

```
/*FUNCIONES VENTANA*/
```

```
-- FUNCIONES DE VENTANA: EJEMPLO SUMA ACUMULATIVA
```

```
with ventas_mes as (      -- ASI CREAMOS UNA VENTANA
```

```
SELECT
```

```
year(orderDate) as anno,
```

```
month(orderDate) as mes,
```

```
sum(importe.importe_pedido) as importes_pedido,
```

```
count(*) as ca_pedidos
```

```
from classicmodels.orders pedidos
```

```
left join (
```

```
  select
```

```
  detalle.ordernumber,
```

```
  sum(quantityordered*priceeach) as importe_pedido
```

```
  from classicmodels.orderdetails detalle
```

```
  group by 1
```

```
  order by 2 desc
```

```
) importe
```

```
on pedidos.ordernumber=importe.ordernumber
```

```
group by 1,2
```

```
)
```

```
SELECT
```

```
base.*,
```

```
sum(importes_pedido) over(partition by anno order by mes) as importes_acumulados,
```

```
sum(ca_pedidos) over(partition by anno order by mes) as pedidos_acumulados
```

```
from ventas_mes base
```

```
-----  
-----
```

```
-- FUNCIONES DE VENTANA: EJEMPLO DE LISTADO DE FUNCIONES DE AGREGACION
```

```
with ventas_mes as (
```

```
SELECT
```

```
year(orderDate) as anno,
```

```
month(orderDate) as mes,
```

```
sum(importe.importe_pedido) as importes_pedido,
```

```
count(*) as ca_pedidos
```

```
from classicmodels.orders pedidos
```

```
left join (
```

```
  select
```

```
  detalle.ordernumber,
```

```
  sum(quantityordered*priceeach) as importe_pedido
```

```
  from classicmodels.orderdetails detalle
```

```
  group by 1
```

```
  order by 2 desc
```

```
) importe
```

```
on pedidos.ordernumber=importe.ordernumber
```

```
group by 1,2
```

```
)
```

```
SELECT
```

```

base.*,
sum(importes_pedido) over ( partition by anno
                           order by mes
                           rows between 2 preceding and current row -- COJO LAS DOS
                           ANTERIORES MAS LA ACTUAL, ES DECIR 3 MESES
                           ) as importe_pedido,

avg(importes_pedido) over ( partition by anno -- si no hago la particion tengo que ordenar por
                           mes y año
                           order by mes
                           rows between 2 preceding and current row -- COJO LAS DOS
                           ANTERIORES MAS LA ACTUAL, ES DECIR 3 MESES
                           ) as importe_trimestral_medio,

max(importes_pedido) over ( partition by anno
                           order by mes
                           rows between 2 preceding and current row -- COJO LAS DOS
                           ANTERIORES MAS LA ACTUAL, ES DECIR 3 MESES
                           ) as importe_trimestral_max,

min(importes_pedido) over ( partition by anno
                           order by mes
                           rows between 2 preceding and current row -- COJO LAS DOS
                           ANTERIORES MAS LA ACTUAL, ES DECIR 3 MESES
                           ) as importe_trimestral_min

from ventas_mes base

-----
-----

-- FUNCIONES DE VENTANA: EJEMPLO DE LISTADO DE VALOR ENESIMO

with ventas_mes as (

SELECT
year(orderDate) as anno,
month(orderDate) as mes,
sum(importe.importe_pedido) as importes_pedido,
count(*) as ca_pedidos
from classicmodels.orders pedidos

left join (
select
detalle.ordernumber,
sum(quantityordered*priceeach) as importe_pedido
from classicmodels.orderdetails detalle
group by 1
order by 2 desc
) importe

on pedidos.ordernumber=importe.ordernumber
group by 1,2

)

SELECT
base.*,
LAG(importes_pedido, 1) over ( order by anno, mes) as anterior,
LEAD(importes_pedido, 1) over ( order by anno, mes) as posterior,
LAST_VALUE(mes) over ( partition by anno
                      order by importes_pedido

```

```

        rows between UNBOUNDED PRECEDING and UNBOUNDED FOLLOWING ) as ultimo,
-- MES EN EL QUE EL IMPORTE ES EL MAXIMO
FIRST_VALUE(mes) over ( partition by anno
                        order by importes_pedido
                        rows between UNBOUNDED PRECEDING and UNBOUNDED FOLLOWING) as primero
-- MES EN EL QUE EL IMPORTE ES EL MINIMO

from ventas_mes base
order by 1,2 -- ORDENAR POR AÑO Y POR MES

-----

-- FUNCIONES DE VENTANA: EJEMPLO DE LISTADO DE ORDENACION (IDEAL PARA ORDENAR POR RESULTADOS O
POR RENDIMIENTO)

with ventas_cliente as (

SELECT
pedidos.customerNumber,
sum(importes.importe_pedido) as importes_pedido,
count(*) as ca_pedidos
FROM classicmodels.orders pedidos
left join (
    select
        detalle.orderNumber,
        sum(detalle.quantityordered*priceeach) as importe_pedido
    from classicmodels.orderdetails detalle
    group by 1
    order by 2 desc
) importes

on pedidos.orderNumber=importes.orderNumber
where year(pedidos.orderdate)=2004
group by 1

)

select
base.*,
round(base.importes_pedido/10000) as importe_redondeado, -- REDONDEA A LA POSICION 5, DE
DERECHA A IZQUIERDA, ES DECIR, A DIEZ MILES
RANK() over (order by round(base.importes_pedido/10000)) as fv_rank,
DENSE_RANK() over (order by round(base.importes_pedido/10000)) as fv_dense_rank,
ROW_NUMBER () over (order by round(base.importes_pedido/10000)) as fv_row_number

from ventas_cliente base

-----

-- FUNCIONES DE VENTANA: EJEMPLO DE LISTADO DE AGRUPACION

with ventas_clientes as (

SELECT
pedidos.customerNumber,
sum(importes.importe_pedido) as importes_pedido,
count(*) as ca_pedidos
FROM classicmodels.orders pedidos
left join (
    select

```

```

    detalle.orderNumber,
    sum(detalle.quantityordered*priceeach) as importe_pedido
  from classicmodels.orderdetails detalle
  group by 1
  order by 2 desc
) importes

on pedidos.orderNumber=importes.orderNumber
where year(pedidos.orderdate)=2004
group by 1

)

SELECT
base.*,
PERCENT_RANK() over (order by importes_pedido) as percentil_decimal, -- VA DE CERO A UNO
round(100*PERCENT_RANK() over(order by importes_pedido)) as percentil, -- X POR CIENTO ESTA
POR DEBAJO DEL CLIENTE DE ESA FILA, ES EL VALOR QUE MARCA FRONTERA
NTILE(10) over (order by importes_pedido) as decil,
CEIL(10*PERCENT_RANK() over (order by importes_pedido)) as decil_bis,
CUME_DIST() over (order by importes_pedido) as distribucion

from ventas_clientes base

-----
-----

/*EJERCICIOS FUNCIONES VENTANA*/

/* OBTENER EL PRODUCTO MÁS VENDIDO POR LÍNEA DE PRODUCTOS*/

with tabla_productos as (

select
productos.productCode,
productos.productName,
productos.productLine,
sum(detalle_pedidos.quantityOrdered) as cantidad -- para sumar la cantidad vendida de cada
producto
from classicmodels.products productos
LEFT JOIN classicmodels.orderdetails detalle_pedidos
ON productos.productCode = detalle_pedidos.productCode
group by 1,2,3
)

select
tabla_productos.*,
row_number() over(partition by tabla_productos.productLine
order by cantidad desc) as orden

from tabla_productos

-- necesito solo quedarme con las filas de orden 1 para asi quedarme con las lineas de
producto con mayor cantidad vendida, para ello:

with tabla_productos as (

select
productos.productCode,
productos.productName,
productos.productLine,
sum(detalle_pedidos.quantityOrdered) as cantidad

```

```

from classicmodels.products productos
LEFT JOIN classicmodels.orderdetails detalle_pedidos
ON productos.productCode = detalle_pedidos.productCode
group by 1,2,3
)
-- tengo que lanzar de nuevo una consulta con select, al hacer la ventana como subconsulta
agrupada

select * from -- en vez de asterico puedo seleccionar las columnas que quiera

    (select
    tabla_productos.*,
    row_number() over( partition by tabla_productos.productLine
                        order by cantidad desc) as orden -- si ordeno ascendente me quedaria
con los menos vendidos

    from tabla_productos) as tabla
where tabla.orden=1 -- para agrupar la ventana simplemente tengo que ponerle unos parentesis
-----

/*CALCULAR LA VARIACIÓN DEL IMPORTE DE LOS PEDIDOS RESPECTO AL MES ANTERIOR*/

with ventas_mes as (

select
year(orderdate) as anno,
month(orderdate) as mes,
sum(importe.importe_pedidos) as importe_pedidos,
count(*) as ca_pedidos
from classicmodels.orders pedidos
left join (
    select
    detalle.orderNumber,
    sum(quantityordered*priceeach) as importe_pedidos
    from classicmodels.orderdetails detalle
    group by 1
) as importe
on pedidos.orderNumber=importe.orderNumber
group by 1,2

)

    select
    base.*,
    LAG(base.importe_pedidos,1) over(order by base.anno, base.mes) as importes_anterior,
    100*((base.importe_pedidos/LAG(base.importe_pedidos,1) over(order by
base.anno,base.mes))-1) as porc_var_mes

from ventas_mes base -- puedo poner en vez del mes anterior, el año anterior, poniendo 12,
para ver la mejora respecto al mismo mes del año anterior (mas sentido)
-----

/*AVERIGUAR QUIEN ES EL EMPLEADO DEL MES PARA TODOS LOS MESES DE 2004*/

with ventas_cliente as (

Select
pedidos.customerNumber,
month(pedidos.orderDate) as mes,

```

```

sum(detalle.importe_pedido) as importe_gastado,
count(*) as ca_pedida
from classicmodels.orders pedidos

left join (
    select detalle.orderNumber,
    sum(quantityordered*priceeach) as importe_pedido
    from classicmodels.orderdetails detalle
    group by 1
    order by 2 desc
) as detalle

on pedidos.orderNumber=detalle.orderNumber
where year(pedidos.orderdate)=2004 -- me cuesta entender que puedo poner aqui un where, al
final del todo
group by 1,2
) -- defino esta consulta con un with, porque ahora voy a definir otra :)

,ventas_empleado as (          -- esto es otra consulta

    select -- esta es la que manda, por eso pongo ahora lo que quiero que salga en la
tabla, siendo casi todo lo mismo que la de arriba
    clientes.salesrepemployeenumber as numero_empleado, -- lo unico que cambio con
esto es el numero de cliente por el de empleado
    ventas_cliente.mes,
    sum(importe_gastado) as importe_gastado,
    sum(ca_pedida) as ca_pedidos
    from ventas_cliente

    left join classicmodels.customers clientes
    on ventas_cliente.customernumber=clientes.customerNumber

    group by 1,2
)

select * from -- esta es la funcion ventana

(
    select -- voy a elegir un rank en vez de row number por si acaso hay empate entre
dos empleados
    base.*,
    RANK () over ( partition by base.mes
                    order by base.importe_gastado desc) as ranking
    from ventas_empleado base
) tabla

where ranking=1 -- se podria cruzar con la tabla de empleados para poner el nombre mejor que
el numero pero...

-----
-----

/* OBTENER EL PROMEDIO DE VENTAS TOTALES PARA CADA MES EN LOS ÚLTIMOS 12 MESES Y SU DESVIACIÓN
ESTÁNDAR*/

with ventas_totales as (

select
year(orderdate) as año,
month(orderdate) as mes,
sum(detalle.importes_vendidos) as importes
from classicmodels.orders pedidos

```

```
left join (
    select
        detalle.ordernumber,
        sum(quantityordered*priceeach) as importes_vendidos
    from classicmodels.orderdetails detalle
    group by 1
) detalle
on pedidos.orderNumber=detalle.orderNumber
group by 1,2

)

select
base.*,
avg(base.importes) over ( partition by base.año
                        order by base.mes
                        rows between 11 preceding and current row) as promedio_anual,
STDDEV(base.importes) over ( partition by base.año
                        order by base.mes
                        rows between 11 preceding and current ROW) as desviación_anual
from ventas_totales base
```


/* OBTENER EL NÚMERO DE PEDIDOS Y EL TOTAL DE VENTAS PARA CADA CLIENTE Y SU RANKING RESPECTO A OTROS CLIENTES POR NÚMERO DE PEDIDOS (Y SI EMPATA POR IMPORTE) PARA 2004*/

```
with ventas_clientes as (

select
pedidos.customernumber,
sum(detalle.importe_vendido) as importe,
count(*) as ca_pedidos
from classicmodels.orders pedidos
left join (
    select
        detalle.ordernumber,
        sum(detalle.quantityordered*detalle.priceeach) as importe_vendido
    from classicmodels.orderdetails detalle
    group by 1
) detalle
on pedidos.orderNumber=detalle.orderNumber
where year(pedidos.orderdate)=2004
group by 1
order by 2 desc

)
```

```
select
base.*,
RANK () over (order by ca_pedidos desc, importe desc) as ranking -- como algunos empatan por
ca_pedidos, ordeno tambien por importe
from ventas_clientes base
```

