Joins – shows you a brief overview of joins in PostgreSQL.

PostgreSQL `Joins` clause statement syntax

PostgreSQL join is used to combine columns from one (self-join) or more tables based on the values of the common columns between the tables. The common columns are typically the primary key columns of the first table and foreign key columns of the second table.

Setting up sample tables

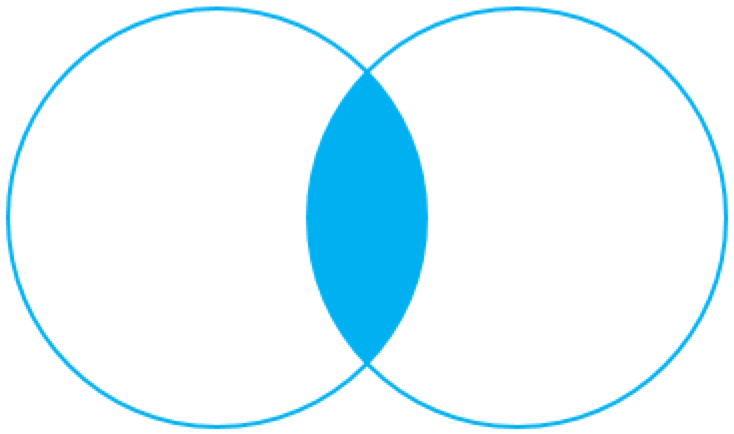
Suppose we have two tables called `basket_a` and `basket_b` that stores fruits:

```
CREATE TABLE basket_a (
  id INT PRIMARY KEY,
  fruit VARCHAR (100) NOT NULL
);

CREATE TABLE basket_b (
  id INT PRIMARY KEY,
  fruit VARCHAR (100) NOT NULL
);

INSERT INTO basket_a (id, fruit)
VALUES
   (1, 'Apple'),
   (2, 'Orange'),
   (3, 'Banana'),
   (4, 'Cucumber');

INSERT INTO basket_b (id, fruit)
VALUES
   (1, 'Orange'),
   (2, 'Apple'),
   (3, 'Watermelon'),
   (4, 'Pear');
```

The tables have some common fruits such as apple and orange. Let's call the `basket_a` is the left table and `basket_b` is the right table.

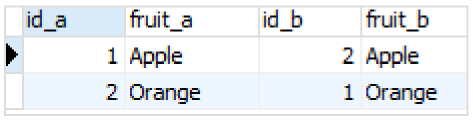
Joins – shows you a brief overview of joins in PostgreSQL.

PostgreSQL inner join



The following statement joins the left table with the right table using the values in the `fruit` column:

```
SELECT
  a.id id_a,
  a.fruit fruit_a,
  b.id id_b,
  b.fruit fruit_b
FROM
  basket_a a
INNER JOIN basket_b b ON a.fruit = b.fruit;
```

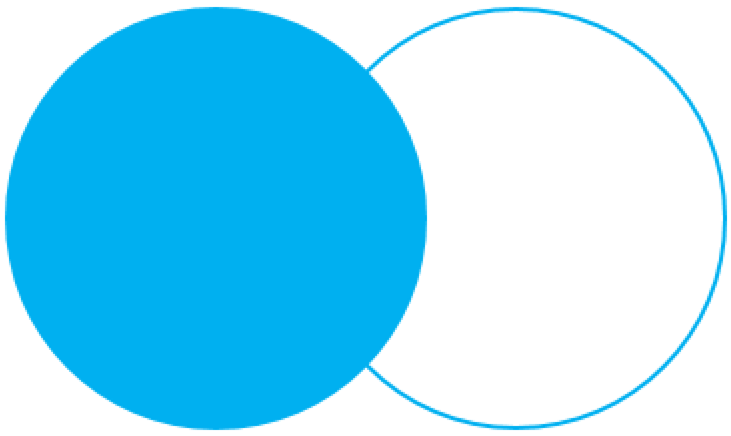


id_a	fruit_a	id_b	fruit_b
1	Apple	2	Apple
2	Orange	1	Orange

Joins – shows you a brief overview of joins in PostgreSQL.

PostgreSQL left join

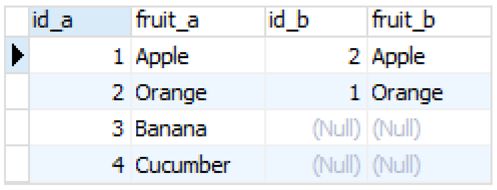
The left join returns a complete set of rows from the left table with the matching rows if avail able from the right table. If there is no match, the right side will have null values.



LEFT OUTER JOIN

The following statement joins the left table with the right table using left join (or left outer join):

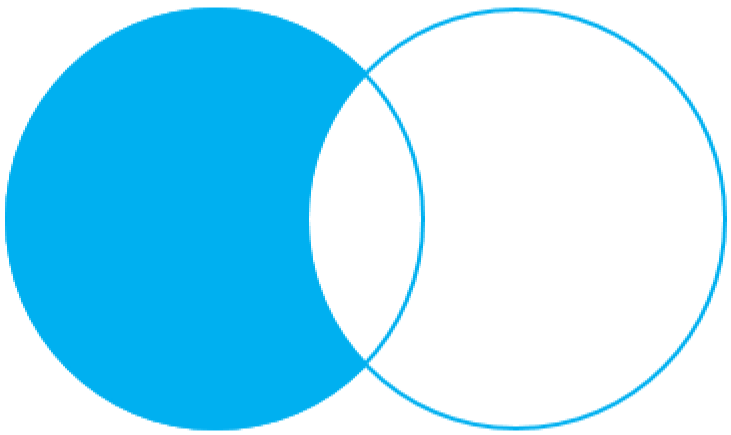
```
SELECT
  a.id id_a,
  a.fruit fruit_a,
  b.id id_b,
  b.fruit fruit_b
FROM
  basket_a a
LEFT JOIN basket_b b ON a.fruit = b.fruit;
```



id_a	fruit_a	id_b	fruit_b
1	Apple	2	Apple
2	Orange	1	Orange
3	Banana	(Null)	(Null)
4	Cucumber	(Null)	(Null)

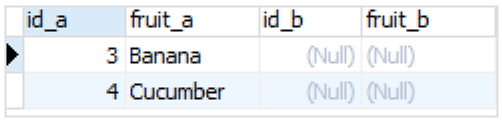
Joins – shows you a brief overview of joins in PostgreSQL.

left join with only rows from the left table:


LEFT OUTER JOIN – only rows from the left table

```
SELECT
  a.id id_a,
  a.fruit fruit_a,
  b.id id_b,
  b.fruit fruit_b
FROM
  basket_a a
LEFT JOIN basket_b b ON a.fruit = b.fruit
WHERE b.id IS NULL;
```

The output is:

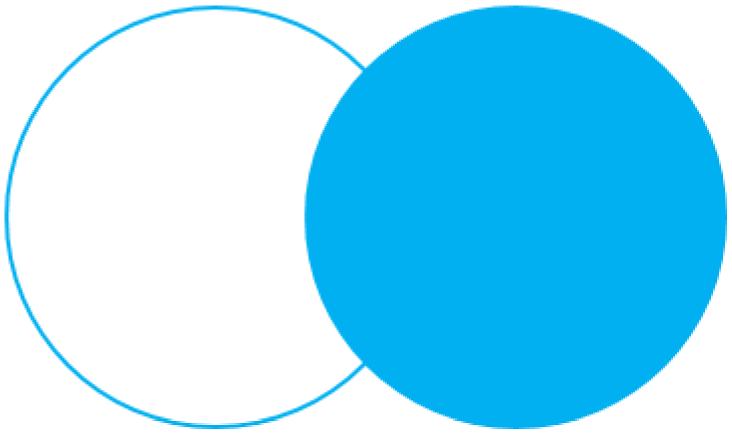


id_a	fruit_a	id_b	fruit_b
3	Banana	(Null)	(Null)
4	Cucumber	(Null)	(Null)

Joins – shows you a brief overview of joins in PostgreSQL.

PostgreSQL right join

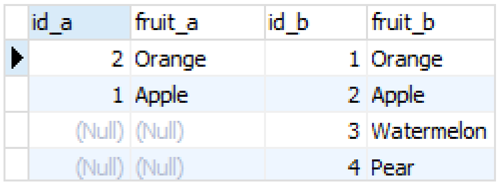
contains all rows from the right table with matching rows from the left table. If there is no match, the left side will contain null values.



RIGHT OUTER JOIN

The following statement performs the right join between the left and the right tables:

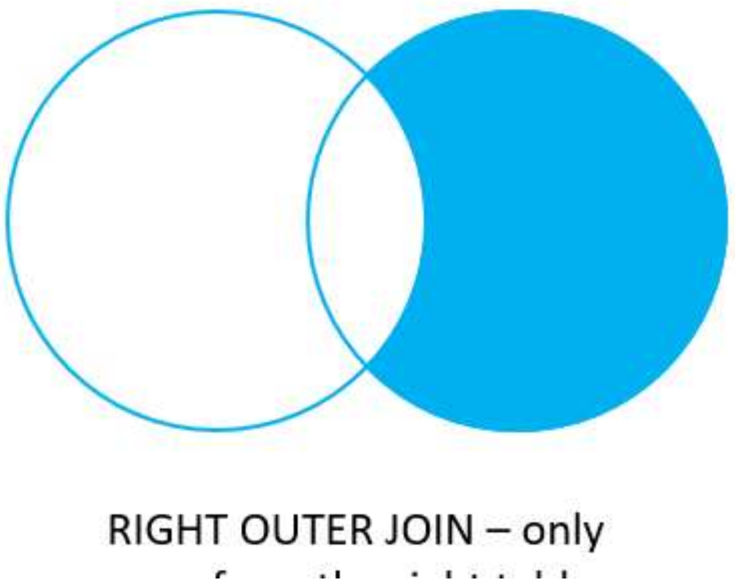
```
SELECT
  a.id id_a,
  a.fruit fruit_a,
  b.id id_b,
  b.fruit fruit_b
FROM
  basket_a a
RIGHT JOIN basket_b b ON a.fruit = b.fruit;
```



id_a	fruit_a	id_b	fruit_b
2	Orange	1	Orange
1	Apple	2	Apple
(Null)	(Null)	3	Watermelon
(Null)	(Null)	4	Pear

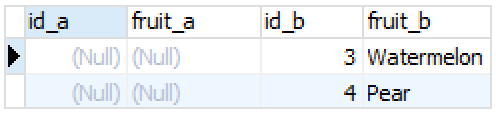
Joins – shows you a brief overview of joins in PostgreSQL.

right join with only rows from the right table:



Similarly, you can get only rows from the right table but not from the left table by adding a `WHERE`clause as follows:

```
SELECT
  a.id id_a,
  a.fruit fruit_a,
  b.id id_b,
  b.fruit fruit_b
FROM
  basket_a a
RIGHT JOIN basket_b b ON a.fruit = b.fruit
WHERE a.id IS NULL;
```

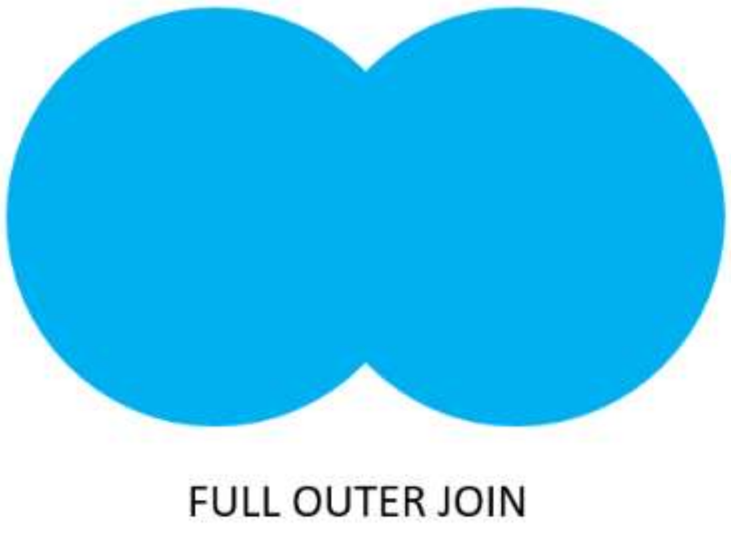


id_a	fruit_a	id_b	fruit_b
(Null)	(Null)	3	Watermelon
(Null)	(Null)	4	Pear

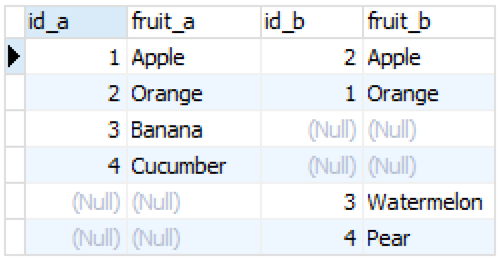
Joins – shows you a brief overview of joins in PostgreSQL.

PostgreSQL full outer join

The full outer join or full join produces a result set that contains all rows from both the left and right tables, with the matching rows from both sides where available. If there is no match, the missing side contains null values.



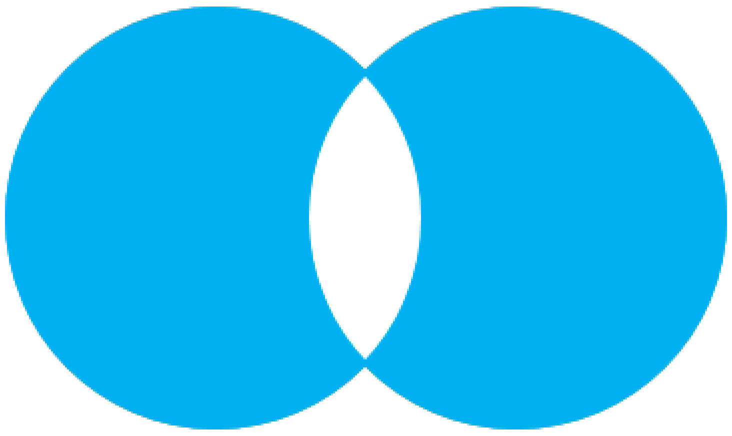
```
SELECT
  a.id id_a,
  a.fruit fruit_a,
  b.id id_b,
  b.fruit fruit_b
FROM
  basket_a a
FULL OUTER JOIN basket_b b ON a.fruit = b.fruit;
```



id_a	fruit_a	id_b	fruit_b
1	Apple	2	Apple
2	Orange	1	Orange
3	Banana	(Null)	(Null)
4	Cucumber	(Null)	(Null)
(Null)	(Null)	3	Watermelon
(Null)	(Null)	4	Pear

Joins – shows you a brief overview of joins in PostgreSQL.

Unique from both tables:

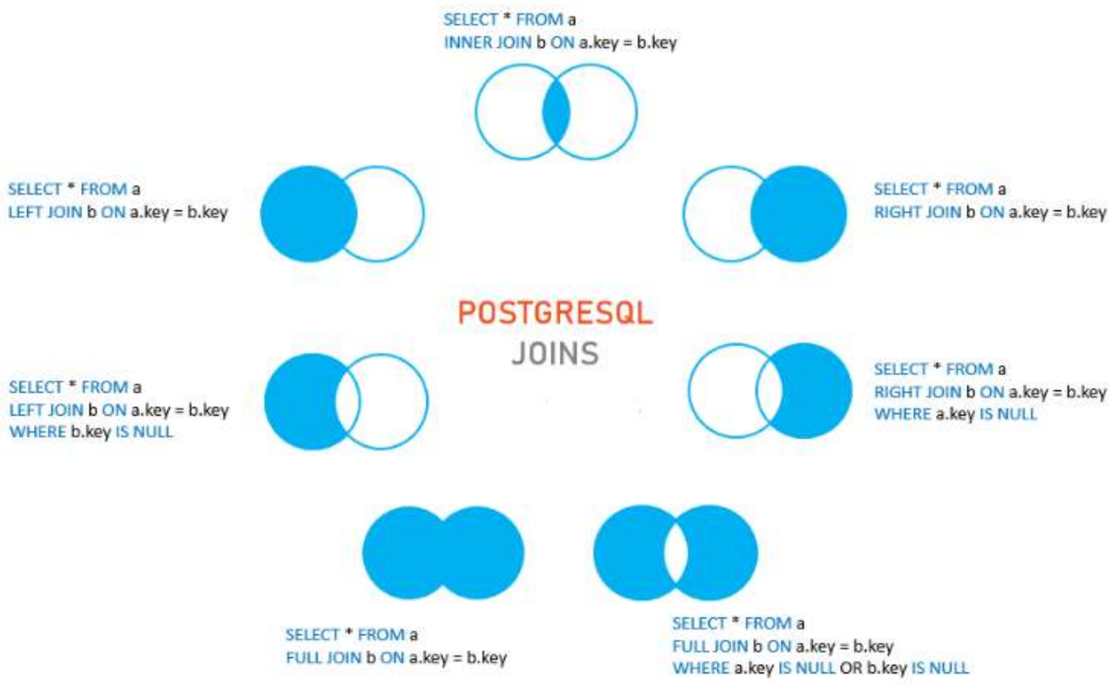


FULL OUTER JOIN – only rows unique to both tables

```
SELECT
  a.id id_a,
  a.fruit fruit_a,
  b.id id_b,
  b.fruit fruit_b
FROM
  basket_a a
FULL JOIN basket_b b ON a.fruit = b.fruit
WHERE a.id IS NULL OR b.id IS NULL;
```

id_a	fruit_a	id_b	fruit_b
3	Banana	(null)	(null)
4	Cucumber	(null)	(null)
(null)	(null)	3	Watermelon
(null)	(null)	4	Pear

Joins – shows you a brief overview of joins in PostgreSQL.


SELECT * FROM a
INNER JOIN b ON a.key = b.key
SELECT * FROM a
LEFT JOIN b ON a.key = b.key
SELECT * FROM a
RIGHT JOIN b ON a.key = b.key
POSTGRESQL
JOINS
SELECT * FROM a
LEFT JOIN b ON a.key = b.key
WHERE b.key IS NULL
SELECT * FROM a
RIGHT JOIN b ON a.key = b.key
WHERE a.key IS NULL
SELECT * FROM a
FULL JOIN b ON a.key = b.key
SELECT * FROM a
FULL JOIN b ON a.key = b.key
WHERE a.key IS NULL OR b.key IS NULL