

```
/*EJERCICIOS SOBRE CONSULTAS RECURSIVAS*/
```

```
/*CREA UN CALENDARIO DESDE 2002 A 2025. GUARDALO EN UNA TABLA LLAMADA  
SANDBOX.TUSINICIALES_CALENDARIO*/
```

```
create table sandbox.calendar2 as (  
  
with recursive cte_calendar2 as (  
    select date('2003-01-01') as calendar_date -- fecha de inicio  
    union all  
    select date_add(calendar_date, interval 1 day) as calendar_date from cte_calendar2  
    where year(date_add(calendar_date, interval 1 day)) <= 2025 -- condicion de fin, fecha  
final  
)
```

```
select  
calendar_date as fecha,  
year(calendar_date) as fx_anno,  
month(calendar_date) as fx_mes,  
day(calendar_date) as fx_day,  
date_format(calendar_date, '%Y%m') as fx_anno_mes,  
date_format(calendar_date, '%x-%v') as semana -- formato  
from cte_calendar2  
  
)
```

```
select * from sandbox.calendar2
```

```
-----  
-----  
  
/*GENERA LA JERARQUÍA DE EMPLEADOS DE CLASSICMODELS*/
```

```
with recursive jerarquia_empleados as (  
    select employeenumber,  
    reportsto as managernumber,  
    cast(null as char(50)) as managertitle,  
    cast(null as char(50)) as managerofficecode,  
    officecode,  
    jobTitle,  
    1 nivel  
  
    from classicmodels.employees  
    where reportsto is null  
  
    union all  
  
        select e.employeenumber,  
        e.reportsto,  
        ep.jobTitle,  
        ep.officecode,  
        e.officecode,  
        e.jobTitle,  
        nivel+1  
  
        from employees e  
        inner join jerarquia_empleados ep  
        on ep.employeenumber = e.reportsto /* Para cada empleado busca si hay alguno del cual es  
su superior*/  
/*No hace falta condicion de parada porque parará cuando un empleado no tenga colaboradores a  
su cargo*/  
)  
select empleados.employeenumber,
```

```
empleados.nivel,
empleados.jobTitle,
oficinas.city,
empleados.managernumber,
empleados.managerTitle,
oficinas_mng.city as managercity
from jerarquia_empleados empleados
inner join offices oficinas
on empleados.officeCode=oficinas.officeCode
left join offices oficinas_mng
on empleados.managerofficeCode=oficinas_mng.officeCode
order by empleados.nivel, oficinas.city
```

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

show indexes from classicmodels.orders

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

```
/*EJERCICIOS FINALES DE REPASO*/
```

```
/*REALIZA UN ANÁLISIS SOBRE LA EVOLUCIÓN DE LAS VENTAS, CON DETALLE DE COMERCIAL, PAÍS Y
CIUDAD DEL MISMO, GERENTE DE VENTAS Y PAÍS Y CIUDAD DEL CLIENTE.
REVISANDO EL RENDIMIENTO DE LOS GERENTES Y BUSCANDO SI HAY ALGUNA CIUDAD DONDE DEBEMOS ABRIR
OFICINA*/
```

```
SELECT
month(pedidos.orderDate) as mes,
year(pedidos.orderDate) as año,
DATE_FORMAT(pedidos.orderDate,'%m-%Y') as mes_año_pedido,
clientes.city as ciudad_cliente,
clientes.country as pais_cliente,
pedidos.customerNumber,
pedidos.orderNumber,
detalle.importe_ventas,
count(*) as pedidos,
concat(empleados.firstname, ' ', empleados.lastname) as nombre_empleado,
empleados.employeeNumber,
oficinas.city as ciudad_oficina,
oficinas.country as pais_oficina,
empleados.jobtitle as cargo_superior
FROM classicmodels.orders pedidos

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

left join classicmodels.customers clientes
on clientes.customerNumber=pedidos.customerNumber

left join classicmodels.employees empleados
on empleados.employeeNumber=clientes.salesrepemployeeNumber

left join classicmodels.offices oficinas
on oficinas.officeCode=empleados.officecode
```

group by 7

 /*ANALIZA LOS PRODUCTOS PARA VER SU EVOLUCIÓN POR LÍNEA DE PRODUCTO Y REVISANDO QUÉ PRODUCTOS FUNCIONAN MEJOR Y PEOR, ASÍ CON CUÁL OBTENEMOS MÁS BENEFICIO*/

-- MI CONSULTA HECHA POR MI :)

```
select
pedidos.orderdate as fecha_pedido,
DATE_FORMAT(pedidos.orderDate,'%Y-%m') as fx_anno_mes_pedido,
clientes.city as ciudad_cliente,
clientes.country as pais_cliente,
empleados.employeeNumber as numero_empleado,
CONCAT(empleados.firstname, ' ', empleados.lastname) as nombre_empleado,
oficinas.city as ciudad_empleado,
oficinas.country as pais_empleado,
superior.jobTitle as cargo_superior,
productos.productCode,
productos.productName,
productos.productLine,
sum(detalle.quantityOrdered) as unidades,
sum(detalle.quantityOrdered*detalle.priceEach) as importe_venta,
sum(productos.buyPrice*detalle.quantityOrdered) as importe_compra,
sum(detalle.quantityOrdered*detalle.priceEach) -
sum(productos.buyPrice*detalle.quantityOrdered) as margen
from classicmodels.products productos

    left join classicmodels.orderdetails detalle
    on productos.productCode=detalle.productCode

    left join classicmodels.orders pedidos
    on pedidos.orderNumber=detalle.ordernumber

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

    left join classicmodels.employees empleados
    on empleados.employeeNumber=clientes.salesrepemployeenumber

    left join classicmodels.offices oficinas
    on oficinas.officeCode=empleados.officecode

    left join classicmodels.employees superior
    on empleados.reportsTo=superior.employeeNumber
```

group by 1,2,3,4,5,6,7,8,9,10,11,12

-- MISMA CONSULTA PERO HECHA POR EL PROFESOR, NO SE PORQUE A MI NO ME SALE POR ORDEN LA FECHA, ADEMÁS QUE ME SALE UNA FILA MÁS NULA

```
SELECT
base.orderDate as fx_pedido,
DATE_FORMAT(base.orderDate,'%Y-%m') as fx_anno_mes_pedido,
clientes.city as ciudad_cliente,
clientes.country as pais_cliente,
empleados.employeeNumber,
concat(empleados.Firstname, ' ', empleados.LastName) as Nombre_empleado,
oficinas.city as ciudad_empleado,
```

```
oficinas.country as pais_empleado,
superior.jobTitle as cargo_superior,
productos.productcode as codigo_producto,
productos.productName as nombre_producto,
productos.productLine as linea_producto,
sum(detalle.quantityOrdered) as unidades,
sum(detalle.quantityOrdered*detalle.priceEach) as importe_venta,
sum(detalle.quantityOrdered*productos.buyprice) as importe_compra,
sum(detalle.quantityOrdered*detalle.priceEach) -
sum(detalle.quantityOrdered*productos.buyprice) as margen

from classicmodels.orders base

    left join classicmodels.orderdetails detalle
    on base.orderNumber=detalle.orderNumber

    left join classicmodels.products productos
    on detalle.productCode=productos.productCode

    left join classicmodels.customers clientes
    on base.customerNumber=clientes.customerNumber

    left join classicmodels.employees empleados
    on clientes.salesRepEmployeeNumber=empleados.employeeNumber

    left join classicmodels.offices oficinas
    on empleados.officeCode=oficinas.officeCode

    left join classicmodels.employees superior
    on empleados.reportsTo=superior.employeeNumber

group by 1,2,3,4,5,6,7,8,9,10,11,12
```
