

Consultas SQL - Consultas Avanzadas



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 *  
FROM pelicula p  
WHERE p.codigo_pelicula  
      IN (SELECT r.codigo_pelicula  
          FROM renglon_entrega r  
          WHERE r.nro_entrega  
                IN (SELECT e.nro_entrega  
                    FROM entrega e  
                    WHERE EXTRACT(YEAR FROM e.fecha_entrega) = 2006  
                    AND p.idioma LIKE 'Inglés'));
```

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 (*)  
FROM entrega e INNER JOIN distribuidor d ON d.id_distribuidor = e.id_distribuidor  
WHERE tipo LIKE 'N';
```

Consultas SQL - Consultas Avanzadas

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 d.nombre
FROM unc_esq_peliculas.departamento d
WHERE d.id_departamento IN (SELECT e.id_departamento
FROM unc_esq_peliculas.empleado e
WHERE e.id_tarea NOT IN (SELECT t.id_tarea
FROM unc_esq_peliculas.tarea t
WHERE (t.sueldo_maximo - t.sueldo_minimo) <
(t.sueldo_maximo*0.4)));
```

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

```
SELECT p.codigo_pelicula, p.titulo
FROM unc_esq_peliculas.pelicula p
WHERE p.codigo_pelicula IN (SELECT codigo_pelicula
FROM unc_esq_peliculas.renglon_entrega re
WHERE re.nro_entrega IN (SELECT nro_entrega
FROM unc_esq_peliculas.entrega
WHERE id_distribuidor IN (SELECT id_distribuidor
FROM unc_esq_peliculas.distribuidor d
WHERE d.tipo = 'I'
)));
```

1.5. Determinar los jefes que poseen personal a cargo y cuyos departamentos (los del jefe) se encuentren en la Argentina.

```
SELECT e.id_empleado, e.nombre, e.apellido
FROM empleado e
WHERE e.id_jefe IN (
SELECT d.jefe_departamento
FROM departamento d
WHERE d.id_ciudad IN (
SELECT c.id_ciudad
FROM ciudad c
WHERE c.id_pais IN(
SELECT p.id_pais
FROM pais p
WHERE nombre_pais LIKE 'ARGENTINA')));
```

Consultas SQL - Consultas Avanzadas

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 *  
  
from empleado e join departamento d on (e.id_departamento=d.id_departamento  
    and e.id_distribuidor=d.id_distribuidor)  
join ciudad c on (d.id_ciudad=c.id_ciudad )  
join pais p on (c.id_pais=p.id_pais)  
where UPPER(nombre_pais)='ARGENTINA' AND  
    exists (select 1  
        from empleado e1  
        where id_empleado = d.jefe_departamento and  
            e.porc_comision >= e1.porc_comision*0.1)  
limit 15;
```

Consultas que involucran agrupamiento:

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

```
SELECT p.genero, COUNT(*)  
    FROM unc_esq_peliculas.pelicula p JOIN unc_esq_peliculas.renglon_entrega re ON  
    (p.codigo_pelicula = re.codigo_pelicula) JOIN unc_esq_peliculas.entrega e ON  
    (re.nro_entrega = e.nro_entrega)  
WHERE EXTRACT(YEAR FROM e.fecha_entrega) >= 2000  
GROUP BY p.genero;
```

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 v.razon_social, e.fecha_entrega, COUNT(*) as cant_entregada  
    FROM unc_esq_peliculas.video v JOIN unc_esq_peliculas.entrega e ON  
    (v.id_video = e.id_video) JOIN unc_esq_peliculas.renglon_entrega re ON  
    (e.nro_entrega = re.nro_entrega)  
GROUP BY v.razon_social, e.fecha_entrega  
ORDER BY e.fecha_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 distinct c.nombre_ciudad  
    FROM ciudad c JOIN departamento d ON (c.id_ciudad = d.id_ciudad)  
    WHERE (d.id_distribuidor, d.id_departamento) IN (SELECT d.id_distribuidor,  
d.id_departamento
```

Consultas SQL - Consultas Avanzadas

```
FROM departamento d JOIN empleado e ON
    (d.id_departamento = e.id_departamento) AND
    (d.id_distribuidor = e.id_distribuidor)
WHERE AGE(e.fecha_nacimiento) >= '18 years'
GROUP BY d.id_departamento, d.id_distribuidor
HAVING COUNT(*) > 30
GROUP BY c.nombre_ciudad
ORDER BY c.nombre_ciudad;
```

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(*)
FROM institucion i INNER JOIN voluntario v
    ON (i.id_institucion = v.id_institucion)
WHERE v.horas_aportadas > 0
GROUP BY i.nombre_institucion
ORDER BY i.nombre_institucion;
```

- 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(*) as numero_de_coordinadores, p.id_pais, p.id_continente
FROM pais p INNER JOIN direccion d ON (p.id_pais = d.id_pais)
    INNER JOIN institucion i ON (d.id_direccion = i.id_direccion)
    INNER JOIN voluntario v ON (i.id_institucion = v.id_institucion)
GROUP BY p.id_pais, p.id_continente;
```

- 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 v.apellido, v.nombre, v.fecha_nacimiento
FROM voluntario v
WHERE v.id_institucion IN (SELECT v1.id_institucion
    FROM voluntario v1
    WHERE v1.apellido LIKE 'Zlotkey') AND v.apellido != 'Zlotkey';
```

- 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 v.nro_voluntario, v.apellido, AVG(horas_aportadas) as hs_aportadas
FROM voluntario v
WHERE v.horas_aportadas > (SELECT AVG(v1.horas_aportadas)
    FROM voluntario v1)
GROUP BY v.nro_voluntario, v.apellido
ORDER BY hs_aportadas;
```

Consultas SQL - Consultas Avanzadas

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 "codigo_pais" 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 "codigo_pais" con el valor correspondiente existente en la tabla "Internacional".

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