Steve Stedman
Freelance SQL Server Consultant
http://stevestedman.com

FULL OUTER JOIN vs CROSS JOIN

As I have been working on the SQL Server JOIN Types poster, I have received several questions around the difference between a CROSS JOIN, and a FULL OUTER JOIN. After looking at the Venn diagrams for the two, they are both shown as the same, however they are not the same by any means.

Yes, they both include all rows from both the LEFT and RIGHT side of the JOIN, however they are matched up or JOINed in a very different way.

Lets take a look, first at the Venn diagrams, you can see below that the Venn diagrams show that all rows from Table1 are included in the results, and all rows from Table2 are included in the results. This is indeed correct, the differences come in the number of rows and how the results are matched up.

Now lets take a look at some sample cod that shows the differences. First we will create a database and create a couple tables to use for the demo, and each table has 3 rows inserted to it:

Look at this as a database to store people and their favorite color, in this example, there are 3 people , and 3 colors. One person does not have a favorite color, and one color has not been favorited by anyone.

Now lets take a look at the differences between the two.

CROSS JOIN

The CROSS JOIN gives you the Cartesian product of the two tables, by matching every row from one table with every row from another table. If there are X rows in the first table, and Y rows in the second table, the result set will have exactly X times Y rows. Notice on the CROSS JOIN, there is no ON clause specified. This is because there is not matching.

CROSS JOIN Sample

Which gives us the following result set:

In the above result set, you can see that for each row in the People table (3) it was matched up with each row from the Colors table (3) which gives us a total of 3 x 3 = 9 rows in the result set.

FULL OUTER JOIN

Now for the full outer JOIN, this is like asking for a LEFT OUTER JOIN and a RIGHT OUTER JOIN in the same query. You are asking to get every row from Table1 matching what could be matched in Table2, for those that don't match on the Table1 side, just display the Table2 side, with NULLs on the LEFT side, and for those that don't match on the TABLE2 side, just display the Table1 side with NULLs on the RIGHT side.

FULL OUTER JOIN Sample

In this case everything from Table1 will be represented, and everything from Table2 will be shown, but they will be matched up base on the ON clause, rather than with the CROSS JOIN which gives every possible combination.

I hope that this helps answer some of the questions with the differences between a CROSS JOIN and a FULL OUTER JOIN.

If you haven't already taken a look, please check out the free download of the SQL Server JOIN Types poster.

For More Information:

- TSQL INNER JOIN
- TSQL OUTER JOINs (LEFT and RIGHT)
- SQL Server JOIN Types Poster