

# Accesos rápidos

ctrl + l: limpia la consola

## Comandos

COMANDOS	SIGNIFICADO
--comentario	
create table nombreElemento ( );	
nombreAtributo varchar (numeroCaracteresMaximos),	varchar es cualquier tipo de carácter
primary key (nombreClavePrimaria)	
foreign key (nombreClaveReferencia) reference elementoDeReferencia (nombreClaveReferencia)	
insert into nombreTabla values (‘valor’,‘valor2’,null);	para insertar valores. el null se puede poner o no (va entre comillas). todo los varchar van entre comillas simples.
<pre>postgres=&gt; select * from equipo; codequ   nomequ   numx -----+-----+----- e1        cuspedrinos   e2        gambusinos    e3        croques       e4        reboludos     e5        novo          (5 rows)  postgres=&gt;</pre>	ver todos los valores
*	te muestra toda la tabla de tipos
from	de
select	target
select * from nombreTaboa	consulta a totalidad

```

postgres=> select * from equipo;
codequ | nomequ | numx
-----+-----+-----
e5     | novo  |
e1     | cuspedrinos | 5
e3     | croques | 3
e4     | reboludos | 5
e2     | gambusinos | 2
(5 rows)

postgres=> insert into equipo values ('e6','vello',null);
INSERT 0 1
postgres=> select * from equipo;
codequ | nomequ | numx
-----+-----+-----
e5     | novo  |
e1     | cuspedrinos | 5
e3     | croques | 3
e4     | reboludos | 5
e2     | gambusinos | 2
e6     | vello  |
(6 rows)

```

para insertar un nuevo valor  
si quiero borrarlo de momento solo lo sabria  
hacer volviendo a ejecutarlo.

tambien poner el punto y coma despues de  
darle a enter ( si le doy sin querer)z

select \* from nombreTabla

```

(15 rows)

postgres=> select nomx, codequ from xogador;
nomx   | codequ
-----+-----
luis   | e1
juan   | e3
pedro  | e4
alberto | e2
juan   | e1
luis   | e1
carlos | e3
pedro  | e4
bent   | e4
manuel | e4
antonio | e3
eliseo | e4
juan   | e2
federico | e1
leon   | e1
(15 rows)

```

para mostrar solo las columnas que me  
interesan. en el orden que quiera.

se llama consulta a proyeccion.

li x

en x arrastrar el archivo .sql

```

postgres=> select * from equipo where numx>2;
codequ | nomequ | numx
-----+-----+-----
e1     | cuspedrinos | 5
e3     | croques | 3
e4     | reboludos | 5
(3 rows)

```

consulta condicional:

muestrame los equipos cuyo numero de  
jugador sea mayor que 2.

la condicion es todo lo que va despues del  
where.

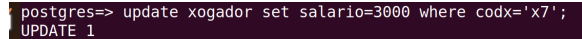
null no es cero, es indefinido.

cada vez que añado una nueva condicion  
lo hago con and.

select nomx from xogador where  
salario>2000 and salario<2500;

select \* from nombreTabla where

consulta condicional: para mostrar los nulos

nombreColumna is null;	
select * from nombreTabla where nombreColumna is not null;	consulta condicional: para consultar todos lo que no son nulos
!=	no igual. tambien se puede escribir <>
order by salario; order by salario desc;	ordenar según salario
nombreColumna or nombreColumna2	escoger una o otra
\d	listar tablas. muestra todas las tablas creadas
\d nombreTabla	te la descripcion de una de las tablas
select sum(nomeColumna) from nomeTabla	suma la columna entera
select count(nomeColumna) from nomeTabla	las null no las cuenta
coalesce(nomeColumna, 0)	los valores que sean null de esa columna los convierte a cero
select operacionSuma	operacionSuma=5 + 3 te va a sumar un y poner un 8
...having...	se usa para poner una condicion entre grupos. (cd se pone group by)  NO ES WHERE
delete from nomeTabla where nomeColumna = 'identificadorQueQuieroBorrar'	Ejemplo: delete from interven codpar = 'c1' and codc = 'p1' borraria la fila de la tabla interven donde el codpar sea c1 y el codc sea p1.
update nombreTabla set nomeCampo = 3000;	modificar el valor de un campo  ej: pongo el salario en 3000 en donde codx sea igual a x7. si hay un null que entra en la condicion, no se modifica el salario de ese. 
substr(cadena,posicion,cantidad)	extraer caracteres a partir de una posicion  a partir de que posicion quieres extraer, los caracteres que quieres extraer de la cadena

	<pre>postgres=&gt; select substr('ola que tal',5,2); substr ----- qu (1 row)</pre>
strpos(cadenaDondeBuscar, 'caracter')	<p>Me devuelve la posicion a partir de un caracter. por ejemplo me puede servir para extraer caracteres de un correo electronico sin que me cuente lo de @gmail...</p> <pre>postgres=&gt; select substr(email,strpos(email,'@')) from cliente; substr ----- @hotmail.com @gmail.com @hotmail.com @gmail.com @hotmail.com @gmail.com @yahoo.com @gmail.com (8 rows)</pre>
TO_NUMBER(cadenaTexto, '99')	<p>el '99' es el numero de numeros qu enos mostrara. si es '9999' serian 4</p>

## Categorías

### Consultas condicionales

```
postgres=> select * from xogador;
codx |  nomx  | codequ | salario | comision |  datan
-----+-----+-----+-----+-----+-----
x1   | luis   | e1     | 2000    |          | 1990-01-03
x2   | juan   | e3     | 1500    | 0        | 1991-04-13
x3   | pedro  | e4     | 1000    |          | 1990-11-12
x4   | alberto | e2     | 1000    | 5        | 1990-07-01
x5   | juan   | e1     | 1200    | 10       | 1990-08-23
x6   | luis   | e1     | 800     | 20       | 1990-09-12
x7   | carlos | e3     |         |          | 1990-10-08
x8   | pedro  | e4     | 3000    | 5        | 1990-06-07
x9   | bent   | e4     | 1500    | 10       | 1990-04-23
x10  | manuel | e4     |         | 10       | 1990-12-25
x11  | antonio | e3     | 2000    |          | 1990-06-16
x12  | eliseo | e4     | 1350    | 10       | 1990-05-14
x13  | juan   | e2     | 2000    | 5        | 1990-04-07
x14  | federico | e1     | 1000    | 10       | 1990-01-05
x15  | leon   | e1     |         |          | 1990-11-27
(15 rows)
```

```
postgres=> select nomx from xogador where salario>2000;
nomx
-----
pedro
(1 row)
```

select nomx from xogador where salario>2000 and salario<2500;

select nomx from xogador where salario='e1';

```
postgres=> select nomx,salario from xogador where salario>1000 and salario<2500 and nomx!='juan' order by salario desc;
nomx | salario
-----+-----
luís  |    2000
antonio |    2000
bent   |    1500
eliseo |    1350
(4 rows)
```

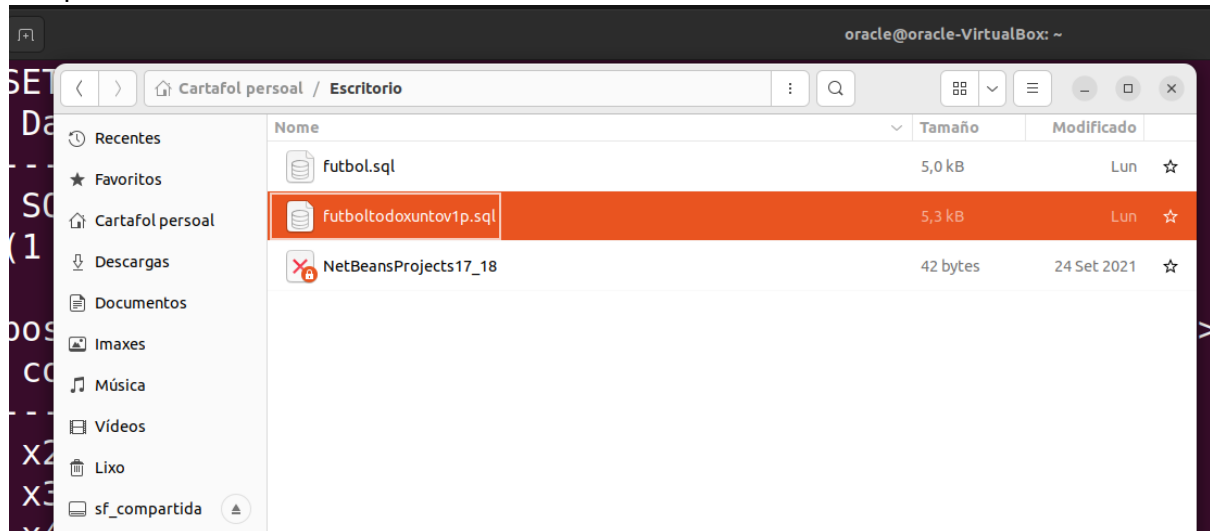
```
postgres=> select nomx,salario from xogador where salario>1000 or salario is null;
nomx | salario
-----+-----
luís  |    2000
juan  |    1500
juan  |    1200
carlos |
pedro |    3000
bent   |    1500
manuel |
antonio |    2000
eliseo |    1350
juan  |    2000
leon  |
(11 rows)
```

otro ejemplo

```
postgres=> select codx,nomx,datan from xogador where datan>'7/5/1990';
codx | nomx | datan
-----+-----+-----
x2   | juan | 1991-04-13
x3   | pedro | 1990-11-12
x4   | alberto | 1990-07-01
x5   | juan | 1990-08-23
x6   | luís  | 1990-09-12
x7   | carlos | 1990-10-08
x8   | pedro | 1990-06-07
x10  | manuel | 1990-12-25
x11  | antonio | 1990-06-16
x12  | eliseo | 1990-05-14
x15  | leon  | 1990-11-27
(11 rows)
```

si te das cuenta el formato de fecha no es el que queremos. esto es porque el terminal tiene un formato predeterminado para mostrar los datos en general.

aunque el terminal este basado en este documento:



no guarda el formato de presentación que hemos aplicado dentro del documento:

```
1
2 set datestyle = "ISO, DMY";
3 set datestyle to DMY, SQL;
4 show datestyle;
5
6 drop table if exists adestra cascade;
7 drop table if exists xoga cascade ;
8 drop table if exists interven cascade ;
9 drop table if exists colexiado cascade ;
10 drop table if exists partido cascade ;
```

para que me ponga ese formato tengo que copiar esas lineas y pegarlo en el terminal

```
1
2 set datestyle = "ISO, DMY";
3 set datestyle to DMY, SQL;
4 show datestyle;
5
6 drop table if exists adestra cascade;
7 drop table if exists xoga cascade ;
8 drop table if exists interven cascade ;
9 drop table if exists colexiado cascade ;
10 drop table if exists partido cascade ;
```

```
postgres=> select codx,nomx,datan from xogador where datan>'7/5/1990';
```

codx	nomx	datan
x2	juan	13/04/1991
x3	pedro	12/11/1990
x4	alberto	01/07/1990
x5	juan	23/08/1990
x6	luis	12/09/1990
x7	carlos	08/10/1990
x8	pedro	07/06/1990
x10	manuel	25/12/1990
x11	antonio	16/06/1990
x12	eliseo	14/05/1990
x15	leon	27/11/1990

(11 rows)

otro ejem

```
postgres=> select codx,nomx,salario from xogador order by nomx,salario;
```

codx	nomx	salario
x4	alberto	1000
x11	antonio	2000
x9	bent	1500
x7	carlos	
x12	eliseo	1350
x14	federico	1000
x5	juan	1200
x2	juan	1500
x13	juan	2000
x15	leon	
x6	luis	800
x1	luis	2000
x10	manuel	
x3	pedro	1000
x8	pedro	3000

(15 rows)

## Operaciones

si se suma un nulo va a dar un nulo, las operaciones con nulo dan nulo

```
postgres=> select codx,nomx,salario+comision from xogador;
```

codx	nomx	?column?
x1	luis	
x2	juan	1500
x3	pedro	
x4	alberto	1005
x5	juan	1210
x6	luis	820
x7	carlos	
x8	pedro	3005
x9	bent	1510
x10	manuel	
x11	antonio	
x12	eliseo	1360
x13	juan	2005
x14	federico	1010
x15	leon	

(15 rows)

si ahora quiero que me sume tambien las nulas como cero para que me muestro todos los salarios?

si el salario es nullo lo iguala a cero y lo suma

```
postgres=> select codx,nomx,salario+coalesce(comision,0) from xogador;
```

codx	nomx	?column?
x1	luis	2000
x2	juan	1500
x3	pedro	1000
x4	alberto	1005
x5	juan	1210
x6	luis	820
x7	carlos	
x8	pedro	3005
x9	bent	1510
x10	manuel	
x11	antonio	2000
x12	eliseo	1360
x13	juan	2005
x14	federico	1010
x15	leon	

(15 rows)

para cambiar el nombre a la columna (cambiamos COMIDIOS por TOTAL)



```
postgres=> select codx,nomx,salario+coalesce(comision,0) total from xogado
codx |  nomx  | total
-----+-----+-----
x1   | luis   | 2000
x2   | juan   | 1500
x3   | pedro  | 1000
x4   | alberto| 1005
x5   | juan   | 1210
x6   | luis   | 820
x7   | carlos |
x8   | pedro  | 3005
x9   | bent   | 1510
x10  | manuel |
x11  | antonio| 2000
x12  | eliseo | 1360
x13  | juan   | 2005
x14  | federico| 1010
x15  | leon   |
(15 rows)
```

para contar todos los luis que hay

```
postgres=> select count(nomx) from xogador where nomx='luis'
count
-----
      2
(1 row)
```

otro ejem

mostrar todos los nombres distintos que hay.

para contarlos seria igual poniendo despues de select count(distinct nomx)...

```
postgres=> select distinct nomx from xogador;
nomx
-----
pedro
leon
bent
carlos
federico
alberto
antonio
juan
manuel
eliseo
luis
(11 rows)
```

## Grupaes

para contar cuantos jugadores hay en e1

```
postgres=> select count(*) from xogador where codequ='e1';
```

lo tendria que hacer asi para buscar todos los que hay en e2 y 23 etc. pero como hago para que me los muestre todos a la vez?

cuenta cuantos jugadores hay con el mismo salario.

```
postgres=> select salario,count(xogador) from xogador group by salario;
salario | count
-----+-----
        |      3
    3000 |      1
    2000 |      3
     800 |      1
    1000 |      3
    1500 |      2
    1350 |      1
    1200 |      1
(8 rows)

postgres=>
```

sumar los salarios en funcion del grupo

```
postgres=> select codequ,sum(salario) from xogador group by codequ;
codequ | sum
-----+-----
    e1 | 5000
    e3 | 3500
    e4 | 6850
    e2 | 3000
(4 rows)
```

lo podemos poner de las dos formas (poe que el \* cuenta por filas)

```
postgres=> select codest,count(ano) from adestra group by codest;
codest | count
-----+-----
    es1 |      4
    es2 |      2
    es3 |      1
(3 rows)

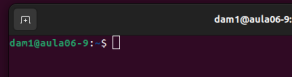
postgres=> select codest,count(*) from adestra group by codest;
codest | count
-----+-----
    es1 |      4
    es2 |      2
    es3 |      1
(3 rows)

postgres=>
```

otro ejem

```
postgres=> select codequ,sum(salario) from xogador group by codequ;
codequ | sum
-----+-----
e1      | 5000
e3      | 3500
e4      | 6850
e2      | 3000
(4 rows)

postgres=> select codequ,sum(salario) from xogador group by codequ having sum(salario)>4000;
codequ | sum
-----+-----
e1      | 5000
e4      | 6850
(2 rows)
```



24/10/22

```
postgres=> select * from interven;
codc | codpar | funcion
-----+-----+-----
c3    | p2     | 1
c2    | p2     | 2
c3    | p1     | 2
c1    | p1     | 3
c1    | p2     | 3
c5    | p3     | 2
c1    | p3     | 3
(7 rows)
```

\*1 explicacion está en libreta

```
postgres=> select * from xogador order by codx;
```

codx	nomx	codequ	salario	comision	datan
x1	luis	e1	2000		1990-01-03
x10	manuel	e4		10	1990-12-25
x11	antonio	e3	2000		1990-06-16
x12	eliseo	e4	1350	10	1990-05-14
x13	juan	e2	2000	5	1990-04-07
x14	federico	e1	1000	10	1990-01-05
x15	leon	e1			1990-11-27
x2	juan	e3	1500	0	1991-04-13
x3	pedro	e4	1000		1990-11-12
x4	alberto	e2	1000	5	1990-07-01
x5	juan	e1	1200	10	1990-08-23
x6	luis	e1	800	20	1990-09-12
x7	carlos	e3	3000		1990-10-08
x8	pedro	e4	3000	5	1990-06-07
x9	bent	e4	1500	10	1990-04-23

(15 rows)

ordena por caracteres. no por numeros como tal

04/10

```
postgres=> select * from xogador;
```

codx	nomx	codequ	salario	comision	datan
x1	luis	e1	2000		1990-01-03
x2	juan	e3	1500	0	1991-04-13
x3	pedro	e4	1000		1990-11-12
x4	alberto	e2	1000	5	1990-07-01
x5	juan	e1	1200	10	1990-08-23
x6	luis	e1	800	20	1990-09-12
x8	pedro	e4	3000	5	1990-06-07
x9	bent	e4	1500	10	1990-04-23
x10	manuel	e4		10	1990-12-25
x11	antonio	e3	2000		1990-06-16
x12	eliseo	e4	1350	10	1990-05-14
x13	juan	e2	2000	5	1990-04-07
x14	federico	e1	1000	10	1990-01-05
x15	leon	e1			1990-11-27
x7	carlos	e3	3000		1990-10-08

(15 rows)

avg hace la media. hace suma de salarios (pero no tiene en cuenta los nulos) entre los jugadores, para que estuviera bien abria que poner el coaleste e igualar los numeros a cero.

```
postgres=> select avg (salario) from xogador where codequ='e1';
          avg
-----
1250.00000000000000000000
(1 row)

postgres=> █
```

ahora estaria bien hecho:

```
postgres=> select avg (coalesce(salario,0)) from xogador where codequ='e1';
          avg
-----
1000.00000000000000000000
(1 row)

postgres=>
```

\*\*\*\*\*ej. amosar o codigo e nome dos xogadores do equipo e1.

```
postgres=> select codx , nomx from xogador where codequ='e1';
codx |  nomx
-----+-----
x1   | luis
x5   | juan
x6   | luis
x14  | federico
x15  | leon
(5 rows)
```

\*\*\*\*\*ej2 amosar codigo e nome dos xogadores do equipo 'cuspedrinos'

donde esta el nombre del equipo? en equipos.

hay algun vinculo entre tabla de xogadores y equipo? si

```
postgres=> select * from equipo;
codequ |  nomequ  | numx
-----+-----+-----
e5     | novo     |
e1     | cuspedrinos | 5
e3     | croques  | 3
e4     | reboludos | 5
e2     | gambusinos | 2
(5 rows)
```

pondria esto mirandolo en la tabla

```
postgres=> select nomx, nomx form xogador where codequ='e1';
```

pero que pasa si yo lo quiero buscar con el nombre del equipo?

```
postgres=> select codequ from equipo where nomequ='cuspedrinos';
codequ
-----
e1
(1 row)
```

pero ahora quiero juntar las dos cosas

```
postgres=> select codx, nomx from xogador where codequ in (select codequ from equipo where
nomequ='cuspedrinos');
codx | nomx
-----+-----
x1   | luis
x5   | juan
x6   | luis
x14  | federico
x15  | leon
(5 rows)
```

\*\*\*\*ej.3amosar codigo e nome dos partidos xogados no estadio 'playero'

```
postgres=> select codpar, nompar from partido where codest in(select codest from estadio wh
ere nomest='playero');
codpar | nompar
-----+-----
p1     | cuspedrinoscroques
p2     | croquesreboludos
p6     | croquesreboludos
(3 rows)
```

en este caso daría lo mismo. no ponemos entre comillas simple lo que está entre parentesis ya que cuando pongo el select me devuelve el valor directamente entre comillas. Entonces diríamos que '=' y es lo mismo que 'in'??? NO, no es lo mismo. Aquí nos funciona igualmente pero luego ya veremos que importa la diferencia.

```
postgres=> select codpar, nompar from partido where codest =(select codest from estadio whe
re nomest='playero');
```

\*\*\*\*ej.4amosar codigo de colegiado que intervinieron en p2

```
postgres=> select codc,codpar from interven where codpar='p2';
codc | codpar
-----+-----
c3   | p2
c2   | p2
c1   | p2
```

\*\*\*\*ej.5amosar codigos de colexiados que intervinon no partido 'croquereboludos'

```
postgres=> select codc from interven where codpar in (select codpar from partido where nom
par='croquesreboludos');
codc
-----
c3
c2
c1
(3 rows)
```

ahora igual no vale. porque hay 3 croquesreboludos. el igual compara un valor con otro directamente. El in lo que hace es igualar un valor en un grupo.

```
postgres=> select codc from interven where codpar = (select codpar from partido where nompar='croquesreboludos');
ERROR: more than one row returned by a subquery used as an expression
postgres=>
```

observamos que hay 3 croques revoludos

```
postgres=> select * from partido
postgres-> ;
```

codpar	nompar	data	codest
p1	cuspedrinoscroques	2010-03-02	es1
p2	croquesreboludos	2010-06-02	es1
p3	gambusinoscuspedrinos	2010-06-12	es3
p4	croquesreboludos	2010-07-17	es4
p5	gambusinoscuspedrinos	2010-07-07	es3
p6	croquesreboludos	2010-09-07	es1
p7	cuspedrinoscroques	2010-10-07	es4

(7 rows)

\*\*\*\*ej.6 amorsar nome de colexiados que intervinieron no partido 'croquesreboludos'

```
postgres=> select nomc from colexiado where codc in(select codc from interven where codpar in (select codpar from partido where nompar='croquesreboludos'));
nomc
-----
cfelipe
cpedro
cjuan
(3 rows)
```

\*\*\*\*\*ahora queremos mostrar el codigo y nombre del jugador y el nombre del equipo correspondiente.

1. Como con el siguiente codigo nos muestra el codigo y no el equipo:

```

(15 rows)
postgres=> select codx, nomx, codequ from xogador;
codx |  nomx  | codequ
-----+-----+-----
x1   | luis   | e1
x2   | juan   | e3
x3   | pedro  | e4
x4   | alberto| e2
x5   | juan   | e1
x6   | luis   | e1
x8   | pedro  | e4
x9   | bent   | e4
x10  | manuel | e4
x11  | antonio| e3
x12  | eliseo | e4
x13  | juan   | e2
x14  | federico| e1
x15  | leon   | e1
x7   | carlos  | e3
(15 rows)
postgres=> 

```

me pone el codigo del equipo y lo que quiero es que me muestre el nombre.

2. por tanto tengo que juntar las tablas: su juntas las tablas. con la ',' juntamos tablas:

```

postgres=> select codx, nomx, nomequ from xogador, equipo;

```



codx	nomx	nomequ
x1	luis	novo
x2	juan	novo
x3	pedro	novo
x4	alberto	novo
x5	juan	novo
x6	luis	novo
x8	pedro	novo
x9	bent	novo
x10	manuel	novo
x11	antonio	novo
x12	eliseo	novo
x13	juan	novo
x14	federico	novo
x15	leon	novo
x7	carlos	novo
x1	luis	cuspedrinos
x2	juan	cuspedrinos
x3	pedro	cuspedrinos
x4	alberto	cuspedrinos
x5	juan	cuspedrinos

me daría una tabla con valores erróneos ya que la consulta está incompleta ya que estaría juntando las dos tablas directamente sin tener nada en cuenta. junta los datos en las distintas columnas, es como si mezclara de forma vertical, las columnas de las distintas tablas que tienen el mismo nombre las junta directamente en la misma columna

3. Sería ahora correctamente así: se combinan esas 2 filas.  
con esto le estamos diciendo que JUNTEN las tablas pero no que las COMBINEN de forma que las junta horizontalmente sin mezclar los datos.

```
postgres=> select xogador.*, equipo.* from xogador, equipo;
```

codx	nomx	codequ	salario	comision	datan	codequ	nomequ	n
x1	luis	e1	2000		1990-01-03	e5	novo	
x2	juan	e3	1500	0	1991-04-13	e5	novo	
x3	pedro	e4	1000		1990-11-12	e5	novo	
x4	alberto	e2	1000	5	1990-07-01	e5	novo	
x5	juan	e1	1200	10	1990-08-23	e5	novo	
x6	luis	e1	800	20	1990-09-12	e5	novo	
x8	pedro	e4	3000	5	1990-06-07	e5	novo	
x9	bent	e4	1500	10	1990-04-23	e5	novo	
x10	manuel	e4		10	1990-12-25	e5	novo	
x11	antonio	e3	2000		1990-06-16	e5	novo	
x12	eliseo	e4	1350	10	1990-05-14	e5	novo	
x13	juan	e2	2000	5	1990-04-07	e5	novo	
x14	federico	e1	1000	10	1990-01-05	e5	novo	
x15	leon	e1			1990-11-27	e5	novo	
x7	carlos	e3	3000		1990-10-08	e5	novo	
x1	luis	e1	2000		1990-01-03	e1	cusnedrinos	
x2	juan	e3	1500	0	1991-04-13	e1	CI	
x3	pedro	e4	1000		1990-11-12	e1	CI	
x4	alberto	e2	1000	5	1990-07-01	e1	CI	
x5	juan	e1	1200	10	1990-08-23	e1	CI	

aun así esta tabla no me sirve ya que los datos son erróneos al comparar ambas tablas por que únicamente los junta

4. el código de jugador se corresponde con el código del equipo. sin embargo son ambiguas entonces tenemos que especificar. podemos especificar con un '.' o con parentesis

```
postgres=> select xogador.*, equipo.* from xogador, equipo where codequ=codequ;
ERROR: column reference "codequ" is ambiguous
LINE 1: ...ct xogador.*, equipo.* from xogador, equipo where codequ=cod...
```

5. Ahora especificamos y nos sale una tabla con valores correctos y no mezclados al chou. al poner el punto o el parentesis ESPECIFICAMOS (no tengo claro si está bien)

```

postgres=> select xogador.*, equipo.* from xogador, equipo where equipo.codequ=xogador.code
qu;

```

codx	nomx	codequ	salario	comision	datan	codequ	nomequ	numx
x1	luis	e1	2000		1990-01-03	e1	cuspedrinos	5
x2	juan	e3	1500	0	1991-04-13	e3	croques	3
x3	pedro	e4	1000		1990-11-12	e4	reboludos	5
x4	alberto	e2	1000	5	1990-07-01	e2	gambusinos	2
x5	juan	e1	1200	10	1990-08-23	e1	cuspedrinos	5
x6	luis	e1	800	20	1990-09-12	e1	cuspedrinos	5
x8	pedro	e4	3000	5	1990-06-07	e4	reboludos	5
x9	bent	e4	1500	10	1990-04-23	e4	reboludos	5
x10	manuel	e4		10	1990-12-25	e4	reboludos	5
x11	antonio	e3	2000		1990-06-16	e3	croques	3
x12	eliseo	e4	1350	10	1990-05-14	e4	reboludos	5
x13	juan	e2	2000	5	1990-04-07	e2	gambusinos	2
x14	federico	e1	1000	10	1990-01-05	e1	cuspedrinos	5
x15	leon	e1			1990-11-27	e1	cuspedrinos	5
x7	carlos	e3	3000		1990-10-08	e3	croques	3

```

(15 rows)

postgres=>

```

ahora me mostraria en principio la tabla con los datos corrcotos

6. como ultimo paso especificaremos para que nos muestre las columnas que queriamos al principio: el nombre y codigo del jugador y el nombre del equipo correspondiente
- 7.

07/11

amosar nombre dos jugadores que rematan coa letra: o

```
select codx, nomx from xogador where monx like '/%o';
```

el like lo que hace es

el tanto por ciento significa cualquier numero de caracteres

para todos los que empiezan por a

```
select codx, nomx from xogador where monx like '/a%';
```

que tenga una a en la segunda palabra

```
select codx, nomx from xogador where monx like ' _/a%';
```

que antes o despues puede o no contener la letra e

```
select codx, nomx from xogador where monx like '%e%';
```

mostrar nombres que tengan la a y la u dando igual el orden

```
..nomx like '%a%' and nomx like '%u%'
```

otra opcion

```
...nomx like '%a%u%' or nomx like '%u%a%'
```

10/11

9) amosar a media dos totais de todos os pedidos

```
select avg(coalesce(total, 0)) from pedido;
```

```
postgres=> select avg(coalesce(total, 0)) from
pedido;
          avg
-----
 25.9411764705882353
(1 row)
```

otra opcion (lo que pasa es que trunca el resultado: deberia dar 25.94...)

```
select sum(total)/count(*) from pedido;
```

para que me los tenga en cuenta tengo que castearlo o multiplicar por 1.0

```
select sum(total)::float/count(*)::float from pedido;
```

```
postgres=> select sum(total)::float/count(*)::
float from pedido;
          ?column?
-----
 25.941176470588236
(1 row)
```

14/11

mostrar el nombre de los productos de la marca 'argal' que son fiambres

```
postgres=> select * from produto where codm
in (select codm from marca where nomm='argal'
) and n in(select n from fiambres where produ
to.n=fiambres.n);
 codm | n |  nome  |  prezo
-----+---+-----+-----
  m6  | 1 | chourizo |      5
  m6  | 2 | xamon   |     10
  m6  | 3 | mortadela |      6
(3 rows)
```

El where se utiliza para igualar 2 valores.

El 'in' es para llamar a otra tabla

en el grafo busco la relacion entre producto, marca y fiambres: codm y n



-----  
 mostrar nif, nome, email ordenador por correo electronico

```
postgres=> select nif,nome,email from cliente
order by  substr(email,strpos(email,'@'));
nif  |  nome  |      email
-----+-----+-----
362b | pedro  | pedroc@gmail.com
366f | eva    | evab@gmail.com
368h | carlos | carlosr@gmail.com
364d | borja  | borjap@gmail.com
365e | pedro  | pedroc@hotmail.com
363c | carlos | carlosc@hotmail.com
361a | ana    | anax@hotmail.com
367g | luis   | luis@yahoo.com
(8 rows)
```

-----  
 Ordenar la tabla de xogadores por codx de manera.

como me reocnoce la cadena como string, tengo que quitarle el x previo, y luego pasar a numero la cadena restante mediante el comando TO\_NUMER

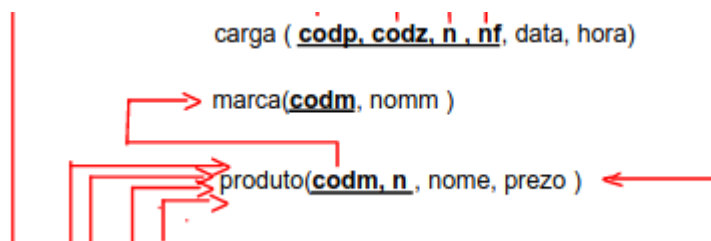
```

postgres=> select codx from xogador order by
TO_NUMBER(substr(codx,2), '99') desc;
codx
-----
x15
x14
x13
x12
x11
x10
x9
x8
x7
x6
x5
x4
x3
x2
x1
(15 rows)

```

amosar nome e marcar de todos os produtos cun prezo maior de 4.

el where producto.codm=marca.codm ya que codm es el que tienen en comun y para que me junte correctamente las 2 tablas. le tengo que decir que el cdom es el que tienen en comun porque si no el programa no sabe a que codm hay que asignar el nomm o el nome o el prezo.



```
postgres=> select produto.nome, marca.nommm fr  
om produto, marca where produto.codm=marca.c  
odm and produto.prezo>4;
```

nome	nommm
chourizo	pozo
xamon	pozo
chourizo	argal
xamon	argal
mortadela	argal
sardinha	argal
polvo	argal
berberecho	argal

(8 rows)

```
postgres=>
```

#### CONSULTAS BALNEARIOS 24/11

amosar codigos de poboacions onde viven os pacientes e o numero de pacientes que ten cada unha desas poboacions

desglosamiento

1. mostrar códigos de poblaciones donde viven pacientes

si solo fuera esto busco los codigos de pobloaciones dd viven los pacientes, esto esta en la tabla de paciones, ya que en la de poblaciones me pone todos los codigos de pobloaciones

```

codp
-----
p21
p2
p4
p7
p8
p9
p9
p8
p7

p4
p7
p9
p10
p10
p9
p5
(18 rows)

```

vemos que de p10 hay 2, de p9 hay 4... y comparamos con la respuesta correspondiente.

```

32
33
34 p10
35 p7
36 p21
37 p4
38 p2
39 p8
40 p9
41 p5
42

```

2. y el numero de pacientes que tiene cada una de esas poblaciones

ahora lo que quiero es que me agrupe por codigo de poblacion y paciente:  
 select codp,count(\*) from pacientes group by codp;

```

2
p10 2
p7 3
p21 1
p4 2
p2 1
p8 2
p9 4

```



—otra consulta

mostrar nombre de poblaciones donde viven pacientes y el numero de pacientes que tiene cada una de esas poblaciones

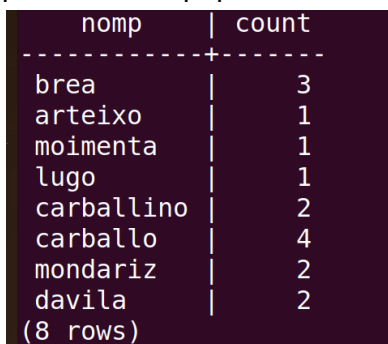
`select poboacions.nomp,count(*) from pacientes,poboacions group by poboacions.nomp;`  
 pasa esto porque no puse cual es la relacion que hay entre las dos tablas, el codp de uno tiene que corresponderse con el del otro ya que si no me lo junta todo



```
oracle@oracle-Virtual:~$ select poboacions.nomp,count(*) from pacientes,poboacions group by poboacions.nomp;
```

nomp	count
cuntis	18
molgas	18
partovia	18
brea	18
vigo	18
guitiriz	18
toxa	18
arteixo	18
moimenta	18
santiago	18
lugo	18
ourense	18
carballino	18
carballo	18
mondariz	18
tuy	18
arnoia	18
davila	18
cangas	18
coruna	18

`select poboacions.nomp,count(*) from pacientes,poboacions where  
 pacientes.codp=poboacions.codp group by poboacions.nomp;`



```
oracle@oracle-Virtual:~$ select poboacions.nomp,count(*) from pacientes,poboacions where  
pacientes.codp=poboacions.codp group by poboacions.nomp;
```

nomp	count
brea	3
arteixo	1
moimenta	1
lugo	1
carballino	2
carballo	4
mondariz	2
davila	2

(8 rows)