

TSQL OUTER JOINS (LEFT and RIGHT)

After learning the INNER JOIN when querying SQL Server, the LEFT OUTER JOIN and RIGHT OUTER JOIN are often what people attempt to learn next. The reason that I say attempt is that most people don't get OUTER JOINS right the first time they learn it. This becomes very clear when interviewing developers and asking the following question:

Can you explain the difference between an INNER JOIN and an OUTER JOIN?

Most candidates get the INNER JOIN right, but very few developer or software engineer candidates are able to accurately answer the OUTER JOIN part of the question.

For many years I had a difficult time trying to explain the difference myself, I would try to say things like:

The INNER JOIN only returns those rows that are an exact match, and the OUTER JOIN (LEFT OR RIGHT) returns the rows that are an exact match, plus all remaining rows from one side or the other based on whether it is a LEFT or RIGHT OUTER JOIN.

This usually ended up with a blank stare, or a really confused look on the face of the person asking me the question. Then I discovered Venn Diagrams to describe JOIN types.

The best way that I have discovered to represent JOINing tables is to use Venn Diagrams that show what overlaps, what tables have in common, and what is missed. This is represented on my Free [SQL Server Join Types Poster](#), if you haven't seen the poster, please check it out, it will help with your understanding of all SQL Server JOIN Types.

The [INNER JOIN was previously discussed in another post](#), so I am not going to repeat it, just visit that post if you need an [INNER JOIN refresher](#).

Keep in mind that LEFT OUTER JOIN is often abbreviated as LEFT JOIN and RIGHT OUTER JOIN is often abbreviated as RIGHT JOIN. For the purpose of this demo, and to be clear, I will always include the word OUTER which is optional in TSQL.

First we will establish a sample database with a couple tables to query against.

So let's take a look at the Venn Diagrams for LEFT OUTER JOIN and RIGHT OUTER JOIN.

-

The LEFT OUTER JOIN diagram shows that the query will return all rows from TABLE1, and those rows from TABLE 2 that match up on the JOIN condition. For the rows in TABLE1 that don't match TABLE2, the TABLE2 side of the results will be filled with NULLs. Keep in mind that if a single row from TABLE 1 matches multiple rows from TABLE2, that will result in multiple rows in the output, not just a single row.

Here is the sample code.

Which produces the following output.

Now lets take a look at the RIGHT OUTER JOIN and see how it compares.

The Venn diagram shows that the RIGHT OUTER JOIN should include all the rows from Table2, and it will match what it can from TABLE1, and return NULLs on the left side of the results where there is no match.

Here is a sample of a RIGHT OUTER JOIN.

Which gives us the following results:

For More Information:

- [TSQL INNER JOIN](#)
- [CROSS JOIN vs FULL OUTER JOIN](#)
- [SQL Server JOIN Types Poster](#)