**-- Visualizar el numero y el nombre del departamento que tenga mas empleados cuyo oficio** **sea empleado;**

select dept\_no, dnombre

from emple natural join depart

where oficio = 'EMPLEADO'

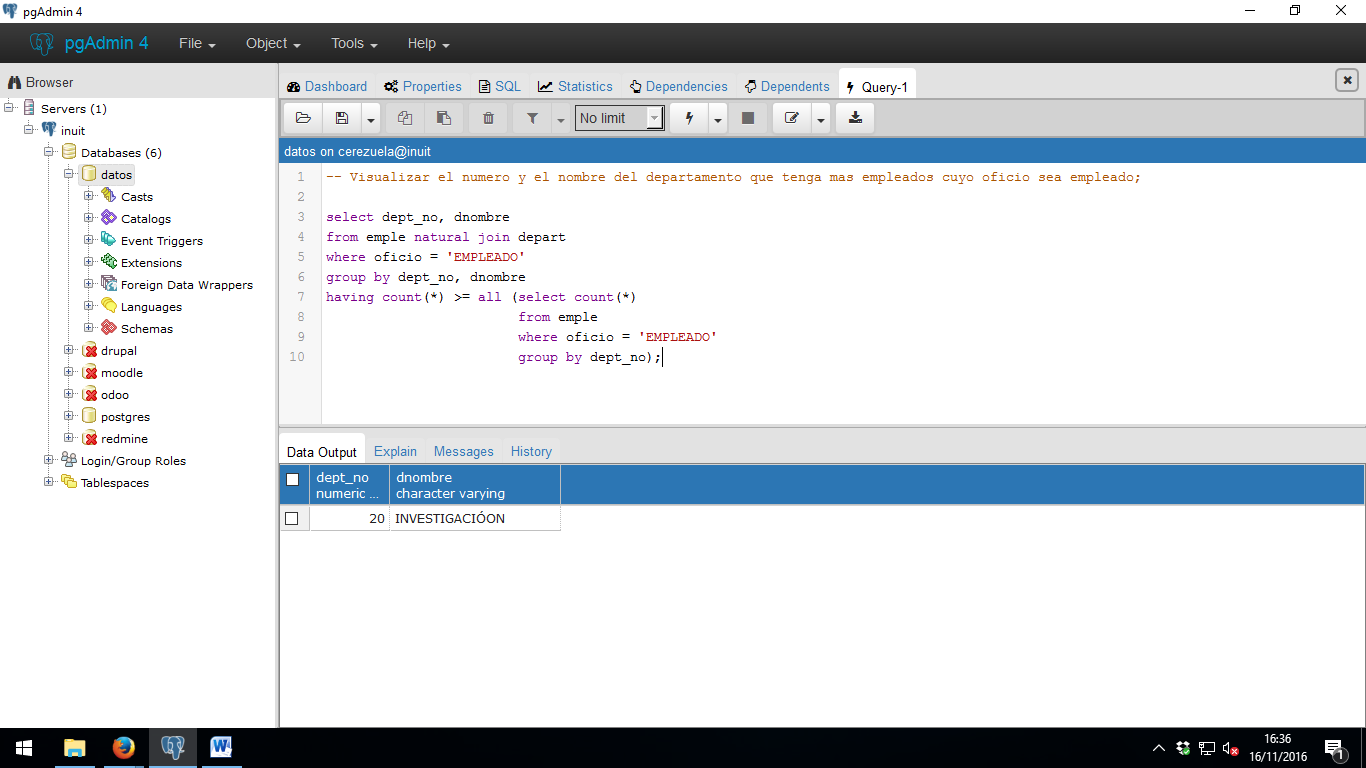
group by dept\_no, dnombre

having count(\*) >= all (select count(\*)

from emple

where oficio = 'EMPLEADO'

group by dept\_no);



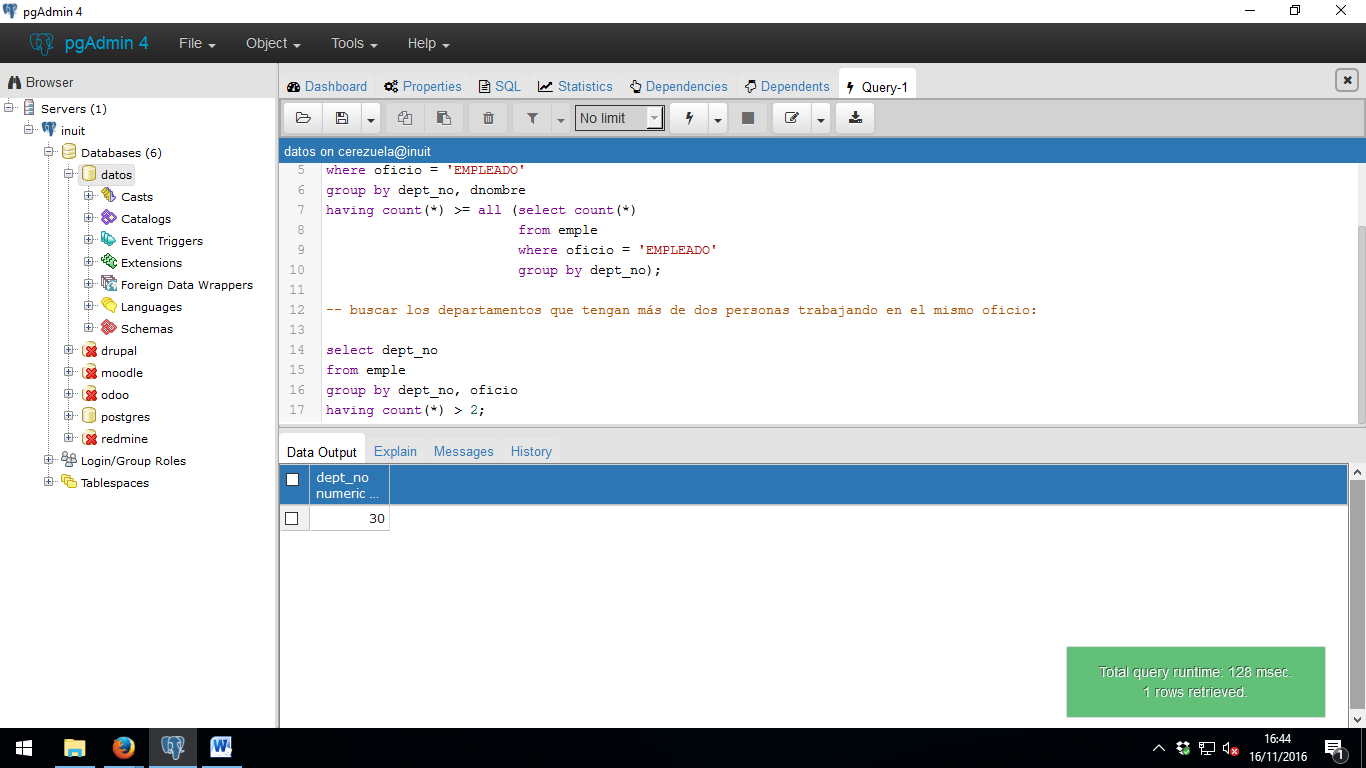
**-- buscar los departamentos que tengan más de dos personas trabajando en el mismo oficio:**

select dept\_no, oficio, count(\*)

from emple

group by dept\_no, oficio

having count(\*) > 2;



**-- Mostrar que departamentos tienen menos de 3 empleados**

select dept\_no, dnombre

from emple natural right join depart

group by dept\_no

having count(emp\_no) < 3;

**-- Mostrar cuantos subordinado tiene cada empleado, si un empleado no tiene subordinados** **,debera indicar que tiene 0 subordinados**

select j.emp\_no, count(e.emp\_no)

from emple e right join emple j on e.dir = j.emp\_no

group by j.emp\_no;

/\*Combinacion externa

Puede ser por la izquierda, por la derecha y completa

Left outer join

Right outer join

Full outer join\*/

**-- mostrar el numero de empleados que hay en cada centos, si el centro carece de empleados tiene que aparecer un cero:**

select cod\_centro, nombre, count(dni)

from centros natural left join personal

group by cod\_centro, nombre

order by cod\_centro;

**-- mostrar el salario total que debe pagar centro:**

select cod\_centro, to\_char(coalesce(sum(salario),0), '999G990D99L')

from centros natural left join personal

group by cod\_centro;

**-- mostrar un listado que muestre cuantos empleados se han dado de alta en cada trimestre**

select extract(quarter from fecha\_alt),count(\*)

from emple

group by extract(quarter from fecha\_alt)

order by 1;

FACTURACION

Tablas

CLI(nif,nombre)

ART (código, denominación, precio)

FAC(número, fecha, nif, código)

**-- Mostrar el total de cada factura , numero de lineas de cada factura y cantidad de articulos en cada factura:**

FM quita ceros a la izquierda , La G puntos de miles, D decimal , L pinta la moneda

Numeric (6,2) 999,99

select numero, to\_char(sum(cantidad\*precio),'FM99G999G999G999D90 L) as total, count(\*) as numero\_lineas, sum(cantidad) as cantidad\_de\_articulos

from lin natural join art

group by numero

order by numero;

**-- Mostrar cuántas unidades se han vendido de cada articulo, el mes pasado**

select codigo , sum(cantidad)

from lin natural join fac

where to\_char(fecha, 'MM-YYYY') = to\_char(current\_date - '1 month' ::interval , 'MM-YYYY')

group by codigo;

-- Mostrar los daros de los clientes que hayan realizado dos compras (facturas)

select nif

from fac

group by nif

having count(\*) = 2 ;

-- mostrar cuantas ventas(facturas) se han realizado en cada me independientemente del año

With n as (select \* from generate\_series(1,12) as mes),

ventas as (select extract(month from fecha) as mes, count(\*) as numeroVentas

from fac

group by extract(month from fecha))

select mes, coalesce(numeroVentas,0) from n natural left join ventas;

La Unión , intersección y diferencia

∪ : union

# ∩ : Intersect

- : except

L {1,2,3,4,5,6,7,8,9, …}

N {a,b,c,d,e,f,g,h,i, … }

L ∪ N {1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i, … }

Condiciones que tienen que tener dos select para poderse unir , intersectar y hacer la diferencia las tablas deben de tener el mismo numero de columnas.

Las columnas de ambas select deben de tener dominios compatibles 2 a 2

select emp\_no, apellido, salario

from emple

No compatible

union

select dept\_no, dnombre, loc

from depart

Esta la primera select cumple con la primera condición pero no con el segundo

Se puede forzar la conversión de un dominio a otro si se expresa explícitamente

por ejemplo un numeric en un var char

ejemplo

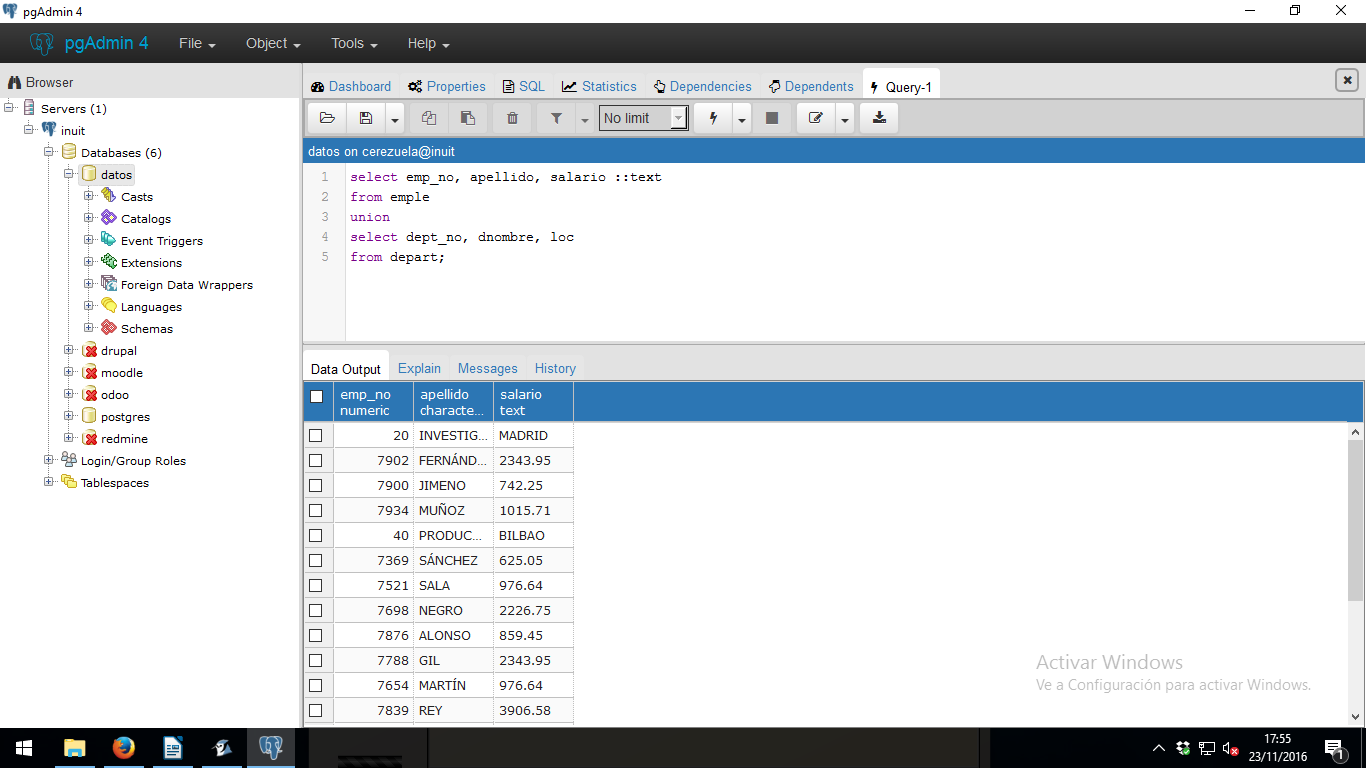
select emp\_no, apellido, salario ::text

from emple

union

select dept\_no, dnombre, loc

from depart;



select \* from emple

union

select \* from emple

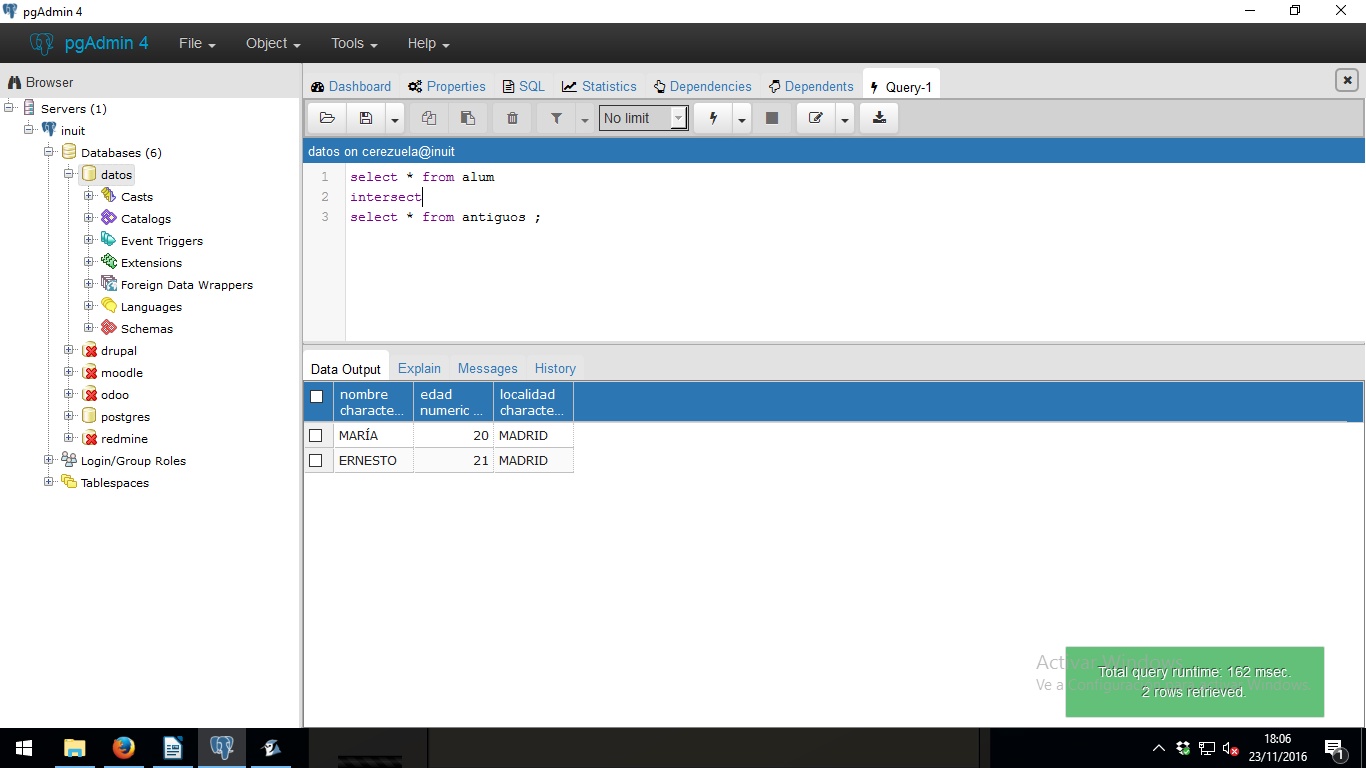
Esto devuelve la misma tabla ya que aunque deberian salir las filas repetidas estas se filtran, porque el resultado de la union de dos conjuntos es otro conjunto, y en un conjunto no hay elementos repetidos, a menos que coloquemos all siguiendo a union

select \* from alum

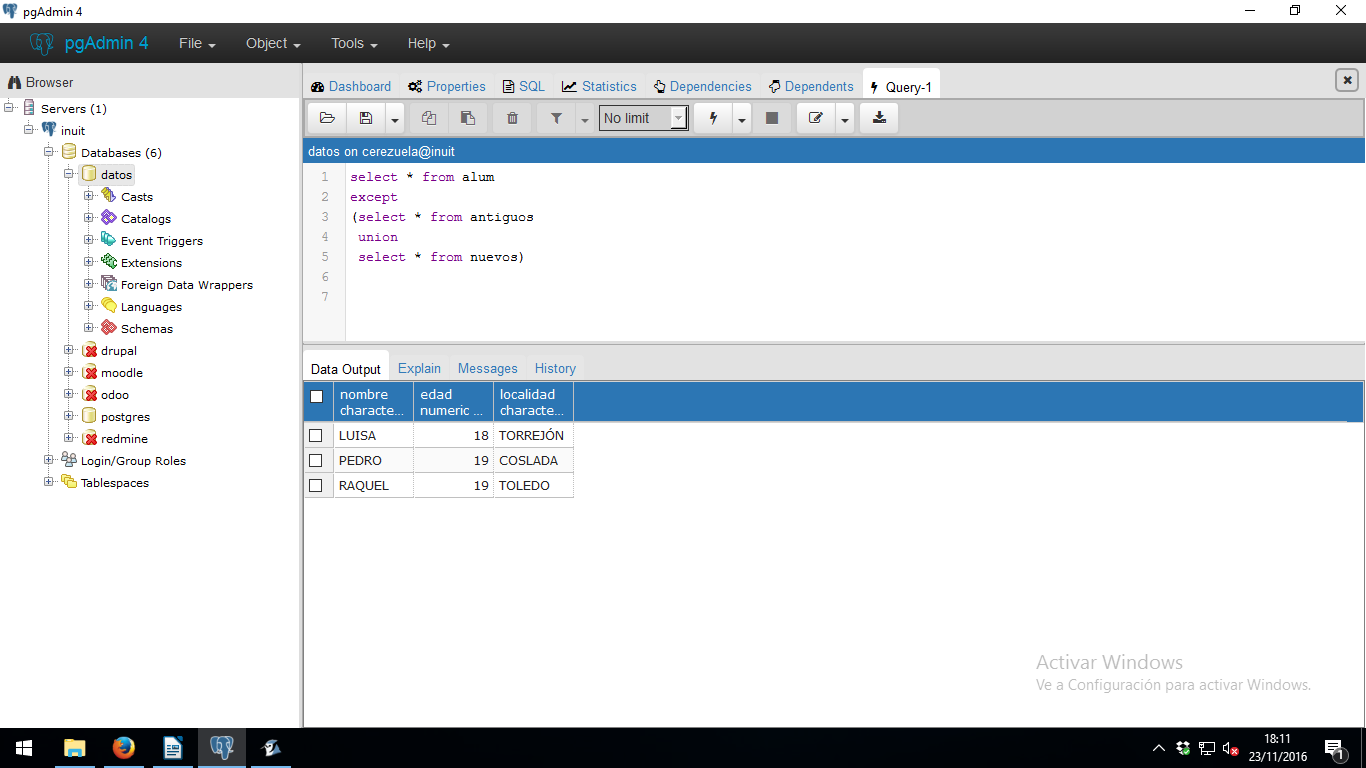
union all

select \* from antiguos ;

**Con Intersect vamos a ver las filas que sean coincidentes en las dos tablas**

****

Se pueden utilizar parentesis



ejemplo

-- mostrar los departamentos que tengan empleados:

select dept\_no

from depart

where dept\_no in (select dept\_no

from emple);

select dept\_no

from depart

intersect

select dept\_no

from emple;

Ejemplo videoclub

N(6)

SOCIO (numero ,nombre ,teléfono)

N(4)

PELÍCULAS (código,titulo,precio\_alq)

ALQUILERES (numero, código, fecha\_alq,fecha\_dev)

-- mostrar los dias acumulados de retraso que tiene cada socio en o que vá de año:

select numero,nombre ,sum(coalesce(fecha\_dev,current\_date) - fecha\_alq -1) as retraso

from socios natural join alquileres

where extract(year from\_fecha\_alq) = extract(year from current\_date) =

group by numero,nombre;

-- Mostrar las películas con el mismo título que otras

select lower(trim(titulo)),

from peliculas

group by lower(trim(titulo)) -> trim() quita espacios delante y detrás , lower(pone en minúsculas)

having count(\*) > 1;

o

-- Comprobar si hay dos películas con el mismo título

select count(\*) != count(distinct titulo)

from peliculas;

**limit número** se debe usar siempre con order by para elegir que filas salen **(limita en número máximo de filas que debe devolver una consulta)**

**offset número (se salta el número de líneas primeras)**

**-- Mostrar el apellido del empleado que más gana en el departamento 20**

**select apellido**

**from emple**

**where dept\_no = 20 and salario = (select max(salario)**

**from emple**

**where dept\_no = 20) ;**

**-- Mostrar el apellido del segundo que mas gana en el departamento 20**

**select apellido**

**from emple**

**where dept\_no = 20 and salario = (select distinct salario**

**from emple**

**where dept\_no = 20**

**order by salario desc**

**limit 1**

**offset 1) ;**

**-- Mostrar el apellido de los tres empleados que mas ganan en el departamento 20**

**select apellido**

**from emple**

**where dept\_no = 20**

**order by salario desc**

**limit 3;**

**-- Mostrar el apellido de los empleados del departamento 20 que tengan uno de los tres salarios mas altos de si departamento**

**select apellido**

**from emple**

**where dept\_no = 20 and salario in (select distinct salario**

**from emple**

**where dept\_no = 20**

**order by salario desc**

**limit 3)**

**order by salario desc;**

**Ejercicio aquileres**

with alq\_importe as (select a.\*, precio\_alq \* (1 + iva / 100.0) as importe

from alquileres a natural join peliculas p)

select \*, importe \* (coalesce(fecha\_dev, current\_date) - fecha\_alq) as total

from alq\_importe;

**-- Calcular el número de dias de retraso totales que tiene cada socio en lo que va de año**

**-- e importe total de dicho retraso, ordenado de mayor a menor por morosidad**

with alq\_pel as

(

select a.\*, round(precio\_alq \* ( 1 + iva / 100.0),2) as precio\_ii,

coalesce(fecha\_dev,current\_date) - fecha\_alq - 1 as dias

from alquileres a natural join peliculas p

where extract (year from fecha\_alq) = extract (year from current\_date)

)

select numero, nombre ,telefono ,sum(coalesce(dias, 0)) as dias\_totales,

sum(coalesce(dias \* precio\_ii, 0)) as importe\_total

from alq\_pel natural right join socios

group by numero , nombre, telefono

order by dias\_totales desc;