A Join A Day – The Left Anti Semi Join

2012-12-14 - A Join A Day, Fundamentals, General, Series

Introduction

This is the fourteenth post in my A Join A Day series about SQL Server Joins. Make sure to let me know how I am doing or ask your burning join related questions by leaving a comment below.

Today we are going to talk about the Left Anti Semi Join. And with that it is time for another short Latin lesson: "Anti"

The prefix "anti" means as much as "opposite of". While most dictionaries list it as originating from the Latin language, is roots are actually deeper in ancient greek: ἀντί.

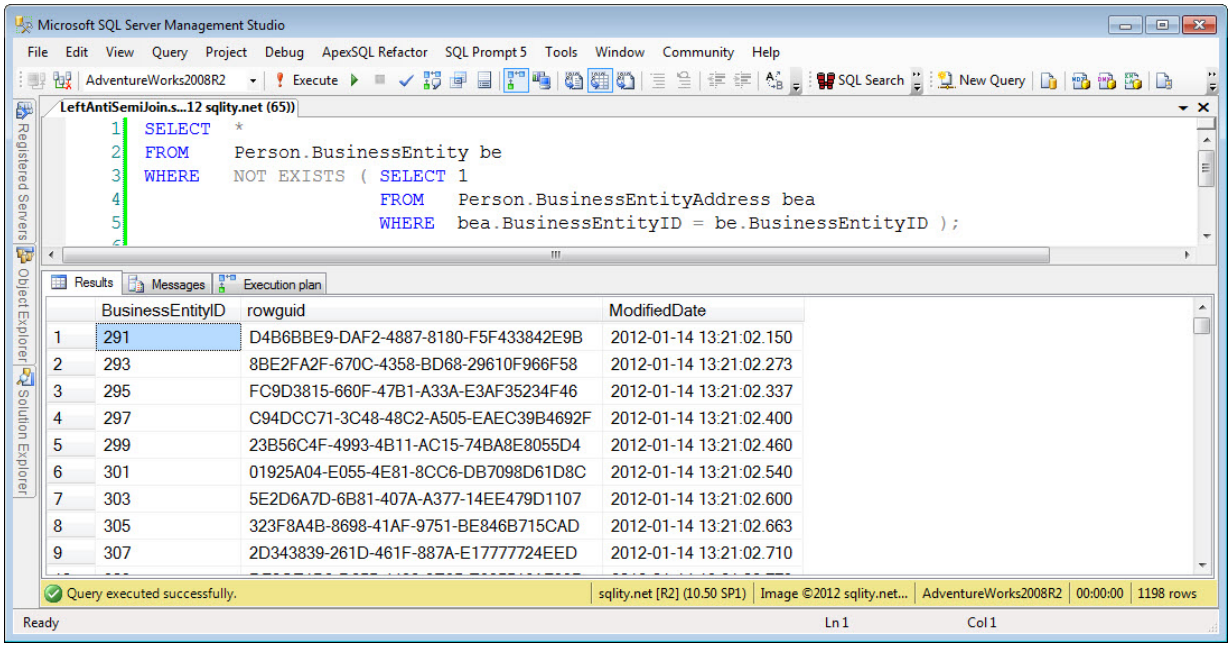
So, the Left Anti Semi Join is the opposite of a Left Semi Join. However, that does not make it a right semi join. Instead "Anti" affects which rows are returned and which aren't. Like the Left Semi Join, the Left Anti Semi Join returns only rows from the left row source. Each row is also returned at most once. And duplicates are also not eliminated. However, other than the Left Semi Join, the Left Anti Semi Join returns only rows for which no match on the right side exists. Let's look at an example.

LEFT ANTI SEMI JOIN Example

The easiest way to write a query with a logical left anti semi join is a `NOT EXISTS()` query like this:

```
[sql] SELECT *
FROM Person.BusinessEntity be
WHERE NOT EXISTS ( SELECT 1
FROM Person.BusinessEntityAddress bea
WHERE bea.BusinessEntityID = be.BusinessEntityID );
[/sql]
```

This query returns the 1198 entities that do not have an address at all:

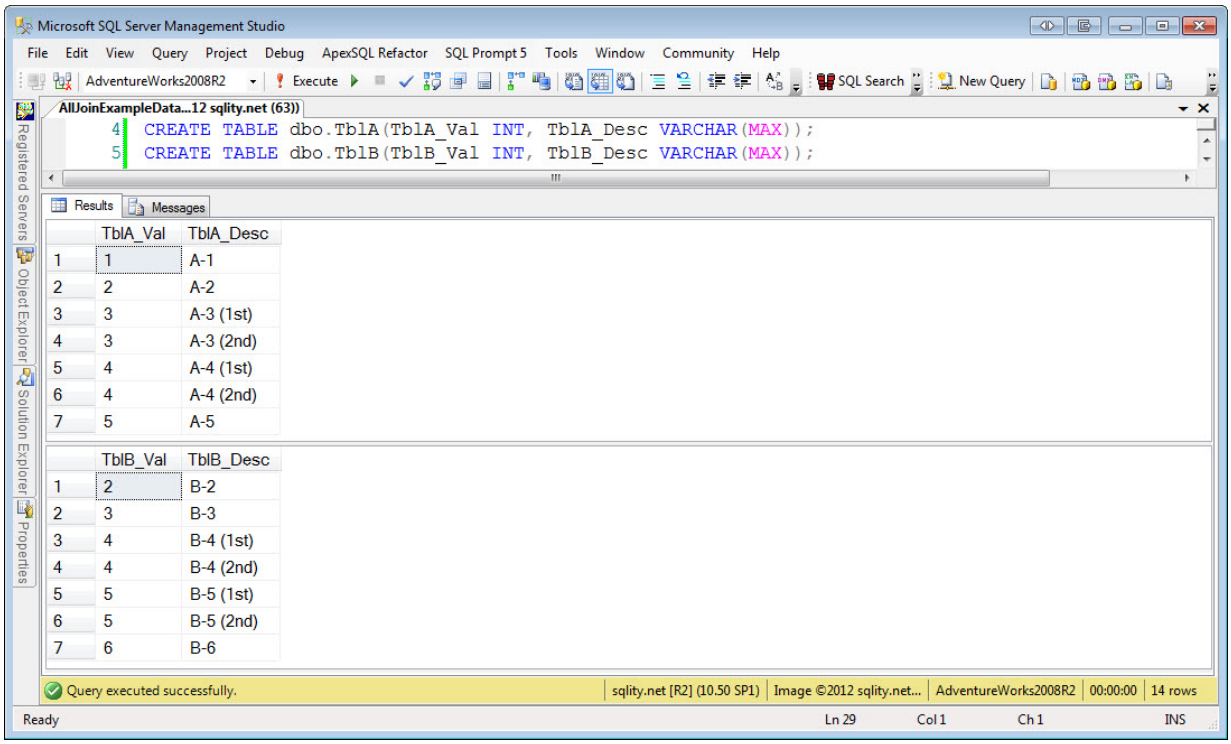


So exactly all the entities that did not get returned by the Left Semi Join are the ones that the Left Anti Semi Join is returning.

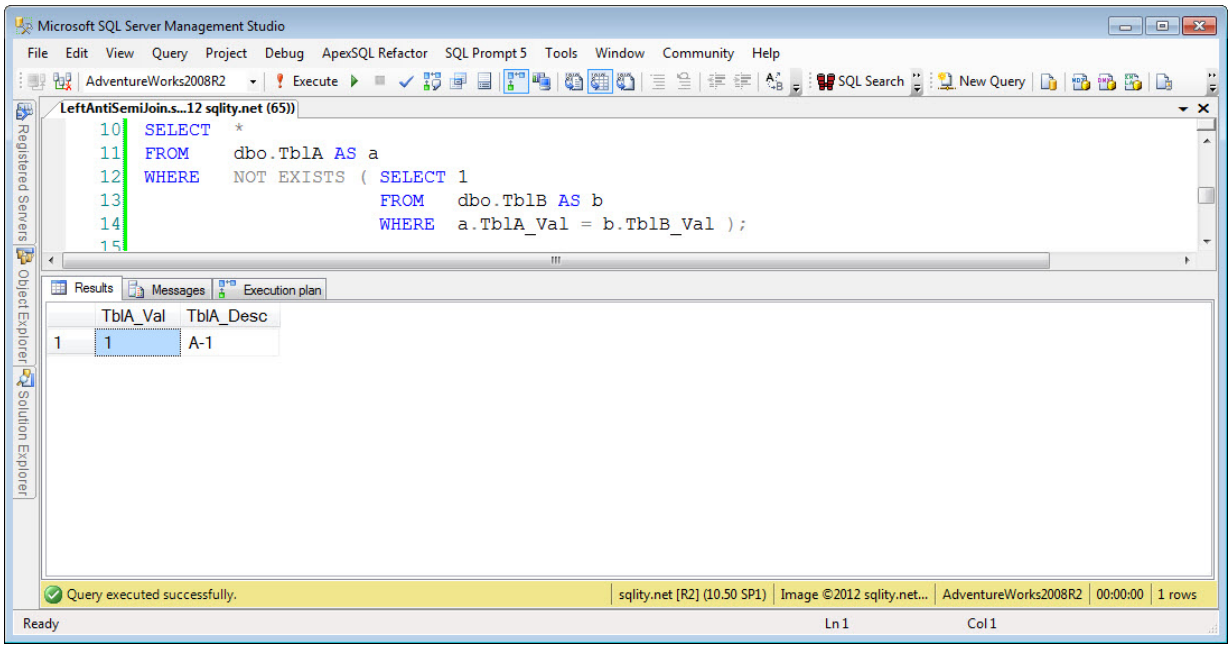
Let's look at the `TblA - TblB` example again with this query:

```
[sql] SELECT *
FROM dbo.TblA AS a
WHERE NOT EXISTS ( SELECT 1
FROM dbo.TblB AS b
WHERE a.TblA_Val = b.TblB_Val );
[/sql]
```

As always, this is the data in the two tables:



The query returns just a single row:



The row `TblA_Val = 1` is the only row that does not have a match in `TblB`. Therefore it is the only one returned by this query.

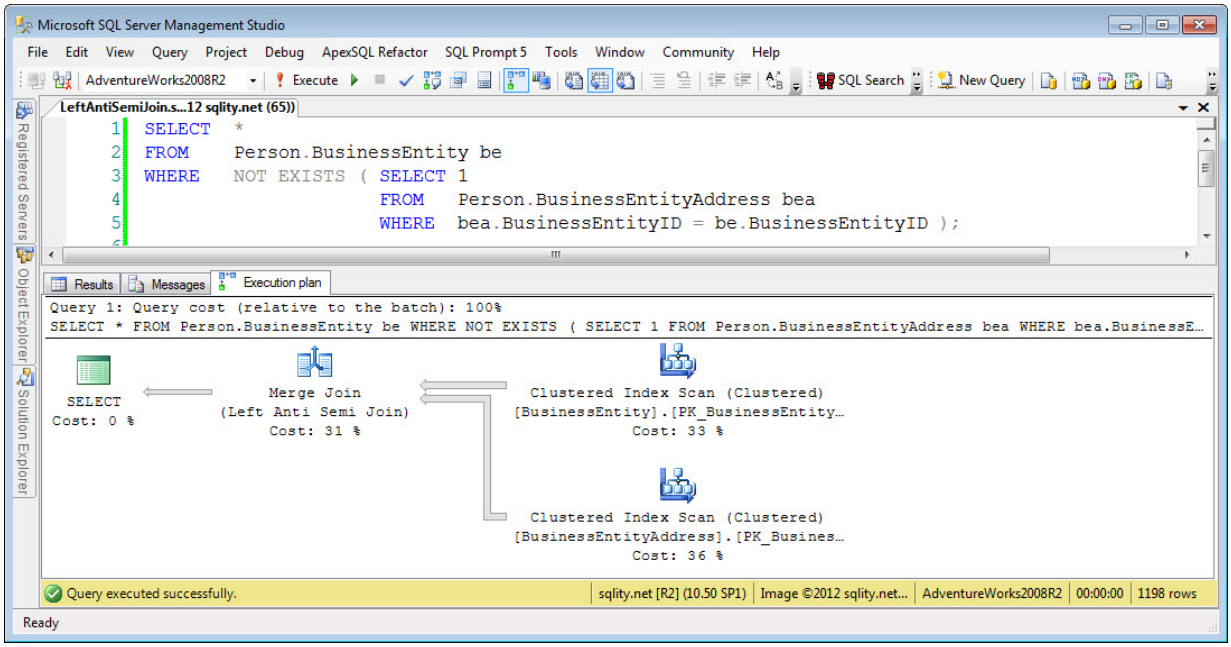
LEFT ANTI SEMI JOIN Syntax

There is again no statement or command that can be directly linked to a Left Anti Semi Join. There are many ways to write a query like this. However, the `NOT EXISTS()` syntax used in both examples above is the most common way to write a Left Anti Semi Join query.

LEFT ANTI SEMI JOIN Operator

Like with the Left Semi Join, there is no command or statement that will always lead to a Left Anti Semi Join operator. The optimizer will consider the Left Anti Semi Join operator for any query that fits the model. However there is no guarantee that the operator will be used.

For above BusinessEntity-without-Address query the optimizer selected a Left Anti Semi Join operator:

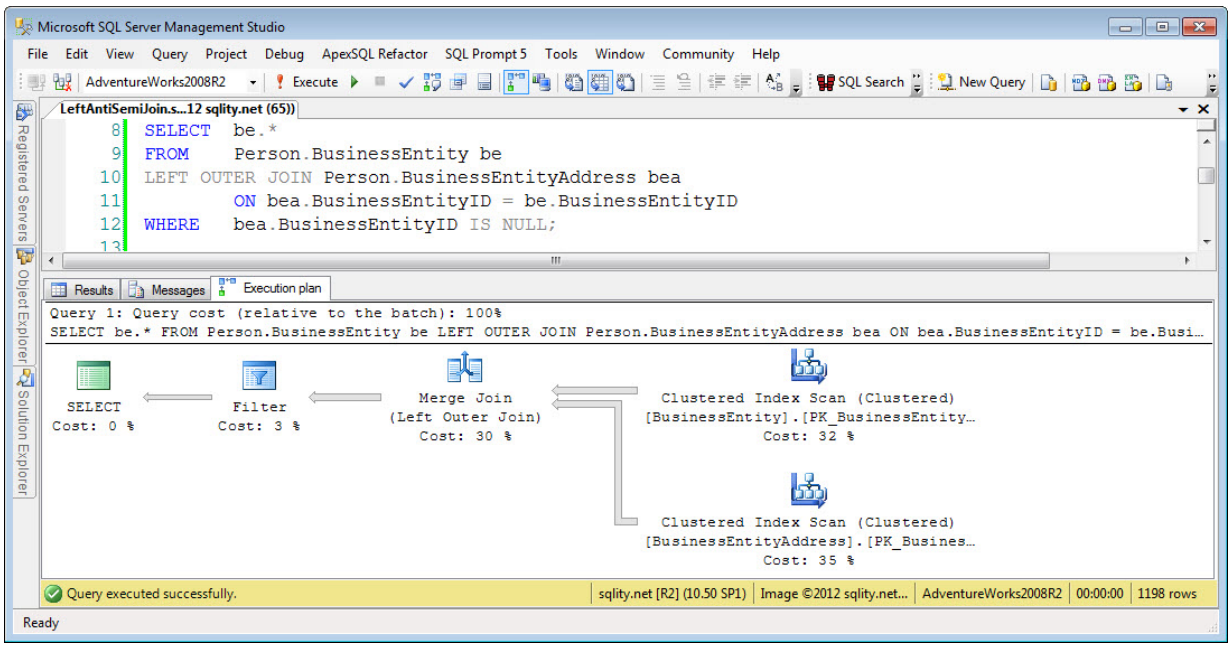


However, let's look at this logically equivalent query:

```
[sql] SELECT be.*
FROM Person.BusinessEntity be
LEFT OUTER JOIN Person.BusinessEntityAddress bea
ON bea.BusinessEntityID = be.BusinessEntityID
WHERE bea.BusinessEntityID IS NULL;
[/sql]
```

It first finds executes a left outer join between the `BusinessEntity` table and the `BusinessEntityAddress` table. Of the 20812 rows that come out of the left outer join, most get filtered out by the where clause. Only the 1198 rows that do not have a match get returned.

Even though this query is logically equivalent it ends up with a different execution plan:



The execution plan reflects exactly the way the query is written. First a Left Outer Join operator joins the two tables together and then an additional Filter operator removes the rows for which a match exists. Because of the additional operator and because of the 20812 rows that get passed into it, only to get discarded en mass (only 1198 rows survive) this execution plan is actually less efficient than the Left Anti Semi Join execution plan.

The SQL Server optimizer cannot do an exhaustive search of all possible plans when compiling a query. Therefore sometimes a sub-optimal plan gets selected. However, the team developing the optimizer is constantly working on improving this already amazing piece of software and it is possible that with the next version or even the next service pack both queries will result in the same plan.

Summary

The Left Anti Semi Join filters out all rows from the left row source that have a match coming from the right row source. Only the orphans from the left side are returned. While there is a Left Anti Semi Join operator, there is no direct SQL command to request this operator. However, the `NOT EXISTS()` syntax shown in the above examples will often result in this operator being used.

A Join A Day

This post is part of my December 2012 "A Join A Day" blog post series. You can find the table of contents with all posts published so far in the introductory post: A Join A Day – Introduction. Check back there frequently throughout the month.

Categories: A Join A Day, Fundamentals, General, Series

6 Responses to *A Join A Day – The Left Anti Semi Join*

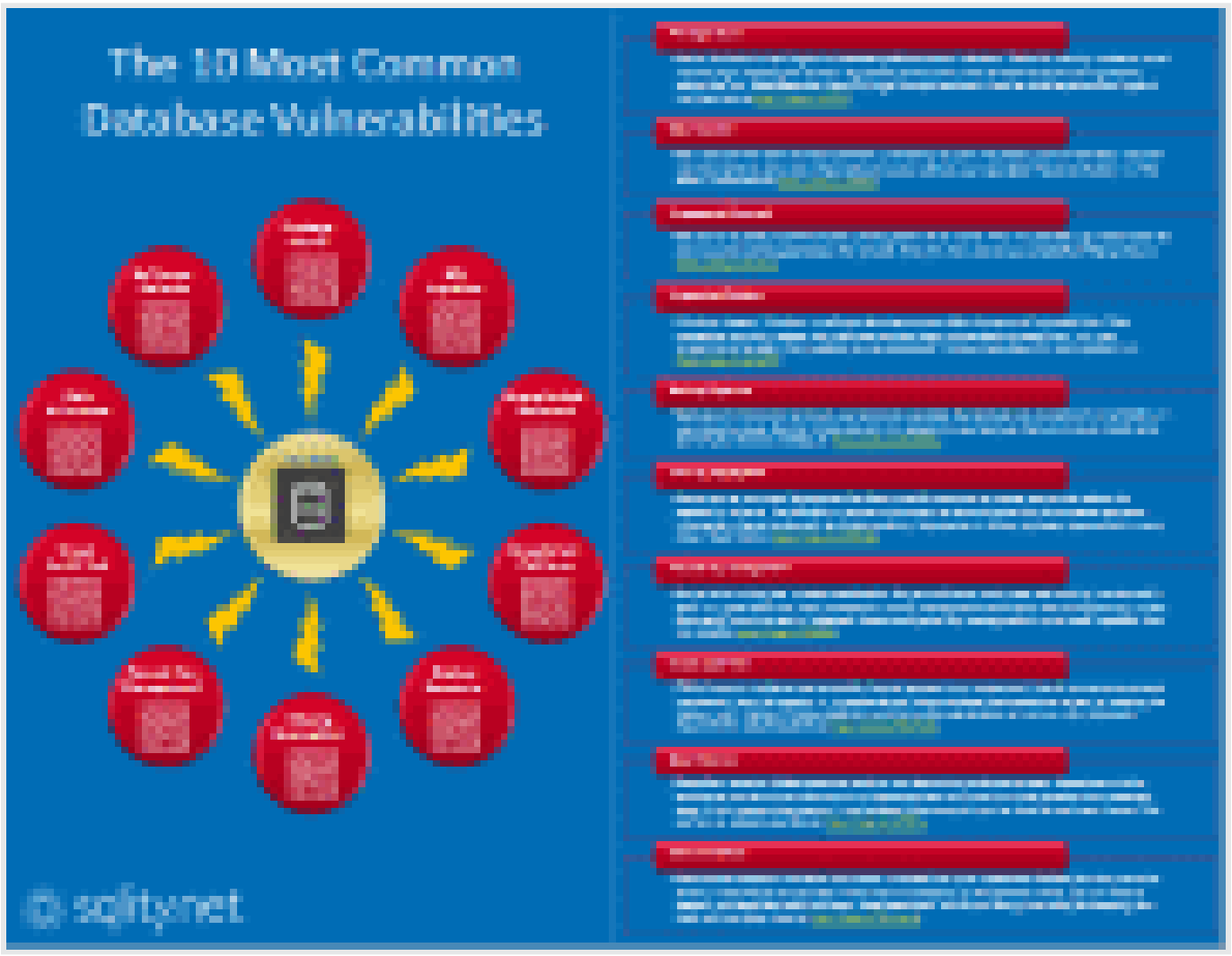
Leave a Reply

You must be logged in to post a comment.

« A JOIN A DAY – THE RIGHT SEMI JOIN

A JOIN A DAY – RIGHT ANTI SEMI JOIN »

Free Poster: The 10 Most Common Database Vulnerabilities



Get Yours Now!