<Conexión Clase> Clase 6.2 lunes, 23 de octubre de 2023 19:32

```
/*CRUCES ENTRE TABLAS (JOINS)*/



/*INNER JOIN*/ /*CONCLUSION: ME SACA 89 CLIENTES QUE SON LOS DE 2004*/ /*SIRVE PARA FILTRAR
REGISTROS Y ENRIQUECERLOS*/
SELECT
clientes.*, -- VALORES TABLA IZQUIERDA
ca_pedidos  -- VALORES TABLA DERECHA
FROM classicmodels.customers clientes -- SIEMPRE TABLA IZQUIERDA(BASE)
inner join (
    select -- BASICAMENTE A LA TABLA DE CLIENTES LE AÑADO LA CONSULTA DE NUMERO DE PEDIDOS EN
2004
    customerNumber,
    COUNT(customerNumber) AS ca_pedidos
    from classicmodels.orders o
    where year(orderdate)=2004
    group by 1
) pedidos -- ALIAS DE ESTA CONSULTA
on clientes.customerNumber=pedidos.customernumber -- EL ON INDICA LAS CONDICIONES DEL CRUCE
-- EN ESTE CASO CRUZAMOS POR EL NUMERO DE CLIENTE QUE SON DATOS QUE COINCIDEN EN AMBAS TABLAS



/*LEFT JOIN*/ /*CONCLUSION: ME SACA TODOS LOS CLIENTES*/ /*SE USA PARA ENRIQUECER OTRA TABLA,
NO PERDEMOS DATOS DE LA TABLA INICIAL*/
SELECT
clientes.*,
COALESCE(ca_pedidos, 0) as ca_pedidos_2004
FROM classicmodels.customers clientes -- TABLA IZQUIERDA(BASE)
left join (
    select
    customerNumber,
    COUNT(customerNumber) AS ca_pedidos
    from classicmodels.orders o
    where year(orderdate)=2004
    group by 1
) pedidos
on clientes.customerNumber=pedidos.customernumber



/*RIGHT JOIN*/ /*SE USA LA ANTERIOR, YA QUE LA ESTRUCTURA ES IZQUIERDA PRINCIPAL Y DERECHA
AÑADIDA*/
SELECT
clientes.*,
COALESCE(ca_pedidos, 0) as ca_pedidos
from(
    select
    customernumber,
    count(customernumber) as ca_pedidos
    from orders o
    where year(orderdate)=2004
    group by 1
) pedidos

right join classicmodels.customers clientes
on clientes.customernumber=pedidos.customernumber -- MEJOR NO HACERLE CASO, USAR LEFT JOIN
PORQUE ESTA ES DEMASIADO LIO
```

```sql
/*FULL JOIN*/ /*POR EJEMPLO PARA QUEDARME CON TODOS LOS CLIENTES DE 2004 Y TODOS LOS CLIENTES
DE 2005, TENGAN O NO PEDIDOS*/
/*HAY QUE HACER UNA UNION DE LEFT + RIGHT PORQUE FULL JOIN NO ESTA EN SQL*/
select -- HAY PEDIDOS A 0 EN 2004
pedidos_2005.customernumber,
coalesce(pedidos_2004.ca_pedidos, 0) as pedidos_2004,
pedidos_2005.ca_pedidos as pedidos_2005

from
(
    select
    customernumber,
    count(customernumber) as ca_pedidos
    from orders o
    where year(orderdate)=2005
    group by 1
) pedidos_2005

left join
(
    select
    customernumber,
    count(customernumber) as ca_pedidos
    from orders o
    where year(orderdate)=2004
    group by 1
) pedidos_2004

on pedidos_2004.customernumber=pedidos_2005.customernumber

union

select
pedidos_2004.customernumber, -- HAY PEDIDOS A 0 EN 2005
pedidos_2004.ca_pedidos as pedidos_2004,
coalesce(pedidos_2005.ca_pedidos, 0) as pedidos_2005

from
(
    select
    customernumber,
    count(customernumber) as ca_pedidos
    from orders o
    where year(orderdate)=2005
    group by 1
) pedidos_2005

right join
(
    select
    customernumber,
    count(customernumber) as ca_pedidos
    from orders o
    where year(orderdate)=2004
    group by 1
) pedidos_2004

on pedidos_2004.customernumber=pedidos_2005.customernumber



/*CROSS JOIN*/ /*CRUZA TODOS CON TODOS*/ /*NO SE USA CASI NUNCA*/
```

```sql
SELECT
X.city as equipo_local, -- QUIERO QUE X1,X2,X3 SE CRUCE CON Y1, QUE X1,X2,X3 SE CRUCE CON
Y2... ASÍ SUCESIVAMENTE
Y.city as equipo_visitante
from classicmodels.offices X
cross join offices Y
where X.city <> Y.city



/*FUNCION MINUS*/
SELECT
clientes.customernumber,
clientes.customername
from customers clientes
left join

(
    SELECT
    a.customernumber
    from orders a
    where year(a.orderdate)=2004
    group by 1
) pedidos

on clientes.customernumber=pedidos.customernumber

where pedidos.customernumber is null -- PARA QUE EL NOMBRE DE CLIENTE DE LA SUBCONSULTA QUE
ESTE REPETIDO CON LA TABLA PRINCIPAL LO ELIMINE
-- NOS QUEDAMOS CON LOS CLIENTES QUE NO TIENEN PEDIDOS EN 2004



/*FUNCION INTERSECT*/ /*ME QUEDO SOLO  CON LOS REGISTROS COMUNES*/
SELECT
clientes.customernumber,
clientes.customername
from customers clientes

inner join

(
    SELECT
    a.customernumber
    from orders a
    where year(a.orderdate)=2004
    group by 1
) pedidos
on clientes.customernumber=pedidos.customernumber



/*EJERCICIOS JOINs*/



/*LISTA LOS PEDIDOS CON EL NOMBRE DEL CLIENTE*/
-- --> HACER UN SELECT ASTERISCO CON LOS CRUCES ES BASTANTE INTERESANTE PARA VERLO MAS CLARO
<--
SELECT
*
FROM classicmodels.orders pedidos -- LA QUE VOY A ENROQUECER CON OTRA TABLA
```

```
left join classicmodels.customers clientes -- AMPLIO POR LA DERECHA
on pedidos.customerNumber=clientes.customerNumber -- PRIMARY KEY

-- SOLUCION FINAL MAS PRECISA
SELECT
clientes.customerNumber,
clientes.customerName,
pedidos.orderNumber,
pedidos.orderdate
FROM classicmodels.orders pedidos
left join classicmodels.customers clientes
on pedidos.customerNumber=clientes.customerNumber

-----------------------------------------------------------------------------------------------
-----------------

/*CALCULA CUÁNTOS EMPLEADOS HAY POR CADA OFICINA, POR CIUDAD Y PAÍS*/
select
oficinas.country,
oficinas.city,
count(*) as ca_empleados
FROM classicmodels.employees empleados
left join classicmodels.offices oficinas
on empleados.officeCode=oficinas.officeCode
group by 1,2
order by 3 desc

-----------------------------------------------------------------------------------------------
-----------------

/*MUESTRA LOS CLIENTES CON UN RANGO DE PEDIDOS (0, 1 A 4, 5 A 10, MÁS DE 10)*/
SELECT
clientes.customernumber,
clientes.customername,
(CASE
    when tabla_pedidos.numero_pedidos is null then '1-) sin pedidos'
    when tabla_pedidos.numero_pedidos between 1 and 4 then '2-) 1 a 4'
    when tabla_pedidos.numero_pedidos between 5 and 10 then '3-) 5 a 10'
    when tabla_pedidos.numero_pedidos >10 then '4-) mas de 10'
    else 'corregir'
END) as rango_pedidos

from classicmodels.customers clientes

left join (
    select
    pedidos.customernumber,
    count(*) as numero_pedidos
    from classicmodels.orders pedidos
    group by 1
) tabla_pedidos

on clientes.customernumber=tabla_pedidos.customernumber
order by 3

-----------------------------------------------------------------------------------------------
-------------------------------------------------

/*CALCULA PARA TODOS LOS CLIENTES, EL NÚMERO DE PEDIDOS, IMPORTE DE LOS MISMOS Y PAGOS PARA EL
AÑO 2004. ANALIZA LA RELACIÓN ENTRE PAGOS Y PEDIDOS*/
select
clientes.customerName,
```

```
clientes.customerNumber,
coalesce(pedidos.ca_pedidos,0) as ca_pedidos,
coalesce(pedidos.importes_pedido,0) as importes_pedido,
coalesce(pagos.importes_pago,0) as importes_pagos
from classicmodels.customers clientes

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

    left join (
        select
        o.orderNumber,
        sum(quantityOrdered*priceEach) as importe_pedido
        from classicmodels.orderdetails o
        group by 1
        order by 2 desc) importes
    on pedidos.orderNumber=importes.orderNumber

    where year(pedidos.orderDate)=2004
    -- and pedidos.status<>'Cancelled'
    group by 1
    ) pedidos
on pedidos.customerNumber=clientes.customerNumber

left join (
    select
    customerNumber, sum(amount) as importes_pago
    from classicmodels.payments p
    where YEAR(paymentDate)=2004
    group by 1
) pagos
on pagos.customerNumber=clientes.customerNumber;

-- COMPROBACIONES DE UN CLIENTE QUE NO COINCIDE
select * from classicmodels.orders o
where customerNumber =141
and year(orderDate)=2004

select * from classicmodels.orderdetails o
where orderLineNumber =10262

-----------------------------------------------------------------------------------------------
-----------------

/*COMPARA CUÁNTOS CLIENTES TIENE ASIGNADA CADA OFICINA Y CALCULA EL RATIO DE CLIENTES POR
EMPLEADO*/
SELECT
oficinas.country,
oficinas.city,
count(distinct empleados.employeenumber) as numero_empleados, -- SI NO PONGO EL DISTINCT
CONTARIA CLIENTES, EL ASTERICO CUENTA FILAS
count(*) as numero_clientes,
count(*)/count(distinct empleados.employeenumber) as ratio
from classicmodels.customers clientes

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

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

group by 1,2

-------------------------------------------------------------------------------------------------
-----------------

/*PARA CADA PEDIDO CALCULA LOS MESES DESDE EL PRIMER PEDIDO DEL CLIENTE*/
SELECT
pedidos.ordernumber,
pedidos.orderdate as fecha_pedido,
primer_pedido.fecha_primerpedido,
floor(DATEDIFF(pedidos.orderdate, primer_pedido.fecha_primerpedido)/30) as meses
from classicmodels.orders pedidos

left join (
    select
    a.customernumber,
    min(a.orderdate) as fecha_primerpedido
    from classicmodels.orders a
    group by 1
) primer_pedido
on pedidos.customernumber=primer_pedido.customernumber

order by 4 desc
-------------------------------------------------------------------------------------------------
-----------------
```