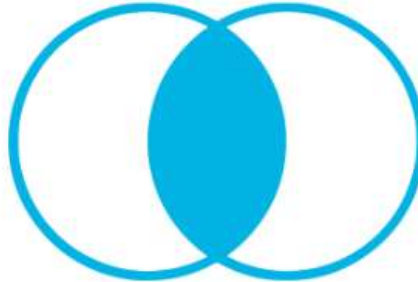


FULL OUTER JOIN

In earlier lessons, we covered inner joins, which produce results for which the join condition is matched in both tables.

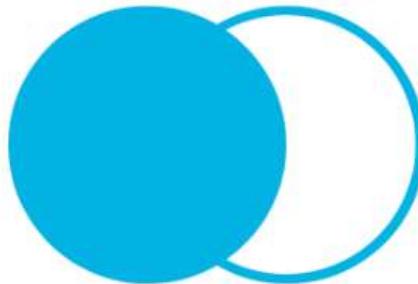
Venn diagrams, which are helpful for visualizing table joins, are provided below along with sample queries. Consider the circle on the left Table A and the circle on the right Table B.



INNER JOIN Venn Diagram

```
SELECT column_name(s)
FROM Table_A
INNER JOIN Table_B ON Table_A.column_name = Table_B.column_name;
```

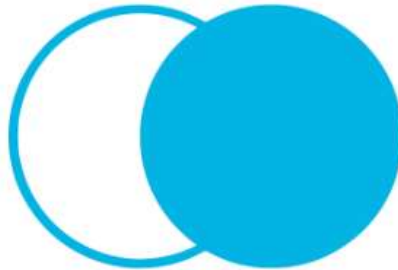
Left joins also include unmatched rows from the left table, which is indicated in the "FROM" clause.



LEFT JOIN Venn Diagram

```
SELECT column_name(s)
FROM Table_A
LEFT JOIN Table_B ON Table_A.column_name = Table_B.column_name;
```

Right joins are similar to left joins, but include unmatched data from the right table -- the one that's indicated in the JOIN clause.



RIGHT JOIN Venn Diagram

```
SELECT column_name(s)
FROM Table_A
RIGHT JOIN Table_B ON Table_A.column_name = Table_B.column_name;
```

In some cases, you might want to include unmatched rows from *both* tables being joined. You can do this with a full outer join.



FULL OUTER JOIN Venn Diagram

```
SELECT column_name(s)
FROM Table_A
FULL OUTER JOIN Table_B ON Table_A.column_name = Table_B.column_name;
```

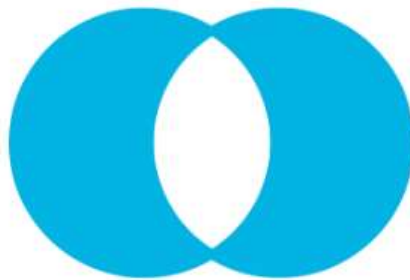
A common application of this is when joining two tables on a timestamp. Let's say you've got one table containing the number of *item 1* sold each day, and another containing the number of *item 2* sold. If a certain date, like January 1, 2018, exists in the left table but not the right, while another date, like January 2, 2018, exists in the right table but not the left:

- a left join would drop the row with January 2, 2018 from the result set
- a right join would drop January 1, 2018 from the result set

The only way to make sure both January 1, 2018 and January 2, 2018 make it into the results is to do a full outer join. A full outer join returns unmatched records in each table with null values for the columns that came from the opposite table.

If you wanted to return unmatched rows only, which is useful for some cases of data assessment, you can isolate them by adding the following line to the end of the query:

```
WHERE Table_A.column_name IS NULL OR Table_B.column_name IS NULL
```



FULL OUTER JOIN with WHERE A.Key IS NULL OR B.Key IS NULL Venn Diagram