

Subconsultas

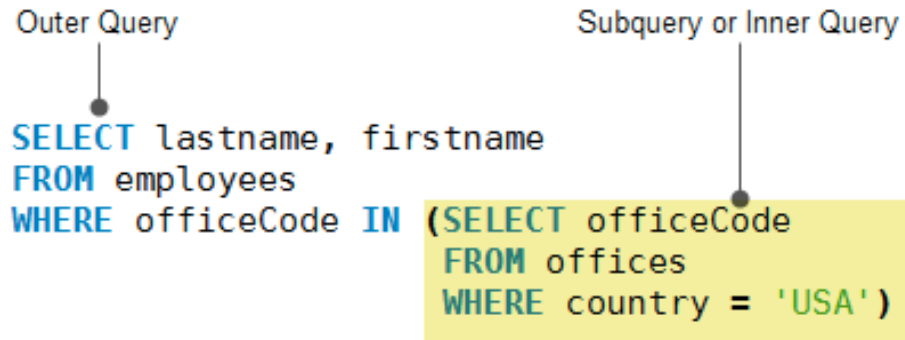
SUBCONSULTAS

En MySQL una subconsulta es una **consulta anidada dentro de otra consulta** tales como **SELECT, INSERT, UPDATE o DELETE**. Además, una subconsulta puede anidarse dentro de otra subconsulta.

Una **subconsulta** de MySQL se denomina **consulta interna**, mientras que la **consulta que contiene la subconsulta** se denomina **consulta externa**. Una subconsulta **puede usarse en cualquier lugar donde se use esa expresión** y debe cerrarse entre paréntesis.

Subconsulta dentro del IN

La siguiente consulta devuelve a los empleados que trabajan en oficinas ubicadas en los EE. UU.



The diagram shows a SQL query with two parts. The 'Outer Query' is labeled with a line pointing to the first part of the query: `SELECT lastname, firstname FROM employees WHERE officeCode IN`. The 'Subquery or Inner Query' is labeled with a line pointing to the second part of the query, which is enclosed in a yellow box: `(SELECT officeCode FROM offices WHERE country = 'USA')`.

```
SELECT lastname, firstname
FROM employees
WHERE officeCode IN (SELECT officeCode
                     FROM offices
                     WHERE country = 'USA')
```

- La subconsulta devuelve todos los códigos de oficina de las oficinas ubicadas en los EE. UU.
- La consulta externa selecciona el apellido y el nombre de los empleados que trabajan en las oficinas cuyos códigos de oficina están en el conjunto de resultados devuelto por la subconsulta.

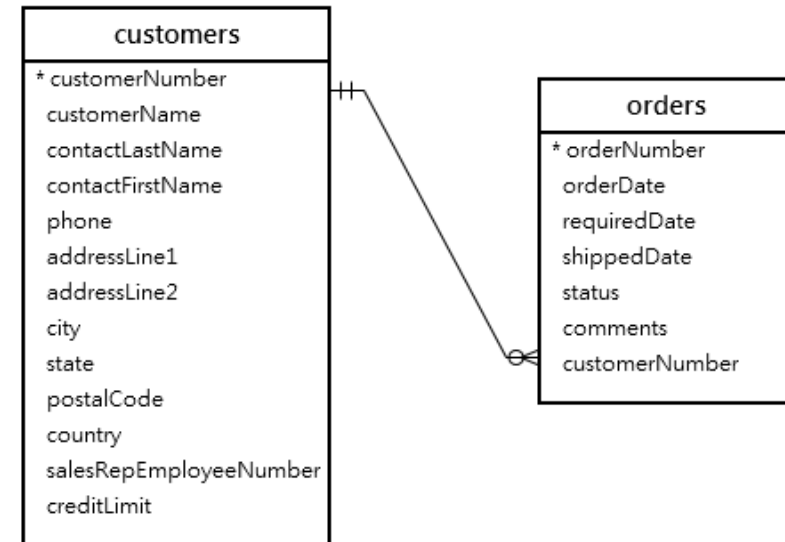
Cuando se ejecuta la consulta, la subconsulta se ejecuta primero y devuelve un conjunto de resultados. Luego, este conjunto de resultados se utiliza como entrada para la consulta externa.

Subconsulta dentro en el NOT IN

Si una subconsulta devuelve más de un valor, puede usar también el operador NOT IN en la cláusula WHERE.

Por ejemplo, puede usar una subconsulta con el operador NOT IN para encontrar los clientes que no han realizado ningún pedido de la siguiente manera:

```
SELECT
    customerName
FROM
    customers
WHERE
    customerNumber NOT IN (SELECT DISTINCT
                           customerNumber
                           FROM
                               orders);
```



	customername
►	Havel & Zbyszek Co
	American Souvenirs Inc
	Porto Imports Co.
	Asian Shopping Network, Co
	Natürlich Autos
	ANG Resellers
	Messner Shopping Network
	Franken Gifts, Co
	BG&E Collectables

Subconsulta en la cláusula WHERE

Puede usar **operadores de comparación**, por ejemplo, =, >, < para comparar un solo valor devuelto por la subconsulta con la expresión en la cláusula **WHERE**.

Por ejemplo, la siguiente consulta devuelve al cliente que tiene el pago máximo.

payments
* customerNumber
* checkNumber
paymentDate
amount

```
SELECT
  customerNumber,
  checkNumber,
  amount
FROM
  payments
WHERE
  amount = (SELECT MAX(amount) FROM payments);
```

	customerNumber	checkNumber	amount
►	141	JE105477	120166.58

Puede usar otros operadores de comparación como mayor que (>), menor que (<).

Por ejemplo, puede encontrar clientes cuyos pagos sean mayores que el pago promedio utilizando una subconsulta:

- Usar una subconsulta para calcular el pago promedio usando la AVG función agregada.
- Luego, consultar los pagos que son mayores que el pago promedio devuelto por la subconsulta en la consulta externa.

```
SELECT
  customerNumber,
  checkNumber,
  amount
FROM
  payments
WHERE
  amount > (SELECT
    AVG(amount)
  FROM
    payments);
```

	customerNumber	checkNumber	amount
►	112	HQ55022	32641.98
	112	ND748579	33347.88
	114	GG31455	45864.03
	114	MA765515	82261.22
	114	NR27552	44894.74
	119	LN373447	47924.19
	119	NG94694	49523.67
	121	DB889831	50218.95
	121	MA302151	34638.14

Subconsulta en la cláusula FROM

Cuando usa una subconsulta en la cláusula FROM, **el conjunto de resultados devuelto por una subconsulta se usa como una tabla temporal**. Esta tabla se conoce como tabla derivada o subconsulta materializada.

La siguiente subconsulta encuentra el número máximo , mínimo y promedio de artículos en los pedidos de venta:

```
SELECT
    MAX(items),
    MIN(items),
    FLOOR(AVG(items))
FROM
    (SELECT
        orderNumber, COUNT(orderNumber) AS items
    FROM
        orderdetails
    GROUP BY orderNumber) AS lineitems;
```

	MAX(items)	MIN(items)	FLOOR(AVG(items))
►	18	1	9

Tener en cuenta que **FLOOR()** se utiliza para eliminar lugares decimales de los valores promedio de los elementos.

Subconsulta correlacionada MySQL

Una **subconsulta** es **independiente** cuando puede ejecutar la subconsulta como una consulta **separada** de la consulta que la utiliza.

Una **subconsulta correlacionada** es una subconsulta que utiliza los datos de la consulta externa. En otras palabras, una subconsulta correlacionada **depende de la consulta externa**. Una subconsulta correlacionada **se evalúa una vez para cada fila en la consulta externa**.

Ejemplo de subconsulta correlacionada

Seleccionamos productos cuyos precios de compra son mayores que el precio de compra promedio de todos los productos en cada línea de productos.

```
SELECT
    productname,
    buyprice
FROM
    products p1
WHERE
    buyprice > (SELECT
        AVG(buyprice)
    FROM
        products
    WHERE
        productline = p1.productline)
```

	productname	buyprice
▶	1952 Alpine Renault 1300	98.58
	1996 Moto Guzzi 1100i	68.99
	2003 Harley-Davidson Eagle Drag Bike	91.02
	1972 Alfa Romeo GTA	85.68
	1962 LanciaA Delta 16V	103.42
	1968 Ford Mustang	95.34
	2001 Ferrari Enzo	95.59
	1958 Setra Bus	77.90

La consulta interna se ejecuta para cada línea de producto porque la línea de producto se cambia para cada fila. Por lo tanto, el precio de compra promedio también cambiará. La consulta externa solo filtra los productos cuyo precio de compra es mayor que el precio de compra promedio por línea de productos de la subconsulta.

Subconsulta con EXISTS y NOT EXISTS

Cuando se utiliza una subconsulta con el operador **EXISTS** o **NOT EXISTS**, una subconsulta **devuelve un valor booleano** de TRUE o FALSE. La siguiente consulta ilustra una subconsulta utilizada con el operador EXISTS:

```
SELECT
    *
FROM
    table_name
WHERE
    EXISTS( subquery );
```

En la consulta anterior, si la subconsulta devuelve alguna fila, **EXISTS subquery** devuelve TRUE, de lo contrario, devuelve FALSE. Los **EXISTS** y **NOT EXISTS** se utilizan a menudo en las subconsultas correlacionadas.

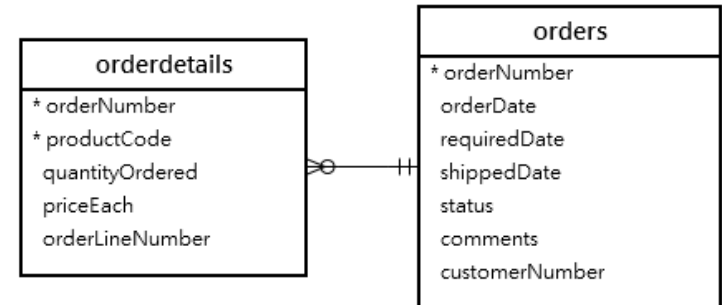
Ejemplo de subconsultas usando el EXISTS / NOT EXISTS

La siguiente consulta busca pedidos de ventas cuyos valores totales son superiores a 60K.

```
SELECT
    orderNumber,
    SUM(priceEach * quantityOrdered) total
FROM
    orderdetails
    INNER JOIN
    orders USING (orderNumber)
GROUP BY orderNumber
HAVING SUM(priceEach * quantityOrdered) > 60000;
```

orderNumber	total
10165	67392.85
10287	61402.00
10310	61234.67

Devuelve 3 filas, lo que significa que hay 3 pedidos de ventas cuyos valores totales son superiores a 60K.



Puede usar la consulta anterior como una subconsulta correlacionada para encontrar clientes que hayan realizado al menos un pedido de ventas con un valor total superior a 60K utilizando el **operador EXISTS**:

```
SELECT
    customerNumber,
    customerName
FROM
    customers
WHERE
    EXISTS( SELECT
        orderNumber, SUM(priceEach * quantityOrdered)
    FROM
        orderdetails
        INNER JOIN
        orders USING (orderNumber)
    WHERE
        customerNumber = customers.customerNumber
    GROUP BY orderNumber
    HAVING SUM(priceEach * quantityOrdered) > 60000);
```

customerNumber	customerName
148	Dragon Souvenirs, Ltd.
259	Toms Spezialitäten, Ltd
298	Vida Sport, Ltd