The Seven Types of SQL Joins

It's unlikely that you'll be working exclusively with data from one table. So, you need to be able to grab data from multiple tables. One of the ways to do this is by using joins. In relational databases, joins are what make relational databases relational. This post provides a quick refresher of the four main types of SQL Joins and three of its variants.

What is a join?

A SQL join is a Structured Query Language (SQL) way of linking data from two or more tables based on a column shared between the tables. For example, you might connect an address table and a phone numbers table based on the person's name (e.g., "Give me the address and phone number for the person named John Smith.").

Why are joins important?

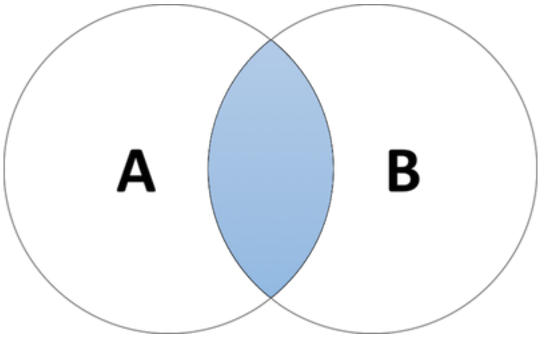
Joins allow you to gather data from multiple tables using one query. It's unlikely that you'll work exclusively with one table. Having one table means that you're either limiting yourself regarding the amount of data you gather. Or, having so much data that the table becomes unwieldy. Joins *relate* one table to another (hence the term *relational database*).

What kinds of SQL joins are there?

The following is a list of the four main types of SQL joins (along with three variants), ordered from most restrictive to most inclusive. We have provided sample SQL snippets that you can use to see what's returned when using a given join in your query.

Inner Join

The **inner join** is probably the most commonly-used type of join in SQL. Inner joins return all rows from two or more tables that meet the join condition.



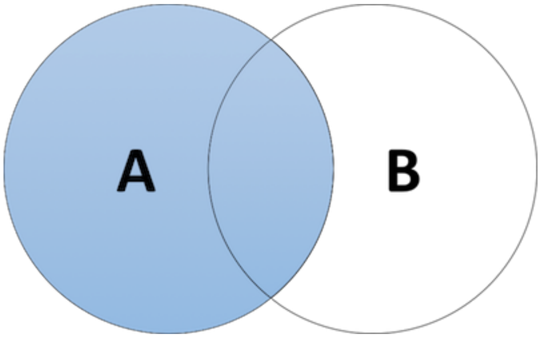
(https://s3.amazonaws.com/prod.io.teamsql.wordpress/wp-content/uploads/2017/04/16133647/0_MHrYXyHi3YLvkH7i_.png)

Sample SQL

```
SELECT columns
FROM TableA
INNER JOIN TableB
ON A.columnName = B.columnName;
```

Left [Outer] Join

The left outer join (sometimes abbreviated to left join) returns all rows from the **left**-hand table specified in the ON condition. And, *only* the rows from the **right**-hand table that meet the join condition.



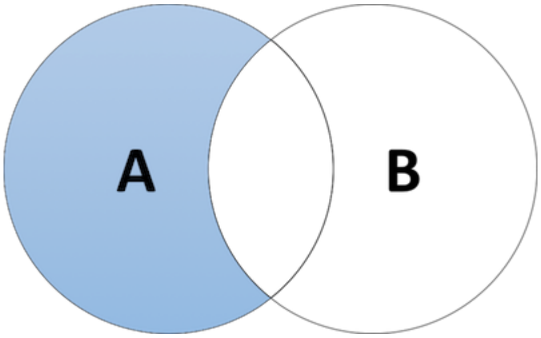
(https://s3.amazonaws.com/prod.io.teamsql.wordpress/wp-content/uploads/2017/04/16133710/0_Nj9TD1XYtLNCj79O_.png)

Sample SQL

```
SELECT columns
FROM TableA
LEFT OUTER JOIN TableB
ON A.columnName = B.columnName
```

Left [Outer] Join without Intersection

This join type is a variant on the basic left outer join. But instead, it returns all rows from the **left**-hand table specified in the ON condition that also meets the join condition. N*one* of the rows from the **right**-hand table that matches the join condition.



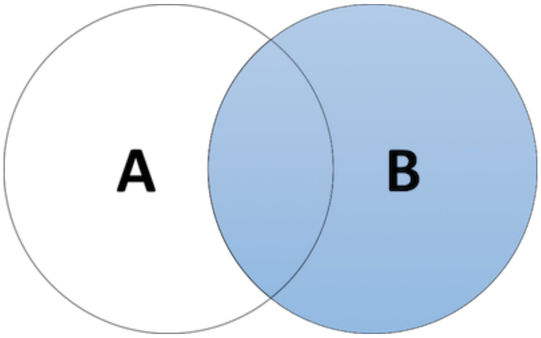
(https://s3.amazonaws.com/prod.io.teamsql.wordpress/wp-content/uploads/2017/04/16133731/0_1zy_6UhueRrkLigl_.png)

Sample SQL

```
SELECT columns
FROM TableA
LEFT OUTER JOIN TableB
ON A.columnName = B.columnName
WHERE B.columnName IS NULL
```

Right [Outer] Join

The right outer join (sometimes abbreviated to right join) returns all rows from the **right**-hand table specified in the ON condition and *only* the rows from the **left**-hand table that meet the join condition.



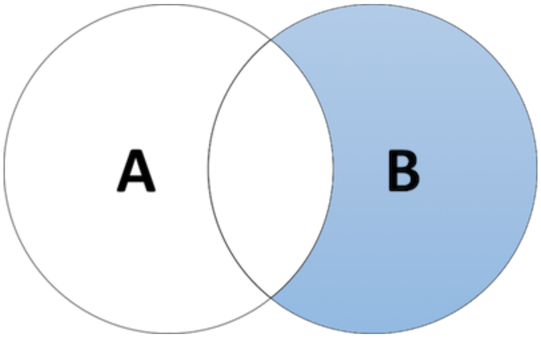
(https://s3.amazonaws.com/prod.io.teamsql.wordpress/wp-content/uploads/2017/04/16133807/0_jM3CoEQ9DAnT1PbD_.png)

Sample SQL

```
SELECT columns
FROM TableA
RIGHT OUTER JOIN TableB
ON A.columnName = B.columnName
```

Right [Outer] Join without Intersection

This join is a variant on the right outer join, but instead, it returns all rows from the **right**-hand table specified in the ON condition that also meets the join condition and *none* of the rows from the **left**-hand table that match the join condition.



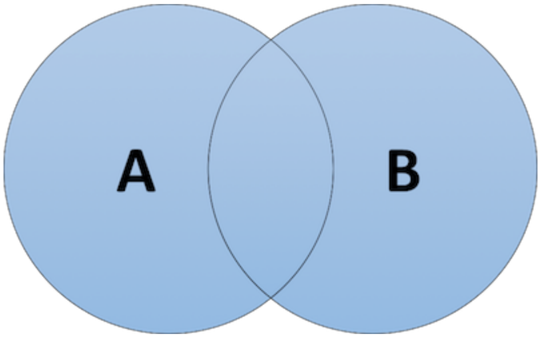
(https://s3.amazonaws.com/prod.io.teamsql.wordpress/wp-content/uploads/2017/04/16133850/0_VJOHPBkd0e0Y4jcs_.png)

Sample SQL

```
SELECT columns
FROM TableA
RIGHT OUTER JOIN TableB
ON A.columnName = B.columnName
WHERE A.columnName IS NULL
```

Full [Outer] Join

The entire outer join (sometimes abbreviated to full join) returns all rows from both tables named in the ON condition where the join condition is *not* met (including NULL values).



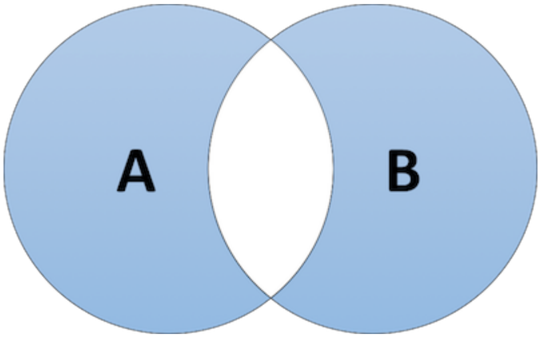
(https://s3.amazonaws.com/prod.io.teamsql.wordpress/wp-content/uploads/2017/04/16133954/0_JjEjDtvyjjCWps2w_-1111.png)

Sample SQL

```
SELECT columns
FROM TableA
FULL JOIN TableB
ON A.columnName = B.columnName
```

Full [Outer] Join without Intersection

This variant of the full outer join (sometimes abbreviated to full join) returns all rows from both tables named in the ON condition where the join condition is *not* met (excluding NULL values).



(https://s3.amazonaws.com/prod.io.teamsql.wordpress/wp-content/uploads/2017/04/16134027/0_6osFFhtC_1MRyk122222222

Sample SQL

```
SELECT columns

FROM TableA

FULL JOIN TableB

ON A.columnName = B.columnName

WHERE A.columnName IS NULL

OR B.columnName IS NULL
```

Takeaways

This post covered the four basic types of SQL joins and three variants on those basic types. Joins are what make a relational database relational, so if you haven't mastered the different types of SQL joins available, you'll want to do so. Launch TeamSQL today and use the provided code snippets to see all the different ways you can bring back different subsets of your data.