Subconsultas

Una subconsulta es una consulta anidada dentro de otra consulta.

Debe tener en cuenta que no existe una única solución para resolver una consulta en SQL. En esta unidad vamos a estudiar cómo podemos resolver haciendo uso de subconsultas, algunas de las consultas que hemos resueltoen las unidades anteriores.

1.1 Tipos de subconsultas

El estándar SQL define tres tipos de subconsultas:

- **Subconsultas de fila**. Son aquellas que devuelven más de una columna, pero una única fila.
- **Subconsultas de tabla**. Son aquellas que devuelve una o varias columnas y cero o varias filas.
- **Subconsultas escalares**. Son aquellas que devuelven una columna y una fila.

1.2 Subconsultas en la cláusula WHERE

Por ejemplo, suponga que queremos conocer el nombre del producto que tiene el mayor precio. En este caso podríamos realizar una primera consulta para buscar cuál es el valor del precio máximo y otra segunda consulta para buscar el nombre del producto cuyo precio coincide con el valor del precio máximo. La consulta sería lasiguiente:

```
SELECT nombre
FROM producto
WHERE precio = (SELECT MAX(precio) FROM producto)
```

En este caso sólo hay un nivel de anidamiento entre consultas, pero pueden existir varios niveles de anidamiento.

1.3 Subconsultas en la cláusula HAVING

Ejemplo: Devuelve un listado con todos los nombres de los fabricantes que tienen el mismo número de productos que el fabricante Asus.

```
SELECT fabricante.nombre, COUNT(producto.codigo)
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
GROUP BY fabricante.codigo
HAVING COUNT(producto.codigo) >= (
    SELECT COUNT(producto.codigo)
    FROM fabricante INNER JOIN producto
    ON fabricante.codigo = producto.codigo_fabricante
    WHERE fabricante.nombre = 'Asus');
```

1.4 Subconsultas en la cláusula FROM

Ejemplo: Devuelve un listado de todos los productos que tienen un precio mayor o igual al precio medio detodos los productos de su mismo fabricante.

```
SELECT *
FROM producto INNER JOIN (
    SELECT codigo_fabricante, AVG(precio) AS media
    FROM producto
    GROUP BY codigo_fabricante) AS t
ON producto.codigo_fabricante = t.codigo_fabricante
WHERE producto.precio >= t.media;
```

1.5 Subconsultas en la cláusula SELECT

Las subconsultas que pueden aparecer en la cláusula SELECTtienen que ser subconsultas de tipo **escalar**, quedevuelven una única fila y column

---

Operadores que podemos usar en las subconsultas

Los operadores que podemos usar en las subconsultas son los siguientes:

- Operadores básicos de comparación (>,>=, <, <=, !=, <>, =).
- Predicados ALLy ANY.
- Predicado INy NOT IN.
- Predicado EXISTS y NOT EXISTS.

2.1 Operadores básicos de comparación

Los operadores básicos de comparación (>,>=, <, <=, !=, <>, =) se pueden usar cuando queremos comparar unaexpresión con el valor que devuelve una subconsulta.

Los operadores básicos de comparación los vamos a utilizar para realizar comparaciones con **subconsultas que devuelven un único valor, es decir, una columna y una fila**.

Ejemplo: Devuelve todos los productos de la base de datos que tienen un precio mayor o igual al producto máscaro del fabricante Asus.

```
SELECT *
FROM producto
WHERE precio >= (SELECT MAX(precio)
    FROM fabricante INNER JOIN producto
    ON fabricante.codigo = producto.codigo_fabricante
    WHERE fabricante.nombre = 'Asus');
```

La consulta anterior también se puede escribir con subconsultas sin hacer uso de INNER JOIN.

```
SELECT *
FROM producto
WHERE precio = (
  SELECT MAX(precio)
  FROM producto
  WHERE codigo_fabricante = (
    SELECT codigo
    FROM fabricante
    WHERE nombre = 'Asus'));
```

2.2 Subconsultas con ALLy ANY

ALL y ANY se utilizan con los operadores de comparación (>,>=, <, <=, !=, <>, =) y nos permiten comparar una expresión con el conjunto de valores que devuelve una subconsulta.

ALL y ANY los vamos a utilizar para realizar comparaciones con **subconsultas que pueden devolver varios valores, es decir, una columna y varias filas**.

Ejemplo: Podemos escribir la consulta que devuelve todos los productos de la base de datos que tienen un precio mayor o igual al producto más caro del fabricante Asus, haciendo uso de ALL. Por lo tanto, estas dos consultas darían el mismo resultado.

```
SELECT *
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
WHERE precio >= (SELECT MAX(precio)
    FROM fabricante INNER JOIN producto
    ON fabricante.codigo = producto.codigo_fabricante
    WHERE fabricante.nombre = 'Asus');
```

```
SELECT *
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
WHERE precio >= ALL (SELECT precio
    FROM fabricante INNER JOIN producto
    ON fabricante.codigo = producto.codigo_fabricante
    WHERE fabricante.nombre = 'Asus');
```

La palabra reservada SOMEes un alias de ANY. Por lo tanto, las siguientes consultas devolverían el mismo resultado:

```
SELECT s1 FROM t1 WHERE s1 <> ANY  (SELECT s1 FROM t2);

SELECT s1 FROM t1 WHERE s1 <> SOME (SELECT s1 FROM t2);
```

2.3 Subconsultas con IN y NOT IN

IN y NOT IN nos permiten comprobar si un valor está o no incluido en un conjunto de valores, que puede ser el conjunto de valores que devuelve una subconsulta.

IN y NOT IN los vamos a utilizar para realizar comparaciones con **subconsultas que pueden devolver varios valores, es decir, una columna y varias filas**.

Ejemplo: Devuelve un listado de los clientes que no han realizado ningún pedido.

```
SELECT *
FROM cliente
WHERE id NOT IN (SELECT id_cliente FROM pedido);
```

Cuando estamos trabajando con subconsultas, IN y = ANY realizan la misma función. Por lo tanto, las siguientes consultas devolverían el mismo resultado:

```
SELECT s1 FROM t1 WHERE s1 = ANY (SELECT s1 FROM t2);

SELECT s1 FROM t1 WHERE s1 IN    (SELECT s1 FROM t2);
```

Ocurre lo mismo con NOT IN y <> ALL. Por lo tanto, las siguientes consultas devolverían el mismo resultado:

```
SELECT s1 FROM t1 WHERE s1 <> ALL (SELECT s1 FROM t2);

SELECT s1 FROM t1 WHERE s1 NOT IN (SELECT s1 FROM t2);
```

Importante:

Tenga en cuenta que cuando hay un valor NULL en el resultado de la consulta interna, la consulta externa no devuelve ningún valor.

Ejemplo: Devuelve un listado con el nombre de los departamentos que no tienen empleados asociados.

```
SELECT nombre
FROM departamento
WHERE codigo NOT IN (
  SELECT codigo_departamento
  FROM empleado);
```

La consulta interna SELECT codigo_departamento FROM empleado, devuelve algunas filas con valores
NULLy por lo tanto la consulta externa no devuelve ningún valor.

La forma de solucionarlo sería quitando los valores NULLde la consulta interna:

```
SELECT nombre
FROM departamento
WHERE codigo NOT IN (
  SELECT codigo_departamento
  FROM empleado
  WHERE codigo_departamento IS NOT NULL);
```

2.4 Subconsultas con EXISTSy NOT EXISTS

Ejemplo: Devuelve un listado de los clientes que no han realizado ningún pedido.

```
SELECT *
FROM cliente
WHERE NOT EXISTS (SELECT id_cliente FROM pedido WHERE cliente.id = pedido.
    id_cliente);
```

Errores comunes

3.1 Número incorrecto de filas en la subconsulta

```
ERROR 1242 (ER_SUBSELECT_NO_1_ROW)
SQLSTATE = 21000
Message = "Subquery returns more than 1 row"
```

```
SELECT * FROM t1 WHERE column1 = (SELECT column1 FROM t2);
```

La consulta anterior sólo se podrá ejecutar cuando la consulta interna SELECT column1 FROM t2devuelva una única fila.