

```
/*REPASO DE LA FUNCION RAND*/  
SELECT  
*  
from classicmodels.customers c  
order by RAND()  
limit 10
```

```
/*EMPEZAMOS CON LAS SUBCONSULTAS Y LAS TABLAS DERIVADAS EN SQL*/  
SELECT  
firstName,  
lastName,  
officeCode  
from classicmodels.employees e  
where officeCode in (SELECT  
officeCode  
from classicmodels.offices o  
where country = 'USA') -- EN VEZ DE TENER QUE MIRAR PRIMERO QUE NUMEROS DE OFICINA TIENE USA Y  
LUEGO  
-- LANZAR LA CONSULTA PONIENDO WHERE OFFICE CODE in (1,2,3) PUEDO HACER UNA SUBCONSULTA  
DIRECTAMENTE ELIGIENDO DE  
-- LA TABLA DE OFFICES, AQUELLAS OFICINAS QUE SOLO SE ENCUENTREN EN ESTADOS UNIDOS.
```

```
/*OTRO EJEMPLO*/  
SELECT  
*  
FROM classicmodels.payments p  
where amount >= (select avg(amount) from classicmodels.payments)  
order by amount desc -- QUE SOLO ME SALGAN LOS PAGOS MAYOR A LA MEDIA DE PAGOS, EN ESTE CASO  
32.
```

```
/*OTRO EJEMPLO*/  
SELECT  
count(ordernumber) as pedidos,  
avg(importe_pedido) as media_importe  
from  
(  
SELECT  
orderNumber,  
sum(quantityOrdered*priceEach) as importe_pedido  
FROM classicmodels.orderdetails o  
group by 1  
) as tabla_derivada -- PRIMERO HE HECHO LA CONSULTA DE ABAJO PARA VER EL IMPORTE DE CADA  
PEDIDO, DESPUES LA  
-- CONSULTA DE ARRIBA PARA QUE ME CUENTE CUANTOS PEDIDOS HAY EN TOTAL Y CUAL ES LA MEDIA DEL  
IMPORTE
```

```
/*CONSULTAS CORRELACIONADAS (SE USAN MUY POCO, YA QUE SON POCO VISUALES Y SE PUEDEN HACER  
MEJOR USANDO IN*/  
SELECT  
productName,  
productLine,  
buyPrice  
FROM classicmodels.products p  
where  
    buyPrice > (SELECT  
        avg (buyPrice)  
        from classicmodels.products p  
        where productLine = p.productLine) -- CALCULA LOS PRECIOS MEDIOS POR LINEA DE PRODUCTO
```

Y LO CRUZA CON LINEA DE PRODUCTO DE LA TABLA DE ARRIBA

**order by 2,3 DESC** -- DE ESTA FORMA EL PRECIO DE COMPRA ES MAYOR QUE LA MEDIA DE CADA LINEA DE PRODUCTO (MOTOS, COCHES, ETC)

/\*OTRO EJEMPLO\*/

**SELECT**

\*

**FROM** customers c

**where EXISTS** (

**SELECT**

customerNumber, **sum**(amount) **as** cantidad

**from** payments p

**where year**(paymentdate)=2004 -- SUMA DE PAGOS HECHA EN 2004 MAYOR QUE 100K CRUZADA CON LA TABLA DE CLIENTES

**and** customerNumber = c.customerNumber

**group by 1**

**having** cantidad>100000

)

/\*EJEMPLO ANTERIOR HECHO CON UN IN, MUCHO MAS CLARO Y MEJOR\*/

**SELECT**

\*

**FROM** classicmodels.customers c

**WHERE** customerNumber **IN** (

**SELECT**

customerNumber

**FROM** payments p

**WHERE YEAR**(PAYMENTDATE)=2004

**GROUP BY 1**

**HAVING SUM**(AMOUNT)>100000 -- TODOS LOS CLIENTES CON UN TOTAL DE PAGOS EN 2004 DE 100K

)

/\*UNION DE TABLAS\*/

**SELECT**

firstname,

lastname

**from** employees e

**UNION**

**SELECT**

contactfirstname,

contactlastname

**from** customers c

-- HECHO POR MI PARA SABER CUALES SON LOS EMPLEADOS QUE TAMBIEN SON CLIENTES

**SELECT**

\*

**from** classicmodels.employees e

**WHERE** firstName **IN** (

**SELECT**

firstName

**FROM** customers c

)

-- OTRA FORMA PARA DISTINGUIR QUIEN ES EMPLEADO Y QUIEN ES CLIENTE\*/

**SELECT**

firstname,

lastname,

```
'Empleado' as tipo -- DIME QUE ESTA SELECCION SON EMPLEADOS
from employees e
UNION
SELECT
contactfirstname,
contactlastname,
'Cliente' as tipo      -- DIME QUE ESTA SELECCION SON CLIENTES.
from customers c      -- EN CASO DE NO PONER 'AS' SQL SABRIA QUE LO TIENE QUE METER EN TIPO
PORQUE ESTÁ EN EL MISMO ORDEN :)
```

```
/*OTRA VARIANTE*/
SELECT
COUNT(*) AS ca_personas
from
(
SELECT
firstname,
lastname,
'Empleado' as tipo
from employees e
UNION
SELECT
contactfirstname,
contactlastname,
'Cliente'
from customers c
) as personas -- NOMBRE OBLIGATORIO CUANDO SE HACE UNA TABLA DERIVADA
```

```
/*OTRA VARIANTE*/
SELECT
tipo, -- PARA HACERME LA CUENTA DE TIPO DE PERSONA, EMPLEADO O CLIENTE
COUNT(*) AS ca_personas
from
(
SELECT
firstname,
lastname,
'Empleado' as tipo
from employees e
UNION
SELECT
contactfirstname,
contactlastname,
'Cliente'
from customers c
) as personas
group by 1
```

```
/*CONSULTA DE EJEMPLO PARA APLICAR CLAUSULA WITH*/
SELECT
tabla.customerGroup,
count(*) as ca_clientes,
sum(sales) as importe_ventas,
avg(sales) as importe_medio

from
(
select -- PRIMERO SE HACE ESTO
    customernumber,
```

```

round(sum(amount)) as sales,
(CASE
  when sum(amount) < 10000 then 'silver'
  when sum(amount) between 10000 and 100000 then 'gold'
  when sum(amount) > 100000 then 'platinum'
END) as customergroup
FROM
payments P
where
year(paymentdate) = 2004
group by customernumber -- HASTA AQUI, ESTO ES LA PRIMERA CONSULTA

) as tabla

group by tabla.customergroup

```

```
/*APLICO CLAUSULA WITH A LA PRIMER CONSULTA ANTERIOR*/
```

```

with rango_pagos as (
select
  customernumber,
  round(sum(amount)) as sales,
  (CASE
    when sum(amount) < 10000 then 'silver'
    when sum(amount) between 10000 and 100000 then 'gold'
    when sum(amount) > 100000 then 'platinum'
  END) as customergroup
FROM
payments P
where
year(paymentdate) = 2004
group by customernumber
)

```

```

SELECT -- AHORA HAGO LA SUBCONSULTA AHORRANDOME DE ESCRIBIR TODO LO ANTERIOR
tabla.customergroup,
count(*) as ca_clientes,
sum(sales) as importe_ventas,
avg (sales) as importe_medio

```

```

from rango_pagos tabla
group by tabla.customergroup -- AL FINAL ES LO MISMO PORQUE TENGO QUE ESCRIBIR LO ANTERIOR
TAMBIEN PARA QUE ME DETECTE LA TABLA RANGO_PAGOS

```

```
/*EJERCICIOS SUBCONSULTAS Y TABLAS DERIVADAS*/
```

```
/*OBTENER LOS CLIENTES SIN COMPRAS*/
```

```

SELECT
customername
from classicmodels.customers c
where customerNumber not in (
  select customernumber
  from classicmodels.orders
)

```

```
/*ENCUENTRA EL NOMBRE DE LOS CLIENTES QUE HAYAN REALIZADO MÁS DE 10 PEDIDOS EN LA TABLA
ORDERS*/
```

```

SELECT
customerName

```

```
FROM classicmodels.customers c
WHERE customerNumber IN (
    SELECT customerNumber
    FROM classicmodels.orders o
    GROUP BY 1
    HAVING COUNT(customerNumber) > 10
)
```

```
/*ENCUENTRA EL NOMBRE DE LOS PRODUCTOS QUE HAYAN SIDO COMPRADOS EN MÁS DE 25 PEDIDOS
DIFERENTES EN LA TABLA ORDERDETAILS*/
```

```
SELECT
*
FROM classicmodels.products p -- PARA CONSULTAR

SELECT * FROM classicmodels.orderdetails o -- PARA CONSULTAR

SELECT
PRODUCTNAME
FROM classicmodels.products p
WHERE productCode IN (
    SELECT
    productCode
    FROM classicmodels.orderdetails o
    GROUP BY 1
    HAVING COUNT(productCode) > 25
)
```

```
/*CREA UNA SEGMENTACIÓN DE CLIENTES PLATA (HACEN PAGOS DE MENOS DE 10K EN AÑO),
ORO (DE 10K A 100K) Y PLATINO (MÁS DE 100K). CALCULA CUÁNTOS CLIENTES DE CADA TIPO
HAY PARA EL AÑO 2004*/
```

```
-- Consulta para calcular ingresos y rango por cliente
SELECT
CUSTOMERNUMBER,
SUM(AMOUNT) AS PAGOS,
(CASE
    WHEN sum(amount) < 10000 THEN 'PLATA'
    WHEN sum(amount) BETWEEN 10000 AND 100000 THEN 'ORO'
    WHEN sum(amount) > 100000 THEN 'PLATINO'
END) as grupo_clientes
FROM classicmodels.payments p
WHERE
    year(paymentDate)=2004
GROUP BY customerNumber
```

```
-- Agregamos esta consulta
```

```
select
tabla.grupo_clientes,
count(*) as ca_clientes,
sum(pagos) as importe_ventas,
avg(pagos) as importe_medio
```

```
from
(
    SELECT
    customerNumber,
    SUM(AMOUNT) AS pagos,
    (CASE
        WHEN sum(amount) < 10000 THEN 'PLATA'
```

```
        WHEN sum(amount) BETWEEN 10000 AND 100000 THEN 'ORO'
        WHEN sum(amount) > 100000 THEN 'PLATINO'
    END) as grupo_clientes
FROM
classicmodels.payments p

WHERE
YEAR(paymentDate) = 2004
GROUP BY customerNumber
) tabla -- ALIAS DE LA SUBCONSULTA
group by tabla.grupo_clientes
```