**SQL Concepts that Beginners Have Trouble With**

How do I Write SQL Queries from Scratch When There Are No Hints?



Photo by [Elena Mozhvilo](https://unsplash.com/es/@miracleday?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral)

As a mentor for an online analytics bootcamp, I‘ve had many mentees ask me “how do I come up with SQL queries in the real job world when there are no hints provided like in my coursework?”

Many websites and courses will teach you how to use SQL, but they don’t teach you how to think through creating SQL queries from scratch. That is, how do you know what the SQL query should look like in real life when there is no handholding, no hints provided by the course.

This article is not meant to teach you SQL, but to provide clarifications for SQL beginners so that you can get started with writing your own queries.

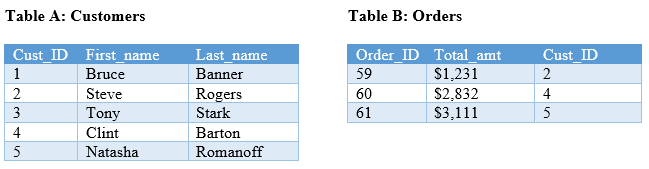
In this article, I will attempt to clear up the confusion with regards to these specific questions that are commonly asked by my mentees.

* How do I know whether to use a left join, right join, or inner join?
* Does it matter which table is on the left and which table is on the right when I do a join?
* Does it matter which column I join on?
* In a real job environment, how do I know how my SQL query should look like?
* What do I do if I’m struggling with SQL?

Again, the purpose of this article is not to teach you how to write SQL, so I assume you know what SQL is and the basics of SQL.

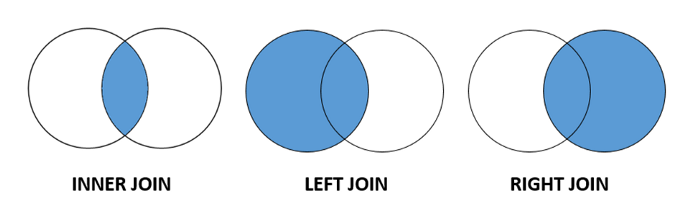
**How do I know whether to use a left join, right join, or inner join?**

Let’s look at a quick example to understand what the outputs are for all three types of joins. Table A, on the left, is a table with customer first and last name. Table B, on the right, is a table with order totals.



Source: Author

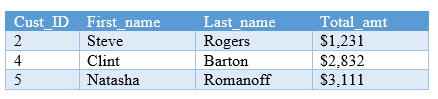
If you’ve forgotten the differences between all the joins, here’s a visual to refresh your memory.



SQL INNER JOIN vs. LEFT JOIN vs. RIGHT JOIN — Source: Author

If you do an inner join, you’ll get all records that match between the two tables. As you can see, Cust\_ID 2, 4, and 5 are in both tables.

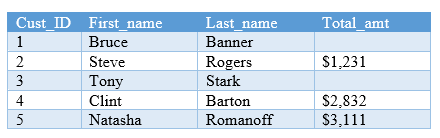
**INNER JOIN**



INNER JOIN OUTPUT — Source: Author

If you do a left join, you’ll get all records that match PLUS all records from the table on the left. We get all customers 1–5 from the left table.

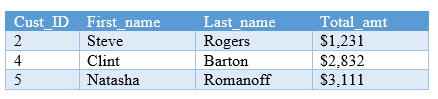
**LEFT JOIN**



LEFT JOIN OUTPUT — Source: Author

If you do a right join, you’ll get all records that match PLUS all records from the table on the right. There are 3 orders for 3 customers.

**RIGHT JOIN**



RIGHT JOIN OUTPUT — Source: Author

The right join gives the same output as the left join. Let’s take a closer look at the the left join and an inner join. What do you notice about the differences between these outputs? Which one should you be using?

The output from the left join contains both the customers that had an order as well as customers that did not have an order. The inner join contains only customers that have an order. The decision on which join to use will depend on what you’re being asked and the purpose of this piece of data.

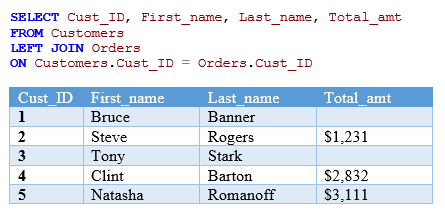
Do we only care about customers that have made an order? Do we want to see which customers don’t have orders? Those are the type of questions that you should think about when deciding what join to do.

**Does it matter which table is on the left and which table is on the right when I do a join?**

It does not matter whether a table goes on the “left side” or the “right side” of the join. You can get the same output regardless of where you place it. What does matter is what type of join you use after you decide where to place the table.

**Example 1**

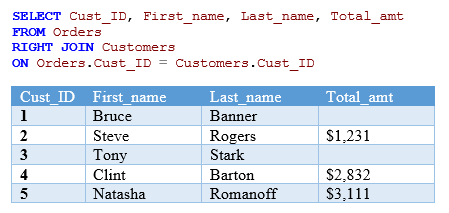
Let’s say that I put table Customers on the left and table Orders on the right and do a left join.



LEFT JOIN OUTPUT — Source: Author

**Example 2**

Now let’s reverse which tables is on the left and right. Put table Orders on the left this time and table Customers on the right. To get the same output as earlier, you want to do a right join because the Customers table is now on the right instead of the left.



RIGHT JOIN OUTPUT — Source: Author

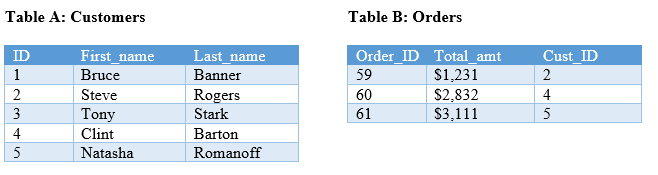
As you can see, the results in example 1 and example 2 are exactly the same. Thus, you can change the position of which side the table is on, but make sure you’re doing the correct join to get accurate results!

**Does it matter which column I join on?**

You would have to choose the correct column to join on in order to accurately obtain the data. We always join two tables based on the column in which they are related.

In the example earlier, we join **ON Orders.Cust\_ID = Customers.Cust\_ID**. This one is easy to spot given that they have the exact column name, therefore it is not a secret that they are related to one another.

One common example you might often see is something like this. Table A has a column called ID. Table B has a column called Cust\_ID. Even though the column name is not exactly the same, we can infer by logic that ID in Table A is related to Cust\_ID in Table B.



If you’re lucky, you may find documentation showing an entity relationship diagram that will tell you all the primary and foreign keys in the database. That will be your blueprint for which columns to join on.

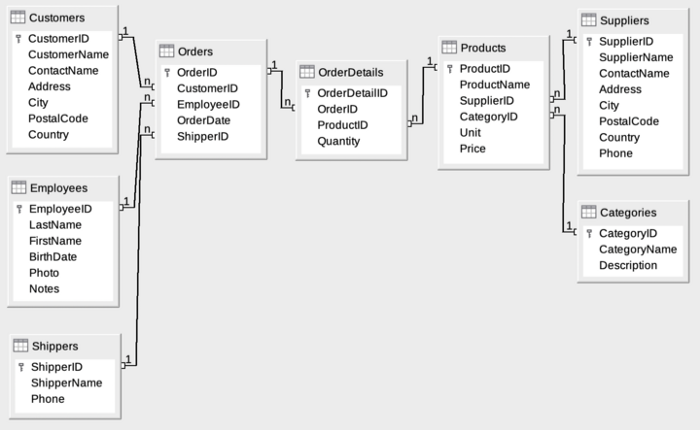
Note that in some cases, you might find that you have to join on multiple columns. So make sure to take a close look at the tables!

**In a real job environment, how do I know how my SQL query should look like?**

To answer this question, there are two aspects I’d like to talk about. The first part that is often confusing for my mentees is “how do I know where to get the data?” That is, what tables should be used?

Well, you’ll have to take a look at the tables and figure out where your data is located. Is the data located in just one table? Is the data located across several tables? If so, you’ll have the join the tables together to get your output.

Here’s an example of what a database might look like. There are 8 different tables here, each containing specific information related to the name of the table.



Source: <https://en.wikiversity.org/wiki/Database_Examples/Northwind#Entity-Relationship_Diagram>

To figure out what tables to use, you’ll examine the tables and determine where the data you want is located. Do you want product information? Do you want employee information?

If you’re unsure about where exactly the data is, I recommend querying the table and seeing a sample output to give you a better idea of what’s in the tables.

The second part that is often confusing for my mentees is knowing what exactly the SQL statement should look like. You’ve learned the SELECT FROM WHERE ORDER BY syntax, but how do you know what goes into this template? You will know what to write based on what the request is. That is, what data are you being asked to get from the database? You’ll essentially translate the request to your SQL statement.

The way I go about writing my queries is first starting with figuring out what the output should look like. That will guide you in determining what your SQL query should look like.

Some of the questions I try to answer initially are the following: What columns are needed? What columns need to be calculated and how should they be calculated. What table(s) are those columns located in?

These are useful questions because it will help you decide things such as whether you need to JOIN tables, whether you need to do a GROUP BY, etc.

**What do I do if I’m struggling with SQL?**

In my opinion, I believe you should learn about relational databases before you even learn about SQL because that will help you to understand how databases and joins work. Try taking a relational database course and see if that helps your understanding of the concepts.

In addition, you’ll only get better with practice. Try out different queries and see how the output changes. That will really help you understand how to manipulate the SQl query to get what you need.

If you’re writing a large query, start small and make sure that each piece is running as you would expect.

Hopefully, this article helped clear some things up for you. For hands on SQL practice, check out these 24 SQL exercises with solutions.

## [SQL exercises on movie Database - Exercises, Practice, Solution - w3resource](https://www.w3resource.com/sql-exercises/movie-database-exercise/joins-exercises-on-movie-database.php" \t "_blank)

### **[Practice with solution of exercises on SQL movie database, INNER JOIN, LEFT JOIN, RIGHT JOIN, RIGHT OUTER JOIN, LEFT…](https://www.w3resource.com/sql-exercises/movie-database-exercise/joins-exercises-on-movie-database.php" \t "_blank)**

[www.w3resource.com](https://www.w3resource.com/sql-exercises/movie-database-exercise/joins-exercises-on-movie-database.php" \t "_blank)