SQL Joins Explained

WHAT IS A SQL JOIN?   SQL JOIN TYPES   PREPARING DATA
DATA SOURCES

What is a SQL join?

A SQL join is a Structured Query Language (**SQL**) instruction to combine data from two sets of data (e.g. two tables). Before we dive into the details of a SQL join, let's briefly discuss what SQL is, and why someone would want to perform a SQL join.

SQL is a special-purpose programming language designed for managing information in a relational database management system (**RDBMS**). The word relational here is key; it specifies that the database management system is organized in such a way that there are clear relations defined between different sets of data.

Typically, you need to extract, transform, and load data into your RDBMS before you're able to manage it using SQL, which you can accomplish by using a tool like Stitch.

Want to learn about setting the data strategy for your organization?

Signup for a free 30 day course to learn what you need in order to succeed with data. We have worked with over 500 companies of all sizes and helped them build their data infrastructure, run analytics, and make data-driven decisions. Learn how the data landscape has changed and what that means for your company.

Email Address Type your email address Get The Course

We will never share your email

Relational Database Example

Imagine you’re running a store and would like to record information about your customers and their orders. By using a relational database, you can save this information as two tables that represent two distinct entities: customers and orders.

Customers

customer_id	first_name	last_name	email	address	city	state	zip
1	George	Washington	gwashington@usa.gov	3200 Mt Vernon Hwy	Mount Vernon	VA	22121
2	John	Adams	jadams@usa.gov	1250 Hancock St	Quincy	MA	02169
3	Thomas	Jefferson	tjefferson@usa.gov	931 Thomas Jefferson Pkwy	Charlottesville	VA	22902
4	James	Madison	jmadison@usa.gov	11350 Constitution Hwy	Orange	VA	22960
5	James	Monroe	jmonroe@usa.gov	2050 James Monroe Parkway	Charlottesville	VA	22902

Here, information about each customer is stored in its own row, with columns specifying different bits of information, like their first name, last name, and email address. Additionally, we associate a unique customer number, or primary key, with each customer record.

Orders

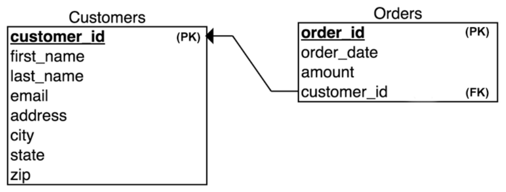
order_id	order_date	amount	customer_id
1	07/04/1776	$234.56	1
2	03/14/1760	$78.50	3

3	05/23/1784	$124.00	2
4	09/03/1790	$65.50	3
5	07/21/1795	$25.50	10
6	11/27/1787	$14.40	9

Again, each row contains information about a specific order. Each order has its own unique identification key `order_id` assigned to it as well.

Relational Model

You've probably noticed that these two examples share similar information. You can see these simple relations diagrammed below:



Note that the orders table contains two keys: one for the order and one for the customer who placed that order. In scenarios when there are multiple keys in a table, the key that refers to the entity being described in that table is called the **Primary Key** (PK) and other key is called a **Foreign Key** (FK).

In our example, `order_id` is a primary key in the orders table, while `customer_id` is both a primary key in the customers table, and a foreign key in the orders table. Primary and foreign keys are essential to describing relations between the tables, and in performing SQL joins.

SQL Join Example

Let's say we want to find all orders placed by a particular customer. We can do this by joining the customers and orders tables together using the relationship established by the `customer_id` key:

```
select order_date, order_amount
```

```
from customers
join orders
   on customers.customer_id = orders.customer_id
where customer_id = 3
```

all-orders-placed-by-a-customer hosted with ♥ by GitHub view raw

Here, we're joining the two tables using the `join` keyword, and specifying what key to use when joining the tables in the `on customers.customer_id = orders.customer_id` line following the join statement. Here is the result of the above SQL query, which includes two orders placed by Thomas Jefferson (customer_id = 3):

order_id	order_date	order_amount
2	3/14/1760	$78.50
4	9/03/1790	$65.50

This particular join is an example of an "inner" join. Depending on the kind of analysis you'd like to perform, you may want to use a different method. There are actually a number of different ways to join the two tables together, depending on your application. The next section will explain inner, left, right, and full joins, and provide examples using the data tables used above.

NEXT PAGE →

14-day free trial | setup in minutes | no ETL scripts necessary