

## Práctica 6. SUBCONSULTAS

### 0. Introducción

En esta práctica se introduce el uso de subconsultas. Una subconsulta es una consulta que aparece en el interior de otra consulta, es decir, es una sentencia **select** que se utiliza dentro de otra **select**. La subconsulta devuelve resultados intermedios que utiliza la consulta principal. Es una alternativa a la forma de notación en SQL de las sentencias en las que intervengan varias tablas. Las **subconsultas** siempre deben estar escritas **entre paréntesis** y se pueden escribir subconsultas con varios niveles de anidamiento.

### 1. Subconsultas no correlacionadas

Las subconsultas permiten el encadenado de varias consultas dentro de la misma sentencia SELECT. Un tipo de estas subconsultas lo forman las **no correlacionadas**. Este tipo de consultas se evalúan de dentro hacia fuera, empezando por la más interna. En muchos casos, permiten expresar de forma más legible operaciones de selección que de otra manera serían más difíciles de entender.

#### 1.1 Subconsultas que devuelven un único valor

La forma habitual de usar la subconsulta es: **expresión comparador (subconsulta)**. La subconsulta en este caso **debe devolver un único valor** (una fila con una columna).

**Ejemplo1:** Sabiendo que nom\_marca es una clave alternativa, seleccionar los códigos y nombres de los modelos de la marca AUDI.

```
SELECT cod_modelo,nom_modelo
FROM practbd.modelos
WHERE cod_marca = ( SELECT cod_marca
                    FROM practbd.marcas
                    WHERE nom_marca = 'AUDI' );
```

COD_MODELLO	NOM_MODELLO
1 A3	1.8
2 A4	1.8
3 A4	2.0
4 A6	2.0
5 A8	3.0

La subconsulta obtiene el código de la marca que se llama AUDI. La consulta principal obtiene aquellos modelos que tienen ese código de marca.

En muchas ocasiones las consultas con subconsultas se pueden escribir como consultas concatenadas y viceversa. El ejemplo anterior expresado en forma de concatenación sería:

```
SELECT cod_modelo,nom_modelo
FROM practbd.modelos JOIN practbd.marcas USING (cod_marca)
WHERE nom_marca = 'AUDI';
```

**Ejemplo2:** Mostrar los nombres de los modelos y su potencia para aquellos modelos cuya potencia sea superior a la media de todos los modelos.

```
SELECT nom_modelo,potencia
FROM practbd.modelos
WHERE potencia > ( SELECT avg(potencia)
                  FROM practbd.modelos );
```

NOM_MODELLO	POTENCIA
A8 3.0	240
VECTRA 2.0	150

La subconsulta obtiene la media de las potencias de los modelos. La consulta principal obtiene el nombre y la potencia de los que superan la media.

## **1.2 Subconsultas que devuelven un conjunto de filas**

Si la consulta devuelve un conjunto de valores no se pueden utilizar los comparadores tradicionales. En este caso se suelen utilizar los siguientes comparadores:

### **a) (atributo1 o valor1, atributo2 o valor2, ..., atributo\_n o valor\_n) IN (subconsulta)**

Devuelve True si el atributo/s o valor/es se encuentra/n dentro de la lista de valores devuelta por la subconsulta. En caso contrario devuelve False.

**Ejemplo 1:** Seleccionar los propietarios de vehículos de los códigos de marca 5 ó 7.

```
SELECT DISTINCT propietario
FROM practbd.vehiculos
WHERE cod_marca IN (5 , 7);
```

```
PROPIETAR
-----
27365138A
```

**Ejemplo 2:** Seleccionar el nombre de los modelos de los vehículos matriculados en abril de 2001.

```
SELECT DISTINCT nom_modelo
FROM practbd.modelos
WHERE (cod_marca,cod_modelo) IN
      (SELECT cod_marca,cod_modelo
       FROM practbd.vehiculos
       WHERE fec_matriculacion BETWEEN '01/04/01' AND '31/04/01');
```

```
NOM_MODELO
-----
VECTRA 2.0
```

Con concatenación sería:

```
SELECT DISTINCT nom_modelo
FROM practbd.modelos JOIN practbd.vehiculos
      USING (cod_marca,cod_modelo)
WHERE fec_matriculacion BETWEEN '01/04/01' AND '31/04/01';
```

**Ejemplo 3:** Obtener las matrículas de las multas impuestas sobre vehículos de la marca AUDI:

```
SELECT DISTINCT matricula
FROM practbd.multas
WHERE matricula IN
      (SELECT matricula
       FROM practbd.vehiculos
       WHERE cod_marca =
             (SELECT cod_marca
              FROM practbd.marcas
              WHERE nom_marca = 'AUDI'))
);
```

```
MATRICULA
-----
7715BFP
```

Con concatenación sería:

```
SELECT DISTINCT matricula
FROM practbd.multas JOIN practbd.vehiculos USING (matricula)
      JOIN practbd.marcas USING (cod_marca)
WHERE nom_marca = 'AUDI';
```

**b) (atributo1 o valor1, atributo2 o valor2, ..., atributo\_n o valor\_n) NOT IN (subconsulta)**

Devuelve True si el atributo/s o valor/es no se encuentra/n dentro de la lista de valores devuelta por la subconsulta, es decir, son distintos de **todos** los valores devueltos por la subconsulta. **También devuelve True cuando la subconsulta no devuelve nada.** En caso contrario devuelve False.

**Ejemplo 1:** Seleccionar los DNI de propietarios de algún vehículo que no es ni de la marca 1 ni de la 5

```
SELECT DISTINCT propietario
FROM practbd.vehiculos
WHERE cod_marca NOT IN (1 , 5);

PROPIETAR
-----
27365138A
```

Obsérvese que la consulta anterior **no** es la solución a la pregunta: "Seleccionar los DNI de propietarios de vehículos que no disponen de ningún vehículo ni de la marca 1 ni de la 5". Esto se debe a que si una persona tiene vehículos, por ejemplo, de la marca 1 y de la marca 7, con la solución anterior sí aparecería en la respuesta. Para esta segunda pregunta la solución sería:

```
SELECT DISTINCT propietario
FROM practbd.vehiculos
MINUS
SELECT DISTINCT propietario
FROM practbd.vehiculos
WHERE cod_marca IN (1 , 5);
```

**Ejemplo 2:** Obtener el numero de personas que no tienen multas:

```
SELECT count(*) AS sin_multas
FROM practbd.personas
WHERE nif NOT IN (SELECT DISTINCT infractor
                  FROM practbd.multas);

SIN_MULTAS
-----
3
```

**c) (atributo1 o valor1, atributo2 o valor2, ..., atributo\_n o valor\_n) comparador ALL (subconsulta)**

Se seleccionará la tupla si se cumple el comparador para **todos** los valores devueltos por la subconsulta. **También devuelve True cuando la subconsulta no devuelve nada.**

**Ejemplo:** Seleccionar los infractores que tienen multas con un importe superior a las del infractor con nif 08124356C

```
SELECT DISTINCT infractor
FROM practbd.multas
WHERE importe > ALL
( SELECT importe
  FROM practbd.multas
  WHERE infractor='08124356C' );

INFRACTOR
-----
27365138A
```

En la mayoría de sistemas si se utiliza la combinación de más de un atributo o valor para efectuar la comparación sólo se pueden utilizar los comparadores = y <>

**d) (atributo1 o valor1, atributo2 o valor2, ..., atributo\_n o valor\_n) comparador ANY (subconsulta)**

Se seleccionará la tupla si se cumple el comparador para **alguno** de los valores devueltos por la subconsulta.. **Devuelve False cuando la subconsulta no devuelve nada.**

**Ejemplo:** Seleccionar los infractores que tienen multas con un importe superior a alguna de las multas del infractor con nif 08124356C

```
SELECT DISTINCT infractor
FROM practbd.multas
WHERE importe > ANY
  ( SELECT importe
    FROM practbd.multas
    WHERE infractor='08124356C');
```

```
INFRACTOR
-----
27365138A
```

En la mayoría de sistemas si se utiliza la combinación de más de un atributo o valor para efectuar la comparación sólo se pueden utilizar los comparadores = y <>. El operador SOME es sinónimo de ANY.

## 2. Subconsultas correlacionadas

Una subconsulta anidada diremos que es correlacionada si se evalúa una vez por cada tupla procesada en la sentencia padre, es decir, la subconsulta se debe recalcularse una vez por cada fila de la consulta principal, estando la subconsulta parametrizada mediante valores de columnas de la consulta principal.

Su sintaxis general es la siguiente:

```
SELECT atributos
FROM tabla1
WHERE expresión operador
  (SELECT atributos
   FROM tabla2
   WHERE tabla2.atributo operador tabla1.atributo
  );
```

**Ejemplo:** Seleccionar toda la información de aquellos vehículos que tengan alguna multa cuando han sido conducidos por alguna persona distinta al propietario del vehículo.

```
SELECT *
FROM practbd.vehiculos a
WHERE propietario <> ANY
  (SELECT infractor
   FROM practbd.multas b
   WHERE a.matricula=b.matricula);
```

MATRICULA	FEC_MATR	PROPIETAR	BASTIDOR	COD_MARCA	COD_MODELO
0001BBB	01/04/01	27365138A		7	1

### 2.1 Subconsultas correlacionadas mediante EXISTS

Seleccionará la tupla de la consulta principal si la subconsulta devuelve algún valor. En caso contrario no se seleccionará la tupla. Con EXISTS, es frecuente utilizar subconsultas anidadas correlacionadas.

**Ejemplo:** Seleccionar toda la información de aquellos vehículos que han intervenido en multas en las que el infractor ha sido siempre el propietario.

```
SELECT *
FROM practbd.vehiculos a
WHERE EXISTS ( SELECT *
                FROM practbd.multas b
                WHERE a.matricula=b.matricula )
AND propietario = ALL ( SELECT infractor
                        FROM practbd.multas b
                        WHERE a.matricula=b.matricula );
```

MATRICULA	FEC_MATR	PROPIETAR	BASTIDOR	COD_MARCA	COD_MODELO
7715BFP	01/12/01	08124356C		1	5

### 3. Uso de subconsultas en sentencias UPDATE y DELETE

Dentro de las sentencias UPDATE y DELETE pueden incluir subconsultas tanto dentro de la cláusula WHERE para actualizaciones y borrados, como dentro de la cláusula SET para las actualizaciones.

**Ejemplo 1:** Actualizar la potencia de los modelos de la marca AUDI de mas de 150 caballos a el modelo de mayor potencia de la marca 7.

```
UPDATE modelos
SET potencia = (SELECT max(potencia) FROM modelos WHERE cod_marca=7)
WHERE potencia >150 AND cod_marca IN
      (SELECT cod_marca FROM marcas WHERE nom_marca='AUDI');
```

**Ejemplo 2:** Borrar todos los modelos de la marca OPEL de mas de 150 caballos de potencia

```
DELETE FROM modelos
WHERE potencia > 150 AND cod_marca IN
      (SELECT cod_marca FROM marcas WHERE nom_marca='OPEL');
```

## Anexo I: Ejercicios

Resuelve las siguientes consultas en SQL, intentando buscar todas las soluciones posibles.

1. Obtener los títulos de películas clasificadas como 'G' en las que no ha intervenido ningún actor

```
TITULO
-----
MINORITY KISS
DWARFS ALTER
DRUMLINE CYCLONE
```

2. Para aquellas películas en las que intervienen más de once actores obtener el título y características especiales de la película, ordenando la salida alfabéticamente por el título.

```
TITLE                                SPECIAL_FEATURES
-----
DRACULA CRYSTAL                      Commentaries
LAMBS CINCINATTI                     Trailers,Behind the Scenes
```

3. Obtener los títulos de las películas que empiezan por 'PE' en las que ha intervenido un actor o actriz que se apellida 'HACKMAN'

```
TITULO
-----
PEACH INNOCENT
PEARL DESTINY
```

4. Obtener el número de películas cuyo precio de alquiler es superior al importe medio de alquiler de todas las películas.

```
PELICULAS
-----
659
```

5. Obtener los títulos y la descripción de las películas en las que han intervenido más actores.

```
TITLE                                DESCRIPTION
-----
LAMBS CINCINATTI A Insightful Story of a Man And a Feminist who ...
```

6. Obtener las diferentes clasificaciones de películas y el número de películas para cada clasificación parental y periodos de alquiler de 3 y 4 días, siempre que existan para dicha clasificación en el periodo seleccionado más de 40 películas. Ordenar la salida alfabéticamente por el código de clasificación y para los mismos códigos, de mayor a menor por el periodo de alquiler.

```
CLASIFICACION  PERIODO  PELICULAS
-----
G              3        49
PG             4        41
PG-13          4        48
R              4        53
R              3        42
```

7. Obtener el número de actores que han intervenido en películas de todas las posibles clasificaciones existentes

```
ACTORES
-----
154
```

8. Obtener el título de la película con el identificador más alto en la que ha intervenido el actor más antiguo de los que han intervenido en películas (el que tiene menor identificador de actor)

```
TITULO
-----
WIZARD COLDBLOODED
```