

# **Instituto Tecnológico de Culiacán**



**Carrera: Ingeniería en Sistemas Computacionales**

**Materia: Taller de Base de Datos**

**Alumnos:**

**Cesar Alfredo Astorga Ochoa**

**José Alfredo García Aguilar**

**Trabajo: U5 Tarea Procedimientos**

**Fecha: 27-Octubre-2021**

**Horario de clase: 08:00 – 09:00 am**

**Profesor: Daniel Esparza Soto**

**1.- AGREGAR A LA TABLA CATEGORIES EL CAMPO TOTALPIEZAS, EL CUAL REPRESENTARÁ EL TOTAL DE PIEZAS VENDIDAS DE CADA CATEGORIA. CREAR UN PROCEDIMIENTO ALMACENADO QUE LLENE DICHO CAMPO.**

Código:

```
alter table categories add total_de_piezas numeric(12,2)
create proc sp_categorias_as
declare @categoria_id varchar(40), @total_de_piezas numeric(12,2), @prod varchar(200)
select @categoria_id = min(categoryid) from categories
while @categoria_id is not null
begin
    select @total_de_piezas = sum(od.quantity)
    from [order details] od
    inner join products prod on prod.productid = od.productid
    where prod.categoryid = @categoria_id

    update categories set total_de_piezas = isnull(@total_de_piezas,0) where categoryid = @categoria_id

    select @categoria_id = min(categoryid) from categories where categoryid > @categoria_id
end
go
select categoryid, total de piezas from categories
```

Ejecución:

```
--ejecucion
exec sp_categorias
go
```

Resultados:

	categoryid	total_de_piezas
1	1	9532.00
2	2	5298.00
3	3	7906.00
4	4	9149.00
5	5	4562.00
6	6	4199.00
7	7	2990.00
8	8	7681.00

## 2.- SP QUE RECIBA LA CLAVE DEL EMPLEADO Y REGRESE POR RETORNO LA EDAD EXACTA DEL EMPLEADO.

Código:

```
alter proc sp_edadempleado(@emp int) as
declare @edad_en_años int, @conta int, @fecha datetime
create table #tabla(emp int, edad_en_años int)

select @fecha = birthdate from employees where employeeid = @emp
select @conta = 0

while @fecha <= getdate()
begin
    select @fecha = dateadd(yy,1,@fecha)

    select @conta = @conta + 1
end
insert #tabla values(@emp, @conta)

select Nombre = e.firstname+' '+e.lastname, Edad = T.edad_en_años
from #tabla T
inner join employees e on e.employeeid = T.emp
```

Ejecución:

```
--Ejecucion
exec sp_edadempleado 2
go
```

Resultados:

	Nombre	Edad
1	Andrew Fuller	70

**3.- PROCEDIMIENTO ALMACENADO QUE RECIBA COMO PARAMETRO UNA FECHA Y REGRESE DOS PARAMETROS: UN PARAMETRO CON EL NOMBRE DE TODOS LOS CLIENTES QUE COMPRARON ESE DIA Y OTRO PARAMETRO CON LA LISTA DE LAS ORDENES REALIZADAS ESE DIA.**

Código:

```
create proc sp_clientesOrdenes(@fecha datetime, @clientes varchar(500) output, @ordenes varchar(500) output) as
declare @clienteid int, @ordenid int

select @clienteid = count(customerid) from orders where orderdate = @fecha
select @clientes = ''
while @clienteid > 1
begin
    select @clientes = c.contactname+', '+@clientes from customers c
    inner join orders o on c.CustomerID = o.CustomerID
    where o.orderdate = @fecha

    select @clienteid = @clienteid - 1
end

select @ordenid = min(orderid) from orders where orderdate = @fecha
select @ordenes = ''
while @ordenid is not null
begin
    select @ordenes = cast(orderid as varchar(20))+', '+@ordenes from orders where orderid = @ordenid

    select @ordenid = min(orderid) from orders where orderdate = @fecha and orderid > @ordenid
end
```

Ejecución:

```
declare @a varchar(500), @b varchar(500)
exec sp_clientesordenes '16-10-1996', @a output, @b output
select @a, @b
```

Resultados

	(No column name)	(No column name)
1	Laurence Lebihan, Carlos González,	10331, 10330,

--4.- PROCEDIMIENTO ALMACENADO QUE REGRESE UNA TABLA CON LA FECHA Y LOS NOMBRES DE LOS CLIENTES QUE COMPRARON ESE DÍA.

--FECHA                      Nombre de jefes

--1-1-2021      JUAN PEREZ, ANA LOPEZ, JOSE LARA

--2-1-2021      JOSE LARA, JUAN PEREZ

```
-- create proc sp_tabla(@compra datetime) as
-- declare @fecha datetime, @nombre_cliente varchar
-- create table #tabla(compra datetime, fecha datetime, nombre_cliente varchar)

-- select @compra = orderdate from orders where orderdate = @compra
-- begin
--     select @compra = dateadd(yy,1,@fecha)
-- end
-- insert #tabla values(@compra, @nombre_cliente)

-- select o.orderdate, t.nombre_cliente
-- from #tabla T
-- inner join orders o on o.orderdate = t.compra
-- --ejecucion
-- exec sp_tabla
-- go
```

5.- SP QUE RECIBA UN AÑO Y REGRESE COMO PARAMETRO DE SALIDA LA CLAVE DEL ARTICULO QUE MAS SE VENDIO ESE AÑO Y CANTIDAD DE PIEZAS VENDIDAS DE ESE PRODUCTO EN ESE AÑO.

Código:

```
create proc sp_añoProducto @año int, @articulo int output , @piezas int output as
declare @masVendido int, @TotalPiezas int, @aux int

select @piezas = 0
select @aux = min(orderid) from orders where year(orderdate) = @año
while @aux is not null
begin
    select @TotalPiezas = quantity from [order details] where orderid = @aux

    if @TotalPiezas > @piezas
    begin
        select @piezas = @totalpiezas
        select @articulo = productid from [order details] where orderid = @aux
    end

    select @aux = min(orderid) from orders where year(orderdate) = @año and orderid > @aux
end
```

Ejecución:

```

declare @b int, @c int
exec sp_añoProducto 1998, @b output, @c output
select @b, @c

```

Resultados:

100 %

Results		Messages
	(No column name)	(No column name)
1	64	130

6.- SP QUE RECIBA LA CLAVE DEL EMPLEADO Y REGRESE COMO PARAMETRO DE SALIDA TODOS LOS NOMBRES DE LOS EMPLEADOS QUE ATIENDEN EL TERRITORIO.

Código:

```

create proc sp_clientesTerritorios @emp int, @empleados varchar(200) output as
declare @territorio varchar(20), @min int, @aux varchar(40)

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

select @empleados = ''

while @min is not null
begin
    select @aux = firstname+' '+lastname from employees where employeeid = @min

    select @empleados = @empleados+', '+@aux

    select @min = min(employeeid) from employeeterritories where territoryid = @territorio and employeeid > @emp
end

```

Ejecución:

```
--ejecucion
declare @b varchar(200)
exec sp_clientesTerritorios 1, @b output
select @b
```

Resultados:

	(No column name)
1	, Nancy Davolio

7.- SP QUE REALICE UN PROCESO DONDE REGRESE LA SIGUIENTE TABLA:

NOMBRE DE EMPLEADO	Nombre de jefes	Jefes Superior

"NOMBRE DE JEFES" SERÁ LA CADENA CON TODOS LOS NOMBRE DE LOS JEFES QUE TIENE EL EMPLEADO. "Jefe Superior" ES EL JEFE QUE SE ENCUENTRA EN LA RAZ DEL ARBOL DE EMPLEADOS.

Código:

```

create proc sp_empleadojefes as
declare @emp int, @jefe int, @empleado varchar(20), @jefes varchar(200), @jefesup varchar(20)
create table #T(empleado varchar(20), jefes varchar(200), jefesup varchar(20))

select @emp = min(employeeid) from employees
while @emp is not null
begin
select @empleado = firstname+' '+lastname from employees where employeeid = @emp
select @jefe = reportsto from employees where employeeid = @emp
select @jefes = firstname+' '+lastname from employees where employeeid = @jefe

while @jefe is not null
begin
select @jefe = reportsto from employees where employeeid = @jefe
select @jefes = @jefes+', '+firstname+' '+lastname from employees where employeeid = @jefe
end

select @jefesup = firstname+' '+lastname from employees where employeeid = @jefe
insert #T values(@empleado, @jefes, @jefesup)
select @emp = min(employeeid) from employees where employeeid > @emp

end
select*from #T

```

Ejecución:

```
exec sp_empleadojefes
```

Resultados:

	empleado	jefes	jefesup
6	Michael Suyama	Steven Buchanan, Andrew Fuller	NULL
7	Robert King	Steven Buchanan, Andrew Fuller	NULL
8	Laura Callahan	Andrew Fuller	NULL
9	Anne Dodsworth	Steven Buchanan, Andrew Fuller	NULL