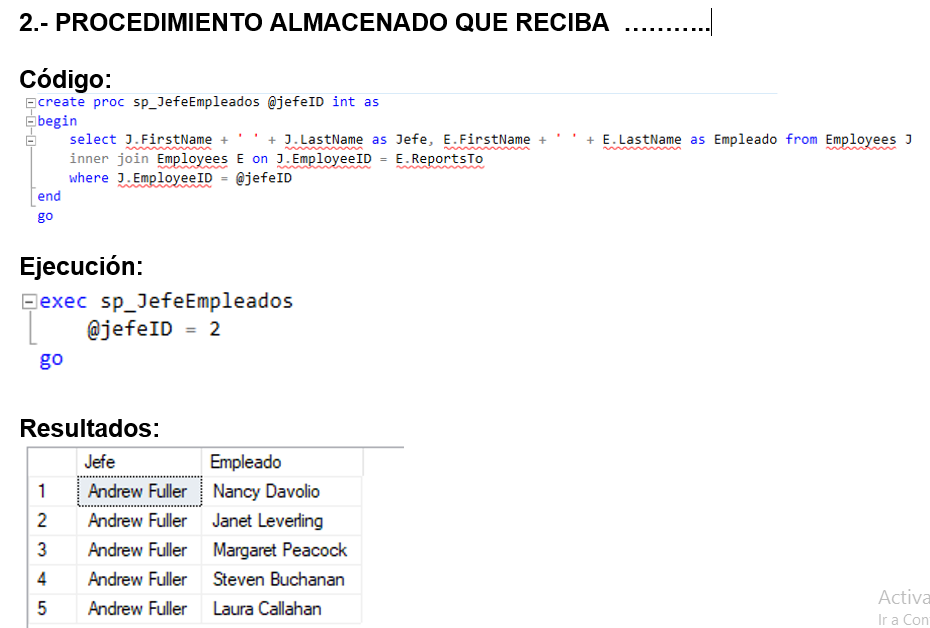
**HECHO POR:**

**GUZMÁN BAJO JORGE HORACIO**

**CASTRO PARRA ALBERTO MANUEL**

**Realizar los siguientes ejercicios y documentarlos de la siguiente manera:**



**Procedimientos almacenados**

1.- AGREGAR A LA TABLA **CUSTOMERS** EL CAMPO **IMPORTETOTAL**, EL CUAL REPRESENTARÁ EL IMPORTE TOTAL DE VENTAS QUE HA TENIDO ESE CLIENTE. CREAR UN PROCEDIMIENTO ALMACENADO QUE LLENE DICHO CAMPO.

Código:

alter table customers add importe numeric (12,2)

create PROC SP\_CLIENTES AS

DECLARE @CTE VARCHAR(50), @IMP NUMERIC(12,2), @PROD VARCHAR(200)

SELECT @CTE = MIN(customerid) FROM Customers

WHILE @CTE IS NOT NULL

BEGIN

select @IMP = sum(d.quantity \* d.unitprice)

from orders o

inner join [order details] d on d.orderid = o.orderid

where o.customerid = @CTE

UPDATE customers SET IMPORTE = isnull(@IMP, 0) WHERE customerid = @CTE

SELECT @CTE = MIN(customerid) FROM customers WHERE customerid>@CTE

END

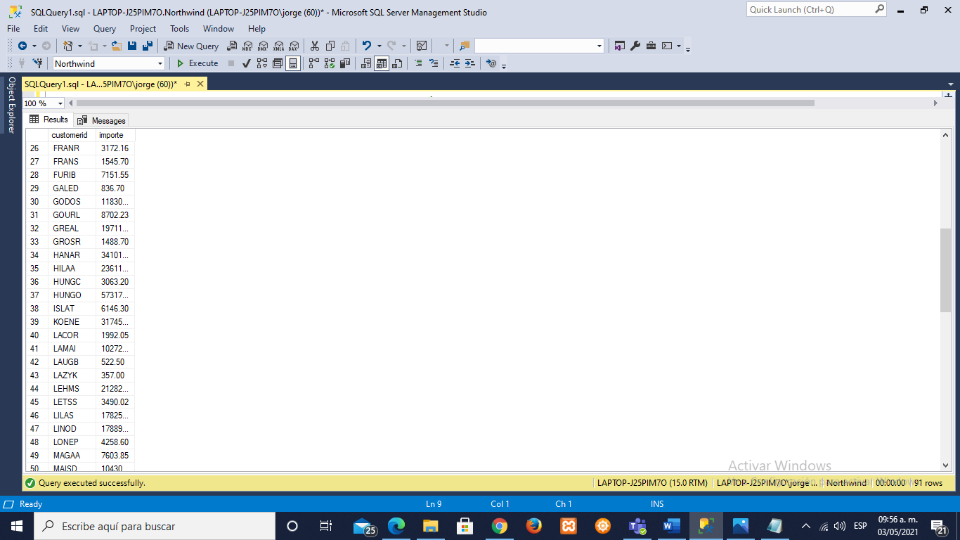
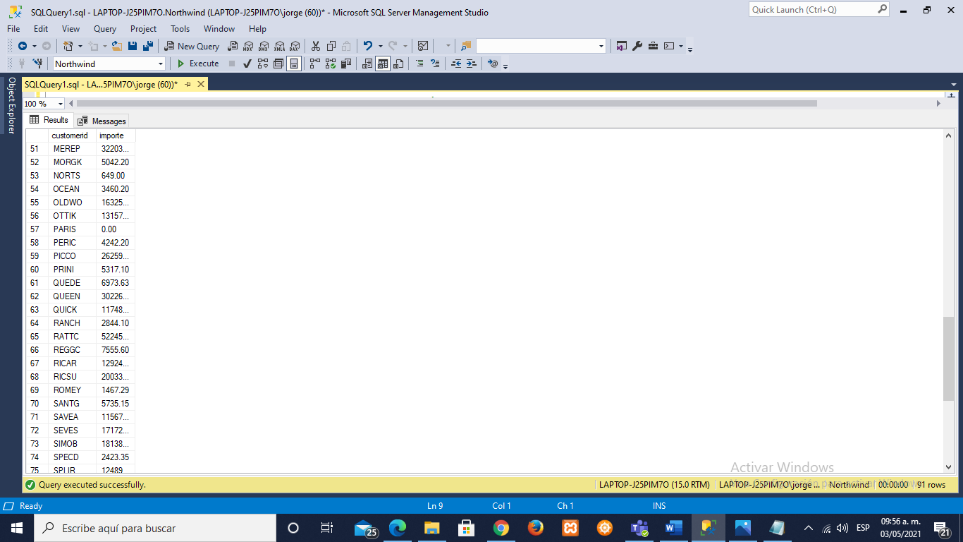
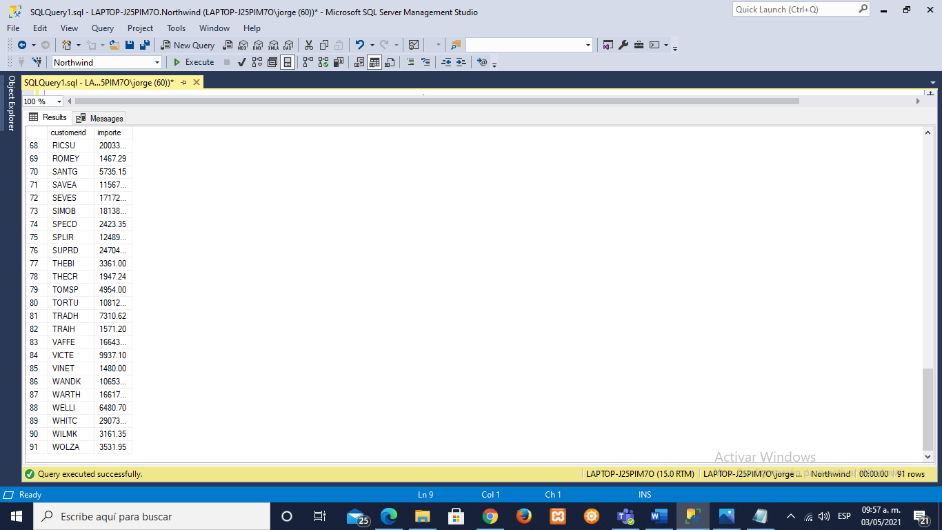
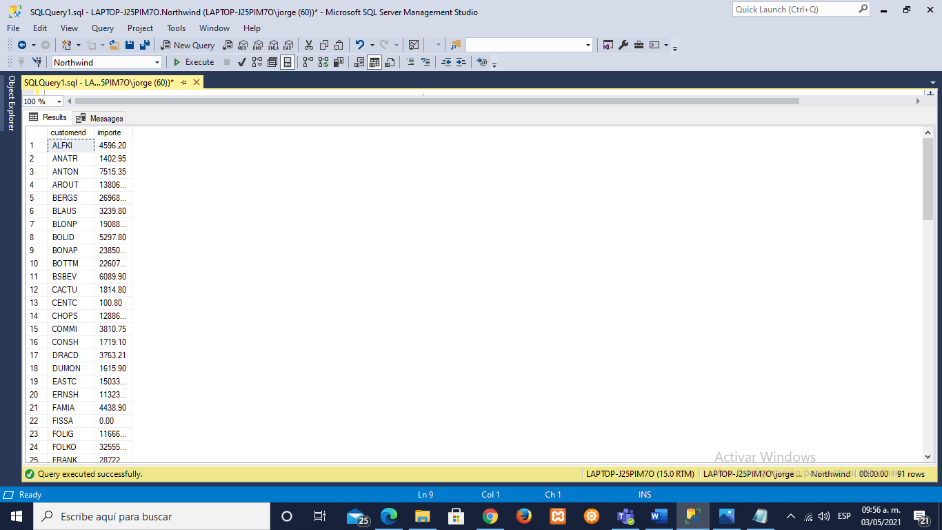
go

select customerid, importe from Customers

EJECUCION:

exec SP\_CLIENTES

RESULTADOS:



2.- **PROCEDIMIENTO ALMACENADO** QUE RECIBA DOS CLAVES DE CLIENTES Y UN AÑO, Y REGRESE **DOS** **PARAMETRO DE SALIDA:** EL NOMBRE DEL CLIENTE QUE MAS HA VENDIDO EN ESE AÑO Y EL IMPORTE TOTAL DE VENTAS.

3.- **PROCEDIMIENTO ALMACENADO** QUE RECIBA LA CLAVE DE JEFE Y REGRESE UNA TABLA CON EL NOMBRE DEL JEFE ENVIADO, EL NOMBRE DE LOS EMPLEADOS A SU CARGO.

|  |  |
| --- | --- |
| JEFE | EMPLEADOS |
| CARLOS PEREZ | ANA LOPEZ, JOSE CASTRO |

Código:

create proc SP\_JefeEmp @cveJefe int as

declare @emp int, @NomJefe varchar(50), @NomEmp varchar(50)

create table #t(Jefe varchar(25), Empleados varchar(25))

select @NomEmp = ''

select @NomJefe = e.firstname + ' ' + e.lastname

from employees e

where e.employeeid = @cveJefe

select @emp = min(employeeid)

from employees

while @emp is not null

begin

select @NomEmp = @NomEmp + e.firstname + ' ' + e.lastname + ', '

from employees e

where e.reportsTo = @cveJefe

select @emp = min(employeeid)

from employees where employeeid < @emp

end

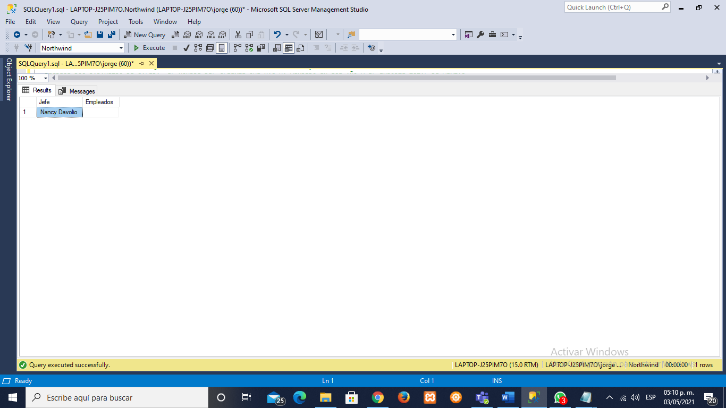
insert into #t values(@NomJefe,@NomEmp)

select \* from #t

Ejecución:

EXEC SP\_JefeEmp 1

Resultados:



4.- **PROCEDIMIENTO ALMACENADO** QUE RECIBA LA CLAVE DE UN EMPLEADO, REGRESE EN UN PARAMETRO DE SALIDA LOS NOMBRES DE LOS CLIENTES QUE HA ATENDIDO.

Código:

create proc SP\_EmpleadoClientes @EmpID int, @Cli nvarchar(1000) output as

declare @Cliente nchar(5),@OrdenID int,@Emp int

set @Cli = ' '

select @OrdenID = min(OrderID)

from Orders

while @OrdenID is not null

begin

select @Emp = EmployeeID

from Orders

where OrderID = @OrdenID

if @Emp = @EmpID

begin

select @Cliente = CustomerID

from Orders

where OrderID = @OrdenID

select @Cli = @Cli + ContactName

from Customers

where CustomerID = @Cliente

select @Cli = @Cli +', '

end

select @OrdenID = min (OrderID)

from Orders

where OrderID > @OrdenID

end

go

Ejecución:

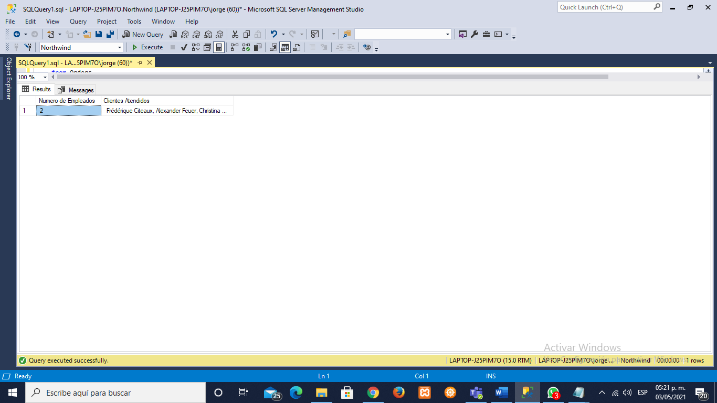
declare @Cli nvarchar(1000),@EmpID int

set @EmpID = 2

exec SP\_EmpleadoClientes @EmpID,@Cli output

select 'Numero de Empleados' = @EmpID, 'Clientes Atendidos' = @Cli

Resultados:



**Funciones**

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

Código:

create function dbo.ProveedorAño(@Prov1 int, @Prov2 int, @Año int)

returns varchar (100)

as

begin

declare @pieza1 int, @pieza2 int,@Res nvarchar (100)

select @pieza1 = sum(ode.Quantity)

from [Order Details] ode

inner join Orders o on o.OrderID = ode.OrderID

inner join Products p on p.ProductID = ode.ProductID

inner join Suppliers s on s.SupplierID = p.SupplierID

where s.SupplierID = @Prov1 and year (o.OrderDate) = @Año

select @pieza2 = sum(ord.Quantity)

from [Order Details] ord

inner join Orders od on od.OrderID = ord.OrderID

inner join Products pr on pr.ProductID = ord.ProductID

inner join Suppliers sup on sup.SupplierID = pr.SupplierID

where sup.SupplierID = @Prov2 and year(od.OrderDate) = @Año

if @pieza1 > @pieza2

select @Res = ContactName + ' VENDIO ' + convert(varchar(20), @pieza1) + ' PIEZAS ' from Suppliers where SupplierID = @Prov1

else

select @Res = ContactName + ' VENDIO ' +convert(varchar(20), @pieza2) + ' PIEZAS ' from Suppliers where SupplierID = @Prov2

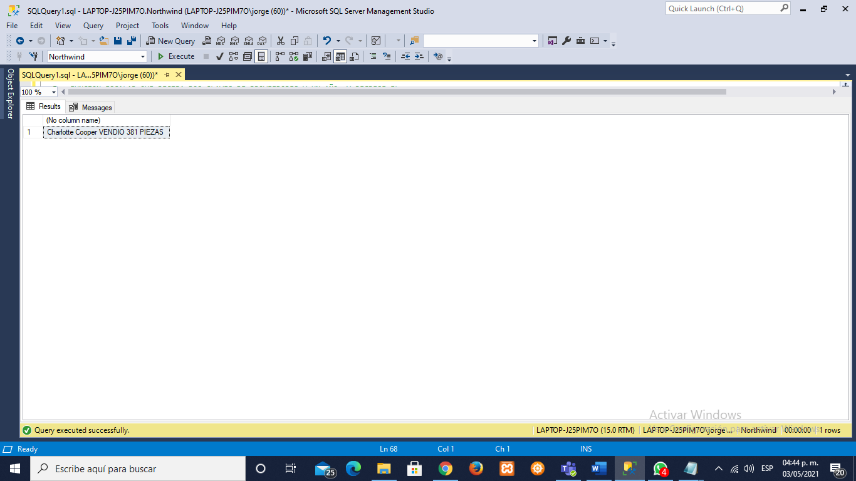
return @Res

end

Ejecución:

select dbo.ProveedorAño(1,3,1996)

Resultados:



6.- **FUNCION DE TABLA EN LINEA** QUE RECIBA EL AÑO Y REGRESE UNA TABLA CON EL NOMBRE DE TODOS LOS CLIENTES, EL IMPORTE DE VENTAS Y EL TOTAL DE ORDENES REALIZADAS ESE AÑO.

Código:

create function dbo.OrdenesAño(@año int)

RETURNS TABLE

AS

RETURN (

select c.CompanyName, 'Importe de ventas' = sum(d.quantity \* d.unitprice), 'Total de Ordenes' = count(o.OrderID)

from orders o

inner join [order details] d on d.orderid = o.orderid

RIGHT OUTER JOIN customers c on c.CustomerID = o.CustomerID and year(o.orderdate)=@año

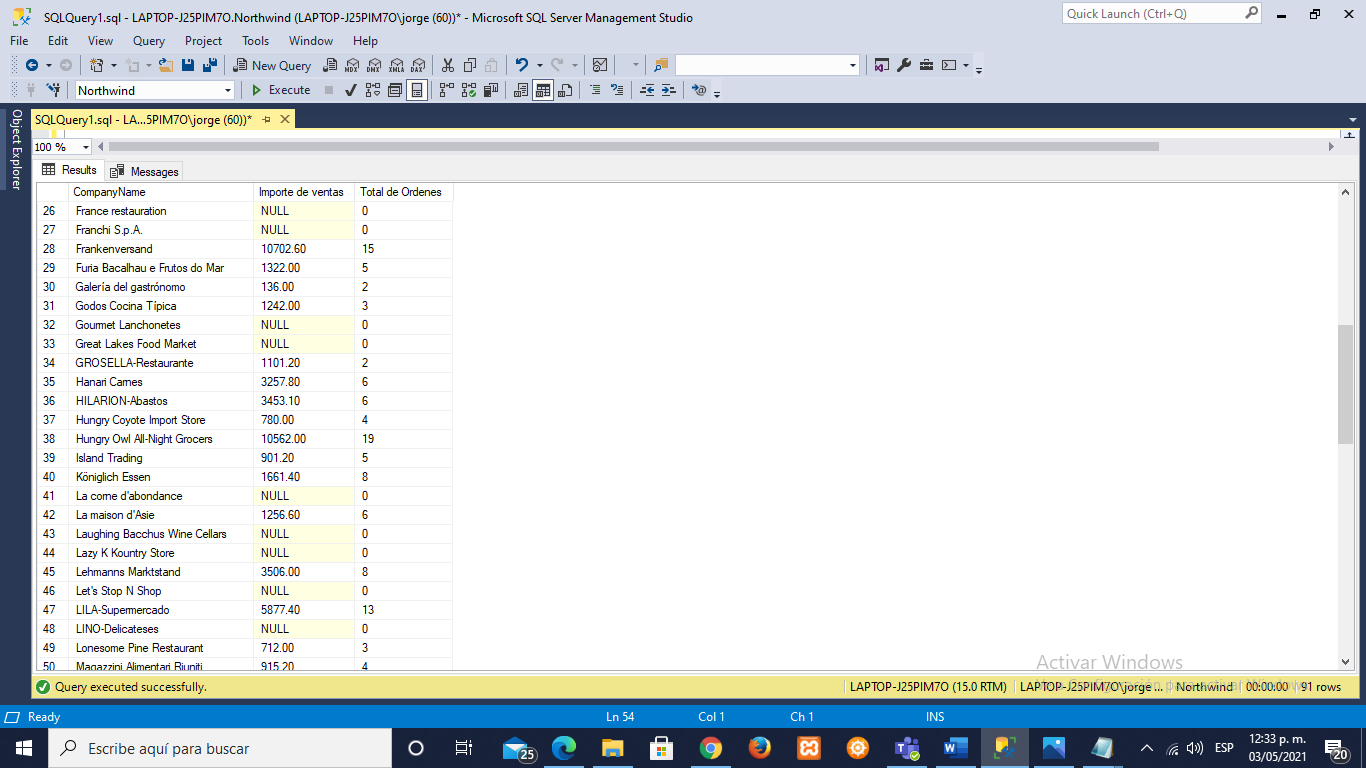
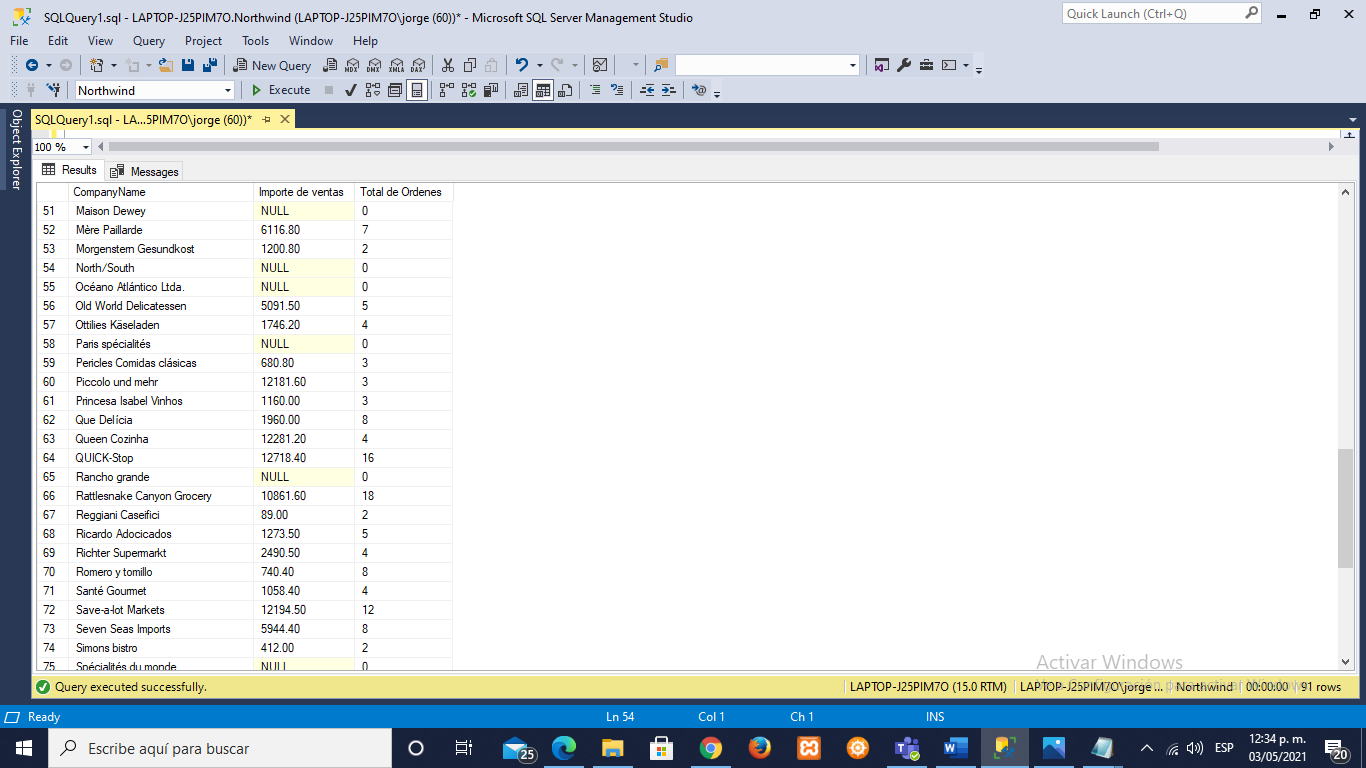
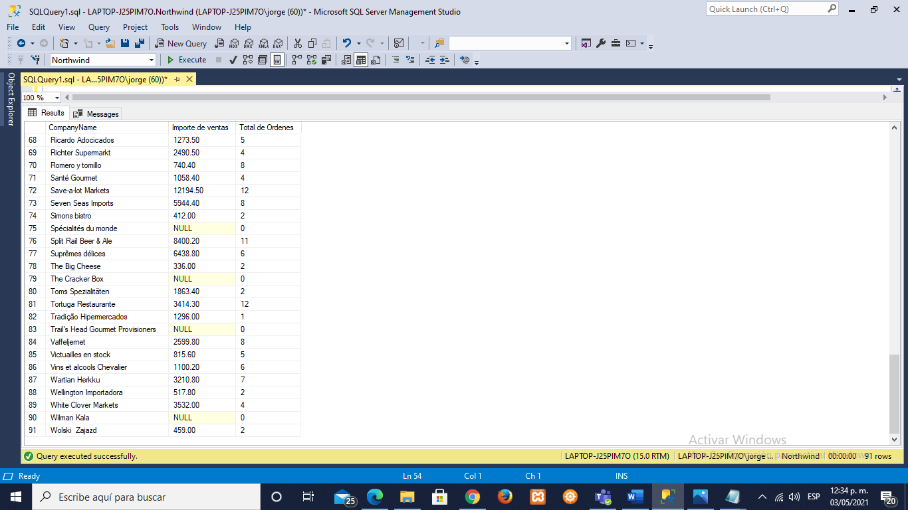
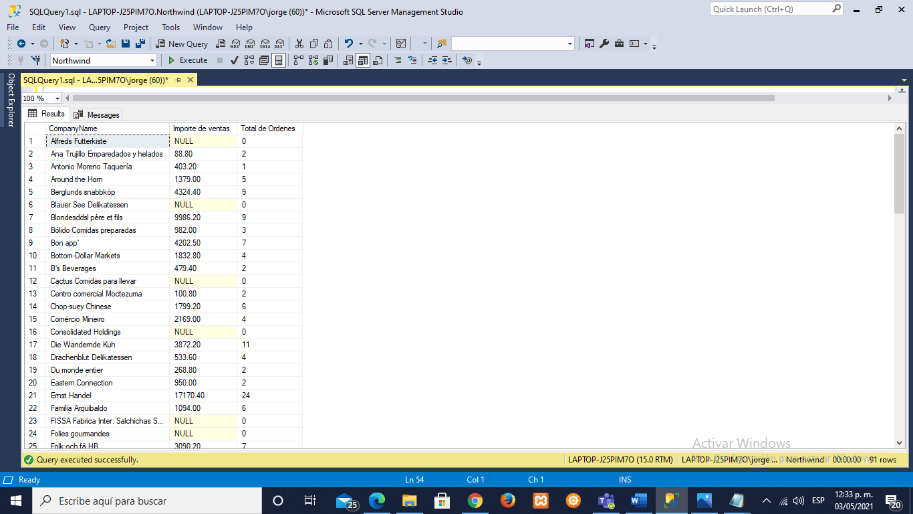
group by c.CompanyName

)

Ejecución:

select \* from dbo.OrdenesAño(1996) order by 1

Resultados:



7.- **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 COMPRADOS POR ESE EMPLEADO Y ESE AÑO.

CÓDIGO:

create function dbo.ProductosAño(@emp int, @año int)

RETURNS TABLE

AS

RETURN (

SELECT p.ProductName, 'Total de productos' = isnull(sum(ode.quantity), 0)

from Products p

inner join [Order Details] ode on ode.ProductID = p.ProductID

inner join Orders o on o.OrderID = ode.OrderID

RIGHT OUTER JOIN Employees e on e.EmployeeID = o.EmployeeID

where e.EmployeeID = @emp and year(o.orderdate) = @año

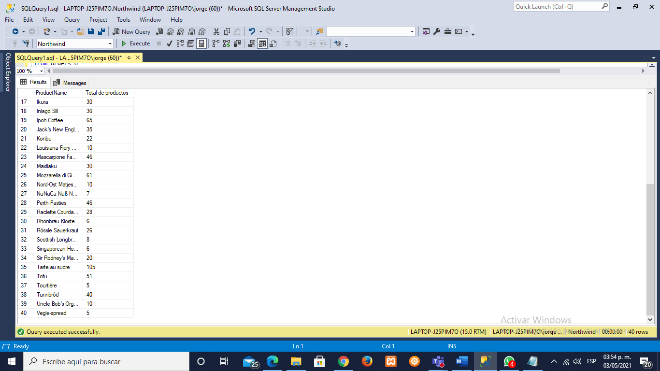
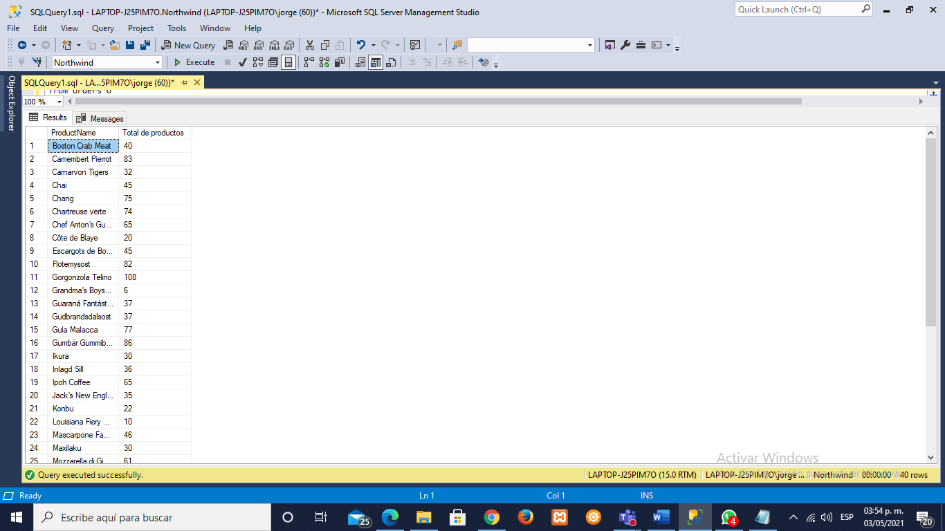
group by p.ProductName

)

Ejecución:

SELECT \* FROM dbo.ProductosAño(1, 1996)

RESULTADOS:



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

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

Código:

select A.ProductName, TotalPiezas96 =coalesce (A.Total, 0), TotalPiezas97 = coalesce(B.TOTAL,0), TotalPiezas98 = coalesce(C.TOTAL,0)

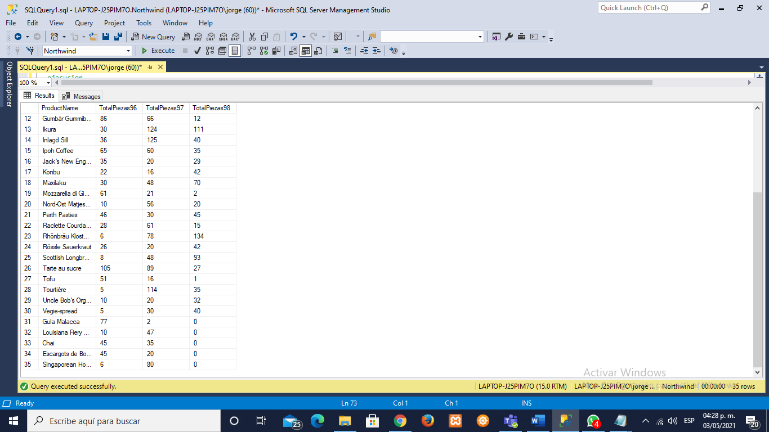
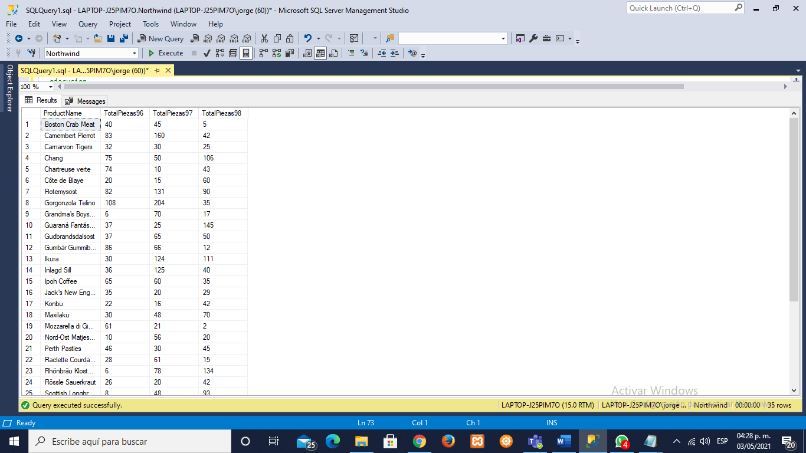
from dbo.ProductosAño(1,1996) A

INNER JOIN dbo.ProductosAño(1,1997) B ON A.ProductName = B.ProductName

LEFT OUTER JOIN dbo.ProductosAño(1,1998) C ON A.ProductName = C.ProductName

go

Resultados:



9.- **FUNCION DE TABLA DE MULTISENTENCIA** (NO LLEVA PARAMETROS DE ENTRADA) QUE REGRESE UNA TABLA CON EL NOMBRE DEL TERRITORIO Y LOS NOMBRES DE LOS EMPLEADOS QUE LO ATIENDEN.

|  |  |
| --- | --- |
| TERRITORIO | EMPLEADOS |
| NORTE | ANA LOPEZ, JOSE CASTRO |
| SUR | CARLOS LOPEZ, ANA CASTRO |

10.- **FUNCION DE TABLA DE MULTISENTENCIA** (NO LLEVA PARAMETROS DE ENTRADA) QUE REGRESE UNA TABLA CON DOS COLUMNAS: LA FECHA Y LOS FOLIOS DE LAS ORDENES QUE SE REALIZARON ESA FECHA.

|  |  |
| --- | --- |
| FECHA | NOMBRE |
| 1-1-2020 | 1 ,2,3,4,5 |
| 2-1-2020 | 6,7,8,9 |