**Types of Joins**

**Visual Studio 2013**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [Visual Studio 2012](http://msdn.microsoft.com/en-us/library/zt8wzxy4(d=printer,v=vs.110).aspx)
* [Visual Studio 2010](http://msdn.microsoft.com/en-us/library/zt8wzxy4(d=printer,v=vs.100).aspx)
* [Visual Studio .NET 2003](http://msdn.microsoft.com/en-us/library/zt8wzxy4(d=printer,v=vs.71).aspx)
* [.NET Framework 3.5](http://msdn.microsoft.com/en-us/library/zt8wzxy4(d=printer,v=vs.90).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/zt8wzxy4(d=printer,v=vs.80).aspx)

When you join tables, the type of join that you create affects the rows that appear in the result set. You can create the following types of joins:

* Inner join   A join that displays only the rows that have a match in both joined tables. (This is the default type of join in the [Query and View Designer](http://msdn.microsoft.com/en-us/library/ms172013.aspx).) For example, you can join the titles and publishers tables to create a result set that shows the publisher name for each title. In an inner join, titles for which you do not have publisher information are not included in the result set, nor are publishers with no titles. The resulting SQL for such a join might look like this:

SELECT title, pub\_name

FROM titles INNER JOIN

publishers ON titles.pub\_id = publishers.pub\_id

|  |
| --- |
| **Note** |
| Columns containing NULL do not match any values when you are creating an inner join and are therefore excluded from the result set. Null values do not match other null values. |

* Outer join   A join that includes rows even if they do not have related rows in the joined table. You can create three variations of an outer join to specify the unmatched rows to be included:
  + Left outer join   All rows from the first-named table (the "left" table, which appears leftmost in the JOIN clause) are included. Unmatched rows in the right table do not appear. For example, the following SQL statement illustrates a left outer join between the titles and publishers tables to include all titles, even those you do not have publisher information for:

SELECT titles.title\_id,

titles.title,

publishers.pub\_name

FROM titles LEFT OUTER JOIN publishers

ON titles.pub\_id

= publishers.pub\_id

* + Right outer join   All rows in the second-named table (the "right" table, which appears rightmost in the JOIN clause) are included. Unmatched rows in the left table are not included. For example, a right outer join between the titles and publishers tables will include all publishers, even those who have no titles in the titles table. The resulting SQL might look like this:

SELECT titles.title\_id,

titles.title,

publishers.pub\_name

FROM titles RIGHT OUTER JOIN publishers

ON titles.pub\_id

= publishers.pub\_id

* + Full outer join   All rows in all joined tables are included, whether they are matched or not. For example, a full outer join between titles and publishers shows all titles and all publishers, even those that have no match in the other table.

SELECT titles.title\_id,

titles.title,

publishers.pub\_name

FROM titles FULL OUTER JOIN publishers

ON titles.pub\_id

= publishers.pub\_id

|  |
| --- |
| **Note** |
| Some databases, such as Oracle, do not support full outer joins. |

* Cross join   A join whose result set includes one row for each possible pairing of rows from the two tables. For example, authors CROSS JOIN publishers yields a result set with one row for each possible author/publisher combination. The resulting SQL might look like this:

SELECT \*

FROM authors CROSS JOIN publishers