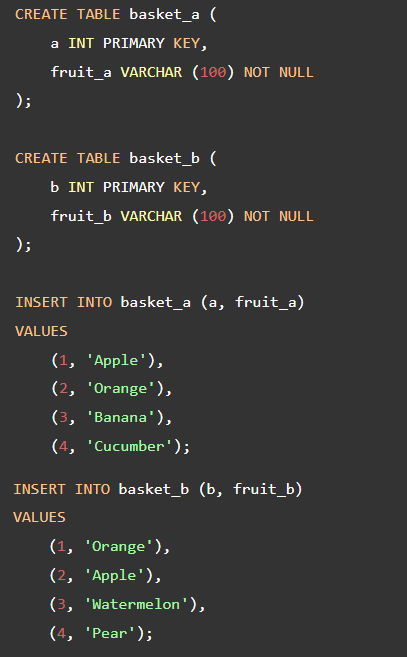
PostgreSQL Joins

PostgreSQL join is used to combine columns from one ([self-join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-self-join/)) or more tables based on the values of the common columns between related tables. The common columns are typically the [primary key](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-primary-key/) columns of the first table and [foreign key](https://www.postgresqltutorial.com/postgresql-foreign-key/) columns of the second table.

PostgreSQL supports [inner join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-inner-join/),[left join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-left-join/), [right join](https://www.postgresqltutorial.com/postgresql-right-join/), [full outer join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-full-outer-join/), [cross join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-cross-join/), [natural join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-natural-join/), and a special kind of join called [self-join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-self-join/).

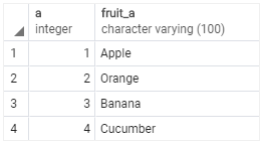
## **Setting up sample tables**

Suppose you have two tables called basket\_a and basket\_b that store fruits:

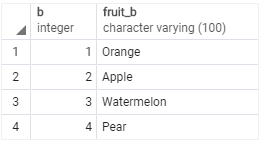


The tables have some common fruits such as apple and orange.

The following statement returns data from the basket\_a table:

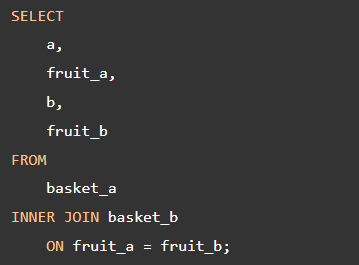


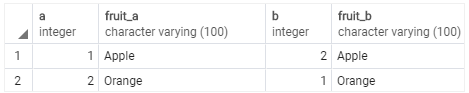
And the following statement returns data from the basket\_b table:



## **PostgreSQL inner join**

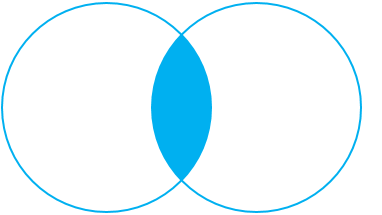
The following statement joins the first table (basket\_a) with the second table (basket\_b) by matching the values in the fruit\_a and fruit\_b columns:





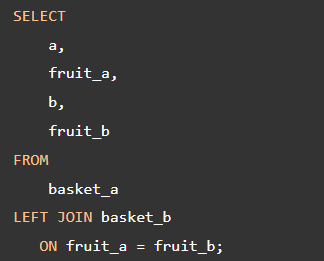
The inner join examines each row in the first table (basket\_a). It compares the value in the fruit\_a column with the value in the fruit\_b column of each row in the second table (basket\_b). If these values are equal, the inner join creates a new row that contains columns from both tables and adds this new row the result set.

The following Venn diagram illustrates the inner join:



## **PostgreSQL left join**

The following statement uses the left join clause to join the basket\_a table with the basket\_b table. In the left join context, the first table is called the left table and the second table is called the right table.



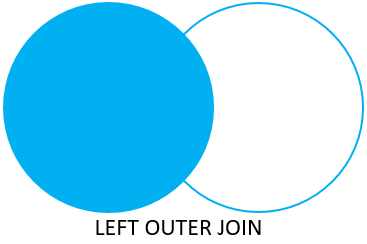


The left join starts selecting data from the left table. It compares values in the fruit\_a column with the values in the fruit\_b column in the basket\_b table.

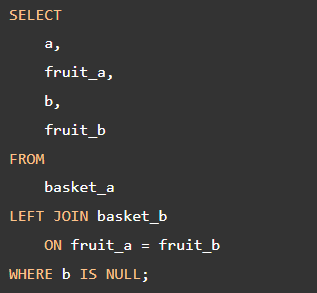
If these values are equal, the left join creates a new row that contains columns of both tables and adds this new row to the result set. (see the row #1 and #2 in the result set).

In case the values do not equal, the left join also creates a new row that contains columns from both tables and adds it to the result set. However, it fills the columns of the right table (basket\_b) with null. (see the row #3 and #4 in the result set).

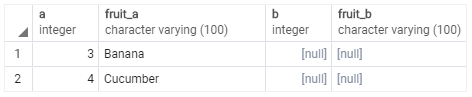
The following Venn diagram illustrates the left join:



To select rows from the left table that do not have matching rows in the right table, you use the left join with a [WHERE](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-where/) clause. For example:

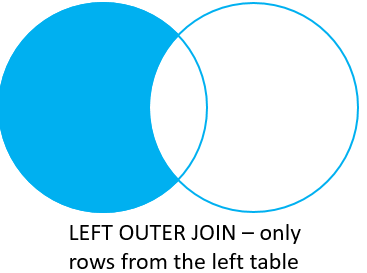


The output is:





The following Venn diagram illustrates the left join that returns rows from the left table that do not have matching rows from the right table:



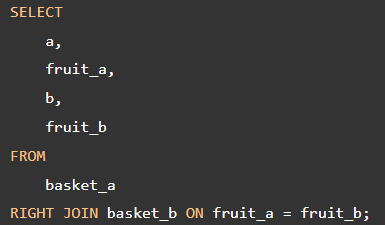
## **PostgreSQL right join**

The [right join](https://www.postgresqltutorial.com/postgresql-right-join/) is a reversed version of the left join. The right join starts selecting data from the right table. It compares each value in the fruit\_b column of every row in the right table with each value in the fruit\_a column of every row in the fruit\_a table.

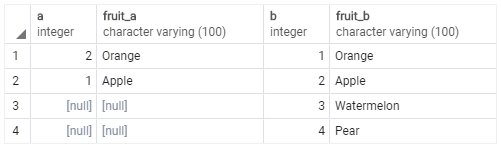
If these values are equal, the right join creates a new row that contains columns from both tables.

In case these values are not equal, the right join also creates a new row that contains columns from both tables. However, it fills the columns in the left table with NULL.

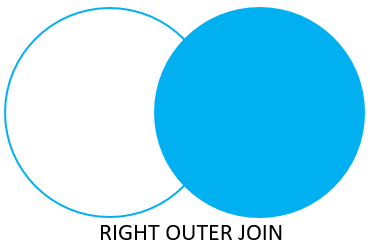
The following statement uses the right join to join the basket\_a table with the basket\_b table:



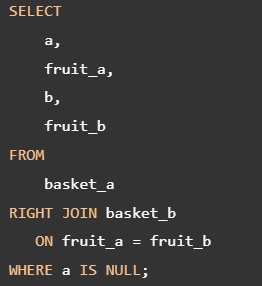
Here is the output:

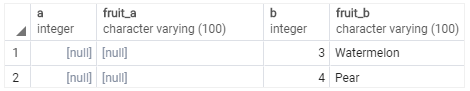


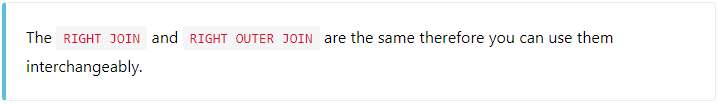
The following Venn diagram illustrates the right join:



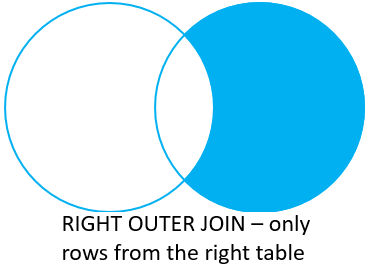
Similarly, you can get rows from the right table that do not have matching rows from the left table by adding a WHERE clause as follows:





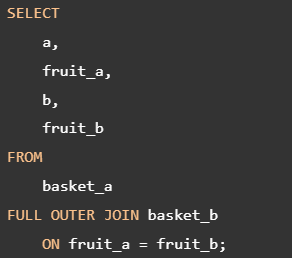


The following Venn diagram illustrates the right join that returns rows from the right table that do not have matching rows in the left table:

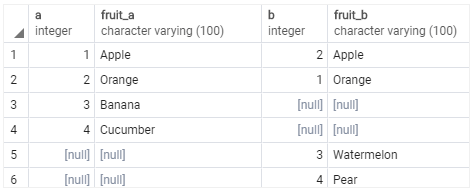


## **PostgreSQL full outer join**

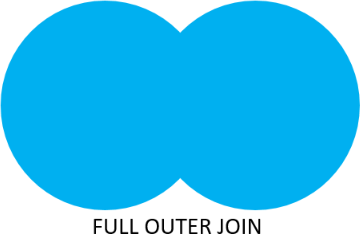
The [full outer join](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-full-outer-join/) or full join returns a result set that contains all rows from both left and right tables, with the matching rows from both sides if available. In case there is no match, the columns of the table will be filled with NULL.



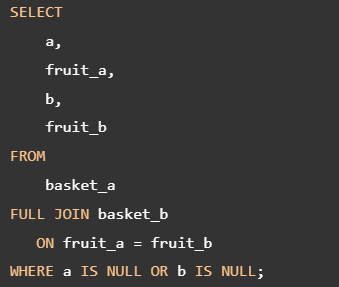
Output:



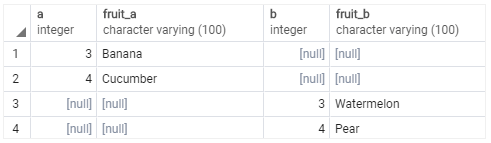
The following Venn diagram illustrates the full outer join:



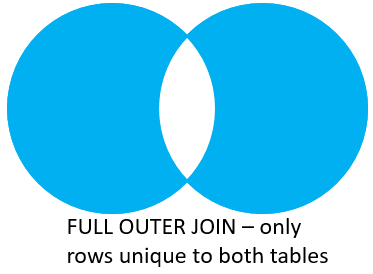
To return rows in a table that do not have matching rows in the other, you use the full join with a WHERE clause like this:



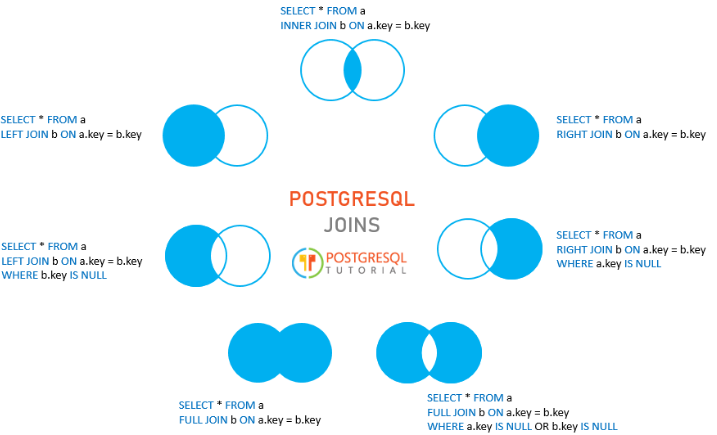
Here is the result:



The following Venn diagram illustrates the full outer join that returns rows from a table that do not have the corresponding rows in the other table:



The following picture shows all the PostgreSQL joins that we discussed so far with the detailed syntax:

  
References

https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-joins/