

71TA11_subconsultas

Comandos SQL Subconsultas

1. Subconsultas

Una **subconsulta** es una consulta dentro de otra consulta. Se puede utilizar en las cláusulas **SELECT**, **FROM**, **WHERE** y **HAVING**. El resultado de una subconsulta se puede utilizar para comparar o devolver valores a la consulta principal.

Ejemplo de uso:

```
sql
SELECT nombre_producto
FROM productos
WHERE precio > (
    SELECT precio
    FROM productos
    WHERE id_producto = 1
);
```

2. ANY

El operador **ANY** se utiliza para comparar un valor con **al menos uno** de los valores devueltos por una subconsulta. El operador **ANY** puede trabajar con operadores de comparación como **=**, **>**, **<**, etc.

Ejemplo de uso:

```
sql
SELECT nombre_producto
FROM productos
WHERE precio > ANY (
    SELECT precio
    FROM productos
    WHERE categoria = 'Electrónica'
);
```

71TA11_subconsultas

3. SOME

El operador **SOME** es sinónimo de **ANY** en la mayoría de las bases de datos. Funciona de la misma manera, permitiendo la comparación de un valor con **al menos uno** de los valores devueltos por la subconsulta.

Ejemplo de uso:

```
sql
SELECT nombre_producto
FROM productos
WHERE precio > SOME (
    SELECT precio
    FROM productos
    WHERE categoria = 'Electrónica'
);
```

4. ALL

El operador **ALL** se utiliza para comparar un valor con **todos** los valores devueltos por una subconsulta. Este operador requiere que la condición sea verdadera para **todos** los valores devueltos por la subconsulta.

Ejemplo de uso:

```
sql
SELECT nombre_producto
FROM productos
WHERE precio > ALL (
    SELECT precio
    FROM productos
    WHERE categoria = 'Electrónica'
);
```

71TA11_subconsultas

5. IN

El operador **IN** se utiliza para verificar si un valor se encuentra dentro de un conjunto de valores devueltos por una subconsulta o lista de valores. Es útil cuando se desea verificar la pertenencia a un conjunto de elementos.

Ejemplo de uso:

```
sql
SELECT nombre_producto
FROM productos
WHERE id_producto IN (
    SELECT id_producto
    FROM ventas
    WHERE id_cliente = 1
);
```

6. EXISTS

El operador **EXISTS** se utiliza para verificar si una subconsulta devuelve **al menos una fila**. Si la subconsulta devuelve alguna fila, la condición es verdadera.

Ejemplo de uso:

```
sql
SELECT nombre_cliente
FROM clientes c
WHERE EXISTS (
    SELECT 1
    FROM ventas v
    WHERE v.id_cliente = c.id_cliente AND v.cantidad > 5
);
```

71TA11_subconsultas

Ejercicios con Enunciados

Ahora que tenemos las tablas y los datos, pasemos a los ejercicios, incluyendo enunciados claros para cada comando SQL.

1. Subconsultas

Ejercicio Fácil:

Enunciado:

Encuentra el nombre de los productos cuyo precio es mayor al precio del producto con `id_producto = 1` (es decir, el precio de la "Laptop").

Ejercicio Medio:

Enunciado:

Encuentra el nombre de los clientes que han comprado más de 3 productos en total.

Explicación:

La subconsulta calcula el total de productos comprados por cada cliente y filtra a aquellos que han comprado más de 3 productos. La consulta principal selecciona los clientes cuyo `id_cliente` está en esa lista.

2. ANY

Ejercicio Fácil:

Enunciado:

Encuentra los clientes que han comprado al menos un producto con un precio mayor a 500 (es decir, al menos un producto caro como la "Laptop" o el "Smartphone").

Ejercicio Medio:

Enunciado:

Encuentra los productos que han sido comprados por al menos un cliente de la ciudad "Madrid".

71TA11_subconsultas

3. SOME

Ejercicio Fácil:

Enunciado:

Encuentra los productos cuyo precio es mayor que el precio de algunos productos comprados por el cliente con `id_cliente = 1` (es decir, el cliente Juan Pérez).

Ejercicio Medio:

Enunciado:

Encuentra los productos cuyo precio es mayor que el precio de **todos** los productos comprados por el cliente con `id_cliente = 1`.

4. ALL

Ejercicio Fácil:

Enunciado:

Encuentra los productos cuyo precio es mayor que el precio de **todos** los productos comprados por el cliente con `id_cliente = 1`.

Ejercicio Medio:

Enunciado:

Encuentra los clientes cuyo total de productos comprados es mayor que el total de productos comprados por el cliente con `id_cliente = 1`.

71TA11_subconsultas

5. IN

Ejercicio Fácil:

Enunciado:

Encuentra los clientes que han comprado el producto con `id_producto = 2` (Smartphone) o el producto con `id_producto = 3` (Tablet).

Ejercicio Medio:

Enunciado:

Encuentra los productos que han sido comprados por los clientes de las ciudades "Madrid" y "Barcelona".

6. EXISTS

Ejercicio Fácil:

Enunciado:

Encuentra los clientes que han comprado al menos un producto con un precio mayor a 300.

Ejercicio Medio:

Enunciado:

Encuentra los productos que han sido comprados por al menos un cliente de la ciudad "Valencia".

71TA11_subconsultas

Base de datos

1. Crear las Tablas

sql

-- Crear la tabla clientes

```
CREATE TABLE clientes (  
    id_cliente INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    ciudad VARCHAR(50)  
);
```

-- Crear la tabla productos

```
CREATE TABLE productos (  
    id_producto INT PRIMARY KEY,  
    nombre_producto VARCHAR(100),  
    precio DECIMAL(10, 2),  
    categoria VARCHAR(50)  
);
```

-- Crear la tabla ventas

```
CREATE TABLE ventas (  
    id_venta INT PRIMARY KEY,  
    id_cliente INT,  
    id_producto INT,  
    cantidad INT,  
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),  
    FOREIGN KEY (id_producto) REFERENCES productos(id_producto)  
);
```

71TA11_subconsultas

2. Insertar Datos en las Tablas

Clientes:

sql

```
-- Insertar datos en la tabla clientes
INSERT INTO clientes (id_cliente, nombre, ciudad) VALUES
(1, 'Juan Pérez', 'Madrid'),
(2, 'María García', 'Barcelona'),
(3, 'Luis Rodríguez', 'Valencia'),
(4, 'Ana López', 'Madrid');
```

Productos:

sql

```
-- Insertar datos en la tabla productos
INSERT INTO productos (id_producto, nombre_producto, precio,
categoria) VALUES
(1, 'Laptop', 1000.00, 'Electrónica'),
(2, 'Smartphone', 500.00, 'Electrónica'),
(3, 'Tablet', 300.00, 'Electrónica'),
(4, 'Monitor', 150.00, 'Electrónica'),
(5, 'Cámara', 450.00, 'Fotografía');
```

Ventas:

sql

```
-- Insertar datos en la tabla ventas
INSERT INTO ventas (id_venta, id_cliente, id_producto, cantidad)
VALUES
(1, 1, 1, 1), -- Juan Pérez compra 1 Laptop
(2, 2, 2, 3), -- María García compra 3 Smartphones
(3, 3, 3, 2), -- Luis Rodríguez compra 2 Tablets
(4, 1, 4, 1), -- Juan Pérez compra 1 Monitor
(5, 4, 5, 1); -- Ana López compra 1 Cámara
```