

SUBCONSULTAS

Las subconsultas se utilizan para realizar filtrados con los datos de otra consulta.

Estos filtros pueden ser aplicados :

- Tanto en la cláusula WHERE para filtrar registros
- Como en la cláusula HAVING para filtrar grupos.

EJEMPLO

- Base de Datos de la NBA
 - ❖ Tabla **Jugadores** (**Código**, **Nombre**, Procedencia, Altura , Peso , Posición, *Nombre_equipo*)
 - ❖ Tabla **Equipos** (**Nombre**, Ciudad, Conferencia, **División**)
- ¿Queremos saber los **nombres de los jugadores** de la división **“SURESTE”**?

EJEMPLO

- Base de Datos de la NBA
 - ❖ Tabla **Jugadores** (**Código**, **Nombre**, Procedencia, Altura , Peso , Posición, *Nombre_equipo*)
 - ❖ Tabla **Equipos** (**Nombre**, Ciudad, Conferencia, **División**)

```
SELECT nombre FROM jugadores ;(Los nombres de todos los jugadores)
```

```
SELECT nombre FROM equipos WHERE división='SUDESTE';
```

```
(Nombres de los equipos que juegan en la división SURESTE)
```

Usamos FOREIGN KEY.... *Nombre_equipo*

- Sacar el nombre de los jugadores , **pero solo de aquellos que juegan en un equipo** que juega en la división SURESTE.

SELECT nombre FROM jugadores

WHERE nombre_equipo IN

(SELECT nombre FROM equipos WHERE
division= 'SURESTE');

La subconsulta es la sentencia select encerrada en el paréntesis.

```
SELECT nombre FROM jugadores  
WHERE nombre_equipo IN  
  (SELECT nombre FROM equipos WHERE division=  
  'SURESTE');
```

TIPOS DE SUBCONSULTAS

- **Test de comparación.** Uso de Operadores de comparación para **filtrar por el único valor** producido por la subconsulta.
- **Test de pertenencia al conjunto.** Uso de operador **IN** para filtrar los registros que coincidan con algún valor producido por la subconsulta.
- **Test de existencia.** Operador **EXISTS**. Permite filtrar los resultados de una consulta si existen filas en la subconsulta asociada.
- **Test cuantificados.** Operadores **ANY** y **ALL**
- **Subconsultas anidadas**

Una restricción importante es que la subconsulta debe estar siempre al lado derecho del operador de comparación.

Campo OPERADOR Subconsulta

Inválido

Subconsulta OPERADOR Campo

Subconsultas: Test de comparación

Consiste en usar los operadores de comparación

= , >= , <= , < > , > , <

para comparar el valor producido con un valor único generado por una subconsulta.

Este valor único producido por la subconsulta es el utilizado para filtrar.

La subconsulta sólo debe dar un ÚNICO VALOR

EJEMPLO

- Consultar el nombre del jugador de mayor altura de la NBA.

```
SELECT nombre, altura FROM jugadores  
WHERE altura = SELECT ( max(altura) FROM  
jugadores);
```

Subconsultas: Test de pertenencia al conjunto.

Consiste en utilizar el operador IN para filtrar los registros cuya expresión coincida **con algún** valor producido por la subconsulta.

```
SELECT ... FROM .....  
WHERE ..... IN (SELECT ..... FROM.....)
```

EJEMPLO

- Conocer las divisiones de la NBA donde juegan jugadores españoles (Porcedencia = 'España').
 - ❖ Tabla Jugadores (**Código**, Nombre, Procedencia, Altura , Peso , Posición, *Nombre_equipo*)
 - ❖ Tabla Equipos (**Nombre**, Ciudad, Conferencia, División)

SELECT division FROM equipos

WHERE nombre IN (SELECT nombre FROM jugadores WHERE procedencia = 'España')

Subconsulta:Test de existencia.EXISTS

El test de existencia permite filtrar los resultados de una consulta si existen filas en la subconsulta asociada, esto es, si la subconsulta genera un número de filas distinto de 0.

Para usar el test de existencia se utiliza el operador EXISTS.

SELECT columnas FROM tabla WHERE EXISTS (Subconsulta)

También puede usarse precedido por NOT. Para filtrar si no existen resultados en la subconsulta.

SELECT columnas FROM tabla WHERE NOT EXISTS (Subconsulta)

EJEMPLO

Conocer los nombre de los equipos que **no tienen** jugadores españoles.

- ❖ Tabla Jugadores (**Código**, Nombre, Procedencia, Altura , Peso , Posición, *Nombre_equipo*)
- ❖ Tabla Equipos (**Nombre**, Ciudad, Conferencia, División)

```
SELECT nombre FROM equipos
```

```
WHERE NOT EXISTS
```

```
(SELECT nombre FROM jugadores
```

```
WHERE
```

```
equipos.nombre = jugadores.nombre_equipo AND  
procedencia = 'España');
```

Subconsulta: Tests cuantificados ALL / ANY

Sirven para calcular la relación entre una expresión y todos los registros de la subconsulta (ALL) o algunos de los registros de la subconsulta (ANY).

Haciendo uso de una subconsulta de este tipo se podría saber:

Los jugadores de la NBA que pesan más que todos los jugadores españoles.

```
SELECT nombre, peso FROM jugadores
```

```
WHERE peso > ALL (SELECT peso from jugadores  
WHERE procedencia = 'España');
```

EJEMPLO

❖ Tabla Jugadores (Código, Nombre, Procedencia, Altura , Peso , Posición, *Nombre_equipo*)

Conocer **nombre**, **peso** de los jugadores que juegan como **bases** y que **pesan más** que **cualquier pivot**.

```
SELECT nombre, peso FROM jugadores
WHERE posicion='base' AND peso > ANY
(SELECT peso FROM jugadores WHERE posicion=
'pivot');
```


SUBCONSULTAS ANIDADAS

- Se puede usar una subconsulta para filtrar los resultados de otra subconsulta.
- De esta manera se anidan subconsultas.
- Las subconsultas anidadas permiten ver la información de la BD de forma estructurada y para muchos autores la S, de SQL viene de esta forma estructurada de generar consultas.

Ejemplo

- Se desea conocer el nombre de la ciudad donde juega el jugador más alto de la NBA.
 - ❖ Tabla Jugadores (**Código**, Nombre, Procedencia, Altura , Peso , Posición, *Nombre_equipo*)
 - ❖ Tabla Equipos (**Nombre**, Ciudad, Conferencia, División)

¿Qué pasos establecerías?

- 1º Obtener la altura del jugador más alto.
- 2º Obtener el nombre del equipo donde juega el jugador más alto.
- 3º Obtener la ciudad

1º Obtener la altura del jugador más alto.

X ← SELECT max(altura) FROM jugadores;

2º Obtener el nombre del equipo donde juega el jugador más alto.

Y ← SELECT Nombre_equipo FROM jugadores
WHERE altura = **X**;

3º Obtener la ciudad

SELECT ciudad from equipos WHERE nombre = **Y**

Ordenando se construye la consulta de
abajo a arriba

SELECT CIUDAD from equipos WHERE nombre = **Y**

Y ← SELECT Nombre_equipo FROM jugadores WHERE
altura = **X** ;

X ← SELECT max(altura) FROM jugadores;

Ordenando se construye la consulta de
abajo a arriba

```
SELECT CIUDAD from equipos
```

```
WHERE
```

```
nombre = (SELECT Nombre_equipo FROM jugadores
```

```
WHERE
```

```
altura = (SELECT max(altura) FROM jugadores));
```

- EJERCICIO DE SUBCONSULTAS BD MUSICA en la Intranet