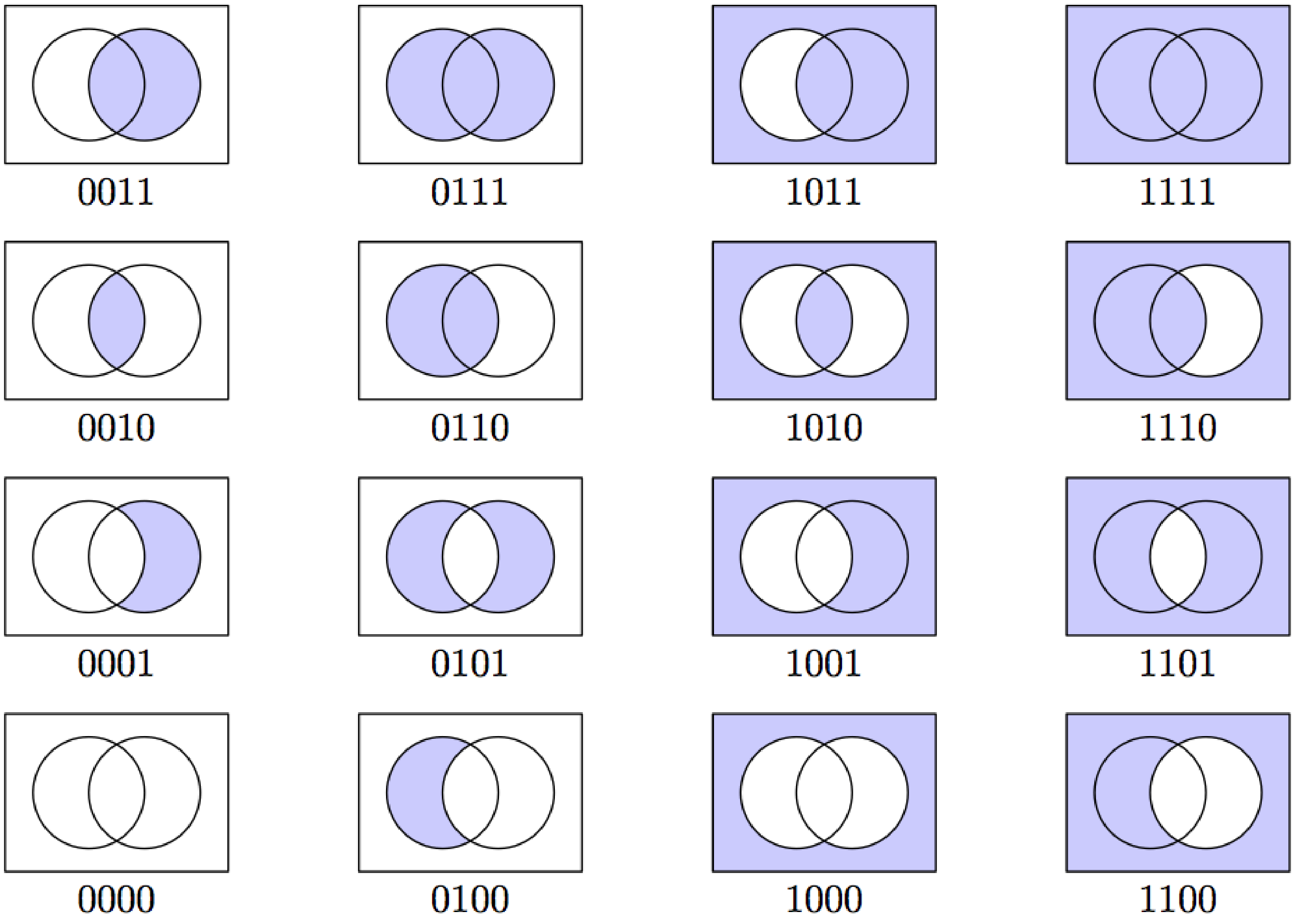
Lecture 06 - Joins

Feb 1



There are 16 of them.

Let’s work through all of them in PostgreSQL.

Tables and data for exampels:

```
drop table if exists marketing_data ;
create table if not exists marketing_data (
    name text,
    amount int,
    phone_no text
);
drop table if exists do_not_call_list ;
create table if not exists do_not_call_list (
    phone_no text,
    when_added timestamp
);
insert into marketing_data (name, amount, phone_no) values
    ( 'philip', 5000, '720-209-7888')
    ,('bob', 25000, '505-444-1212')
    ,('mark', 200, '307-338-1212')
    ,('dave', -200, '307-130-1212')
;
insert into do_not_call_list ( phone_no, when_added ) values
     ('505-444-1212','2021-02-04T13:14:15')
    ,('505-222-1212','2020-01-03T13:14:15')
;
```

```
m4_ omment(

drop table if exists marketing_data ;
create table if not exists marketing_data (
        name text,
        phone_no text
);
drop table if exists do_not_call_list ;
create table if not exists do_not_call_list (
        phone_no text
);
insert into marketing_data (name, phone_no) values
        ( 'philip', '720-209-7888')
        ,('bob', '505-444-1212')
        ,('mark', '307-338-1212')
;
insert into do_not_call_list ( phone_no ) values
         ('505-444-1212')
;

)
```

The JOIN types in Postgres are

- INNER JOIN, or just JOIN - this is the most common.
- The CROSS JOIN
- The LEFT OUTER JOIN
- The RIGHT OUTER JOIN
- The FULL OUTER JOIN

and we can use them do do most of the possible ven diagrams above. The ven diagrams that we can not do are also useful - so I will show you all 16.

On the web: https://www.postgresqltutorial.com/postgresql-joins/ or https://www.tutorialspoint.com/postgresql/postgresql_using_joins.htm .

0 0000 - the set of NO data.

```
select 'a' as "x"
        where 1 = 2
;
```

Useful for checking if a program has a connection to the database.

3 0011 and 6 0110 - Select from a single table.

This is the not-joined condition. Also see below.

```
select * from marketing_data;
select * from do_not_call_list;
```

Output:

```
  name  | amount |   phone_no
--------+--------+--------------
 philip |   5000 | 720-209-7888
 bob    |  25000 | 505-444-1212
 mark   |    200 | 307-338-1212
 dave   |   -200 | 307-130-1212
(4 rows)

   phone_no   |     when_added
--------------+---------------------
 505-444-1212 | 2021-02-04 13:14:15
 505-222-1212 | 2020-01-03 13:14:15
(2 rows)
```

2 0010 - Intersection of 2 sets

This is the `inner` join between 2 sets of data.

```
select name
    from marketing_data as t1
        join do_not_call_list as t2 on ( t1.phone_no = t2.phone_no)
;
```

or INNER JOIN

```
select name
    from marketing_data as t1
        inner join do_not_call_list as t2 on ( t1.phone_no = t2.phone_no)
;
```

or in the where clause

```
select t1.name
    from marketing_data as t1, do_not_call_list as t2
    where t1.phone_no = t2.phone_no
;
```

Output:

```
 name
------
 bob
(1 row)
```

15 - 1111 - Select with no conditions.

This is useful for doing calculations in PostgreSQL.

```
select 12 * 33;
```

(In Oracle....)

```
select 12 * 33 from dual;
```

Or generate a series:

```
SELECT generate_series(3, 100) / 3;
```

8 1000 - Except (minus) Operations.

(this also coves 9, 10, 11, 12, 13, 14 when the 2nd select (after the minus) is one of the other join that we have already shown)

Remove from a fixed list a set of things

```
select t1.phone_no
    from marketing_data as t1
except
    select t2.phone_no
    from do_not_call_list as t2
;
```

In Oracle, DB2, MS-SQL-Server, MariaDB and mySQL:

```
select t1.phone_no
    from marketing_data as t1
minus
    select t2.phone_no
    from do_not_call_list as t2
```

```
;
```

This is really good for things like products that are out of stock!

Output:

```
   phone_no
--------------
 307-130-1212
 720-209-7888
 307-338-1212
(3 rows)
```

1 0001

This gives us just the numbes in the do-not-call list that are not in the set of marketing data.

```
select name, when_added
    from marketing_data as t1
        right join do_not_call_list as t2 on ( t1.phone_no = t2.phone_no)
        where t1.phone_no is null
;
```

Output:

```
 name |     when_added
------+---------------------
      | 2020-01-03 13:14:15
(1 row)
```

4 0100

This is like 1 or the “minus” above with the tables switched.

```
select name, t1.phone_no, when_added
    from marketing_data as t1
        left join do_not_call_list as t2 on ( t1.phone_no = t2.phone_no)
        where t2.phone_no is null
;
```

Output:

```
  name  |   phone_no   | when_added
--------+--------------+------------
 philip | 720-209-7888 |
 mark   | 307-338-1212 |
 dave   | 307-130-1212 |
(3 rows)
```

5 0101

```
select name, when_added
    from marketing_data as t1
        full outer join do_not_call_list as t2 on ( t1.phone_no = t2.phone_no)
        where t1.phone_no is null
            or t2.phone_no is null
;
```

Output:

```
  name  |     when_added
--------+---------------------
 dave   |
 mark   |
        | 2020-01-03 13:14:15
 philip |
(4 rows)
```

7 0111 - full outer join

```
select name, when_added
    from marketing_data as t1
        full join do_not_call_list as t2 on ( t1.phone_no = t2.phone_no)
;
```

FULL OUTER

```
select name, when_added
    from marketing_data as t1
        full outer join do_not_call_list as t2 on ( t1.phone_no = t2.phone_no)
;
```

Output:

```
  name  |     when_added
--------+---------------------
 dave   |
 mark   |
        | 2020-01-03 13:14:15
 bob    | 2021-02-04 13:14:15
 philip |
(5 rows)
```

Cross Join

This is the one that is not in the VEN diagram - that is all of the left table with all of the right table. This is more of a “join-all-together so I can post filer” or “join-all-together so I can construct new data from it” operation.

Given our little set of data this makes no sense. Oh… Well…

```
select *
    from marketing_data as t1
        cross join do_not_call_list as t2
;
```

or (this is the way you would do this in Oracle, mySQL, MariaDB)

```
select *
    from marketing_data as t1
        , do_not_call_list as t2
;
```

Output:

```
  name  | amount |   phone_no   |   phone_no   |     when_added
--------+--------+--------------+--------------+---------------------
 philip |   5000 | 720-209-7888 | 505-444-1212 | 2021-02-04 13:14:15
 bob    |  25000 | 505-444-1212 | 505-444-1212 | 2021-02-04 13:14:15
 mark   |    200 | 307-338-1212 | 505-444-1212 | 2021-02-04 13:14:15
 dave   |   -200 | 307-130-1212 | 505-444-1212 | 2021-02-04 13:14:15
 philip |   5000 | 720-209-7888 | 505-222-1212 | 2020-01-03 13:14:15
 bob    |  25000 | 505-444-1212 | 505-222-1212 | 2020-01-03 13:14:15
 mark   |    200 | 307-338-1212 | 505-222-1212 | 2020-01-03 13:14:15
 dave   |   -200 | 307-130-1212 | 505-222-1212 | 2020-01-03 13:14:15
(8 rows)
```