

Alumnos: Cesar Alfredo Astorga Ochoa y José Alfredo García Aguilar. Clase 8-9AM

Realizar los siguientes ejercicios y documentarlos de la siguiente manera:

2.- PROCEDIMIENTO ALMACENADO QUE RECIBA

Código:

```
--create proc sp_JefeEmpleados @jefeID int as
--begin
select J.FirstName + ' ' + J.LastName as Jefe, E.FirstName + ' ' + E.LastName as Empleado from Employees J
inner join Employees E on J.EmployeeID = E.ReportsTo
where J.EmployeeID = @jefeID
--end
go
```

Ejecución:

```
--exec sp_JefeEmpleados
--@jefeID = 2
go
```

Resultados:

	Jefe	Empleado
1	Andrew Fuller	Nancy Davolio
2	Andrew Fuller	Janet Leverling
3	Andrew Fuller	Margaret Peacock
4	Andrew Fuller	Steven Buchanan
5	Andrew Fuller	Laura Callahan

Activa
Ir a Con

Funciones

1.- **FUNCION ESCALAR** QUE RECIBA DOS CLAVES DE CLIENTES Y UN AÑO, Y REGRESE EL NOMBRE DEL CLIENTE QUE MAS HA VENDIDO PIEZAS DE LOS DOS EN ESE AÑO Y EL TOTAL DE PIEZAS VENDIDAS. POR EJEMPLO, DEBE REGRESAR: EL CLIENTE JUAN PEREZ VENDIO 450 PIEZAS.

CODIGO:

```
create function fn_clientesVentas(@cl1 nchar(5), @cl2 nchar(5), @año int)
returns varchar(70)
as
begin
    declare @t1 int, @t2 int, @nombre varchar(70)

    select @t1 = count(od.quantity)
    from orders o
    inner join [order details] od on o.orderid = od.orderid
    where o.customerid = @cl1 and year(o.orderdate) = @año

    select @t2 = count(od.quantity)
    from orders o
    inner join [order details] od on o.orderid = od.orderid
    where o.customerid = @cl2 and year(o.orderdate) = @año

    if @t1 > @t2
        select @nombre = 'El cliente ' + contactname + ' vendio ' + cast(@t1 as nvarchar(10)) + ' piezas' from customers where customerid = @cl1
    else
        select @nombre = 'El cliente ' + contactname + ' vendio ' + cast(@t2 as nvarchar(10)) + ' piezas' from customers where customerid = @cl2

    return @nombre
end
```

EJECUCIÓN:

```
select dbo.fn_clientesVentas('ALFKI', 'ANATR', 1996)
```

RESULTADOS:

	(No column name)
1	El cliente Ana Trujillo vendio 2 piezas

2.- **FUNCION ESCALAR** QUE RECIBA LA CLAVE DEL EMPLEADO Y REGRESE UNA CADENA CON LOS NOMBRE DE LOS TERRITORIOS QUE ATIENDE.

CODIGO:

```
create function fn_empleadoTerritorio(@emp int)
returns varchar(1000)
as
begin
    declare @tr varchar(1000), @min int

    select @min = min(territoryid) from employeeterritories where employeeid = @emp
    select @tr = ''

    while @min is not null
    begin
        select @tr = territorydescription+', '+@tr from territories where territoryid = @min
        select @min = min(territoryid) from employeeterritories where employeeid = @emp and territoryid > @min
    end
    return @tr
end
```

EJECUCIÓN:

```
select dbo.fn_empleadoTerritorio(2)
```

RESULTADOS:

1	Louisville	, Braintree	, Cambridge	, Boston	, Georgetown	, Bedford	...
---	------------	-------------	-------------	----------	--------------	-----------	-----

3.- **FUNCION DE TABLA EN LINEA** QUE RECIBA LA CLAVE DE UN PRODUCTO Y REGRESE UNA TABLA CON EL NOMBRE DE TODOS LOS CLIENTES QUE HAN COMPRADO ESE PRODUCTO, EL TOTAL DE PRODUCTOS COMPRADO Y EL TOTAL DE ORDENES EN LAS QUE SE HA VENDIDO.

NOMBRE DEL CLIENTE	TOTAL PRODUCTOS	TOTAL DE ORDENES

CODIGO:

```
create function dbo.clientesCompras(@prod int)
returns table
as return(
    select 'Nombre del cliente'=c.contactname, 'Total productos'= sum(od.quantity), 'Total de ordenes'=count(od.orderid)
    from customers c
    inner join orders o on c.customerid = c.customerid
    inner join [order details] od on o.orderid = od.orderid
    where od.productid = @prod
    group by c.contactname, o.orderid
)
```

EJECUCIÓN:

```
select*from dbo.clientesCompras(1)
```

RESULTADOS:

	Nombre del cliente	Total productos	Total de ordenes
88	Yoshi Latimer	45	1
89	Yoshi Tannamuri	45	1
90	Yvonne Moncada	45	1
91	Zbyszek Piestrzeni...	45	1
92	Alejandra Camino	18	1
93	Alexander Feuer	18	1
94	Ana Trujillo	18	1
95	Anabela Domingues	18	1
96	Andrés Fonseca	18	1

4.- **FUNCION DE TABLA EN LINEA** QUE RECIBA LA CLAVE DEL EMPLEADO Y AÑO, REGRESE EN UNA CONSULTA EL NOMBRE DEL PRODUCTO Y TOTAL DE PRODUCTOS VENDIDOS POR ESE EMPLEADO Y ESE AÑO.

NOMBRE DEL PRODUCTO	TOTAL PIEZAS

CODIGO:

```
create function dbo.empleadoVentas(@emp int, @año int)
returns table
as
return(
select 'Nombre' = p.productname, 'Total' = count(p.productid)
from products p
inner join [order details] od on p.productid = od.productid
inner join orders o on od.orderid = o.orderid
where o.employeeid = @emp and year(o.orderdate) = @año
group by p.productname
)
```

EJECUCIÓN:

```
select*from dbo.empleadoVentas(5, 1996)
```

RESULTADOS:

	Nombre	Total
1	Camembert Pierrot	2
2	Chartreuse verte	1
3	Côte de Blaye	1
4	Flotemysost	3
5	Geitost	1
6	Gorgonzola Telino	1
7	Guaraná Fantást...	2
8	Inlagd Sill	1
9	Longlife Tofu	1
10	Manjimup Dried ...	1

5.- UTILIZANDO LA FUNCION ANTERIOR MOSTRAR UNA CONSULTA SIGUIENTE:

NOMBRE DEL PRODUCTO	TOTAL PIEZAS 96	TOTAL PIEZAS 97	TOTAL PIEZAS 98

CODIGO:

```
select a.nombre, 'Total piezas 96' = a.total, 'Total piezas 97' = b.total, 'Total piezas 98' = c.total
from dbo.empleadoVentas(5, 1996) a
inner join dbo.empleadoVentas(5, 1997) b on a.nombre = b.nombre
inner join dbo.empleadoVentas(5, 1998) c on a.nombre = c.nombre
```

EJECUCIÓN:

```
select a.nombre, 'Total piezas 96' = a.total, 'Total piezas 97' = b.total, 'Total piezas 98' = c.total
from dbo.empleadoVentas(5, 1996) a
inner join dbo.empleadoVentas(5, 1997) b on a.nombre = b.nombre
inner join dbo.empleadoVentas(5, 1998) c on a.nombre = c.nombre
```

RESULTADOS:

	nombre	Total piezas 96	Total piezas 97	Total piezas 98
1	Chartreuse verte	1	2	1
2	Gorgonzola Telino	1	3	2
3	Guaraná Fantástica	2	1	2
4	Manjimup Dried A...	1	1	1
5	Mozzarella di Giov...	4	3	1
6	Pâté chinois	1	1	1
7	Pavlova	1	1	1

5.- **FUNCION DE TABLA DE MULTISENTENCIA** (NO LLEVA PARAMETROS DE ENTRADA) QUE REGRESE UNA TABLA CON EL NOMBRE DE LA CATEGORIA Y LOS NOMBRES DE LOS PRODUCTOS QUE PERTENECEN A LA CATEGORIA Y EL TOTAL DE PIEZAS QUE SE HAN VENDIDO DE ESA CATEGORIA

CATEGORIA	PRODUCTOS	PIEZAS VENDIDAS
LACTEOS	QUESO, LECHE, SALCHICHA	
LATERIA	FRIJOLES, CHAMPIÑONES, SALSA	

CODIGO:

```

alter function dbo.fn_productosCategorias()
returns @categoria table(categoria varchar(20), productos varchar(500), piezas int)
as
begin
    declare @cat int, @catname varchar(20), @prodid int, @prod varchar(500), @piezas int

    select @cat = min(categoryid) from categories
    while @cat is not null
    begin

        select @catname = categoryname from categories where categoryid = @cat

        select @prodid = min(productid) from products where categoryid = @cat
        select @piezas = count(quantity) from [order details] where productid = @prodid
        select @prod = ''

        while @prodid is not null
        begin
            select @prod = productname+', '+@prod from products where categoryid = @cat
            select @prodid = min(productid) from products where categoryid = @cat and productid > @prodid
        end
        select @cat = min(categoryid) from categories where categoryid > @cat
        insert @categoria values(@catname, @prod, @piezas)
    end
return
end

```

EJECUCIÓN:

```
select*from dbo.fn_productosCategorias()
```

RESULTADOS:

	categoria	productos	piezas
1	Beverages	Lakkalikööri, Rhönbräu Klosterbier, Outback Lager, Laughing Lumberjack Lager, Ipoh C...	38
2	Condime...	Original Frankfurter grüne Soße, Louisiana Hot Spiced Okra, Louisiana Fiery Hot Pepper...	12
3	Confecti...	Scottish Longbreads, Tarte au sucre, Valkoinen suklaa, Maxilaku, Chocolate, Zaanse k...	43
4	Dairy Pro...	Mozzarella di Giovanni, Flotemysost, Gudbrandsdalsost, Camembert Pierrot, Raclette Co...	38
5	Grains/C...	Wimmers gute Semmelknödel, Ravioli Angelo, Gnocchi di nonna Alice, Filo Mix, Singapo...	14
6	Meat/Po...	Pâté chinois, Tourtière, Perth Pasties, Thüringer Rostbratwurst, Alice Mutton, Mishi Kob...	5
7	Produce	Longlife Tofu, Manjimup Dried Apples, Rössle Sauerkraut, Tofu, Uncle Bob's Organic Dr...	29
8	Seafood	Röd Kaviar, Escargots de Bourgogne, Spegesild, Rogede sild, Jack's New England Cla...	33

6.- FUNCION DE TABLA DE MULTISENTENCIA QUE RECIBA UN AÑO COMO PARAMETRO DE ENTRADA, QUE REGRESE UNA TABLA CON DOS COLUMNAS: DIA DE LA SEMANA, FOLIOS QUE SE VENDIERON ESE DÍA DE SEMANA. NOTA, DEBE MOSTRAR TODOS LOS DIAS DE LA SEMANA AUNQUE NO SE HAYAN REALIZADO ORDENES.

DIA SEMANA	FOLIOS
DOMINGO	1 ,2,3,4,5
LUNES	6,7,8,9

CODIGO:

EJECUCIÓN:

RESULTADOS:

CÓDIGO:

```
create function dbo.fn_diaFolios()  
returns @folios table(diasemana datetime(20), numfolio int)  
as  
begin  
    declare @diasemana datetime, @numfolio int  
  
    select @numfolio = min(orderid) from orders  
    while @numfolio is not null  
    begin  
  
        select @numfolio = orderid from orders where orderid = @numfolio  
        select @diasemana = day(orderdate) from orders where orderid = @numfolio  
    end  
    return  
end
```

EJECUCIÓN:

```
--ejecucion  
select * from dbo.fn_dia_Folios()
```