# **SENTENCIAS**

## **Ejercicio 1**

## Generar una consulta la cual nos permita conocer los programas que ofrecen Presencialmente en la universidad con sus facultades

**select**

pro.nombreprograma,

facul.nombrefacultad,

divi.nombredivicion

**from**

programa pro  
 **join** facultad facul **on** facul.idfacultad = pro.idfacultad

**join** divicion divi **on** divi.iddivicion = facul.iddivicion

**where**

pro.modalidad = 'Presencial'



## **Ejercicio 2**

## Diseñar una consulta la cual nos permita determinar y Conocer los grupos, profesore,créditos, asignaturas, que existen actualmente en el área de medicina

**select** per.nombres,

per.apellidos,

grupo.idgrupo **as** grupo,

asig.nombre **as** Nombre\_Asignatura,

asig.creditos **as** Cantidad\_creditos

**from**

docente d

**join** persona per **on** per.idpersona = d.coddocente

**join** grupo **on** grupo.coddocente = d.coddocente

**join** asignatura asig **on** asig.idasignatura = grupo.idasignatura

**where**

d.**area** = 'Medicina'



## **Ejercicio 3**

## Conocer cuales son los nombres y los apellidos de los estudiantes que están activos hoy en día, también el programa y el nombre de la facultad

**select**

per.nombres,

per.apellidos,

pro.nombreprograma,

facul.nombrefacultad

from

persona per

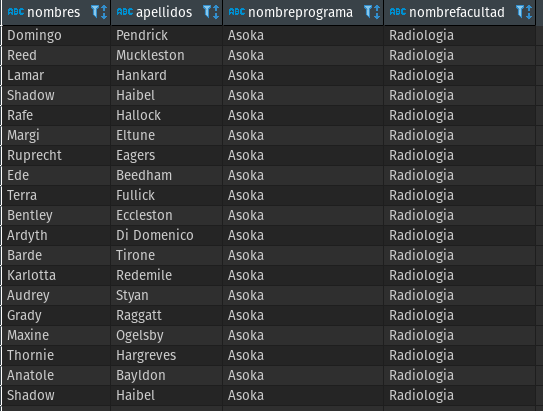
join estudianteprograma estu on estu.idpersona = per.idpersona

join programa pro on pro.idprograma = estu.idprograma

join facultad facul on facul.idfacultad = pro.idfacultad

where

pro.modalidad = 'Presencial' and pro.activo = 'true'



## **Ejercicio 4**

## Consultar los nombres de las materias de la carrera de ingeniería industrial la facultad y el nombre de la asignatura

select

pro.nombreprograma ,

facul.nombrefacultad,

asig.nombre

from

pensum pen

join programa pro on pro.idprograma = pen.idprograma

join facultad facul on facul.idfacultad = pro.idfacultad

join nivel on nivel.idpensum = pen.idpensum

join asignatura asig on asig.idnivelpensum = nivel.idnivelpensum

where

pro.idprograma = 6



## **Ejercicio 5**

## Diseñar una consulta en la cual se pueda evidenciar los los grupos, los docentes y los programas de la facultad de Diseño gráfico para poder tener referencia de los profesores actualmente en la facultad

select

per.nombres,

per.apellidos,

doc.area,

p2.nombreprograma as Programa,

grupo.idgrupo

from

grupo

join docente doc on doc.coddocente = grupo.coddocente

join persona per on per.idpersona =doc.coddocente

join estudianteprograma e on e.idpersona = per.idpersona

join pensum p on p.idprograma = e.idestudianteprograma

join programa on on p2.idprograma = p.idprograma

join facultad f on f.idfacultad = p2.idfacultad

where

f.nombrefacultad = 'Diseño Grafico'



## **Ejercicio 6**

## conocer en qué facultad y en qué división se encuentra el programa ofertado conocido como ‘’Keylex’

select

sede.nombresede as nombre\_sede,

f.nombrefacultad as nombre\_facultad,

divi.nombredivicion as nombre\_division

from

seccional secc

join sede on sede.idseccional = secc.idseccional

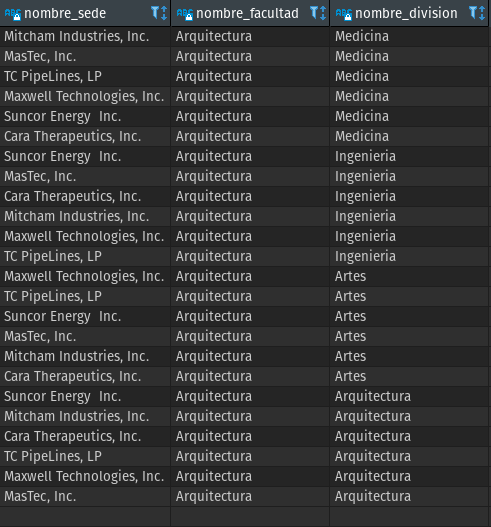
join divicion divi on divi.idseccional = secc.idseccional

join facultad f on f.iddivicion = divi.idseccional

join programa p on p.idfacultad =f.idfacultad

where

p.nombreprograma = 'Keylex' order by divi.nombredivicion desc



## **Ejercicio 7**

Generar una vista donde se conozca los grupos existentes en el semestre 1 para poder tener presente las materias y la cantidad de los grupos que hay en el momento en este

select

asig.nombre as nombre\_asignatura,

count(grupo.idasignatura) as Cantidad\_grupos

from

grupo

join asignatura asig on grupo.idasignatura = asig.idasignatura

join nivel on nivel.idnivelpensum = asig.idnivelpensum

join pensum p2 on p2.idpensum = nivel.idpensum

join programa p on p.idprograma = p2.idprograma

where

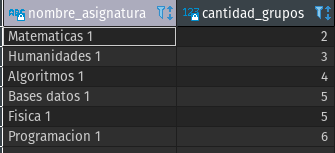
nivel.nivel =1

GROUP BY

asig.idasignatura

ORDER BY

Cantidad\_grupos asc



## 

## **Ejercicio 8**

## Crear una función que imprima el reporte de la tabla facultad de las facultades que pertenezcan a la división de Ingeniería

CREATE FUNCTION reporteFacultades(es INT)

RETURNS TABLE(codfacultad INT, coddivision INT, nomfacultad VARCHAR) as

$$

DECLARE

reg RECORD;

BEGIN

FOR reg IN SELECT \* FROM facultad

WHERE iddivicion=es

LOOP

codfacultad:=reg.idfacultad;

coddivision:=reg.iddivicion;

nomfacultad:=reg.nombrefacultad;

RETURN NEXT;

END LOOP;

RETURN;

END $$ LANGUAGE 'plpgsql'

SELECT \* FROM reporteFacultades(3)

## 

## 

## **Ejercicio 9**

crear una función que imprima el reporte de la tabla división, mostrando el nombre de la división, decano de división y código de la seccional

CREATE FUNCTION reportedivision(OUT codseccional int, OUT nombresosqui varchar, OUT decano varchar) RETURNS SETOF RECORD AS

$$

DECLARE

reg RECORD;

BEGIN

FOR reg IN SELECT \* FROM divicion

LOOP

codseccional:=reg.idseccional;

nombresosqui:=reg.nombredivicion;

decano=reg.decanodivicion;

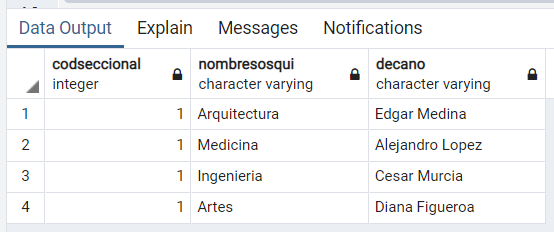
RETURN NEXT;

END LOOP;

RETURN;

END $$ LANGUAGE 'plpgsql'

SELECT\* FROM reportedivision()



## **Ejercicio 10**

Crea una función que actualiza el campo celular en la tabla telefono con el id de la persona para este

CREATE OR REPLACE FUNCTION actualizarpesona100(

persona.idpersona%type,

persona.celular%type)

RETURNS VOID AS

$$

UPDATE persona set celular=$2 WHERE idpersona=$1;

$$ LANGUAGE SQL

SELECT actualizarpesona100('2','314328')

## 

## **Ejercicio 11**

Crear una función que permita insertar empleados sin identificadores de entrada en la tabla personas

CREATE OR REPLACE FUNCTION insertarpersona1(

integer,

integer,

character varying,

character varying,

character varying,

character varying,

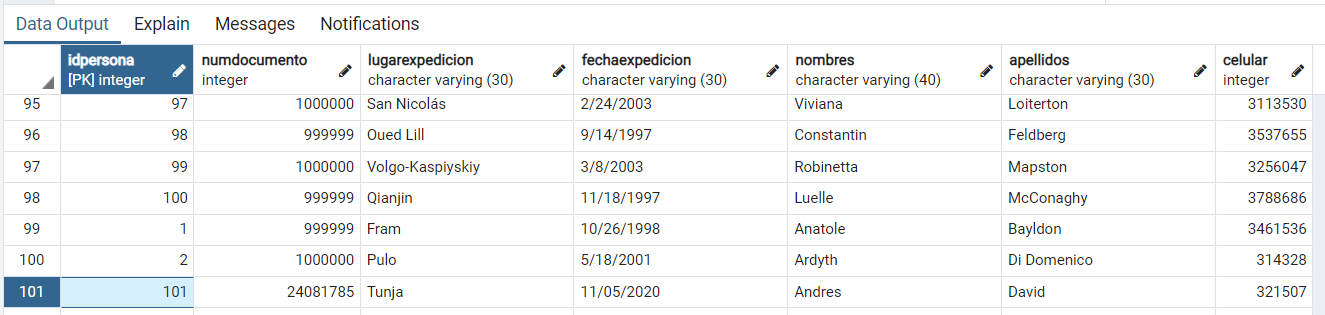
integer)

RETURNS VOID AS $$

INSERT INTO persona VALUES ($1,$2,$3,$4,$5,$6,$7);

$$ LANGUAGE SQL

Select insertarpersona1(101,24081785,'Tunja','11/05/2020','Andres','David',321507)



## 

## **Ejercicio 12**

Crear una función que busque el código de la persona con el identificador del docente

CREATE OR REPLACE FUNCTION buscarDocente10(codDoc Int) Returns INT As

$$

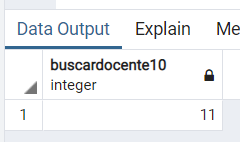
select idpersona from docente

where coddocente=codDoc;

$$

Language sql;

select buscarDocente10(5)



## 

## **Ejercicio 13**

Crear una función que busque por id de la asignatura y regrese él una información detallada de la asignatura, nombre créditos y área de conocimiento a la que pertenece, realizando concatenación de campos.

CREATE FUNCTION descripcion10(codasig Int) Returns varchar As

$$

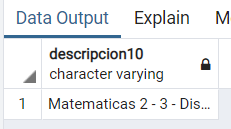
select nombre || ' - ' ||creditos|| ' - ' ||areaconocimientos from asignatura

where idasignatura=codasig;

$$

Language sql;

select descripcion10(2)



**Ejercicio 14**

Crea una tabla que muestra los pensum activos y que cuentan con una oferta académica superior a 20 estudiantes sin identificadores de entrada

CREATE FUNCTION reportePensum10( varchar, varchar)

RETURNS TABLE(codPensum INT,codProgram INT, estado varchar, ofetas VARCHAR) as

$$

DECLARE

reg RECORD;

BEGIN

FOR reg IN SELECT \* FROM pensum

WHERE activo=$1 AND oferta=$2

LOOP

codPensum:=reg.idpensum;

codProgram:=reg.idprograma;

estado:=reg.activo;

ofetas:=reg.oferta;

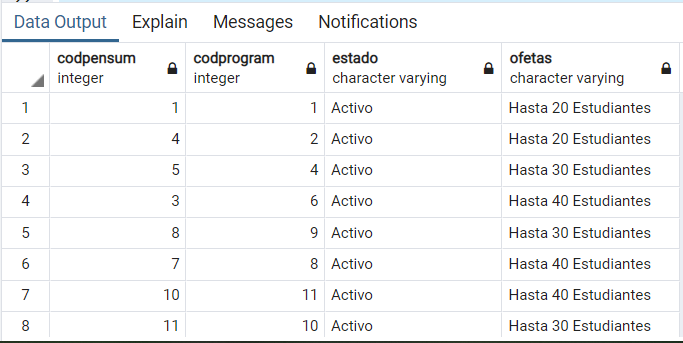
RETURN NEXT;

END LOOP;

RETURN;

END $$ LANGUAGE 'plpgsql'

SELECT \* FROM reportePensum100('Activo','Hasta 20 Estudiantes')



## **Ejercicio 15**

Muestra un reporte de las sedes disponibles dentro de una ciudad mostrando el nombre de la sede y la dirección de esta .

CREATE FUNCTION reporteSede1 (IN city varchar,OUT codsede INT,OUT codsecional INT,OUT nombresede VARCHAR, OUT direc int) RETURNS SETOF RECORD AS

$$

DECLARE

reg RECORD;

BEGIN

FOR reg IN SELECT \* FROM sede

WHERE ciudad=city

LOOP

codsede:=reg.idsede;

codsecional:=reg.idseccional;

nombresede=reg.nombresede;

direc=reg.direccion;

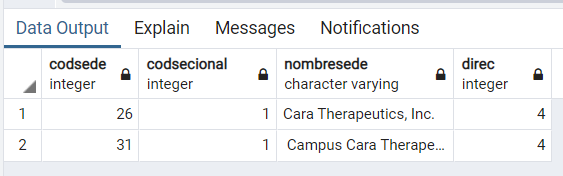
RETURN NEXT;

END LOOP;

RETURN;

END $$ LANGUAGE 'plpgsql'

SELECT \* FROM reporteSede1('Osiek')



## 

## **Ejercicio 16**

Diseñar una consulta la cual nos permite saber la cantidad de programas por facultad y a la ves la cantidad de grupos por facultad tiene activos en el momento para llevar control

select

facul.nombrefacultad as nombre\_facultad,

count(pro.idprograma) as Cantidad\_programas,

count(grupo.idasignatura) as cantidad\_grupos

from

facultad facul

join programa pro on pro.idfacultad = facul.idfacultad

join pensum p on p.idprograma = pro.idprograma

join nivel on nivel.idpensum =p.idpensum

join asignatura a2 on a2.idnivelpensum = nivel.idnivelpensum

join grupo on grupo.idasignatura = a2.idasignatura

GROUP BY

a2.idasignatura,facul.idfacultad order by cantidad\_grupos asc

