

SQL (V)

SUBCONSULTAS

- Una subconsulta es una sentencia SELECT que aparece dentro de otra sentencia SELECT que llamaremos consulta principal.
- Una subconsulta tiene la misma sintaxis que una sentencia SELECT normal exceptuando que aparece **encerrada entre paréntesis**, no puede contener la cláusula **ORDER BY**, ni puede ser la **UNION** de varias sentencias **SELECT**.
- Cuando se ejecuta una consulta que contiene una subconsulta, **la subconsulta se ejecuta por cada fila de la consulta principal**.
- Las consultas que utilizan subconsultas suelen ser **más fáciles de interpretar por el usuario**.

Referencias externas

- Frecuentemente es necesario, dentro del cuerpo de una subconsulta, hacer referencia al valor de una columna en la fila actual de la consulta principal, ese nombre de columna se denomina referencia externa.
- Una **referencia externa** es un nombre de columna que estando en la subconsulta, no se refiere a ninguna columna de las tablas designadas en la **FROM** de la subconsulta sino a una **columna de las tablas designadas en la FROM de la consulta principal**.
- Como la subconsulta se ejecuta por cada fila de la consulta principal, el valor de la referencia externa irá cambiando.

Ejemplo:

```
SELECT num_emp, nom_emp,  
      (SELECT MIN(fecha_pedido) FROM pedidos)  
FROM empleados;
```

- En este ejemplo la consulta principal es **SELECT... FROM empleados**.
- La subconsulta es (**SELECT MIN(fecha_pedido) FROM pedidos**).
- Obtenemos el número de empleado, su nombre y la fecha del primer pedido de todos los que hay en la tabla pedidos.
- En esta subconsulta tenemos una referencia externa (*numemp*), que es un campo de la tabla empleados (origen de la consulta principal).
- Cuando se ejecuta la consulta principal se coge el primer empleado y se calcula la subconsulta sustituyendo num_emp por el valor que tiene en el primer empleado. La subconsulta obtiene la fecha más antigua en los pedidos del empleado = 101,
 - se coge el segundo empleado y se calcula la subconsulta con numemp = 102 (numemp del segundo empleado)... y así sucesivamente hasta llegar al último empleado

Anidar subconsultas

- Las subconsultas pueden **anidarse** de forma que **una subconsulta aparezca en la cláusula WHERE** (por ejemplo) **de otra subconsulta** que a su vez forma parte de otra consulta principal.
- En la práctica, una consulta consume mucho más tiempo y memoria cuantos más niveles de anidamiento.

SELECT num_emp, nom_emp

FROM empleados

WHERE num_emp = (SELECT rep FROM pedidos WHERE clie = (SELECT numclie FROM clientes WHERE nombre = 'Luisa García'))

Subconsulta en las cláusulas WHERE y HAVING

- Se suelen utilizar subconsultas en las cláusulas WHERE o HAVING cuando los datos que queremos visualizar están en una tabla pero para seleccionar las filas de esa tabla necesitamos un dato que está en otra tabla.

Ejemplo:

```
SELECT num_emp, nom_emp
```

```
FROM empleados
```

```
WHERE fecha_contrato = (SELECT MIN(fecha_pedido) FROM pedidos)
```

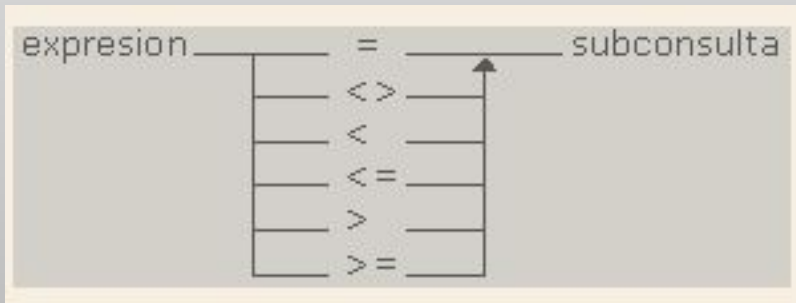
- En el ejemplo listamos el número y nombre de los empleados cuya fecha de contrato sea igual a la primera fecha de todos los pedidos de la empresa.

Condiciones de selección con subconsultas

- Las **condiciones de selección** son las condiciones que pueden aparecer en la cláusula **WHERE** o **HAVING**.

COMPARACIÓN CON SUBCONSULTA

- Se utiliza para comparar un valor de la fila que se está examinado con un único valor producido por la subconsulta. La subconsulta debe devolver una **única columna**, sino se produce un error.



SELECT oficina, ciudad

FROM oficinas

**WHERE objetivo > (SELECT SUM(ventas) FROM empleados WHERE
empleados.oficina = oficinas.oficina).**

Lista las oficinas cuyo objetivo sea superior a la suma de las ventas de sus empleados.

PERTENENCIA A CONJUNTO (IN)

- Examina si el **valor** de la expresión es uno de los valores **incluidos en la lista de valores producida por la subconsulta.**

```
expresion — IN — subconsulta
```

SELECT numemp, nombre, oficina

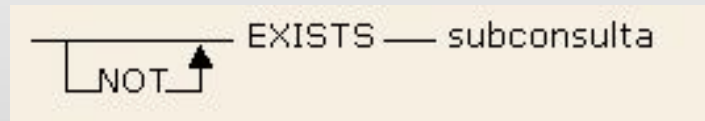
FROM empleados

WHERE oficina IN (SELECT oficina FROM oficinas WHERE region = 'este')

lista los empleados de las oficinas del *este*, ya que con la subconsulta se obtiene la lista de los números de oficina del *este* y con la consulta principal se obtiene los empleados cuyo número de oficina sea uno de los números de oficina del *este*.

EXISTENCIA (EXISTS)

- Se examina si la subconsulta produce alguna fila de resultados.



Ejemplo: N°, nombre de los empleados que trabajan en las oficinas de la región del este:

- SELECT num_emp, nom_emp, oficina**
FROM empleados
WHERE EXISTS (SELECT * FROM oficinas WHERE region = 'ESTE' AND
empleados.oficina = oficinas.cod_oficina)

num_emp integer	nom_emp character varying(30)	oficina integer
101	ANTONIO GARCIA	12
103	JUAN ROVIRA	12
104	JOSE GONZALEZ	12
105	VICENTE MARTINEZ	13
106	LUIS MORENO	11
109	MARIA PEREZ	11

NOT EXISTS

- El el caso contrario al anterior. Ahora el where se cumple si la subconsulta no devuelve ningún resultado.

Ejemplo:

- **SELECT num_emp, nom_emp, oficina
FROM empleados
WHERE NOT EXISTS (SELECT * FROM oficinas WHERE region =
'ESTE' AND empleados.oficina = oficinas.cod_oficina)**

Busca los números, nombres de empleados y oficinas que no pertenecen a la región del ESTE

num_emp integer	nom_emp character varying(30)	oficina integer
102	ALVARO LOPEZ	21
107	JORGE GARCIA	22
108	ANA MARTIN	21
110	JUAN PEREZ	