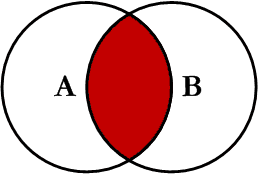
Using the code

I am going to discuss seven different ways you can return data from two relational tables. I will be excluding cross Joins and self referencing Joins. The seven Joins I will discuss are shown below:

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. OUTER JOIN
5. LEFT JOIN EXCLUDING INNER JOIN
6. RIGHT JOIN EXCLUDING INNER JOIN
7. OUTER JOIN EXCLUDING INNER JOIN

For the sake of this article, I'll refer to 5, 6, and 7 as LEFT EXCLUDING JOIN, RIGHT EXCLUDING JOIN, and OUTER EXCLUDING JOIN, respectively. Some may argue that 5, 6, and 7 are not really joining the two tables, but for simplicity, I will still refer to these as Joins because you use a SQL Join in each of these queries (but exclude some records with a WHERE clause).

**Inner JOIN**



This is the simplest, most understood Join and is the most common. This query will return all of the records in the left table (table A) that have a matching record in the right table (table B). This Join is written as follows:

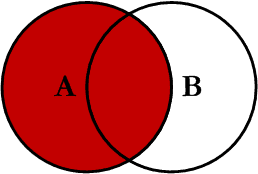
SELECT *<select\_list>*

FROM Table\_A A

INNER JOIN Table\_B B

ON A.Key = B.Key

**Left JOIN**



This query will return all of the records in the left table (table A) regardless if any of those records have a match in the right table (table B). It will also return any matching records from the right table. This Join is written as follows:

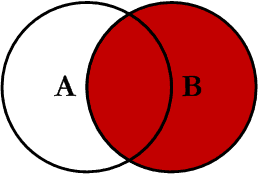
SELECT *<select\_list>*

FROM Table\_A A

LEFT JOIN Table\_B B

ON A.Key = B.Key

**Right JOIN**



This query will return all of the records in the right table (table B) regardless if any of those records have a match in the left table (table A). It will also return any matching records from the left table. This Join is written as follows:

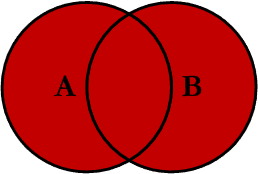
SELECT *<select\_list>*

FROM Table\_A A

RIGHT JOIN Table\_B B

ON A.Key = B.Key

**Outer JOIN**



This Join can also be referred to as a FULL OUTER JOIN or a FULL JOIN. This query will return all of the records from both tables, joining records from the left table (table A) that match records from the right table (table B). This Join is written as follows:

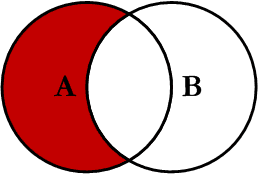
SELECT *<select\_list>*

FROM Table\_A A

FULL OUTER JOIN Table\_B B

ON A.Key = B.Key

**Left Excluding JOIN**



This query will return all of the records in the left table (table A) that do not match any records in the right table (table B). This Join is written as follows:

SELECT *<select\_list>*

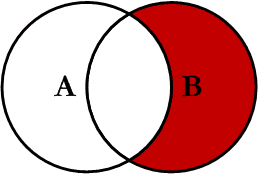
FROM Table\_A A

LEFT JOIN Table\_B B

ON A.Key = B.Key

WHERE B.Key IS NULL

**Right Excluding JOIN**



This query will return all of the records in the right table (table B) that do not match any records in the left table (table A). This Join is written as follows:

SELECT *<select\_list>*

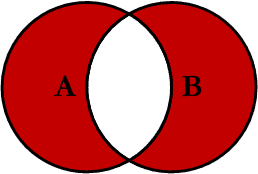
FROM Table\_A A

RIGHT JOIN Table\_B B

ON A.Key = B.Key

WHERE A.Key IS NULL

**Outer Excluding JOIN**



This query will return all of the records in the left table (table A) and all of the records in the right table (table B) that do not match. I have yet to have a need for using this type of Join, but all of the others, I use quite frequently. This Join is written as follows:

SELECT *<select\_list>*

FROM Table\_A A

FULL OUTER JOIN Table\_B B

ON A.Key = B.Key

WHERE A.Key IS NULL OR B.Key IS NULL

**Examples**

Suppose we have two tables, *Table\_A* and *Table\_B*. The data in these tables are shown below:

**TABLE\_A**

PK Value

---- ----------

1 FOX

2 COP

3 TAXI

6 WASHINGTON

7 DELL

5 ARIZONA

4 LINCOLN

10 LUCENT

**TABLE\_B**

PK Value

---- ----------

1 TROT

2 CAR

3 CAB

6 MONUMENT

7 PC

8 MICROSOFT

9 APPLE

11 SCOTCH

The results of the seven Joins are shown below:

**-- INNER JOIN**

SELECT A.PK AS A\_PK, A.Value AS A\_Value,

B.Value AS B\_Value, B.PK AS B\_PK

FROM Table\_A A

INNER JOIN Table\_B B

ON A.PK = B.PK

A\_PK A\_Value B\_Value B\_PK

---- ---------- ---------- ----

1 FOX TROT 1

2 COP CAR 2

3 TAXI CAB 3

6 WASHINGTON MONUMENT 6

7 DELL PC 7

(5 row(s) affected)

**-- LEFT JOIN**

SELECT A.PK AS A\_PK, A.Value AS A\_Value,

B.Value AS B\_Value, B.PK AS B\_PK

FROM Table\_A A

LEFT JOIN Table\_B B

ON A.PK = B.PK

A\_PK A\_Value B\_Value B\_PK

---- ---------- ---------- ----

1 FOX TROT 1

2 COP CAR 2

3 TAXI CAB 3

4 LINCOLN NULL NULL

5 ARIZONA NULL NULL

6 WASHINGTON MONUMENT 6

7 DELL PC 7

10 LUCENT NULL NULL

(8 row(s) affected)

**-- RIGHT JOIN**

SELECT A.PK AS A\_PK, A.Value AS A\_Value,

B.Value AS B\_Value, B.PK AS B\_PK

FROM Table\_A A

RIGHT JOIN Table\_B B

ON A.PK = B.PK

A\_PK A\_Value B\_Value B\_PK

---- ---------- ---------- ----

1 FOX TROT 1

2 COP CAR 2

3 TAXI CAB 3

6 WASHINGTON MONUMENT 6

7 DELL PC 7

NULL NULL MICROSOFT 8

NULL NULL APPLE 9

NULL NULL SCOTCH 11

(8 row(s) affected)

**-- OUTER JOIN**

SELECT A.PK AS A\_PK, A.Value AS A\_Value,

B.Value AS B\_Value, B.PK AS B\_PK

FROM Table\_A A

FULL OUTER JOIN Table\_B B

ON A.PK = B.PK

A\_PK A\_Value B\_Value B\_PK

---- ---------- ---------- ----

1 FOX TROT 1

2 COP CAR 2

3 TAXI CAB 3

6 WASHINGTON MONUMENT 6

7 DELL PC 7

NULL NULL MICROSOFT 8

NULL NULL APPLE 9

NULL NULL SCOTCH 11

5 ARIZONA NULL NULL

4 LINCOLN NULL NULL

10 LUCENT NULL NULL

(11 row(s) affected)

**-- LEFT EXCLUDING JOIN**

SELECT A.PK AS A\_PK, A.Value AS A\_Value,

B.Value AS B\_Value, B.PK AS B\_PK

FROM Table\_A A

LEFT JOIN Table\_B B

ON A.PK = B.PK

WHERE B.PK IS NULL

A\_PK A\_Value B\_Value B\_PK

---- ---------- ---------- ----

4 LINCOLN NULL NULL

5 ARIZONA NULL NULL

10 LUCENT NULL NULL

(3 row(s) affected)

**-- RIGHT EXCLUDING JOIN**

SELECT A.PK AS A\_PK, A.Value AS A\_Value,

B.Value AS B\_Value, B.PK AS B\_PK

FROM Table\_A A

RIGHT JOIN Table\_B B

ON A.PK = B.PK

WHERE A.PK IS NULL

A\_PK A\_Value B\_Value B\_PK

---- ---------- ---------- ----

NULL NULL MICROSOFT 8

NULL NULL APPLE 9

NULL NULL SCOTCH 11

(3 row(s) affected)

**-- OUTER EXCLUDING JOIN**

SELECT A.PK AS A\_PK, A.Value AS A\_Value,

B.Value AS B\_Value, B.PK AS B\_PK

FROM Table\_A A

FULL OUTER JOIN Table\_B B

ON A.PK = B.PK

WHERE A.PK IS NULL

OR B.PK IS NULL

A\_PK A\_Value B\_Value B\_PK

---- ---------- ---------- ----

NULL NULL MICROSOFT 8

NULL NULL APPLE 9

NULL NULL SCOTCH 11

5 ARIZONA NULL NULL

4 LINCOLN NULL NULL

10 LUCENT NULL NULL

(6 row(s) affected)

Note on the OUTER JOIN that the inner joined records are returned first, followed by the right joined records, and then finally the left joined records (at least, that's how my Microsoft SQL Server did it; this, of course, is without using any ORDER BY statement).

You can visit the Wikipedia article for more info [here](http://en.wikipedia.org/wiki/Sql_join) (however, the entry is not graphical).

I've also created a cheat sheet that you can print out if needed. If you right click on the image below and select "Save Target As...", you will download the full size image.

[](https://www.codeproject.com/KB/database/Visual_SQL_Joins/Visual_SQL_JOINS_orig.jpg)

### **Using the code**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#using-the-code)

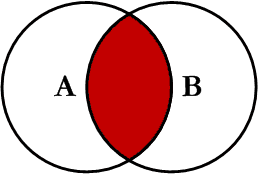
Seven different ways you can return data from two relational tables; excluding cross joins and self referencing joins:

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. OUTER JOIN
5. LEFT JOIN excluding INNER JOIN
6. RIGHT JOIN excluding INNER JOIN
7. OUTER JOIN excluding INNER JOIN



For the sake of this article, 5, 6, and 7 are LEFT EXCLUDING JOIN, RIGHT Excluding JOIN, and OUTER Excluding JOIN, respectively. Some may argue that 5, 6, and 7 are not really joining the two tables, but for simplicity, let’s refer to these as joins because you use a SQL join in each of these queries (but exclude some records with a WHERE clause).

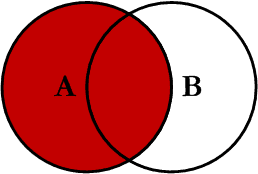
### **INNER JOIN**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#inner-join)



This is the simplest, most understood join and is the most common. This query will return all of the records in the left table (Table\_A) that have a matching record in the right table (Table\_B). This join is written as follows:

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT <select\_list>  FROM Table\_A A  INNER JOIN Table\_B B  ON A.Key = B.Key |

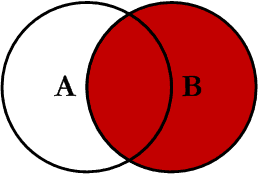
### **LEFT JOIN**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#left-join)



This query will return all of the records in the left table (Table\_A) regardless if any of those records have a match in the right table (Table\_B). It will also return any matching records from the right table. This join is written as follows:

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT <select\_list>  FROM Table\_A A  LEFT JOIN Table\_B B  ON A.Key = B.Key |

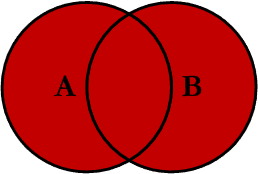
### **RIGHT JOIN**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#right-join)



This query will return all of the records in the right table (Table\_B) regardless if any of those records have a match in the left table (Table\_A). It will also return any matching records from the left table. This join is written as follows:

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT <select\_list>  FROM Table\_A A  RIGHT JOIN Table\_B B  ON A.Key = B.Key |

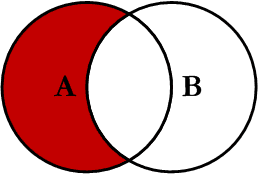
### **OUTER JOIN**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#outer-join)



This Join can also be referred to as a FULL OUTER JOIN or a FULL JOIN. This query will return all of the records from both tables, joining records from the left table (Table\_A) that match records from the right table (Table\_B). This join is written as follows:

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT <select\_list>  FROM Table\_A A  FULL OUTER JOIN Table\_B B  ON A.Key = B.Key |

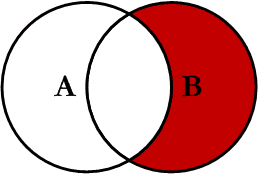
### **LEFT Excluding JOIN**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#left-excluding-join)



This query will return all of the records in the left table (Table\_A) that do not match any records in the right table (Table\_B). This join is written as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT <select\_list>  FROM Table\_A A  LEFT JOIN Table\_B B  ON A.Key = B.Key  WHERE B.Key IS NULL |

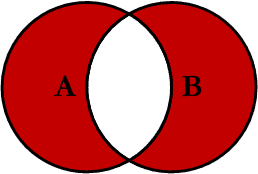
### **RIGHT Excluding JOIN**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#right-excluding-join)



This query will return all of the records in the right table (Table\_B) that do not match any records in the left table (Table\_A). This join is written as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT <select\_list>  FROM Table\_A A  RIGHT JOIN Table\_B B  ON A.Key = B.Key  WHERE A.Key IS NULL |

### **OUTER Excluding JOIN**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#outer-excluding-join)



This query will return all of the records in the left table (Table\_A) and all of the records in the right table (Table\_B) that do not match. I have yet to have a need for using this type of join, but all of the others, I use quite frequently. This join is written as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT <select\_list>  FROM Table\_A A  FULL OUTER JOIN Table\_B B  ON A.Key = B.Key  WHERE A.Key IS NULL OR B.Key IS NULL |

### **Examples**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#examples)

Suppose we have two tables, TABLE\_A and TABLE\_B. The data in these tables are shown below:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | TABLE\_A  PK Value  ---- ----------  1 FOX  2 COP  3 TAXI  6 WASHINGTON  7 DELL  5 ARIZONA  4 LINCOLN  10 LUCENT  TABLE\_B  PK Value  ---- ----------  1 TROT  2 CAR  3 CAB  6 MONUMENT  7 PC  8 MICROSOFT  9 APPLE  11 SCOTCH |

The results of the seven joins are shown below:

**INNER JOIN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | --INNER JOIN  SELECT A.PK AS A\_PK, A.Value AS A\_Value,  B.Value AS B\_Value, B.PK AS B\_PK  FROM Table\_A A  INNER JOIN Table\_B B  ON A.PK = B.PK |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | A\_PK A\_Value B\_Value B\_PK  ---- ---------- ---------- ----  1 FOX TROT 1  2 COP CAR 2  3 TAXI CAB 3  6 WASHINGTON MONUMENT 6  7 DELL PC 7  (5 row(s) affected) |

**LEFT JOIN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | --LEFT JOIN  SELECT A.PK AS A\_PK, A.Value AS A\_Value,  B.Value AS B\_Value, B.PK AS B\_PK  FROM Table\_A A  LEFT JOIN Table\_B B  ON A.PK = B.PK |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | A\_PK A\_Value B\_Value B\_PK  ---- ---------- ---------- ----  1 FOX TROT 1  2 COP CAR 2  3 TAXI CAB 3  4 LINCOLN NULL NULL  5 ARIZONA NULL NULL  6 WASHINGTON MONUMENT 6  7 DELL PC 7  10 LUCENT NULL NULL  (8 row(s) affected) |

**RIGHT JOIN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | --RIGHT JOIN  SELECT A.PK AS A\_PK, A.Value AS A\_Value,  B.Value AS B\_Value, B.PK AS B\_PK  FROM Table\_A A  RIGHT JOIN Table\_B B  ON A.PK = B.PK |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | A\_PK A\_Value B\_Value B\_PK  ---- ---------- ---------- ----  1 FOX TROT 1  2 COP CAR 2  3 TAXI CAB 3  6 WASHINGTON MONUMENT 6  7 DELL PC 7  NULL NULL MICROSOFT 8  NULL NULL APPLE 9  NULL NULL SCOTCH 11  (8 row(s) affected) |

**OUTER JOIN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | --OUTER JOIN  SELECT A.PK AS A\_PK, A.Value AS A\_Value,  B.Value AS B\_Value, B.PK AS B\_PK  FROM Table\_A A  FULL OUTER JOIN Table\_B B  ON A.PK = B.PK |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | A\_PK A\_Value B\_Value B\_PK  ---- ---------- ---------- ----  1 FOX TROT 1  2 COP CAR 2  3 TAXI CAB 3  6 WASHINGTON MONUMENT 6  7 DELL PC 7  NULL NULL MICROSOFT 8  NULL NULL APPLE 9  NULL NULL SCOTCH 11  5 ARIZONA NULL NULL  4 LINCOLN NULL NULL  10 LUCENT NULL NULL  (11 row(s) affected) |

**LEFT Excluding JOIN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | --LEFT EXCLUDING JOIN  SELECT A.PK AS A\_PK, A.Value AS A\_Value,  B.Value AS B\_Value, B.PK AS B\_PK  FROM Table\_A A  LEFT JOIN Table\_B B  ON A.PK = B.PK  WHERE B.PK IS NULL |

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | A\_PK A\_Value B\_Value B\_PK  ---- ---------- ---------- ----  4 LINCOLN NULL NULL  5 ARIZONA NULL NULL  10 LUCENT NULL NULL  (3 row(s) affected) |

**RIGHT Excluding JOIN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | -RIGHT EXCLUDING JOIN  SELECT A.PK AS A\_PK, A.Value AS A\_Value,  B.Value AS B\_Value, B.PK AS B\_PK  FROM Table\_A A  RIGHT JOIN Table\_B B  ON A.PK = B.PK  WHERE A.PK IS NULL |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | A\_PK A\_Value B\_Value B\_PK  ---- ---------- ---------- ----  NULL NULL MICROSOFT 8  NULL NULL APPLE 9  NULL NULL SCOTCH 11  (3 row(s) affected) |

**OUTER Excluding JOIN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | --OUTER EXCLUDING JOIN  SELECT A.PK AS A\_PK, A.Value AS A\_Value,  B.Value AS B\_Value, B.PK AS B\_PK  FROM Table\_A A  FULL OUTER JOIN Table\_B B  ON A.PK = B.PK  WHERE A.PK IS NULL  OR B.PK IS NULL |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | A\_PK A\_Value B\_Value B\_PK  ---- ---------- ---------- ----  NULL NULL MICROSOFT 8  NULL NULL APPLE 9  NULL NULL SCOTCH 11  5 ARIZONA NULL NULL  4 LINCOLN NULL NULL  10 LUCENT NULL NULL  (6 row(s) affected) |

### **Conclusion**[**¶**](https://ugoproto.github.io/ugodoc/Introduction%20to%20SQL%20JOINs/#conclusion)

Note on the OUTER JOIN that the inner joined records are returned first, followed by the right joined records, and then finally the left joined records (at least, that’s how my Microsoft SQL Server did it; this, of course, is without using any ORDER BY statement).