Consultas SQL - Consultas Avanzadas

1. Utilizando los mismos esquemas de la Parte 1 del Práctico 4

resuelva las siguientes consultas.

### Ejercicio 1

Consultas con anidamiento (usando IN, NOT IN, EXISTS, NOT EXISTS):

1.1. Listar todas las películas que poseen entregas de películas de idioma inglés durante

el año 2006. (P)

SELECT DISTINCT titulo, idioma, *extract*(YEAR FROM fecha\_entrega) ANIO FROM pelicula p INNER JOIN renglon\_entrega r

ON p.codigo\_pelicula=r.codigo\_pelicula

INNER JOIN entrega e ON r.nro\_entrega=e.nro\_entrega WHERE idioma='Inglés' AND *extract*(YEAR FROM fecha\_entrega)=2006 ORDER BY titulo;

Trae 44 registros (Ok)

1.2. Indicar la cantidad de películas que han sido entregadas en 2006 por un distribuidor

nacional. Trate de resolverlo utilizando ensambles.(P)

SELECT *count*(DISTINCT codigo\_pelicula) FROM unc\_esq\_peliculas.renglon\_entrega r INNER JOIN unc\_esq\_peliculas.entrega e ON

(r.nro\_entrega=e.nro\_entrega) INNER JOIN unc\_esq\_peliculas.distribuidor d on d.id\_distribuidor = e.id\_distribuidor

WHERE *EXTRACT* (YEAR FROM e.fecha\_entrega)=2006 AND d.tipo='N';

Trae 1866 (Ok)

1.3. Indicar los departamentos que no posean empleados cuya diferencia de sueldo

máximo y mínimo (asociado a la tarea que realiza) no supere el 40% del sueldo máximo.

(P) (Probar con 10% para que retorne valores)

SELECT DISTINCT d.id\_departamento FROM unc\_esq\_peliculas.departamento d WHERE NOT *EXISTS*

(SELECT 1 FROM unc\_esq\_peliculas.empleado e WHERE d.id\_departamento=e.id\_departamento GROUP BY e.id\_departamento

HAVING *MAX* (e.sueldo)-*MIN* (e.sueldo)<=0.4\**MAX*(e.sueldo));

Trae 100

select e.id\_departamento from unc\_esq\_peliculas.empleado e inner join unc\_esq\_peliculas.departamento d on

d.id\_departamento = e.id\_departamento inner join unc\_esq\_peliculas.tarea t on t.id\_tarea = e.id\_tarea

WHERE (sueldo\_maximo - sueldo\_minimo) > (sueldo\_maximo \* 0.4) group by e.id\_departamento;

--resolución compartida por whatsapp

Esta consulta devuelve el nombre de los departamentos que no tienen empleados cuya diferencia de sueldo máximo y mínimo es mayor al 40% del sueldo máximo. La subconsulta encuentra los empleados de cada departamento y calcula la diferencia entre sus sueldos máximo y mínimo. Luego, la cláusula `NOT EXISTS` selecciona los departamentos que no tienen ningún empleado que cumpla con la condición especificada.

1.4. Liste las películas que nunca han sido entregadas por un distribuidor nacional.(P)

SELECT codigo\_pelicula, tipo FROM renglon\_entrega r INNER JOIN entrega e ON r.nro\_entrega=e.nro\_entrega INNER JOIN distribuidor d ON e.id\_distribuidor = d.id\_distribuidor WHERE tipo!='N';

1.5. Determinar los jefes que poseen personal a cargo y cuyos departamentos (los del

jefe) se encuentren en la Argentina.

SELECT DISTINCT d.jefe\_departamento FROM unc\_esq\_peliculas.departamento d WHERE id\_ciudad IN

(SELECT c.id\_ciudad FROM unc\_esq\_peliculas.ciudad c WHERE id\_pais='AR') AND *EXISTS*

(SELECT 1 FROM unc\_esq\_peliculas.empleado e WHERE e.id\_empleado IS NOT NULL);

Trae 6 filas (Ok)

Esta consulta utiliza una subconsulta en la cláusula WHERE para verificar si existe al menos un empleado en cada departamento que pertenece a un jefe en Argentina. Si existe al menos un empleado, entonces se devuelve el ID del jefe del departamento.

1.6. Liste el apellido y nombre de los empleados que pertenecen a aquellos departamentos de Argentina y donde el jefe de departamento posee una comisión de más del 10% de la que posee su empleado a cargo.

SELECT e.nombre, e.apellido, e.id\_departamento, e.id\_distribuidor, d.jefe\_departamento

FROM unc\_esq\_peliculas.empleado e JOIN unc\_esq\_peliculas.departamento d ON

(e.id\_departamento = d.id\_departamento AND e.id\_distribuidor = d.id\_distribuidor)

JOIN unc\_esq\_peliculas.empleado e2 ON (d.jefe\_departamento = e2.id\_empleado)

WHERE d.id\_ciudad IN (SELECT c.id\_ciudad FROM unc\_esq\_peliculas.ciudad c WHERE c.id\_pais = 'AR')

AND e2.porc\_comision > (e.porc\_comision \* 1.1);

Trae 17 filas

### Consultas que involucran agrupamiento:

1.7. Indicar la cantidad de películas entregadas a partir del 2010, por género.

SELECT *count*(e.nro\_entrega) , genero FROM unc\_esq\_peliculas.entrega e INNER JOIN unc\_esq\_peliculas.renglon\_entrega r ON

e. nro\_entrega=r.nro\_entrega INNER JOIN unc\_esq\_peliculas.pelicula p ON r.codigo\_pelicula=p.codigo\_pelicula WHERE

*EXTRACT*(YEAR FROM fecha\_entrega)>=2010 GROUP BY genero;

SELECT genero, *count* (\*) AS cantidad FROM unc\_esq\_peliculas.entrega e INNER JOIN unc\_esq\_peliculas.renglon\_entrega r ON

e. nro\_entrega=r.nro\_entrega INNER JOIN unc\_esq\_peliculas.pelicula p ON r.codigo\_pelicula=p.codigo\_pelicula WHERE

*EXTRACT*(YEAR FROM fecha\_entrega)>=2010 GROUP BY genero;

trae 23 filas

1.8. Realizar un resumen de entregas por día, indicando el video club al cual se le

realizó la entrega y la cantidad entregada. Ordenar el resultado por fecha.

**SELECT d.nombre, e.fecha\_entrega, *count*(\*) cantidad\_entregada FROM unc\_esq\_peliculas.distribuidor d INNER JOIN unc\_esq\_peliculas.entrega e**

**ON d.id\_distribuidor = e.id\_distribuidor INNER JOIN unc\_esq\_peliculas.renglon\_entrega r ON e.nro\_entrega = r.nro\_entrega**

**GROUP BY d.nombre, e.fecha\_entrega ORDER BY e.fecha\_entrega;**

trae 5022 filas

Suponiendo que tenemos dos tablas llamadas "Entregas" y "VideoClubs", donde la tabla "Entregas" tiene una columna "Fecha" que representa la fecha de cada entrega y una columna "IdVideoClub" que indica a cuál video club se realizó la entrega, la consulta SQL para obtener un resumen de entregas por día podría ser la siguiente:

SELECT e.Fecha, v.NombreVideoClub, COUNT(\*) as CantidadEntregada

FROM Entregas e

INNER JOIN VideoClubs v ON e.IdVideoClub = v.IdVideoClub

GROUP BY e.Fecha, v.NombreVideoClub

ORDER BY e.Fecha

Esta consulta une las tablas "Entregas" y "VideoClubs" mediante la columna "IdVideoClub" y agrupa los resultados por fecha y nombre del video club. Luego, la función de agregación `COUNT(\*)` cuenta la cantidad de entregas realizadas para cada combinación de fecha y video club. Finalmente, la cláusula `ORDER BY` ordena los resultados por fecha de entrega.

1.9. Listar, para cada ciudad, el nombre de la ciudad y la cantidad de empleados

mayores de edad que desempeñan tareas en departamentos de la misma y que posean al

menos 30 empleados.

SELECT c.nombre\_ciudad, *count* (id\_empleado) FROM unc\_esq\_peliculas.ciudad c INNER JOIN unc\_esq\_peliculas.departamento d

ON c.id\_ciudad = d.id\_ciudad INNER JOIN unc\_esq\_peliculas.empleado e

ON (d.id\_departamento=e.id\_departamento AND d.id\_distribuidor=e.id\_distribuidor)

WHERE (*EXTRACT*(YEAR FROM *AGE*(fecha\_nacimiento)))>18 GROUP BY c.nombre\_ciudad HAVING *count*(\*)>30;

Trae 726 (OK)

Ejercicio 2

Considere la base de Voluntarios del Práctico 3 y resuelva las siguientes consultas (pueden

Involucrar anidamiento y/o agrupamiento).

2.1. Muestre, para cada institución, su nombre y la cantidad de voluntarios que realizan

aportes. Ordene el resultado por nombre de institución.

SELECT i.nombre\_institucion, *count*(nro\_voluntario) FROM unc\_esq\_voluntario.institucion i INNER JOIN

unc\_esq\_voluntario.voluntario v ON i.id\_institucion=v.id\_institucion WHERE horas\_aportadas>0 GROUP BY i.nombre\_institucion;

Trae 11 filas

2.2. Determine la cantidad de coordinadores en cada país, agrupados por nombre de

país y nombre de continente.Etiquete la primer columna como Número de coordinadores.

SELECT *count*(id\_coordinador) AS "Número de coordinadores"

, nombre\_pais, nombre\_continente FROM unc\_esq\_voluntario.voluntario v INNER JOIN unc\_esq\_voluntario.institucion i ON v.id\_institucion=i.id\_institucion

INNER JOIN unc\_esq\_voluntario.direccion d ON d.id\_direccion = i.id\_direccion INNER JOIN unc\_esq\_voluntario.pais p

ON p.id\_pais = d.id\_pais INNER JOIN unc\_esq\_voluntario.continente c ON c.id\_continente = p.id\_continente

GROUP BY nombre\_pais, nombre\_continente;

SELECT *count*(\*) AS "Número de coordinadores", nombre\_pais, nombre\_continente FROM unc\_esq\_voluntario.voluntario v INNER JOIN unc\_esq\_voluntario.institucion i ON v.id\_institucion=i.id\_institucion

INNER JOIN unc\_esq\_voluntario.direccion d ON d.id\_direccion = i.id\_direccion INNER JOIN unc\_esq\_voluntario.pais p

ON p.id\_pais = d.id\_pais INNER JOIN unc\_esq\_voluntario.continente c ON c.id\_continente = p.id\_continente

GROUP BY nombre\_pais, nombre\_continente;

Trae 4 filas

2.3. Escriba una consulta para mostrar el apellido, nombre y fecha de nacimiento de

cualquier voluntario que trabaje en la misma institución que el Sr. de apellido Zlotkey.

Excluya del resultado a Zlotkey.

(SELECT nombre, apellido, fecha\_nacimiento FROM unc\_esq\_voluntario.voluntario v INNER JOIN unc\_esq\_voluntario.institucion i ON

v.id\_institucion=i.id\_institucion

WHERE nombre\_institucion= (SELECT nombre\_institucion FROM unc\_esq\_voluntario.institucion i INNER JOIN unc\_esq\_voluntario.voluntario v ON

i.id\_institucion=v.id\_institucion WHERE apellido='Zlotkey')) EXCEPT (SELECT nombre, apellido, fecha\_nacimiento FROM unc\_esq\_voluntario.voluntario

WHERE apellido='Zlotkey');

Trae 33 filas

2.4. Cree una consulta para mostrar los números de voluntarios y los apellidos de todos

los voluntarios cuya cantidad de horas aportadas sea mayor que la media de las horas

aportadas. Ordene los resultados por horas aportadas en orden ascendente.

SELECT nro\_voluntario, apellido FROM unc\_esq\_voluntario.voluntario v WHERE horas\_aportadas>

(SELECT *AVG*(horas\_aportadas) FROM unc\_esq\_voluntario.voluntario) ORDER BY horas\_aportadas ASC;

Trae 51 filas

Ejercicio 3

Dada la siguiente tabla y basándose en el esquema de películas,

CREATE TABLE DistribuidorNac

(

id\_distribuidor numeric(5,0) NOT NULL,

nombre character varying(80) NOT NULL,

direccion character varying(120) NOT NULL,

telefono character varying(20),

nro\_inscripcion numeric(8,0) NOT NULL,

encargado character varying(60) NOT NULL,

id\_distrib\_mayorista numeric(5,0),

CONSTRAINT pk\_distribuidorNac PRIMARY KEY (id\_distribuidor)

);

3.1 Se solicita llenarla con la información correspondiente a los datos completos de

todos los distribuidores nacionales.

3.2 Agregar a la definición de la tabla DistribuidorNac, el campo &quot;codigo\_pais&quot; que

indica el código de país del distribuidor mayorista que atiende a cada distribuidor

nacional.(codigo\_pais character varying(5) NULL)

3.3. Para todos los registros de la tabla DistribuidorNac, llenar el nuevo campo

&quot;codigo\_pais&quot; con el valor correspondiente existente en la tabla &quot;Internacional&quot;.

3.4. Eliminar de la tabla DistribuidorNac los registros que no tienen asociado un

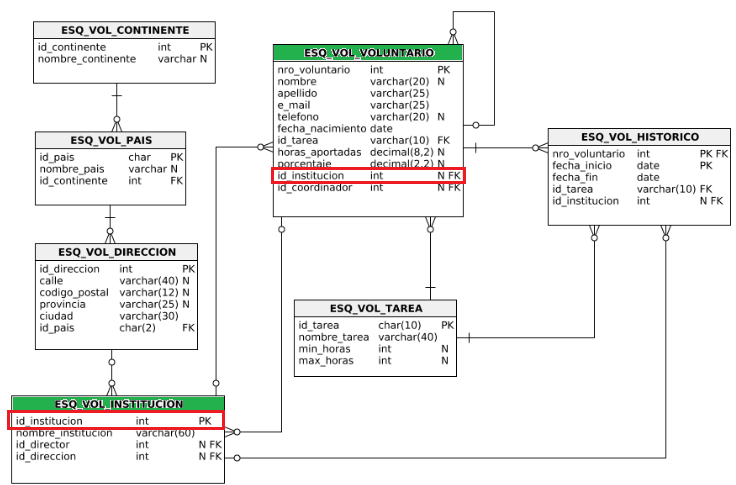
distribuidor mayorista.

CONSULTAS

1. Seleccionar: nro\_voluntario, nombre, id\_insitucion y nombre\_insitucion cuya institución se llame BOSQUEDUCA.

SELECT nro\_voluntario, nombre, i.id\_institucion, nombre\_institucion

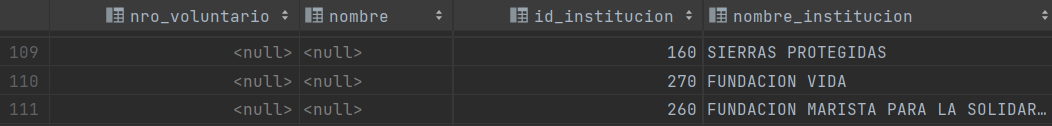
FROM voluntario v INNER JOIN institucion i ON v.id\_institucion = i.id\_institucion WHERE nombre\_institucion = 'BOSQUEDUCA';



1. Traer todos los voluntarios de cada institución. Incluso los voluntarios que no tenga institución.

SELECT nro\_voluntario, nombre, i.id\_institucion, nombre\_institucion FROM voluntario v

LEFT JOIN institucion i ON v.id\_institucion = i.id\_institucion;



Se queda con todos los datos de la tabla de la derecha

1. Traeme todos los voluntarios de las instituciones aunque sean nulos voluntarios o instituciones.

SELECT nro\_voluntario, nombre, i.id\_institucion, nombre\_institucion

FROM voluntario v FULL JOIN institucion i ON v.id\_institucion = i.id\_institucion;

1. Traeme todos los datos de los voluntarios de cada institución.

SELECT v.\* FROM voluntario v JOIN institucion i on v.id\_institucion = i.id\_institucion;

1. Selecciona todos los datos de los voluntarios cuya institución se llame REGAZZO.

SELECT v.\* FROM voluntario v WHERE v.id\_institucion =

(SELECT id\_institucion FROM institucion WHERE nombre\_institucion LIKE '%REGAZO%');

1. Dame todos los voluntarios que tengan fecha nacimiento mayor a Russel

SELECT v.\* FROM voluntario v WHERE v.fecha\_nacimiento >(SELECT fecha\_nacimiento FROM voluntario v1 WHERE v1.apellido LIKE '%Russell');