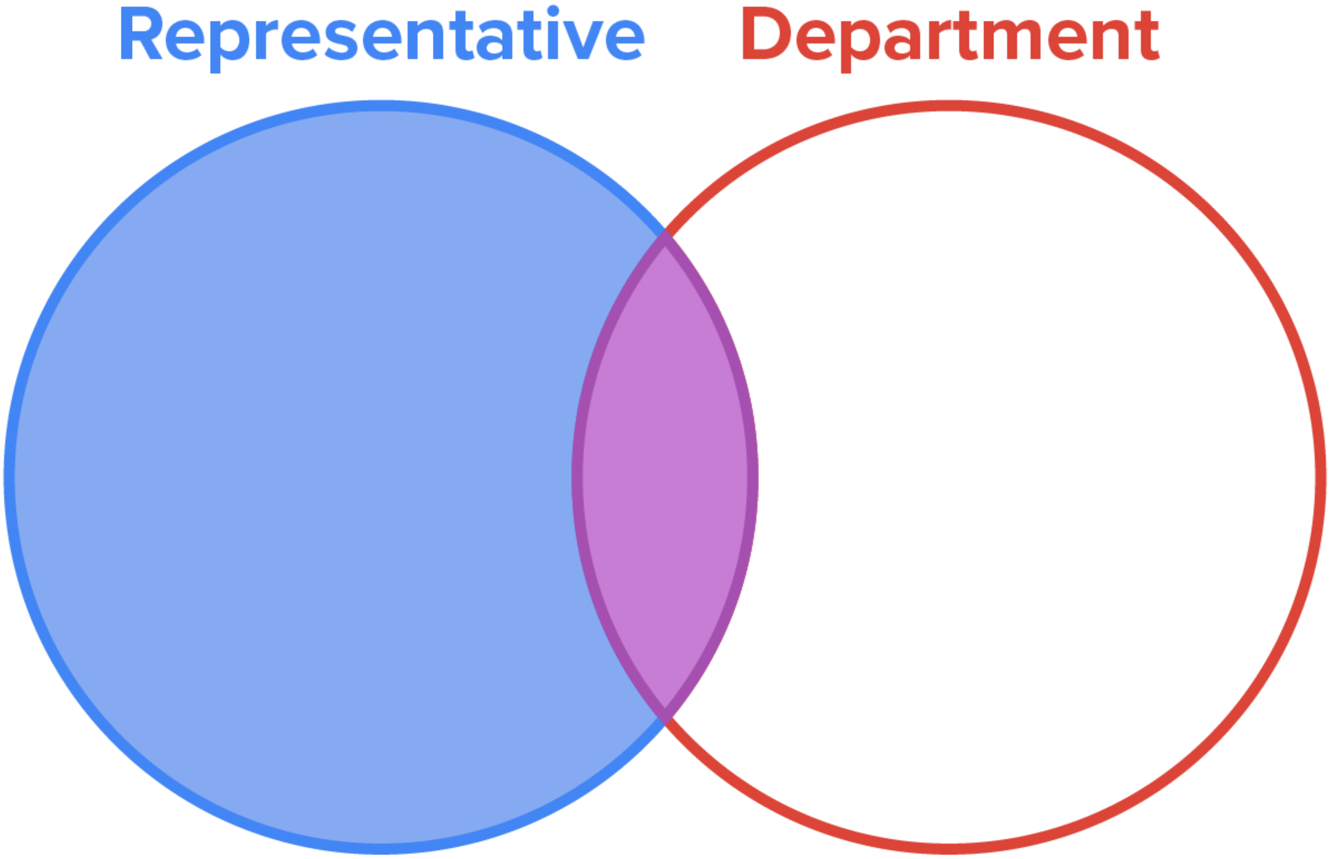


Query Results

Row count: 4

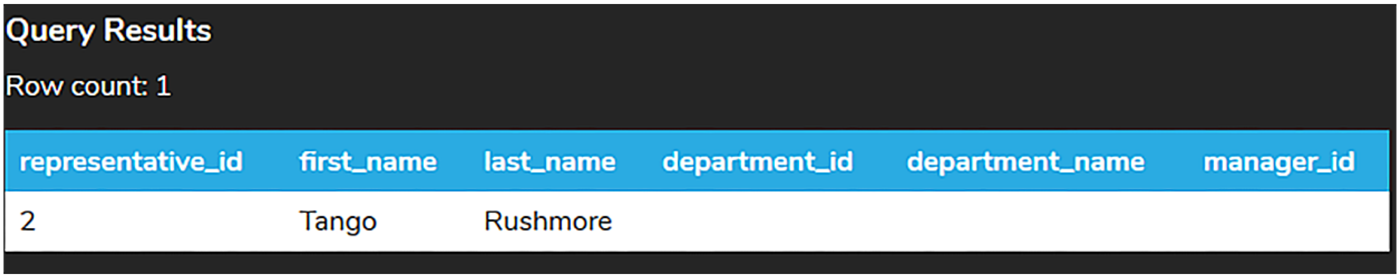
representative_id	first_name	last_name	department_id	department_name	manager_id
1	Bob	Evans	1	Sales	1
3	Danika	Arkane	2	Marketing	3
4	Mac	Anderson	3	IT	4
2	Tango	Rushmore			

The LEFT JOIN starts to select data from the left table. It compares the representative_id from the representative table with the manager_id in the department table. If those values are equal, the LEFT JOIN creates a new row that contains the columns of both tables and adds the new row in the result set, as you can see in the first three rows returned in the image above. If the values are not equal, the LEFT JOIN also creates a new row containing columns from both tables but fills in the columns of the right table (department) with a NULL value, as you can see in the 4th row. The Venn diagram of the LEFT JOIN looks like the following:



We can optionally add a WHERE clause to find only records from the left-hand table that do not have a matching value in the right-hand table. For example, you could use the following query to return only records where manager_id is NULL—in other words, where there is no matching manager_id in the right-hand table:

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

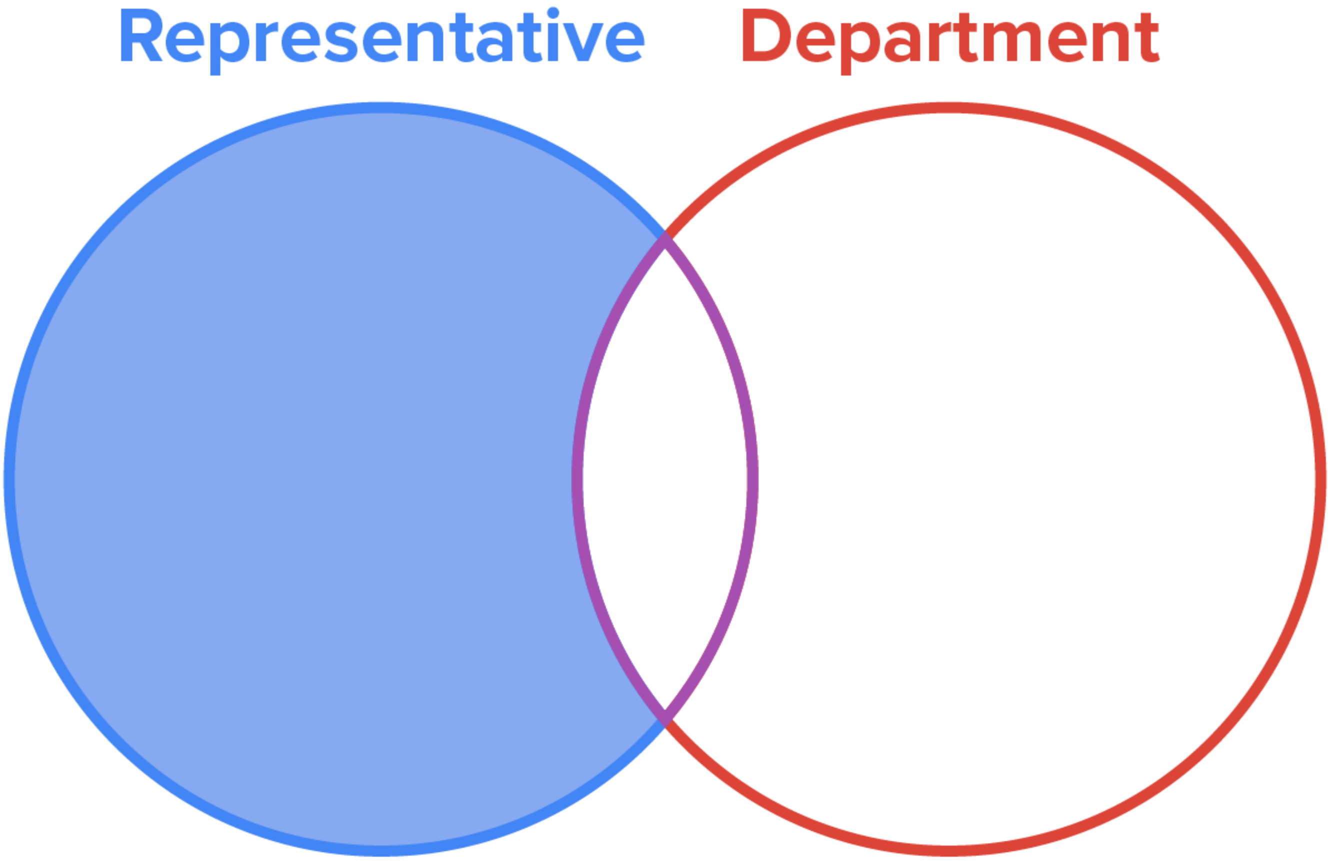


Query Results

Row count: 1

representative_id	first_name	last_name	department_id	department_name	manager_id
2	Tango	Rushmore			

This lesson's video references this type of JOIN as a "LEFT OUTER JOIN," but don't be confused by that. There is no separate LEFT OUTER JOIN clause in SQL (although SQL will accept that syntax; it will just process it as a regular LEFT JOIN). The type of JOIN shown in the above query, and in the Venn diagram below, is accomplished by starting with LEFT JOIN and adding the WHERE clause, not by using a different JOIN clause.



|alt=VENN Diagram Left Join Representative, First name, Last name, Department ID, Department name, Manager ID]]

WATCH

TRY IT

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

In this lesson, you learned how to use a **LEFT JOIN** to retrieve all the records from the left table and only the records from the right table that match them. This type of query can help analyze data based on a specific condition. For example, an employee table and a department table could be joined with a LEFT JOIN to retrieve data about all employees and their associated departments, including employees who do not yet have a department assigned.

In the **examples** section, you learned the syntax for two types of LEFT JOINs. In the first example, a LEFT JOIN is used to include all records from the left-hand table and the data from any corresponding records from the right-hand table. In the second example, you learned how to add a WHERE clause to show only records from the left-hand table that have no corresponding records on the right. This is sometimes called a LEFT OUTER JOIN, and it is referenced as such in this lesson's video.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND FAITHE WEMPEN (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.

TERMS TO KNOW

LEFT JOIN

A clause that combines data from two tables based on a specified condition. It retrieves all records from the left (or first) table and matching records from the right (or second) table. In the result set, any unmatched rows from the left table will have NULL values.