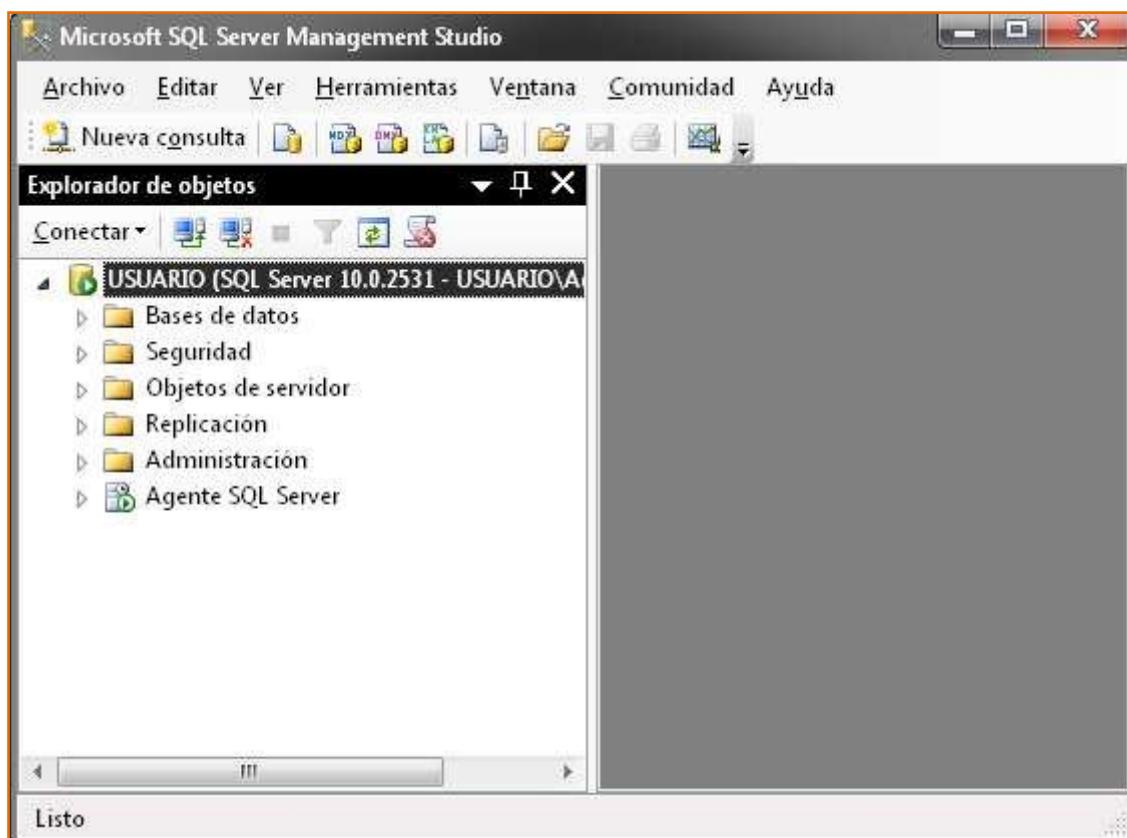


Manual de SQL

MANUAL SQL 2008



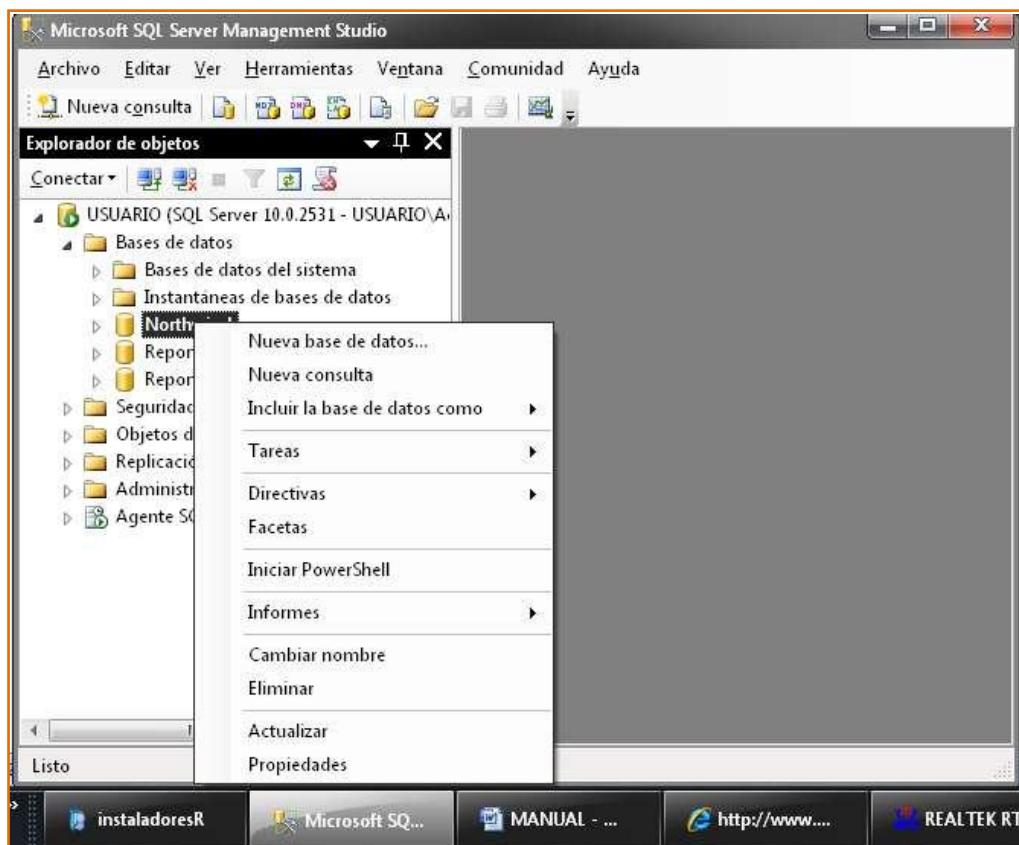
Antes de empezar a trabajar en SQL 2008, tenemos que especificar la manera de arrancar el sql 2008, para ello en tipo de servidor, escogemos motor de base de datos. Y en nombre de servidor escribimos un PUNTO (.) o el nombre de usuaria de la PC. y en autentificación escogemos AUTENTIFICACIÓN DE WINDOWS, y luego clic en CONECTAR.



SQL – TRANSACT (T-SQL)

SQL-TRANSACT, es una extensión del lenguaje estándar SQL implementado por Microsoft. Este lenguaje no diferencia las mayúsculas de las minúsculas.

Para utilizar el SQL TRANSACT (T-SQL). Abrimos una nueva consulta.



COMANDOS O SENTENCIAS

SELECT:

Sirve para seleccionar campos

Ejemplo 1: mostrar todos los empleados y todos sus atributos.

A screenshot of the SQL Server Management Studio query window titled "(local). Northwind - SQLQuery1.sql". The query is: "SELECT * from employees". The results pane shows a table with 9 rows of employee data. The columns are EmployeeID, LastName, FirstName, Title, TitleOfCourtesy, and BirthDate. The data is as follows:

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate
1	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000
7	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000
8	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000
9	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000

Ejemplo 2: mostrar todos los productos y todos sus atributos.

The screenshot shows a SQL query window with the following content:

```
(local) Northwind - SQLQuery1.sql* (local).Northwind - instnwnd.sql Summary
-- mostras todos los productos
SELECT *
from products
```

The results pane displays the following data from the products table:

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitType
1	1	Chai	1	1	10 boxes x 20 bags	18,00	35
2	2	Chang	1	1	24 - 12 oz bottles	19,00	11
3	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10,00	10
4	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22,00	50
5	5	Chef Anton's Gumbo Mix	2	2	36 boxes	21,35	0
6	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25,00	12
7	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30,00	18
8	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40,00	6
9	9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97,00	25
10	10	Ikura	4	8	12 - 200 ml jars	31,00	3
11	11	Queso Cabrales	5	4	1 kg pkg.	21,00	22
12	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38,00	80

Ejemplo 3: Mostrar el código, nombre y categoría de los productos:

Los atributos se muestran en el mismo orden en que se solicitan en SELECT, independientemente de su posición en la tabla.

The screenshot shows a SQL query window with the following content:

```
(local) Northwind - SQLQuery1.sql* (local).Northwind - instnwnd.sql Summary
-- solo mostramos el ID, nombre y categoria de productos
SELECT ProductID,ProductName,CategoryID
from products
```

The results pane displays the following data from the products table:

	ProductID	ProductName	CategoryID
1	1	Chai	1
2	2	Chang	1
3	3	Aniseed Syrup	2
4	4	Chef Anton's Cajun Seasoning	2
5	5	Chef Anton's Gumbo Mix	2
6	6	Grandma's Boysenberry Spread	2
7	7	Uncle Bob's Organic Dried Pears	7
8	8	Northwoods Cranberry Sauce	2
9	9	Mishi Kobe Niku	6
10	10	Ikura	8
11	11	Queso Cabrales	4
12	12	Queso Manchego La Pastora	4
13	13	Konbu	8

FROM: Especifica la tabla que utiliza el comando SELECT, para mostrar los campos

WHERE: especifica una condición que debe cumplirse para que devuelva cierta fila.

Ejemplo: mostrar los productos que pertenecen a la categoría 2.

```
(local). Northwind - SQLQuery1.sql* [Summary]
-- productos de la categoria 2
SELECT *
    from products
where CategoryID=2
```

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	Units
1	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10,00	13
2	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22,00	53
3	5	Chef Anton's Gumbo Mix	2	2	36 boxes	21,35	0
4	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25,00	120
5	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40,00	6
6	15	Genen Shouyu	6	2	24 - 250 ml bottles	15,50	39
7	44	Gula Malacca	20	2	20 - 2 kg bags	19,45	2

Ejemplo: productos que pertenece a la categoría 2 y 4:

```
(local). Northwind - SQLQuery1.sql* [Summary]
-- productos de la categoria 2 y 4
SELECT *
    from products
where CategoryID=2 or
categoryid =4
```

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	Units
1	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10,00	13
2	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22,00	53
3	5	Chef Anton's Gumbo Mix	2	2	36 boxes	21,35	0
4	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25,00	120
5	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40,00	6
6	11	Queso Cabrales	5	4	1 kg pkg.	21,00	22
7	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38,00	86
8	15	Genen Shouyu	6	2	24 - 250 ml bottles	15,50	39
9	31	Gorgonzola Telino	14	4	12 - 100 g pkgs	12,50	0
10	32	Mascarpone Fabioli	14	4	24 - 200 g pkgs.	32,00	9
11	33	Geitost	15	4	500 g	250	112

```
(local). Northwind - SQLQuery1.sql* [Summary]
SELECT ProductID,ProductName,UnitPrice
    from products
where UnitPrice > 100
```

	ProductID	ProductName	UnitPrice
1	29	Thüringer Rostbratwurst	123,79
2	38	Côte de Blaye	263,50

Ejemplo: Mostrar el producto tofu:

(local). Northwind - SQLQuery1.sql* Summary

```
SELECT ProductID,ProductName,UnitPrice
  from dbo.Products
 where ProductName = 'tofu'
```

Results Messages

	ProductID	ProductName	UnitPrice
1	14	Tofu	23.25

Operador IN, evalúa si un valor está contenido en una lista.

```
select *
  from Products
 where CategoryID IN (1, 3, 5)
```

(local). Northwind - SQLQuery1.sql* Summary

```
-- productos de la categoria 2 , 4 y 6
SELECT *
  from products
 where CategoryID IN (2, 4, 6)
```

Results Messages

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit
1	3	Aniseed Syrup	1	2	12 - 550 ml bottles
2	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars
3	5	Chef Anton's Gumbo Mix	2	2	36 boxes
4	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars
5	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars
6	9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.
7	11	Queso Cabrales	5	4	1 kg pkg.
8	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.
9	15	Genen Shouyu	6	2	24 - 250 ml bottles
10	17	Alice Mutton	7	6	20 - 1 kg tins

(local) Northwind - SQLQuery1.sql* Summary

```
SELECT *
  from products
 where SupplierID IN (3,5,8)
```

Results Messages

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit
1	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars
2	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.
3	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars
4	11	Queso Cabrales	5	4	1 kg pkg.
5	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.
6	19	Teatime Chocolate Biscuits	8	3	10 boxes x 12 pieces
7	20	Sir Rodney's Marmalade	8	3	30 gift boxes

Operador BETWEEN: necesita dos condiciones (Límite inferior y límite superior)

Ejemplo: Mostrar los productos cuyo precio oscilan entre 30 y 60 dólares

(local) Northwind - SQLQuery1.sql* Summary

```
SELECT ProductID,ProductName,UnitPrice
  from products
 where UnitPrice between 30 and 60
```

Results Messages

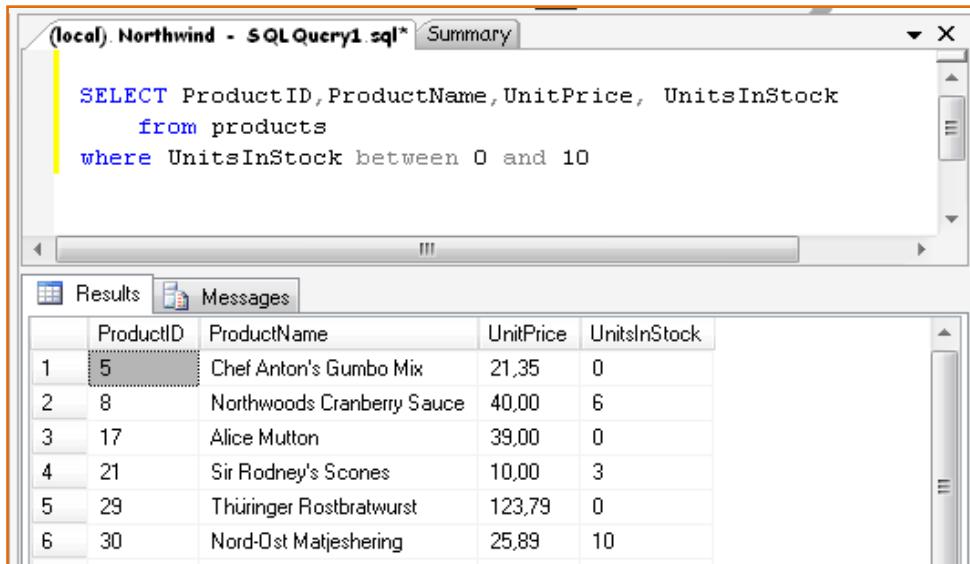
	ProductID	ProductName	UnitPrice
1	7	Uncle Bob's Organic Dried Pears	30,00
2	8	Northwoods Cranberry Sauce	40,00
3	10	Ikura	31,00
4	12	Queso Manchego La Pastora	38,00
5	17	Alice Mutton	39,00
6	26	Gumbär Gummibärchen	31,23

Ejemplo: Mostrar los productos cuyo stock es entre 0 y 10

(local) Northwind - SQLQuery1.sql* Summary

```
SELECT ProductID,ProductName,UnitPrice, UnitsInStock
  from products
 where UnitsInStock >= 0 and UnitsInStock<=10
```

Equivale



The screenshot shows a SQL Server Management Studio window titled '(local). Northwind - SQLQuery1.sql*'. The query window contains the following SQL code:

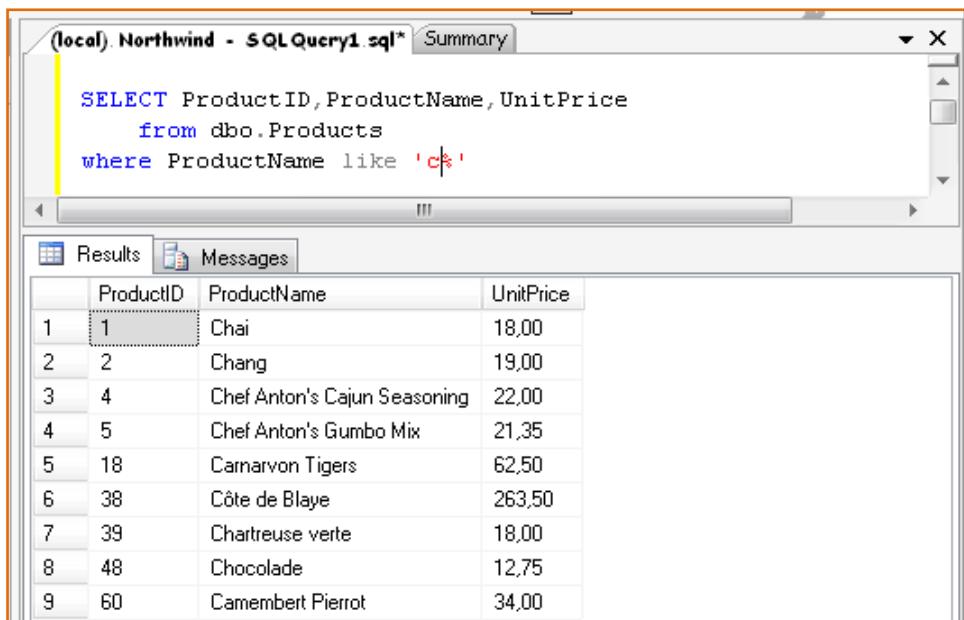
```
SELECT ProductID,ProductName,UnitPrice, UnitsInStock
  from products
 where UnitsInStock between 0 and 10
```

The results grid below shows the following data:

	ProductID	ProductName	UnitPrice	UnitsInStock
1	5	Chef Anton's Gumbo Mix	21,35	0
2	8	Northwoods Cranberry Sauce	40,00	6
3	17	Alice Mutton	39,00	0
4	21	Sir Rodney's Scones	10,00	3
5	29	Thüringer Rostbratwurst	123,79	0
6	30	Nord-Ost Matjeshering	25,89	10

USAR EL OPERADOR LIKE: Sirve para hacer búsquedas con comodines %, que reemplaza a una letra o cadena de caracteres.

Ejemplo: Mostrar todos los productos que empiezan con 'c'



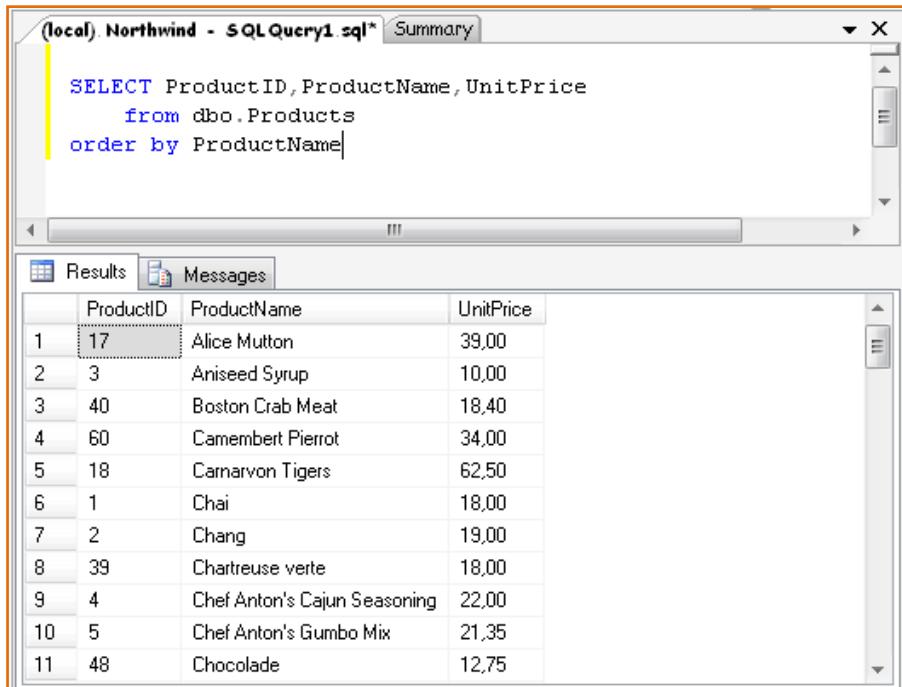
The screenshot shows a SQL Server Management Studio window titled '(local). Northwind - SQLQuery1.sql*'. The query window contains the following SQL code:

```
SELECT ProductID,ProductName,UnitPrice
  from dbo.Products
 where ProductName like 'c%'
```

The results grid below shows the following data:

	ProductID	ProductName	UnitPrice
1	1	Chai	18,00
2	2	Chang	19,00
3	4	Chef Anton's Cajun Seasoning	22,00
4	5	Chef Anton's Gumbo Mix	21,35
5	18	Carnarvon Tigers	62,50
6	38	Côte de Blaye	263,50
7	39	Chartreuse verte	18,00
8	48	Chocolade	12,75
9	60	Camembert Pierrot	34,00

ORDE BY: sirve para ordenar las filas según el o los campos especificados.

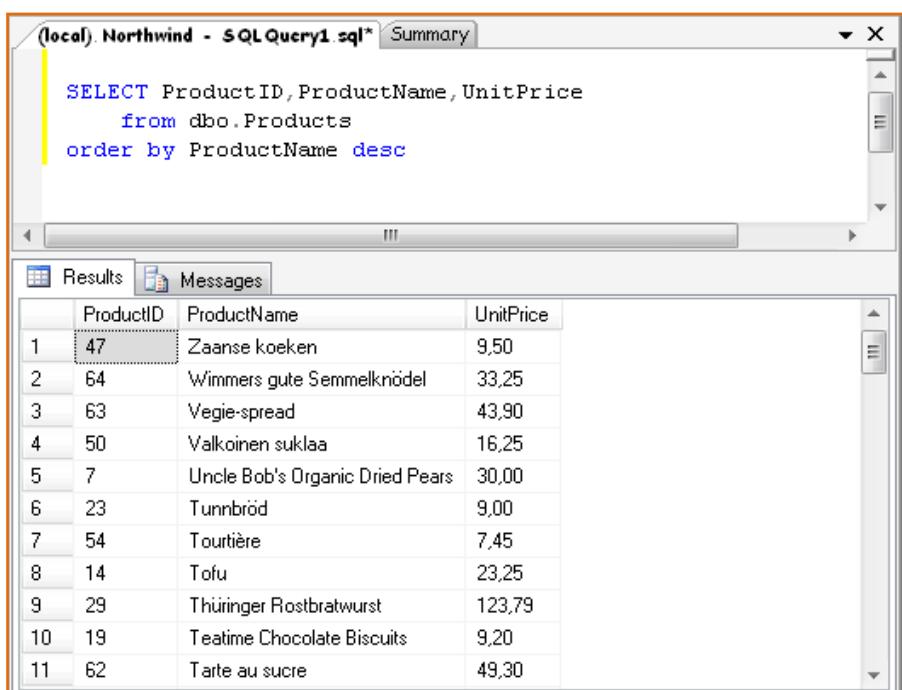


(local) Northwind - SQLQuery1.sql* Summary

```
SELECT ProductID,ProductName,UnitPrice
  from dbo.Products
order by ProductName|
```

Results

	ProductID	ProductName	UnitPrice
1	17	Alice Multon	39,00
2	3	Aniseed Syrup	10,00
3	40	Boston Crab Meat	18,40
4	60	Camembert Pierrot	34,00
5	18	Carnarvon Tigers	62,50
6	1	Chai	18,00
7	2	Chang	19,00
8	39	Chartreuse verte	18,00
9	4	Chef Anton's Cajun Seasoning	22,00
10	5	Chef Anton's Gumbo Mix	21,35
11	48	Chocolateade	12,75



(local) Northwind - SQLQuery1.sql* Summary

```
SELECT ProductID,ProductName,UnitPrice
  from dbo.Products
order by ProductName desc|
```

Results

	ProductID	ProductName	UnitPrice
1	47	Zaanse koeken	9,50
2	64	Wimmers gute Semmelknödel	33,25
3	63	Vegie-spread	43,90
4	50	Valkoinen suklaa	16,25
5	7	Uncle Bob's Organic Dried Pears	30,00
6	23	Tunnbröd	9,00
7	54	Tourtière	7,45
8	14	Tofu	23,25
9	29	Thüringer Rostbratwurst	123,79
10	19	Teatime Chocolate Biscuits	9,20
11	62	Tarte au sucre	49,30

Desc: ordena de forma descendente.

Asc: ordena de forma ascendente.

EJERCICIO BÁSICOS DE SQL TRANSACT (MICROSOFT SQL SERVER 2008)

Usando la base de datos NORTHWIND, realizar las siguientes consultas:

PREGUNTA N° 1

Seleccione todos los campos de la tabla cliente, ordenado por nombre del contacto de la compañía, alfabéticamente:

```
select *  
from dbo.Customers  
order by ContactName
```

PREGUNTA N° 2:

Seleccione todos los campos de la tabla órdenes, ordenados por fecha de la orden, descendenteamente.

```
select *  
from dbo.Orders  
order by OrderDate desc
```

PREGUNTA N° 3

Seleccione todos los campos de la tabla detalle de la orden, ordenada por cantidad pedida. Ascendentemente.

```
select *  
from [Order Details]  
order by Quantity asc
```

PREGUNTA N° 4

Obtener todos los productos, cuyo nombre comienzan con la letra P y tienen un precio unitario comprendido entre 10 y 120:

```
select *  
from Products  
where ProductName like 'p%' and UnitPrice between 10 and 120
```

PREGUNTA N° 5

Obtener todos los clientes de los países de: USA, Francia y UK:

```
select *  
from dbo.Customers  
where Country in('usa','francia','uk')  
--tambien puede ser where country ='usa' or country='francia' or 'uk'
```

PREGUNTA N° 6

Obtener todos los productos descontinuados y sin stock, que pertenecen a la categoría 1, 3, 4 y 7.

```
select *  
from Products  
where (Discontinued=1 and UnitsInStock=0) and CategoryID in(1,3,4,7)
```

PREGUNTA N° 7

Obtener todas las órdenes hechas por el empleado con código: 2, 5 y 7 en el año 1996

```
select *  
from Orders  
where EmployeeID in(2,5,7) AND YEAR(OrderDate)='1996'
```

PREGUNTA N° 8

Seleccionar todos los clientes que cuenten con Fax

```
select *
  from dbo.Customers
 where Fax is not null
--is not null ->sirve para el NULL
```

PREGUNTA N° 9

Seleccionar todos los clientes que no cuenten con FAX, del País de USA

```
select *
  from dbo.Customers
 where Country ='usa' and Fax is null
```

PREGUNTA N° 10

Seleccionar todos los empleados que cuentan con un jefe.

```
select *
  from dbo.Employees
 where ReportsTo is not null
```

PREGUNTA N° 11

Seleccionar todos los campos del cliente, cuya compañia empiecen con letra A hasta la D y pertenezcan al pais de USA, ordenarlos por la direccion.

```
select *
  from dbo.Customers
 where CompanyName like ('[abcd]%' )
---tambien puede ser '[a-d]%'
order by Address
```

PREGUNTA N° 12

Seleccionar todos los campos del cliente, cuya compañia empiecen con las letras de la B a la G, y pertenezcan al pais de UK, ordenarlos por nombre de la compañia

```
select *
  from Suppliers
 where CompanyName not like ('[b-G]%' ) and Country='uk'
order by Address
```

PREGUNTA N° 13

Seleccionar los productos vigentes cuyos precios unitarios están entre 35 y 250, sin stock en almacen, pertenecientes a las categorias 1, 3, 4, 7 y 8, que son distribuidos por los proveedores, 2, 4, 6, 7, 8 y 9

```
select *
  from dbo.Products
 where UnitPrice between 35 and 250 and unitsinstock =0 and
categoryid in(1,3,4,7,8) and SupplierID in(2,4,6,7,9)
```

PREGUNTA N° 14

seleccionar todos los campos de los productos descontinuados, que pertenezcan a los proveedores con códigos: 1, 3, 7, 8 y 9, que tengan stock en almacén, y al mismo tiempo que sus precios unitarios estén entre 39 y 190, ordenados por código de proveedores y precio unitario de manera ascendente.

```
select *
from dbo.Products
where Discontinued=1 and SupplierID in(1,3,7,8,9) and UnitsInStock >0
and
UnitPrice between 39 and 190
order by SupplierID, unitprice asc
--en order by, usamos dos condiciones para ordenar
```

PREGUNTA N° 15

Seleccionar los 7 productos con precio más caro, que cuenten con stock en almacén.

```
select top 7 *
--top 7, saca los primeros 7 registros después del ordenamiento
from Products
where UnitsInStock>0
order by UnitPrice desc
```

PREGUNTA N° 16

Seleccionar los 9 productos, con menos stock en almacén, que pertenezcan a la categoría 3, 5 y 8.

```
select top 9 *
--top 7, saca los primeros 7 registros después del ordenamiento
from Products
where CategoryID in(3,5,8)
order by UnitsInStock asc
```

PREGUNTA N° 17

Seleccionar las órdenes de compra, realizadas por el empleado con código entre el 2 y 5, además de los clientes con código que comienzan con las letras de la A hasta la G, del 31 de julio de cualquier año.

```
select *
from dbo.Orders
where EmployeeID between 2 and 5 and CustomerID like ('[a-g]%'') and
DATEPART(dd,OrderDate)=31 and DATEPART(mm,OrderDate)=7
```

PREGUNTA N° 18

Seleccionar las órdenes de compra, realizadas por el empleado con código 3, de cualquier año pero solo de los últimos 5 meses (agosto-Diciembre)

```
select *
from dbo.Orders
where CustomerID='3' and MONTH(OrderDate) in(8,9,10,11,12)
```

PREGUNTA N° 19

Seleccionar los detalles de las órdenes de compra, que tengan un monto de cantidad pedida entre 10 y 250.

```
select orderid, quantity  
from [Order Details]  
where Quantity between 10 and 250
```

PREGUNTA N° 20

Seleccionar los detalles de las órdenes de compras, cuyo monto del pedido esten entre 10 y 100.

```
select orderid, quantity, UnitPrice, UnitPrice* Quantity AS TOTAL  
--AS TOTAL... estamos dando el nombre a la columna calculada  
from [Order Details]  
where UnitPrice*Quantity between 10 and 300
```

PREGUNTA N° 21

```
select distinct country  
/* DISTINCT... filtra solo uno por tipo de dato, especificado en el  
select */  
from dbo.Customers
```

PREGUNTA N° 22

Mostrar los 15 productos más vendidos e incluyendo a los empates en el ultimo registro(PD. El operador TOP solo saca 'N' registros).

```
select top 15 WITH TIES OrderID, ProductID, Quantity  
--WITH TIES... saca los empates del ultimo registro  
from [Order Details]  
order by Quantity desc
```

COMBINACIÓN DE VARIAS TABLAS

OPERADOR INNER JOIN

El operador JOIN, sirve para combinar dos tablas o más entre sí, utilizando un atributo común.

CROSS JOIN: es un producto entre tablas

Ejemplo: nombre de la compañía del proveedor y el precio

```
select CompanyName, ProductName, UnitPrice
  from Suppliers
    inner join Products /*se colocan INNER JOIN tantas veces como tablas
    se le quiera unir a la tabla del from*/
      on Suppliers.SupplierID=Products.SupplierID
      --ON especifica la condición de la combinación o el campo en común.
```

Utilizando alias (las primeras letras de la tabla)

```
select s.companyname, p.productname, unitprice
  from Suppliers AS s
    INNER JOIN Products AS p
      on s.SupplierID=p.SupplierID
```

Ambigüedad: Se soluciona colocando el nombre de la tabla del campo.

Código de la orden de compra, fecha de la orden de compra, el código del producto, el nombre de producto, y la cantidad pedida.

```
select OrderID, OrderDate, ProductID,ProductName,Quantity
  from Orders As o
    INNER JOIN [Order Details] As od on o.OrderID=od.OrderID
    INNER JOIN Products As p on od.ProductID=p.ProductID
```

```
select Ord.OrderID, Ord.OrderDate, Pr.ProductID, Pr.ProductName,
od.Quantity
  from Orders as ord
    INNER JOIN [Order Details] as od
      on ord.OrderID=od.OrderID
    INNER JOIN Products as pr
      on od.ProductID=Pr.ProductID
```

UNION

Crea una tabla, unirlas campos a unir tiene que tener el mismo dominio:

```
select (firstname + ' ' + lastname) as name, city, postalcode
  from Employees
  union
  select Customers.CompanyName, city, postalcode
  from Customers
```

```

select P.ProductName as prod, P.UnitPrice as price,
       (unitprice * 1.1) as tax
into pricetable
--crea la tabla pricetable
from Products as p

select *
from pricetable

```

COUNT

```

select COUNT(ReportsTo) as total
--count muestra los no null
from Employees

```

GROUP BY

```

select ProductID, SUM(Quantity) as total_quantity
from [Order Details]
where UnitPrice< 10
group by ProductID

```

HAVING

HAVING SE APLICA DESPUES DE AGRUPAR CON LOS NUEVOS REGISTROS (HAVING DEPENDE DEL GROUP BY) SE DIFERENCIA DEL WHERE YA QUE ESTE SE APLICA EN LA TABLA ORIGINAL

```

select ProductID, SUM(Quantity) as total_quantity
from [Order Details]
group by ProductID
having ProductID= 2

```

```

select CategoryID, COUNT(*) as total
from Products as prod
where CategoryID=4
group by prod.CategoryID

```

Equivalencia con el comando having

```

select CategoryID, COUNT(*) as total
from Products as prod
group by prod.CategoryID
having CategoryID=4

```

```

select CategoryID, COUNT(*) as total
from Products as prod
where UnitPrice between 10 and 60
group by prod.CategoryID

```

```

select CategoryID, COUNT(*) as total
from Products as prod
group by prod.CategoryID
having COUNT(*)<=10

```

MUESTRAR LA CANTIDAD DE PRODUCTOS POR CATEGORIAS

```
select CategoryID, COUNT(*) as total  
from Products as prod  
group by CategoryID
```

MOSTRAR EL NUMERO, MAXIMO Y MININO POR CATEGORIA DE LOS PRECIOS

```
Select ca.CategoryName, MAX(pro.UnitPrice) as maximo,  
MIN(pro.UnitPrice) as minimo  
FROM Categories AS CA  
INNER JOIN Products AS PRO ON CA.CategoryID=PRO.CategoryID  
GROUP BY CA.CategoryName
```

MOSTRA AL EMPLEADO Y EL NUMERO DE ORDENES

```
select em.LastName, COUNT(*)  
from dbo.Employees as em  
inner join orders as o on o.EmployeeID=em.EmployeeID  
group by em.LastName
```

MOSTRAR: CUANTAS ORDENES A REALIZADO CADA EMPLEADO(MOSTRAR EL NOMBRE, APELLIDOS Y NUMERO DE PEDIDOS) QUE SEAN MAYORES A 100

```
select em.FirstName,em.LastName, COUNT(*)  
from dbo.Employees as em  
inner join orders as o on o.EmployeeID=em.EmployeeID  
group by em.FirstName,em.LastName  
having COUNT(*)>100
```

MOSTRAR CUANTOS PRODUCTOS A COMPRADO CADA UNO DE LOS CLIENTES (MOSTRAR EL NOMBRE DEL CLIENTE)

```
select cu.CompanyName,SUM(d.Quantity) as 'numero de productos'  
from Customers as cu  
inner join orders as o on cu.CustomerID=o.CustomerID  
inner join [Order Details] as d on d.OrderID=o.OrderID  
group by cu.CompanyName  
order by cu.CompanyName
```

MOSTRAR EL TOTAL PAGADO POR CADA ORDEN DE COMPRAS SOLO DEL MES DE ENERO DEL 1996

```
select o.OrderID, SUM(d.UnitPrice*d.Quantity*2.82) as total  
from orders as o  
inner join [Order Details] as d on d.OrderID=o.OrderID  
where month(o.OrderDate)=12 and YEAR(o.OrderDate)=1996  
group by o.OrderID
```

MOSTRAR EL NUMERO DE PRODUCTOS QUE VENDEN CADA UNO DE LOS PROVEEDORES (MOSTRAR SU NOMBRE DE LA EMPRESA-PROVEEDOR) SOLO CONSIDERAR A LOS PRODUCTOS VIGENTES.

```
select s.CompanyName, count(p.ProductID) as total  
from Products as p  
inner join Suppliers as s on s.SupplierID=p.SupplierID  
where p.Discontinued=0 and p.UnitsInStock>0  
group by s.CompanyName
```

EJERCICIOS INNER JOIN, GROUP BY, HAVING Y WHERE SQL TRANSACT (Microsoft SQL SERVER 2008)

USANDO LA BASE DE DATOS NORTHWIND, REALIZAR LAS SIGUIENTES CONSULTAS:

1. Visualizar el máximo y mínimo precio de los productos por categoría, mostrar el nombre de la categoría.

```
from Products as pro
inner join Categories as ca on ca.CategoryID=pro.CategoryID
group by ca.CategoryName
```

	CategoryName	maximo	minimo
1	Beverages	263,50	4,50
2	Condiments	43,90	10,00
3	Confections	81,00	9,20
4	Dairy Products	55,00	2,50
5	Grains/Cereals	38,00	7,00
6	Meat/Poultry	123,79	7,45
7	Produce	53,00	10,00
8	Seafood	62,50	6,00

2. Visualizar el máximo y mínimo precio de los productos por proveedor, mostrar el nombre de la compañía proveedora.

```
Select sup.CompanyName, MAX(pro.UnitPrice) as maximo,
MIN(pro.UnitPrice) as minimo
from Products as pro
inner join Suppliers as sup on pro.SupplierID=sup.SupplierID
group by sup.CompanyName
```

	CompanyName	maximo	minimo
1	Aux joyeux ecclésiastiques	263,50	18,00
2	Bigfoot Breweries	18,00	14,00
3	Cooperativa de Quesos 'Las Cabras'	38,00	21,00
4	Escargots Nouveaux	13,25	13,25
5	Exotic Liquids	19,00	10,00
6	Forêts d'érables	49,30	28,50
7	Formaggi Fortini s.r.l.	34,80	12,50
8	Gai pâturage	55,00	34,00
9	G'dai Mate	53,00	7,00

3. Seleccionar las categorías que tengan más de 5 productos. Mostrar el nombre de la categoría y el número de productos.

```
select ca.CategoryName, COUNT(pro.CategoryID) as cantidad
from Categories as ca
inner join Products as pro on pro.CategoryID=ca.CategoryID
group by ca.CategoryName
having count(pro.ProductName)>5
```

	CategoryName	cantidad
1	Beverages	12
2	Condiments	12
3	Confections	13
4	Dairy Products	10
5	Grains/Cereals	7
6	Meat/Poultry	6
7	Seafood	12

4. Calcular cuántos clientes existen en cada País.

```
select cu.Country as pais, COUNT(country) as [numero de clientes]
from Customers as cu
group by Country
```

	pais	numero de clientes
1	Argentina	3
2	Austria	2
3	Belgium	2
4	Brazil	9
5	Canada	3
6	Denmark	2
7	Finland	2
8	France	11
9	Germany	11
10	Ireland	1
11	Italy	3
12	Mexico	5

5. Calcular cuántos clientes existen en cada Ciudad.

```
Select cu.City as ciudad, COUNT(city) as [numero de clientes]
from Customers as cu
group by cu.City
```

The screenshot shows a SQL Server Management Studio window with the 'Resultados' tab selected. The results are displayed in a table with two columns: 'ciudad' and 'numero de clientes'. The data shows 17 cities, each with a value of 1 in the 'numero de clientes' column. The table has 17 rows, corresponding to the cities listed in the first column. The bottom status bar indicates the query was executed by 'USUARIO (10.0 SP1)' on 'USUARIO\Administrador ...' at 'Northwind' with a duration of '00:00:00' and 69 rows affected.

	ciudad	numero de clientes
7	Bergamo	1
8	Berlin	1
9	Bern	1
10	Boise	1
11	Bräcke	1
12	Brandenburg	1
13	Bruxelles	1
14	Buenos Aires	3
15	Butte	1
16	Campinas	1
17	Caracas	1

Consulta ejecutada por USUARIO (10.0 SP1) en USUARIO\Administrador ... en Northwind el 00:00:00 con 69 filas.

6. Calcular cuántos proveedores existen en cada Ciudad y País.

```
select city, COUNT( su.City) as numero, country, COUNT( su.Country) as numero
from Suppliers as su
group by City, Country
```

The screenshot shows a SQL Server Management Studio window with the 'Resultados' tab selected. The results are displayed in a table with four columns: 'city', 'numero', 'country', and 'numero'. The data shows 11 entries, each with a value of 1 in both 'numero' columns. The table has 11 rows, corresponding to the cities and countries listed in the first two columns. The bottom status bar indicates the query was executed by 'USUARIO (10.0 SP1)' on 'USUARIO\Administrador ...' at 'Northwind' with a duration of '00:00:00' and 29 rows affected.

	city	numero	country	numero
1	Melbourne	1	Australia	1
2	Sydney	1	Australia	1
3	Sao Paulo	1	Brazil	1
4	Montréal	1	Canada	1
5	Ste-Hyacinthe	1	Canada	1
6	Lyngby	1	Denmark	1
7	Lappeenranta	1	Finland	1
8	Annecy	1	France	1
9	Montceau	1	France	1
10	Paris	1	France	1
11	Berlin	1	Germany	1

Consulta ejecutada por USUARIO (10.0 SP1) en USUARIO\Administrador ... en Northwind el 00:00:00 con 29 filas.

7. Calcular el stock total de los productos por cada categoría. Mostrar el nombre de la categoría y el stock por categoría.

```
select ca.CategoryName, SUM( pro.UnitsInStock)
from Categories as ca
inner join Products as pro on pro.CategoryID=ca.CategoryID
group by ca.CategoryName
```

	CategoryName	(Sin nombre de columna)
1	Beverages	559
2	Condiments	507
3	Confections	386
4	Dairy Products	393
5	Grains/Cereals	308
6	Meat/Poultry	165
7	Produce	100
8	Seafood	701

Consulta ejec... USUARIO (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 8 filas

8. Calcular el stock total de los productos por cada categoría. Mostrar el nombre de la categoría y el stock por categoría. Solamente las categorías 2, 5 y 8.

```
select c.CategoryID,c.CategoryName, SUM(pro.UnitsInStock)
from products as pro
inner join Categories as c on pro.CategoryID=c.CategoryID
where c.CategoryID in( 2, 5 , 8)
group by c.CategoryID,c.CategoryName
```

	CategoryID	CategoryName	(Sin nombre de columna)
1	2	Condiments	507
2	5	Grains/Cereals	308
3	8	Seafood	701

Consulta ejec... USUARIO (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 3 filas

Lin 49 Col 2 Car 2 INS

9. Obtener el nombre del cliente, nombre de proveedor, nombre del empleado y el nombre de los productos que están en la orden 10250.

```
select c.ContactName, s.ContactName, e.FirstName, p.ProductName,
o.OrderID
from orders as o
inner join Employees as e on e.EmployeeID=o.EmployeeID
inner join Customers as c on c.CustomerID = o.CustomerID
inner join [Order Details] as od on od.OrderID=o.OrderID
inner join Products as p on p.ProductID=od.ProductID
inner join Suppliers as s on s.SupplierID=p.SupplierID
where o.OrderID = 10250
```

	ContactName	ContactName	ProductName	FirstName	ProductName	OrderID
1	Mario Pontes	Robb Merchant	Jack's New England Clam Chowder	Margaret	Jack's New England Clam Chowder	10250
2	Mario Pontes	Wendy Mackenzie	Manjimup Dried Apples	Margaret	Manjimup Dried Apples	10250
3	Mario Pontes	Shelley Burke	Louisiana Fiery Hot Pepper Sauce	Margaret	Louisiana Fiery Hot Pepper Sauce	10250

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 3 filas

10. Mostrar le numero de ordenes realizadas de cada uno de los clientes por año.

	CompanyName	numero de ordenes	(Sin nombre de columna)
7	Antonio Moreno Taquería	5	1997
8	Antonio Moreno Taquería	1	1998
9	Around the Horn	2	1996
10	Around the Horn	7	1997
11	Around the Horn	4	1998
12	Berglunds snabbköp	3	1996
13	Berglunds snabbköp	10	1997
14	Berglunds snabbköp	5	1998
15	Blauer See Delikatessen	4	1997
16	Blauer See Delikatessen	3	1998

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 234 filas

11. Mostrar el número de órdenes realizadas de cada uno de los empleados en cada año.

The screenshot shows a SQL query window titled "SQLQuery1.sql - [L...dministrador (55)]". The query selects the first name of employees and the count of their orders, grouped by year and employee name. The results are displayed in a table with three columns: FirstName, numero de ordenes, and (Sin nombre de columna). The table contains 10 rows of data. Below the table, a message indicates the query was executed successfully, and the status bar shows the execution time and number of rows affected.

	FirstName	numero de ordenes	(Sin nombre de columna)
1	Andrew	16	1996
2	Andrew	41	1997
3	Andrew	39	1998
4	Anne	5	1996
5	Anne	19	1997
6	Anne	19	1998
7	Janet	18	1996
8	Janet	71	1997
9	Janet	38	1998
10	Laura	19	1996

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 27 filas Lin 16 Col 40 Car 40 INS

12. Mostrar el número de órdenes realizadas de cada uno de los clientes por cada mes y año.

```
select c.ContactName as cliente,COUNT( o.OrderID) as [numero de ordenes],  
MONTH(o.OrderDate) as mes,  YEAR( o.OrderDate) as año  
from Customers as c  
inner join orders as o on o.CustomerID=c.CustomerID  
group by c.ContactName, month(o.OrderDate), YEAR( o.OrderDate)
```

The screenshot shows a SQL query window titled "SQLQuery1.sql - [L...dministrador (55)]". The query selects the contact name of customers and the count of their orders, grouped by year, month, and customer name. The results are displayed in a table with four columns: cliente, numero de ordenes, mes, and año. The table contains 9 rows of data. Below the table, a message indicates the query was executed successfully, and the status bar shows the execution time and number of rows affected.

	cliente	numero de ordenes	mes	año
1	Alejandra Camino	1	3	1998
2	Alejandra Camino	1	4	1998
3	Alejandra Camino	2	8	1996
4	Alejandra Camino	1	9	1996
5	Alexander Feuer	1	3	1998
6	Alexander Feuer	1	6	1997
7	Alexander Feuer	1	8	1996
8	Alexander Feuer	1	10	1997
9	Alexander Feuer	1	12	1997

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 636 filas

13. Contar el número de órdenes que se han realizado por año y meses.

```
Select COUNT(o.OrderID) as [Número de órdenes],
MONTH(o.OrderDate) as mes, YEAR(o.OrderDate) as año
from orders as o
group by MONTH(o.OrderDate), YEAR(o.OrderDate)
```

	Número de órdenes	mes	año
1	22	7	1996
2	25	8	1996
3	23	9	1996
4	26	10	1996
5	25	11	1996
6	31	12	1996
7	33	1	1997
8	29	2	1997
9	30	3	1997

14. Seleccionar el nombre de la compañía del cliente, el código de la orden de compra, la fecha de la orden de compra, código del producto, cantidad pedida del producto, nombre del producto, el nombre de la compañía proveedora y la ciudad del proveedor.

```
select c.CompanyName as cliente, o.OrderID, o.OrderDate, pd.ProductID ,
od.Quantity, pd.ProductName as proveedor, su.CompanyName, su.City
from customers as c
inner join Orders as o on o.CustomerID=c.CustomerID
inner join [Order Details] as od on od.OrderID=o.OrderID
inner join Products as pd on pd.ProductID=od.ProductID
inner join Suppliers as su on su.SupplierID=pd.SupplierID
```

cliente	OrderID	OrderDate	ProductID	Quantity	proveedor	CompanyName	City
Vins et alcools Chevalier	10248	1996-07-04	11	12	Queso Cabrales	Cooperativa de Quesos 'Las Cabras'	Oviedo
Vins et alcools Chevalier	10248	1996-07-04	42	10	Singaporean Hokkien Fried Mee	Lekka Trading	Singapore
Vins et alcools Chevalier	10248	1996-07-04	72	5	Mozzarella di Giovanni	Fornaggi Fortini s.r.l.	Ravenna
Toms Spezialitäten	10249	1996-07-05	14	9	Tofu	Majum's	Osaka
Toms Spezialitäten	10249	1996-07-05	51	40	Manjimup Dried Apples	G'day, Mate	Sydney
Hanai Cárnicas	10250	1996-07-08	41	10	Jack's New England Clam Chowder	New England Seafood Company	Boston

15. Seleccionar el nombre de la compañía del cliente , nombre del contacto, el código de la orden de compra, la fecha de la orden de compra, código del producto, cantidad pedida del producto, nombre del producto y el nombre de la compañía proveedora, usar Join. Solamente las compañías proveedoras que comienzan con la letra de la A hasta la letra G, además la cantidad pedida del producto debe estar entre 18 y 190.

```
select c.CompanyName,c.ContactName as contacto,
o.OrderID,o.OrderDate,pd.ProductID ,
od.Quantity,pd.ProductName as proveedor,su.CompanyName
from customers as c
inner join Orders as o on o.CustomerID=c.CustomerID
inner join [Order Details] as od on od.OrderID=o.OrderID
inner join Products as pd on pd.ProductID=od.ProductID
inner join Suppliers as su on su.SupplierID=pd.SupplierID
where (c.CompanyName like '[a-g]%' )and od.Quantity
between 18 and 190
order by c.CompanyName
```

	CompanyName	contacto	OrderID	OrderDate	ProductID	Quantity	proveedor	CompanyName
1	Alfreds Futterkiste	Maria Anders	10643	1997-08-25 ...	39	21	Chartreuse verte	Aux joyeux ecclésiastiques
2	Alfreds Futterkiste	Maria Anders	10692	1997-10-03 ...	63	20	Vegie-spread	Pavlova, Ltd.
3	Alfreds Futterkiste	Maria Anders	11011	1998-04-09 ...	58	40	Escargots de Bourgogne	Escargots Nouveaux
4	Alfreds Futterkiste	Maria Anders	11011	1998-04-09 ...	71	20	Flotemysost	Norske Meierier
5	Anthoni Daniels	Anthoni Daniels	10077	1997-09-20 ...	20	20	Gouda Cheddar	Uncle Bob's Organic Dill Lite

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 397 filas

16. Seleccionar cuantos proveedores tengo en cada país, considerando solo a los nombre de los proveedores que comienzan con la letra E hasta la letra P, además de mostrar solo los países donde tenga más de 2 proveedores.

```
select su.Country, COUNT(su.Country)
from suppliers as su
where su.CompanyName like '[e-p]%'
group by su.Country
having COUNT(su.country)>2
```

	Country	(Sin nombre de columna)
1	Germany	3
2	USA	3

(local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 2 filas

17. Obtener el número de productos, por cada categoría. Mostrando el nombre de la categoría, el nombre del producto, y el total de productos por categoría, solamente de las categorías 3, 5 y 8. Ordenar por el nombre de la categoría.

```
select pd.CategoryID as categoria ,count( pd.ProductID) as [numero de productos]
from products as pd
inner join Categories as ca on ca.CategoryID=pd.CategoryID
group by pd.CategoryID
```

	categoria	numero de productos
1	1	12
2	2	12
3	3	13
4	4	10
5	5	7
6	6	6
7	7	5
8	8	12

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 8 filas

18. Mostar el nombre de la orden, la fecha de la orden y el total en soles de cada orden.

```
select o.CustomerID,o.OrderDate,od.Quantity*od.UnitPrice as total
from orders as o
inner join [Order Details] as od on od.OrderID=o.OrderID
```

	CustomerID	OrderDate	total
1	VINET	1996-07-04 00:00:00.000	168,00
2	VINET	1996-07-04 00:00:00.000	98,00
3	VINET	1996-07-04 00:00:00.000	174,00
4	TOMSP	1996-07-05 00:00:00.000	167,40
5	TOMSP	1996-07-05 00:00:00.000	1696,00
6	HANAR	1996-07-08 00:00:00.000	77,00
7	HANAR	1996-07-08 00:00:00.000	1484,00
8	HANAR	1996-07-08 00:00:00.000	252,00
9	VICTE	1996-07-08 00:00:00.000	100,80

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 2155 filas

19. Mostrar el nombre de producto y en cuantas órdenes de compra se encuentra.

```
select pd.ProductName, COUNT(od.OrderID) as [productos en ordenes]
from products as pd
inner join [Order Details] as od on od.ProductID=pd.ProductID
group by pd.ProductName
```

	ProductName	productos en ordenes
1	Alice Mutton	37
2	Aniseed Syrup	12
3	Boston Crab Meat	41
4	Camembert Pierrot	51
5	Carnarvon Tigers	27
6	Chai	38
7	Chang	44
8	Chartreuse verte	30
9	Chef Anton's Cajun Seasoning	20

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 77 filas

UPDATE

subir el precio de los productos

UPDATE

SUBIR EL PRECIO DE LOS PRODUCTOS

```
Update Products
set UnitPrice=UnitPrice*1.1

Select min(unitprice)as minimo,MAX(unitprice) as maximo
from Products as p
--where p.CategoryID=1
group by p.CategoryID
having p.CategoryID=1
```

```
update Products
set UnitPrice=UnitPrice*1.5
where CategoryID=1

Select *
from Products as p
where p.CategoryID=1
```

/* cambiar el nombre del pais de UK pro el de perusalem*/

```
update Suppliers
set Country='perusalen'
where Country='uk'

Select *
from Suppliers as s
where s.Country='uk'
```

CAMBIAR EL NOMBRE DE LA CATEGORIA 5 POR JUGUETES

```
update Categories  
set CategoryName='juguetes'  
where CategoryID=5  
  
select *  
from Categories as c  
where c.CategoryID=5
```

CAMBIAR Y COLOCAR EL NUMERO DE FAX '0' A LOS QUE SON NULOS

```
update Customers  
set Fax ='0'  
where Fax is NULL  
  
select *  
from Customers as C  
where C.Fax is NULL
```

CAMBIAR EL NOMBRE Y APELLIDO DEL EMPREADO CODIGO 6 CON SU NOMBRE Y APELLIDO

```
update Employees  
set FirstName ='ronald ', lastname = 'alcantara '  
where employeeid=6  
  
select *  
from Employees as e  
where e.EmployeeID=6
```

CAMBIAR EL CODIGO DEL CLIENTE ALFKI POR UNCFI

```
update Customers  
set CustomerID ='UNCFI'  
where CustomerID='ALFKI'  
  
select *  
from Customers as c  
where c.CustomerID='alfki'
```

CAMBIAR EL codigo DE LA ORDEN 10250, POR 99999

```
update Orders  
set OrderID =99999  
where OrderID=10250  
  
select *  
from Orders as O  
where O.OrderID=10250
```

```
DELETE Borra registros
```

```
delete  
from  
customer-- limpia la tabla (elimina los registros)
```

ELIMINAR LA ORDEN 10255

eliminamos el detalle (se elimina la relacion de muchos)

```
DELETE  
FROM [Order Details]  
WHERE OrderID=10255  
  
--luego eliminamos la orden (la relacion de uno)  
  
DELETE  
FROM Orders  
WHERE OrderID=10255
```

ELIMINAR LA ORDEN 10260

eliminamos el detalle (se elimina la relacion de muchos)

```
DELETE  
FROM [Order Details]  
WHERE OrderID=10255  
  
--luego eliminamos la orden (la relacion de uno)  
  
DELETE  
FROM Orders  
WHERE OrderID=10255
```

ELIMINAR TODOS LOS PRODUCTOS DISCONTINUADOS

```
select *  
from [Order Details] as o  
where ProductID in (select p.ProductID  
                     from Products as p  
                     where p.Discontinued=1)
```

--eliminamos los resultados del select (se elimina la relacion de muchos)

```
DELETE  
FROM [Order Details]  
WHERE ProductID in (select p.ProductID  
                     from Products as p  
                     where p.Discontinued=1)
```

--luego eliminamos la orden (la relacion de uno)

```
DELETE  
FROM Products  
WHERE Discontinued=1
```

muestre los productos cuyo precio es mayor al promedio de precio de todos los productos

```
select *
from Products as p
where p.UnitPrice >
(select AVG(p.UnitPrice) --saca el promedio de los precios
from Products as p)
```

INSERT

```
-----  
insert into NOMBRE DE LA TABLA  
CAMPO1, CAMPO2, CAMPO3,...  
VALUES (VALOR1, VALOR2, VALOR3, ...
```

Propiedades de columna	
Conjunto de columnas	No
Descripción	
Determinístico	Sí
Disperso	No
■ Especificación de columna calculada	
■ Especificación de identidad	Sí
(Identidad)	Sí
Incremento de identidad	1
Inicialización de identidad	1
■ Especificación de texto completo	No
Indizable	Sí
Intercalación	<base de datos predeterminada>
No disponible para replicación	No

```
select *
from alumnos
```

```
insert into alumnos
VALUES (003, 'ronald', 'alcantara quispe', 12)

insert into alumnos
(codigo,apellidos, nombres,nota)
VALUES (005, 'ana', 'estrada ', 13)
```

```
select *
from articulos
```

```
insert into articulos
(nombre,precio,stock)--especificamos porque el nroarticulo es
autogenerado
values (25,'laptop hp',1200,10)
```

PROCEDIMIENTOS ALMACENADOS

Crear procedimientos almacenados no se tienen que aglomerar todos los procedimientos almacenados ya que hace mas lenta al servidor

```
create proc xxx
as
insert into articulos
(nombre,precio,stock) --especificamos porque el nroarticulo es
autogenerado
values (30,'laptop hp',1200,10)
```

```
create proc SPQ_PRODUCTOS
@categoria int --Variable entera
as
select *
from Products
where CategoryID=@categoria
```

----- llamado
exec SPQ_PRODUCTOS 2

```
---modificado-----
alter proc SPQ_PRODUCTOS
as
select *
from Products
where CategoryID=1
```

----- llamado
exec SPQ_PRODUCTOS

DECLARAR VARIABLES

Declare @var tipo_dato

Ejemplo:

Declare @c int

Asignar valores a variables

Set @c=@c+1

CREA UNA PROCEDIMIENTO QUE OBTENGA TODOS LOS PRODUCTOS

```
create proc productos --sp en el nombre indica que es un procedimiento
del sistema
--declaracion de parametros
@cod int
as
select *
from Products
where SupplierID=@cod --estamos agregando un parametro
--USE para cambiar la base de datos
```

```
--formas de llamar el procedimiento almacenado
exec productos 5 --hacemos el llamado del procedimiento almacenado
--o tambien puede ser
execute productos 5
--o tambien puede ser
productos 5
```

Resultados									
	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	Re
1	11	Queso Cabrales	5	4	1 kg pkg.	21,00	22	30	30
2	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38,00	86	0	0

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:01 2 filas

MODIFICAMOS EL PROCEDIMIENTO

```
ALTER proc productos --sp_ indica que es un procedimiento del
sistema
--declaracion de parametros
@cod int
as
select P.ProductID, P.ProductName, UnitPrice, SupplierID
from Products AS P
where SupplierID=@cod --estamos agregando un parametro
```

SOLO EJECUTARA HASTA ANTES DEL EXEC PORQUE SINO SE VUELVE RECURSIVA
EXEC productos 1

Resultados			
	ProductID	ProductName	UnitPrice
1	1	Chai	18,00
2	2	Chang	19,00
3	3	Aniseed Syrup	10,00

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 3 filas

ELIMINACION DE PROCEDIMIENTOS ALMACENADOS
---HACER UN PROCEDIMIENTOS ALMACENADO QUE ELIMINE UNA ORDEN DE COMPRA

```
create proc eli_orden
@cod int
as
--aqui se declaran las variables (se ejecutan con el programa)

select *
from Orders
where OrderID=@cod

delete
from [Order Details]
where OrderID=@cod

delete
from Orders
where OrderID=@cod
```

```
select *
from Orders
where OrderID=@cod
-----SOLO EJECUTARA HASTA ANTES DEL EXEC PORQUE SINO SE VUELVE
RECURSIVA
```

```
EXEC eli_orden 10571
```

CREAR EL SP PARA CAMBIAR EL STOCK DE UN PRODUCTO*/

```
create proc CAM_STOCK
@cod int,
@val int
as
select *
from Products
where ProductID=@cod

update Products
set UnitsInStock=@val
where ProductID=@cod

select *
from Products
where ProductID=@cod
```

```
--SOLO EJECUTARA(F5) HASTA ANTES DEL EXEC PORQUE SINO SE VUELVE
RECURSIVA
```

```
EXEC cam_stock 50,50
```

```
crear un sp para insertar crear tabla instrucciones, nombre apellidos,  
correo, direccion */
```

```
create proc nuevo_inscrito  
@dni char(8),  
@nombre varchar(20),  
@apellido varchar(25),  
@email varchar(40),  
@sexo char(1)  
  
as  
insert into inscripciones  
values (@dni, @nombre, @apellido, @email, @sexo)  
  
select *  
from dbo.inscripciones  
  
nuevo_inscrito '18071111', 'ronald ivan', 'alcantara quispe',  
'ron@hotmail.com', 'm'
```

crear un sp que muestre los productos que se encuentra en un rango de precios*/

```
create proc rango_precios
@mini float,
@maxi float
as
select *
from Products as p
where p.UnitPrice between @mini and @maxi
order by UnitPrice
```

```
exec rango_precios 10,20
```

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	Reorder
1	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10,00	13	70	25
2	21	Sir Rodney's Scones	8	3	24 pkgs. x 4 pieces	10,00	3	40	5
3	74	Longlife Tofu	4	7	5 kg pkg.	10,00	4	20	5
4	46	Spegesild	21	8	4 - 450 g glasses	12,00	95	0	0
5	31	Gorgonzola Telino	14	4	12 - 100 g pkgs	12,50	0	70	20
6	68	Scottish Longbreads	8	3	10 boxes x 8 piec...	12,50	6	10	15
7	48	Chocolade	22	3	10 pkgs.	12,75	15	70	25
8	77	Original Frankfurter...	12	2	12 boxes	13,00	32	0	15
9	50	Escarmont de Rour	27	9	24 piezas	13,25	62	0	20

```
create proc empleado_un
@anio int
as
select year(orderdate) as anio,      (FirstName + ' ' + lastname),
SUM(od.Quantity)
AS total
from Employees as e
inner join Orders as o on e.EmployeeID=o.EmployeeID
inner join [Order Details] as od on od.OrderID=o.OrderID
where year(OrderDate)=@anio
group by e.FirstName, e.LastName, year(o.OrderDate)
order by e.FirstName
```

```
exec empleado_un 1996
```

	anio	(Sin nombre de columna)	total
1	1996	Andrew Fuller	1085
2	1996	Anne Dodsworth	575
3	1996	Janet Leverling	940
4	1996	Laura Callahan	923
5	1996	Margaret Peacock	2212
6	1996	Michael Suyama	963
7	1996	Nancy Davolio	1620
8	1996	Robert King	485
9	1996	Steven Buchanan	778

/*los productos mas vendidos por año segun el numero de registros k se quiera mostrar y mostrar en monto vendido*/

```
create proc monto_vendido
@anio int,
@num int
as
select top (@num) p.ProductName, SUM(Quantity*od.UnitPrice) as monto
from Products as p
inner join [Order Details] as od on od.ProductID=p.ProductID
inner join Orders as o on o.OrderID=od.OrderID
where year(OrderDate)=@anio
group by p.ProductName
order by monto
```

--LLAMADO AL PROCEDIMIENTO

```
exec monto_vendido 1996, 4
```

/*-----
PRODUCTOS NO VENDIDOS EN UN DETERMINADO AÑO*/

```
select *
from Products
where ProductID not in (select ProductID
from [Order Details]
)
```

/*-----
PRODUCTOS NO VENDIDOS EN UN DETERMINADO AÑO*/

```
create proc productonovendio1
@anio int
as
select p.ProductName
from Products as p
where ProductID not in
(
    select ProductID
    from [Order Details] as od
    inner join Orders as o on o.OrderID=od.OrderID
    where YEAR(o.OrderDate)=@anio)
```

--LLAMADO AL PROCEDIMIENTO

```
exec productonovendio1 1996
```

The screenshot shows a 'Resultados' (Results) tab in a SQL Server Management Studio window. It displays a table with three rows, each containing a product name. The columns are labeled 'ProductName'. The rows are numbered 1, 2, and 3, corresponding to the results of the stored procedure execution.

ProductName
1 Mishi Kobe Niku
2 Sirap d'ériable
3 Chocolade

Consulta ejecutada correctamente. (local) (10.0 SP1) USUARIO\Administrador ... Northwind 00:00:00 3 filas

```

/*
----- buscar frase en productos frase no exacta, como resultado el empleado
y la cantidad de productos vendidos y el nombre del productos*/

```

```

create procedure busqueda
@prod nvarchar(40)= '%'
as
select (e.LastName+ ' '+ e.FirstName) as empleado ,p.ProductName,
SUM(od.Quantity)
from Employees as e
inner join Orders as o on o.EmployeeID=e.EmployeeID
inner join [Order Details] as od on od.OrderID=o.OrderID
inner join Products as p on p.ProductID=od.ProductID

where p.ProductName like @prod
group by e.LastName, e.FirstName, p.ProductName
order by e.LastName, p.ProductName

```

```

exec busqueda '%queso%'

```

mostra la orden, y los dias transcurridos entre el requerimiento y el espacho. en cierto año

```

create proc diffecha
@dif int,
@anio int
as
select count(OrderID) as cantidad, day(o.ShippedDate-o.RequiredDate)
as dias
from Orders as o
where day(o.ShippedDate-o.RequiredDate)<@dif
and YEAR(requireddate)=@anio
group by day(o.ShippedDate-o.RequiredDate)

```

```

exec diffecha 10,1996

```

```

/*otra forma*/

```

```

create proc diffechal
@dif int,
@anio int
as
select count(OrderID) as cantidad,
DATEDIFF (day,o.ShippedDate,o.RequiredDate) as dias
from Orders as o
where datediff ( day,o.ShippedDate,o.RequiredDate)<@dif
and YEAR(requireddate)=@anio
group by datediff (day,o.ShippedDate,o.RequiredDate)

```

```

exec diffechal 10,1996

```

```

create proc codigo
@pat varchar(30),
@mat varchar(30),
@nom varchar(30)
as
declare @nr as int, @cod as char (4)
select @nr=MAX(substring(codigo,2,3))
from alumnos
if @nr is Null set @nr =0;
set @nr=@nr+1;
if @nr <=9
    set @cod='A00'+LTRIM(str(@nr))
else if @nr<=99
    set @cod='A0'+STR(@nr)
else if @nr<=999
    set @cod ='A'+STR(@nr)
;
insert into alumnos(codigo, paterno,materno,nombres)
values (@cod,@pat,@mat,@nom);

exec codigo 'alcantara','quispe','ronald'

select *
from alumnos

```

```

create proc ingresor2
@nom varchar(30),
@pre INT,
@sto INT
as
declare @nr as int, @codi as char (5)

select @nr=MAX(substring( cod ,3,4))
from dbo.productos

if @nr is Null set @nr =0;
set @nr=@nr+1;

if @nr <=9
    set @codi='P-00'+LTRIM(str(@nr))

else if @nr<=99
    set @codi='P-0'+LTRIM(str(@nr))

else if @nr<=999
    set @codi = 'P-'+LTRIM(str(@nr))
;
insert into productos(cod, nombre,precio,stock)
values (@codi,@nom,@pre,@sto);

```

```
exec ingresor2 'leche',12,30
```

```

select *
from productos

delete
from productos
where cod='P-0'

```

```

create proc ingreso_pro
@n varchar(20),
@p money,
@s int
as

declare @codi char(5)
declare @nr int

set @codi='p'

select @nr =COUNT(*)+1
from productos

if @nr <=9
set @codi=rtrim( @codi) + '0' + LTRIM( str(@nr ) )

else set @codi=rtrim( @codi)+ ltrim(str(@nr ))
;
insert into productos
value (@codi,@n,@p,@s);

```

```

create proc ingresoc
@n varchar(20),
@a varchar(30),
@d varchar(30),
@e varchar(35)

as
declare @cod char(8)
declare @nr int

select @nr =COUNT(*)+1
from cliente

if (@nr <=9)
set @cod=rtrim(@cod) +'000'+ LTRIM( str(@nr ) )
if (@nr <=99)
set @cod=rtrim(@cod) +'00'+ LTRIM( str(@nr ) )
else
set @cod=rtrim(@cod) +'0' +ltrim(str(@nr ))
;
insert into cliente
values (@cod,@n,@a,@d,@e);

```

ingresoc 'ronlad','alcantar','la paz','ron'

```

create proc participatnesr
@n varchar(30),
@a varchar(30),
@g char(1)

as
declare @c char(7)
declare @nr int

select @nr=COUNT(*)
from participantes
where genero=@g

if @nr<9
set @c=left(@n,2)+left(@a,2)+'00'+ ltrim(str(@nr+1))

else if @nr<99
set @c=left(@n,2)+left(@a,2)+'0'+ ltrim(str(@nr+1))

else
set @c=left(@n,2)+left(@a,2)+ ltrim(str(@nr+1))

insert into participantes
values (@c, @n,@a,@g)

```

```

participatnesr 'flor','alcantara','f'

select *
from participantes

```

crear un procedimiento almacenado para actualizar los precios de los productos solo a los productos de la categoria ingresada (ingresar el porcentaje de aumentos de los precios) solamente se actualizaran los productos continuos

```

create proc actu_p
@c int,
@incr float

as --inicio del cuerpo del procedimiento

update Products
set UnitPrice=UnitPrice+UnitPrice*@incr/100
where Discontinued=0 and CategoryID=@c

```

```

exec actu_p 1,5

select *
from dbo.Products

```

```

create procedure insetar_alumnos
@n varchar(30) ,
@a varchar(30)

as
declare @nr int
declare @c char(3)

--select @nr=MAX(substring(codigo,2,2))+1
select @nr=MAX(right(codigo,2))+1
from alumno2

if @nr<=9

    set @c='p0' + LTRIM(str(@nr))
else
    set @c='p' + LTRIM(str(@nr))

insert into alumno2
values (@c,@n,@a)

```

```

exec insetar_alumnos 'ana','lopez'
select *
from alumno2

```

CREAR UNA PROCEDIMIENTO ALMACENADO EN LA TABLA DE EMPLEADOS, SORTEAR

```

CREATE proc azar
as
declare @num int
set @num = cast(RAND()*10 as int)
/*CAST permite convertir valores a otro tipo de dato */
select EmployeeID,LastName,FirstName
from dbo.Employees
where EmployeeID=@num

```

```
exec azar
```

	EmployeeID	LastName	FirstName
1	2	Fuller	Andrew

```
CREATE PROC SORTEO
```

```
@NUM INT
```

```
AS
```

```

select top (@NUM) customerid,companyname, NEWID()
FROM Customers
ORDER BY NEWID()
/* NEWID genera una valor único */

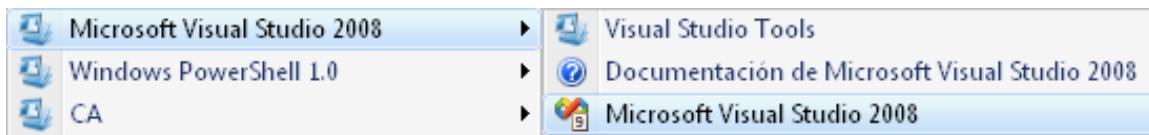
```

```
EXEC SORTEO 3
```

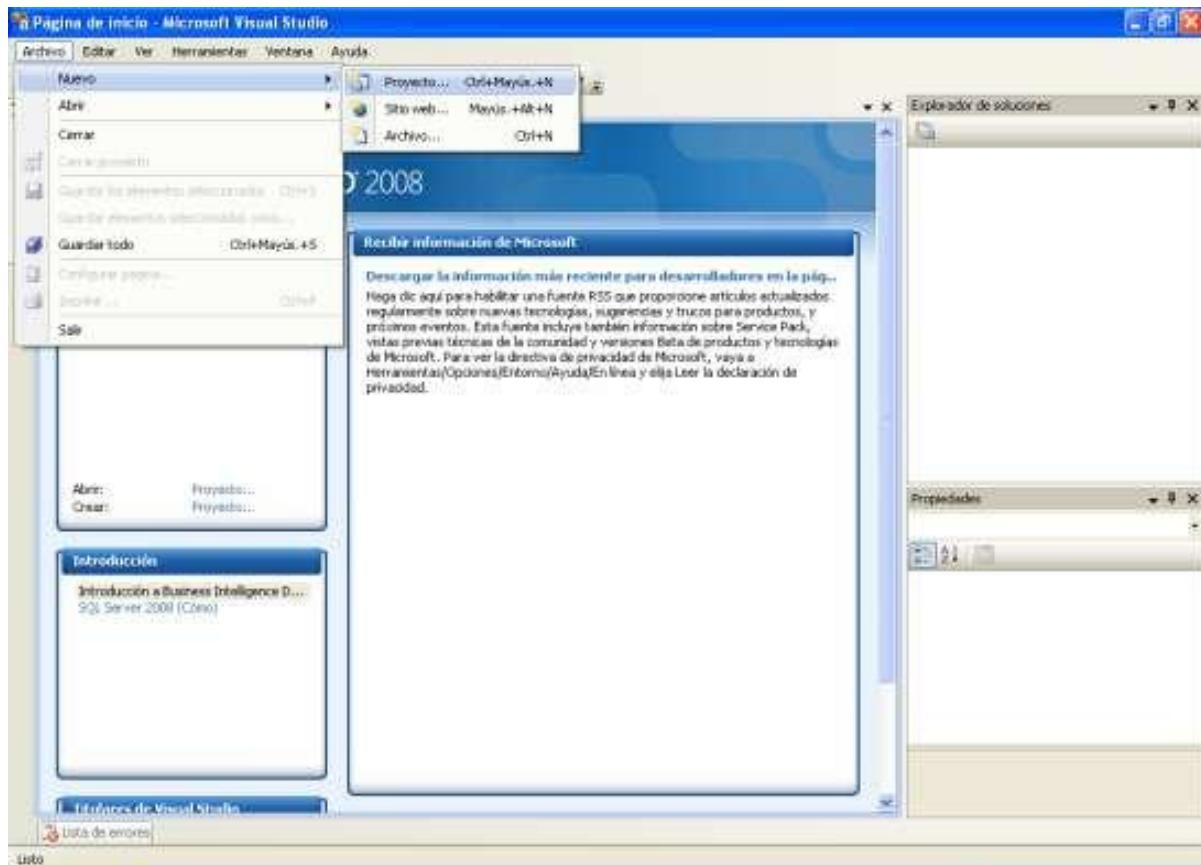
	customerid	companyname	[Sin nombre de columna]
1	LILAS	LILA-Supermercado	9AE49145-26A3-4E0C-817C-0240A572BBB4
2	MEREP	Mère Paillarde	70612FD4-6E41-48E8-B038-02D15EC058F8
3	NORTS	North/South	A1CBB097-CB44-4AC6-AE3D-05F8E05CBFAC

CREACIÓN DE REPORTING SERVICES EN SQL SERVER 2008

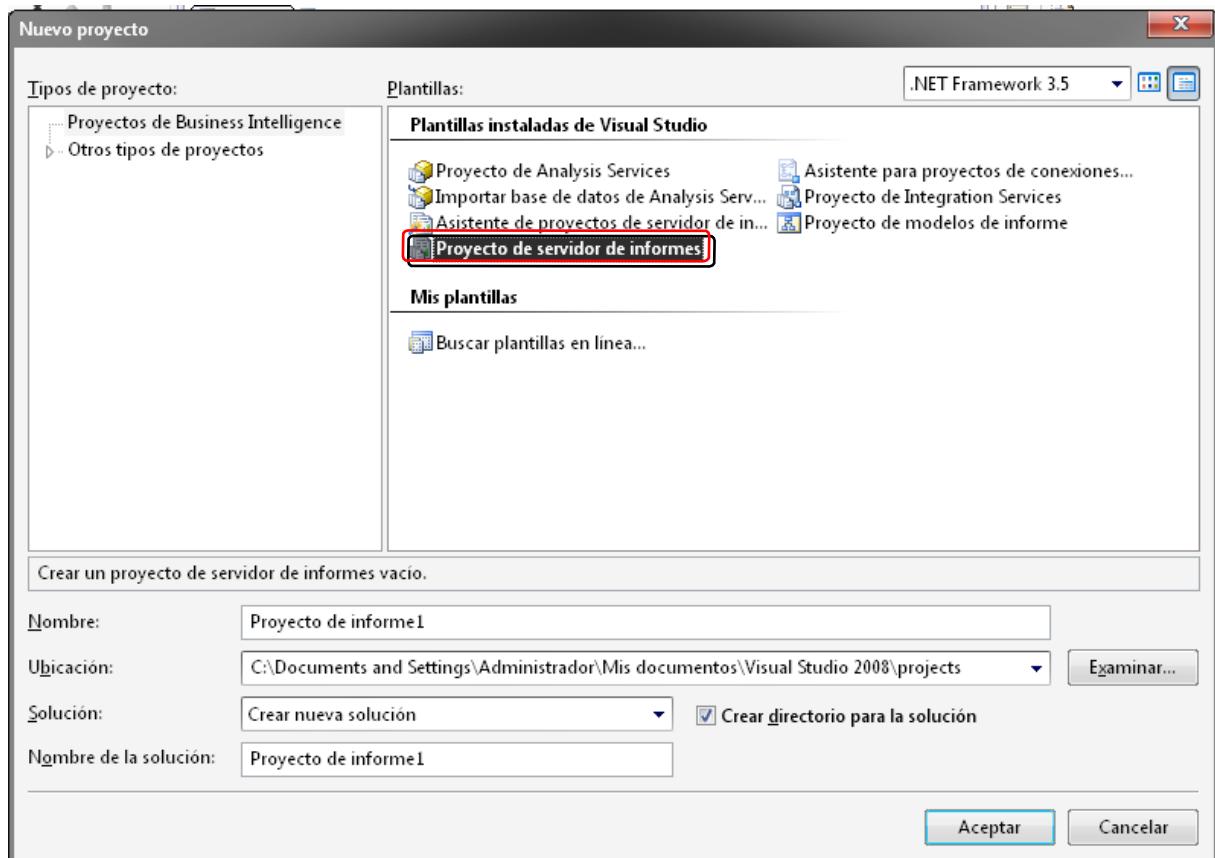
INICIAMOS “Microsoft Visual Studio 2008”



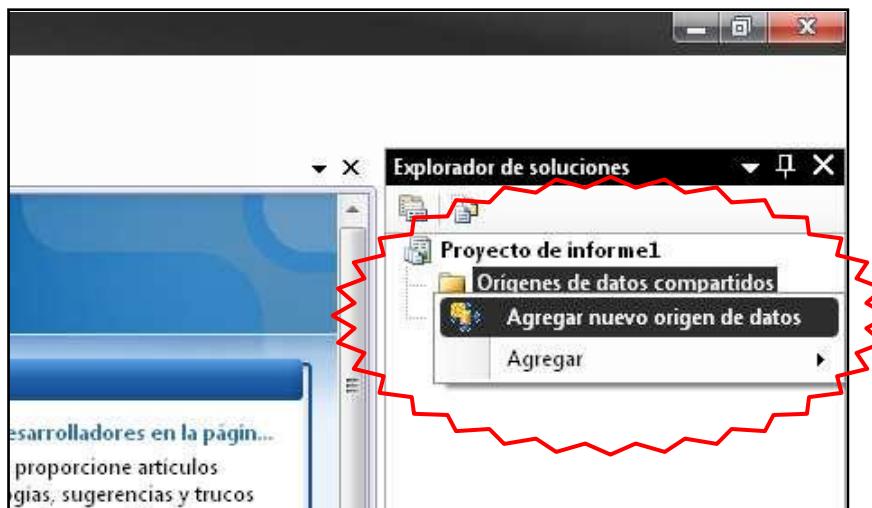
Luego creamos un nuevo proyecto



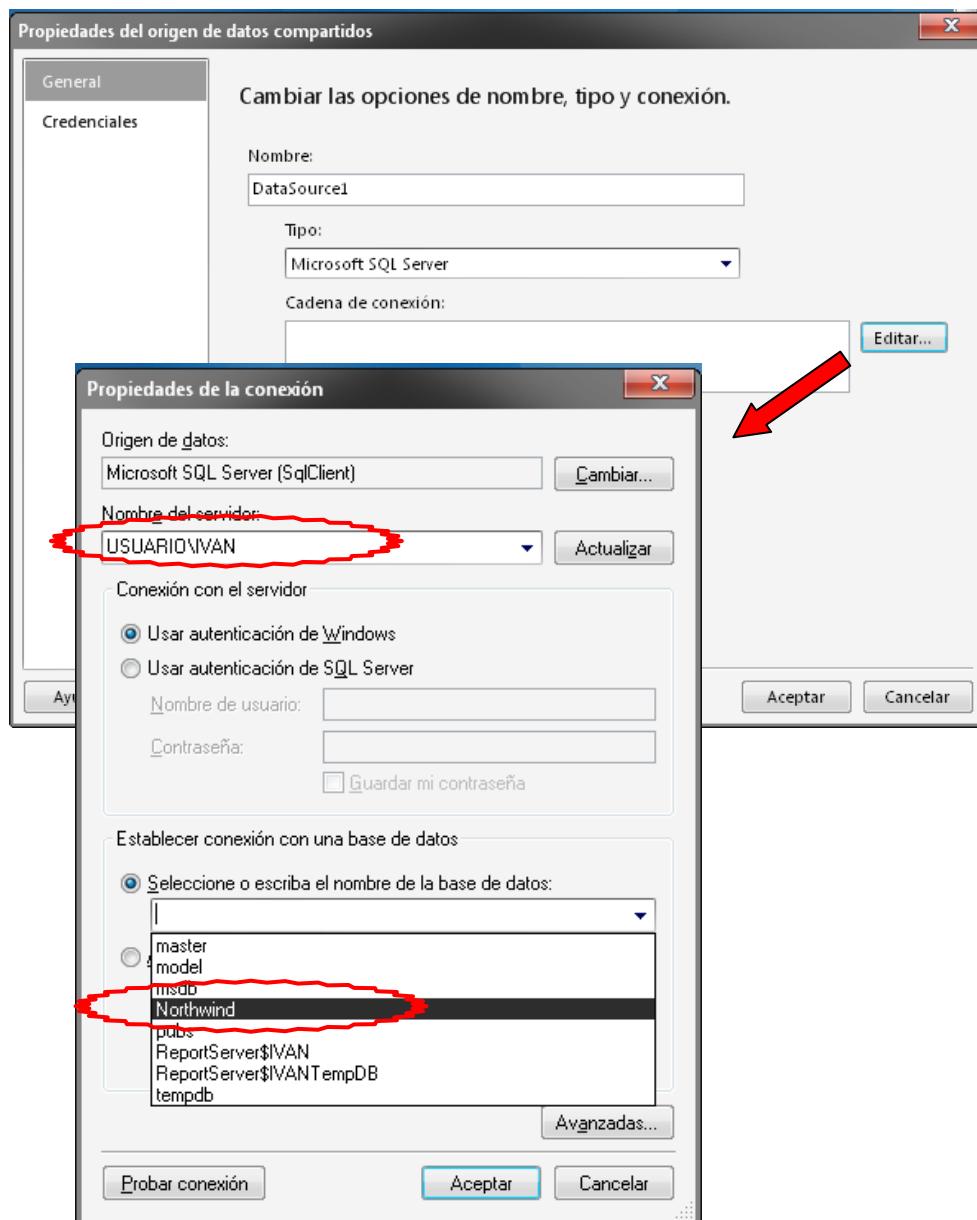
Luego escogemos la opción de proyecto de servidor de informe



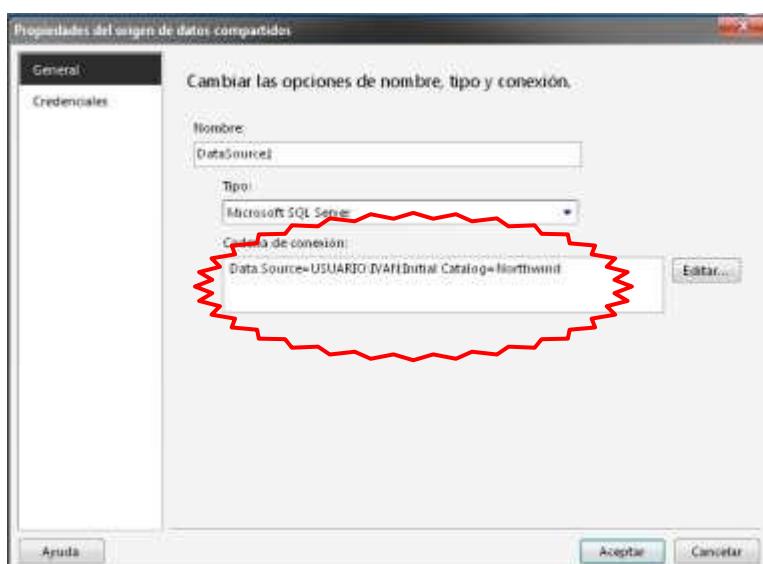
Crear un origen de datos compartidos:

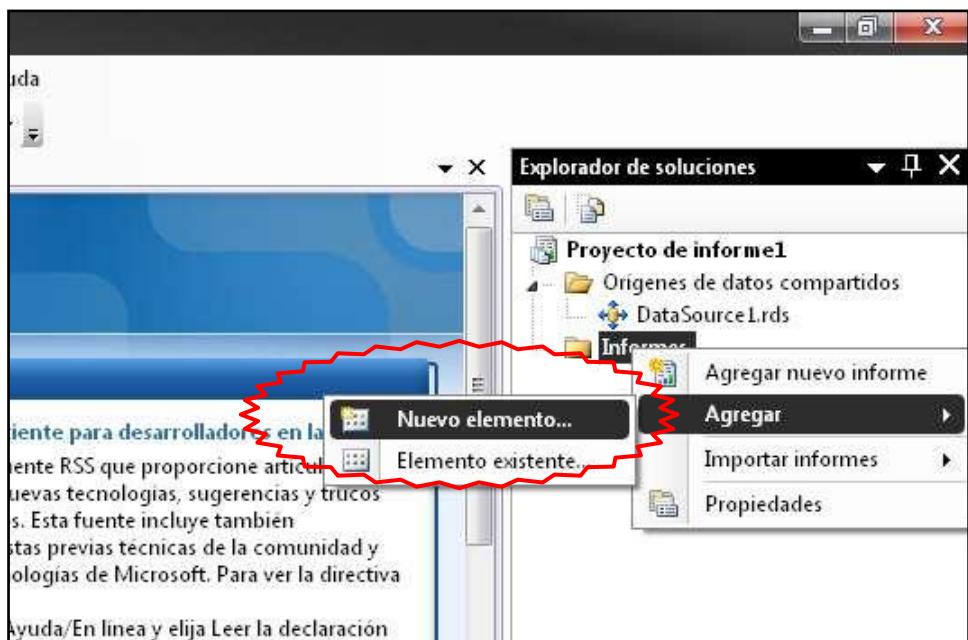


Escogemos el nombre del servidor (USUARIO\IVAN) y escogeos la base de datos



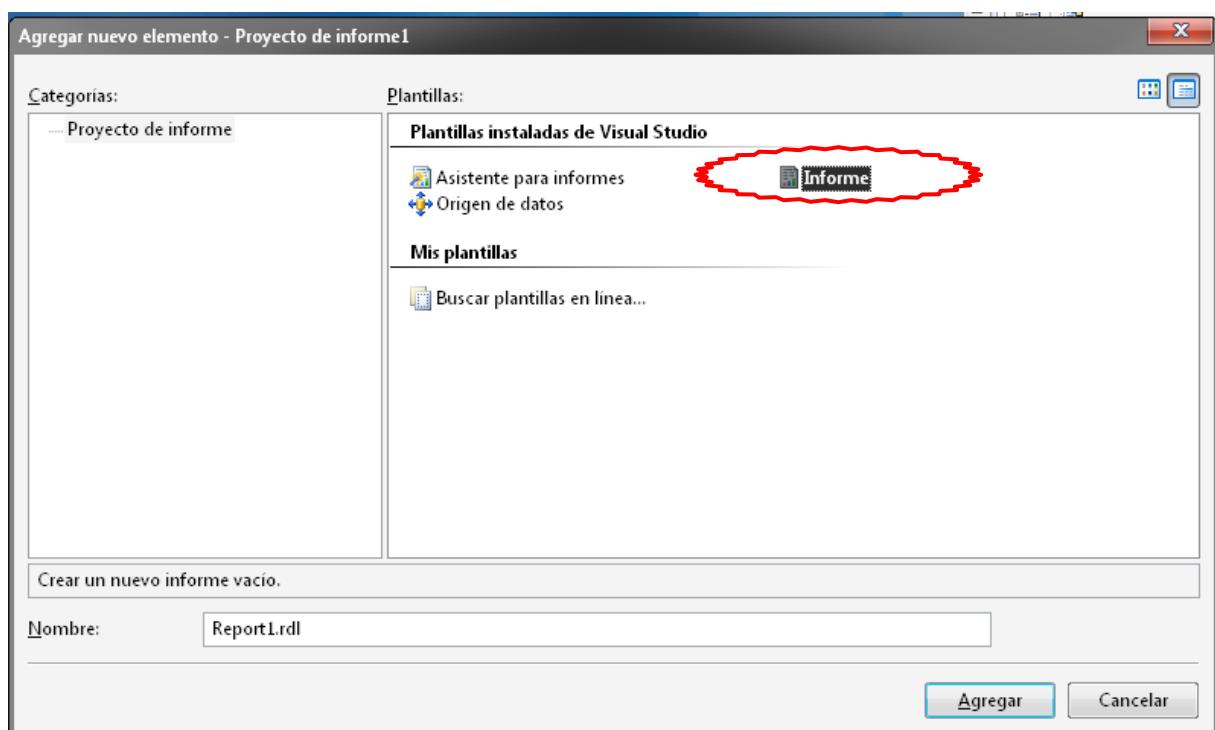
Como demos ver ya se tiene la cadena de selección



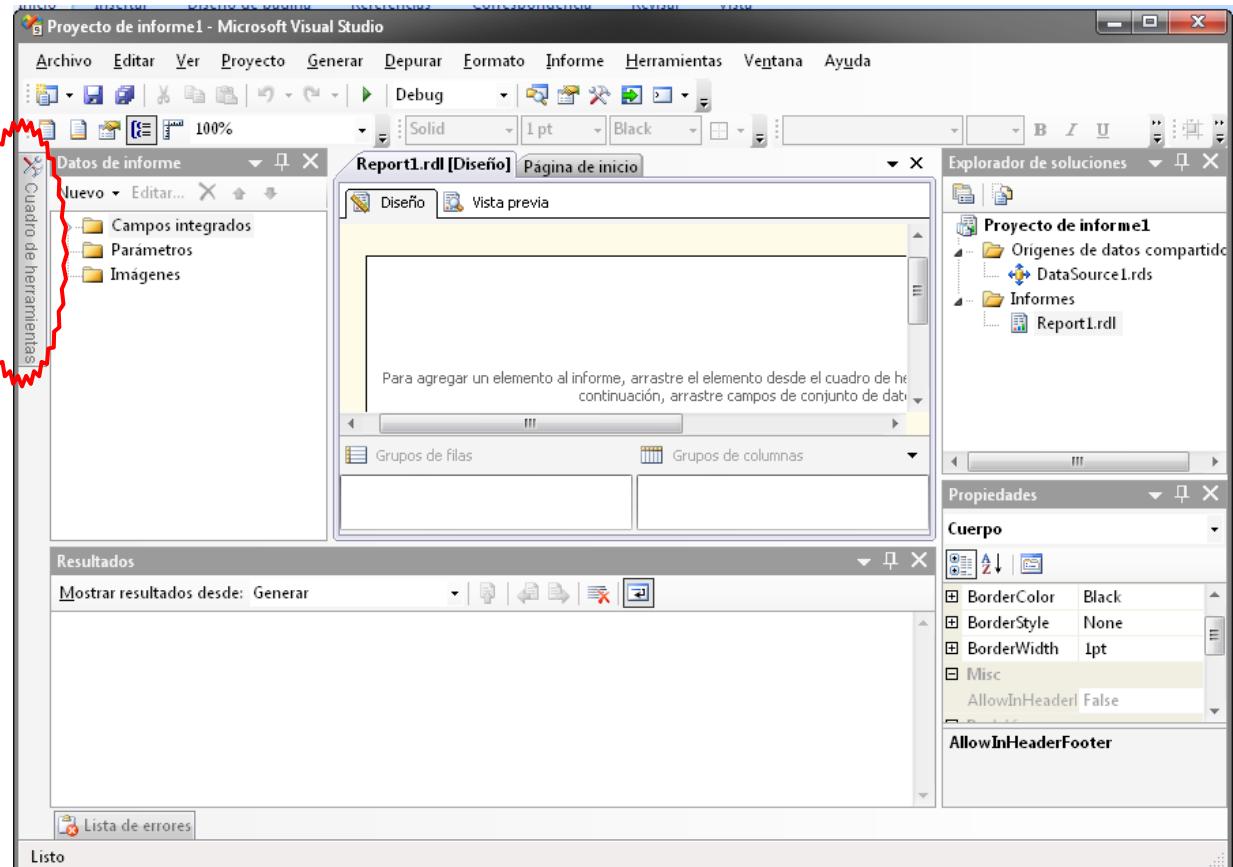


En la opción informe hacemos clic derecho y en el menú contextual escogemos agregar, nuevo elemento.

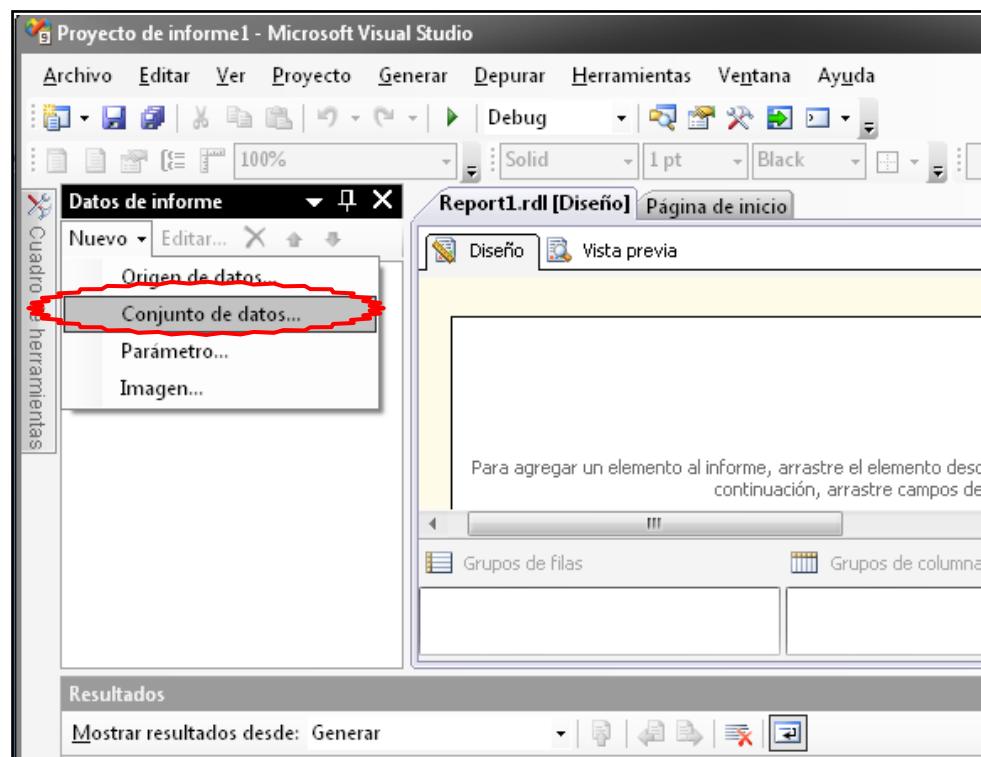
En la nueva ventana escogemos informe



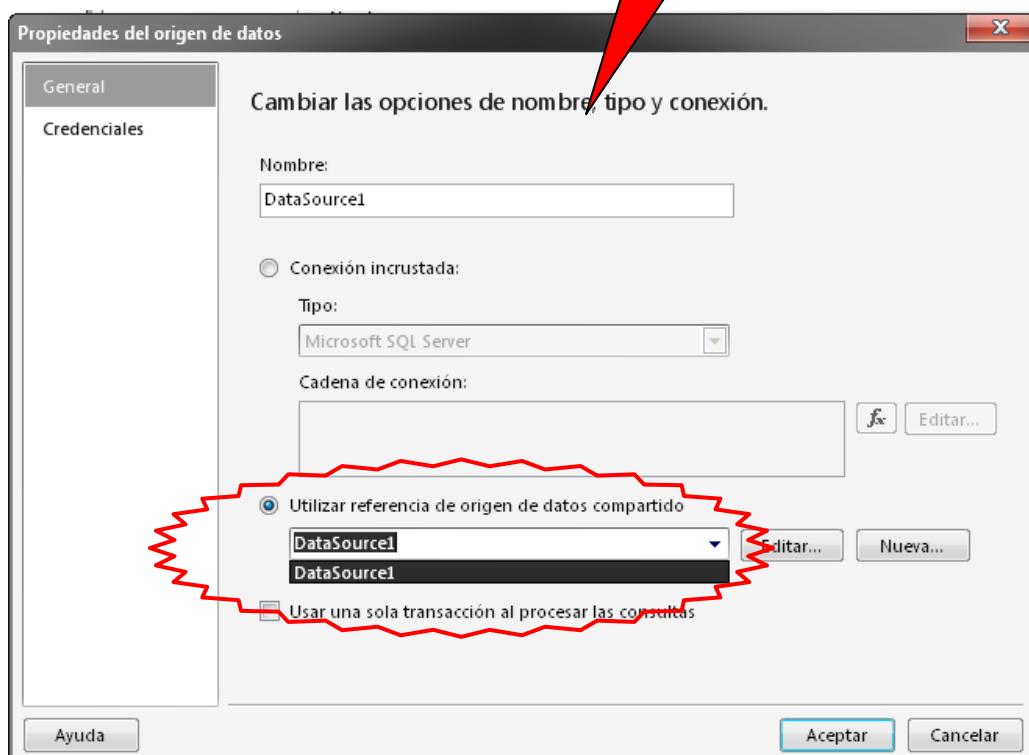
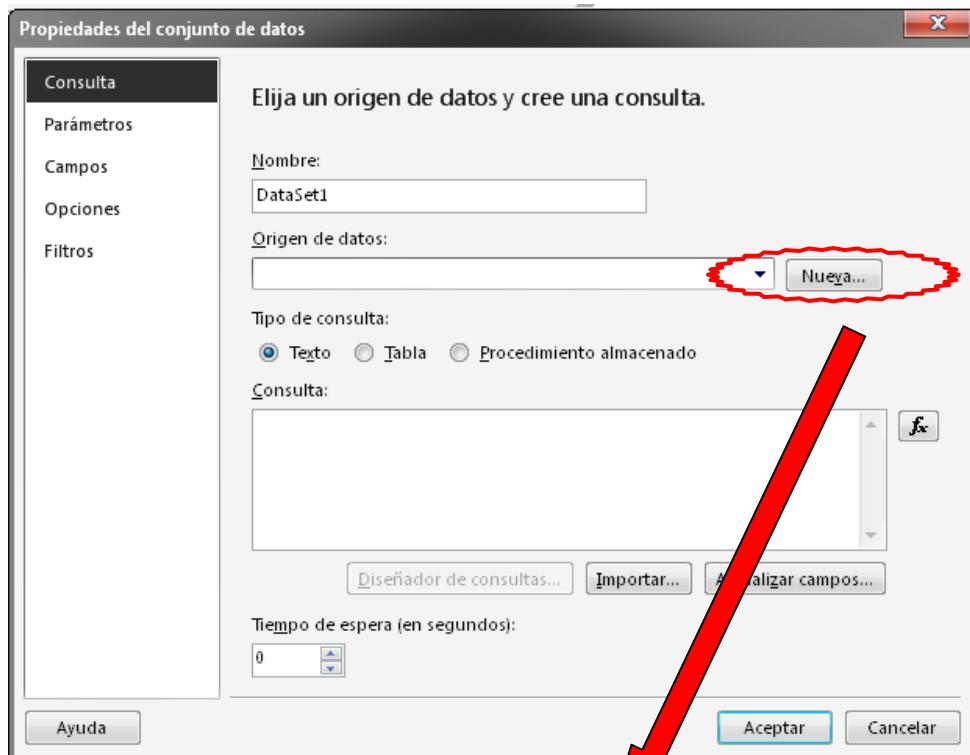
Esta será la nueva ventana, donde se diseñara el informe, en la parte izquierda encontramos el panel de herramientas en donde encontramos los controles para diseñar el informe



A continuación agregamos el conjunto de datos (procedimientos almacenados, en el caso que lo tuviésemos)

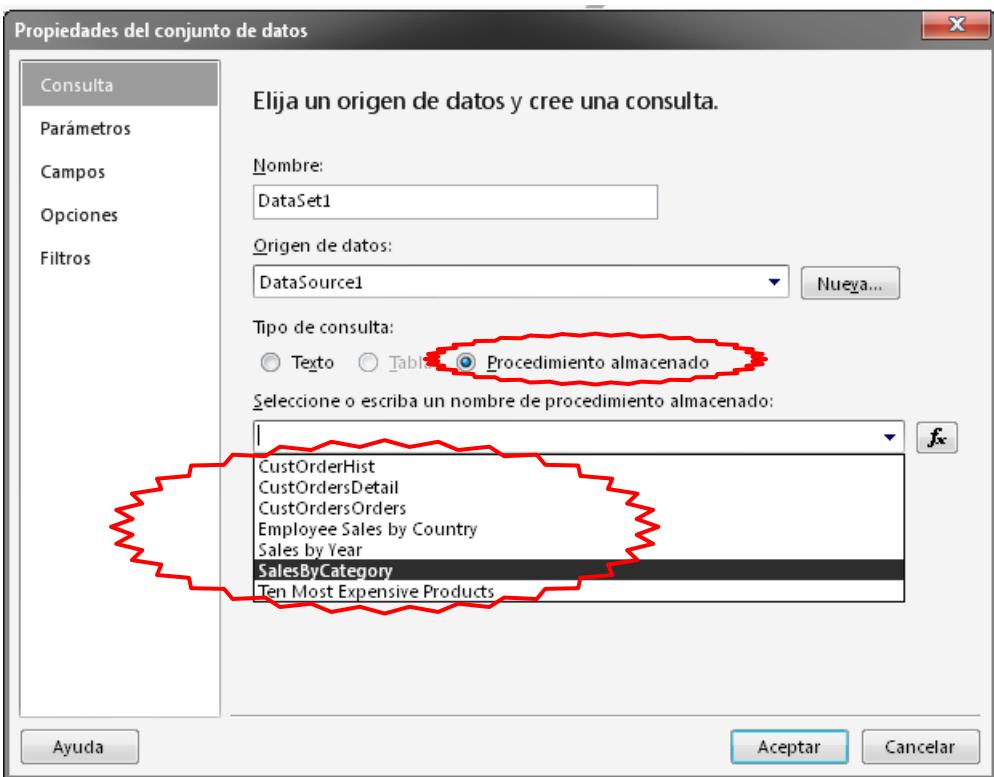


Escogemos el origen de datos ya creado

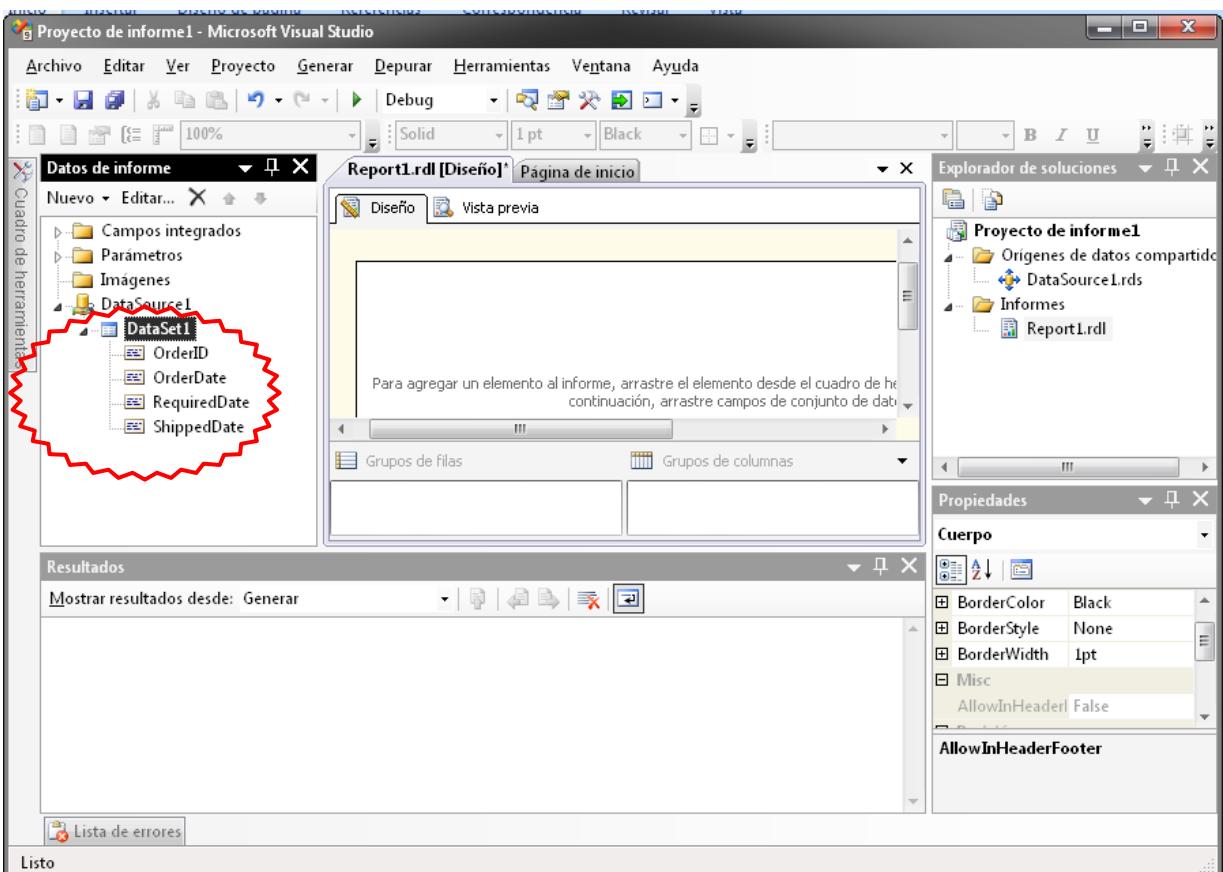


Utilizamos: utilizar referencia de origen de datos compartido (esto quiere decir que podemos utilizar un mismo origen de datos para diversos informes) _ luego aceptamos

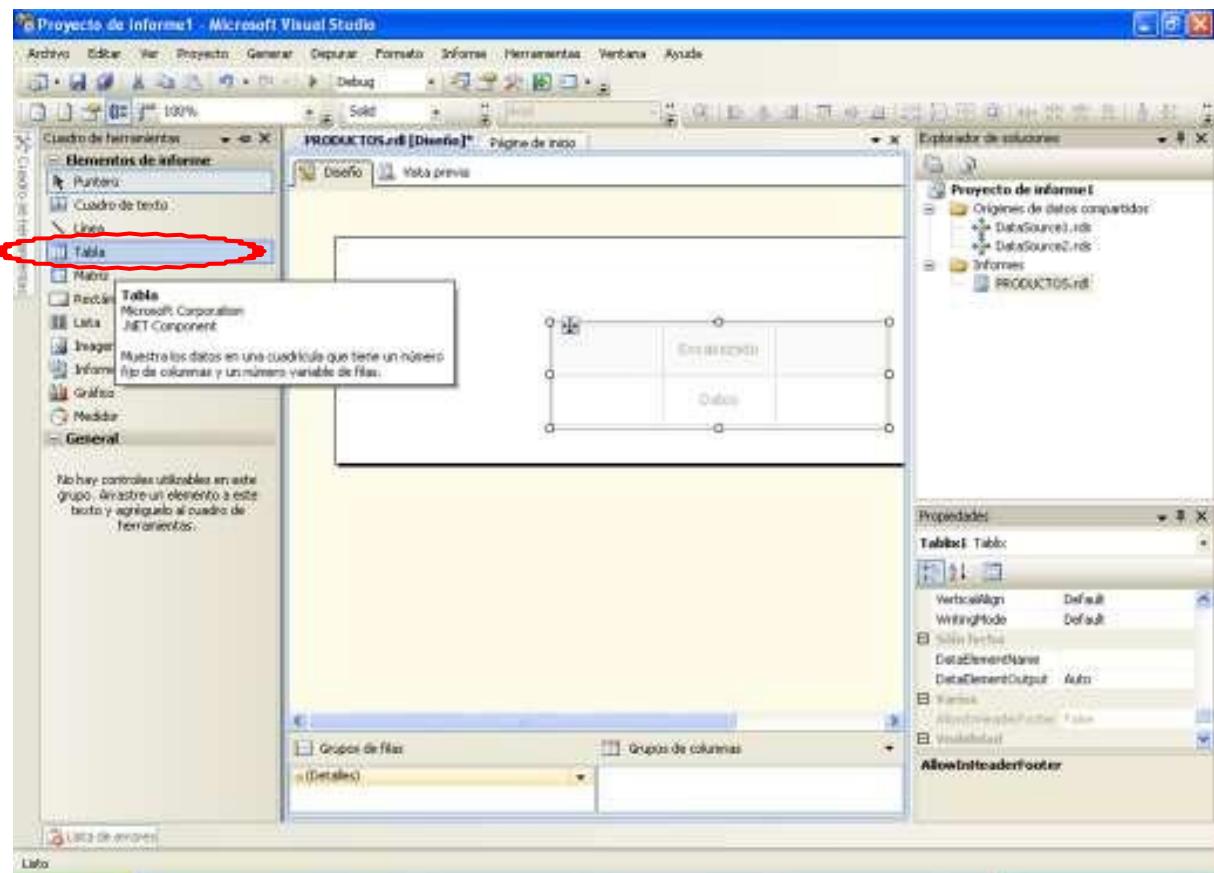
A continuación escogemos procedimiento almacenado (en caso que se trabajo con procedimientos almacenados) y buscamos el procedimiento ya almacenado.



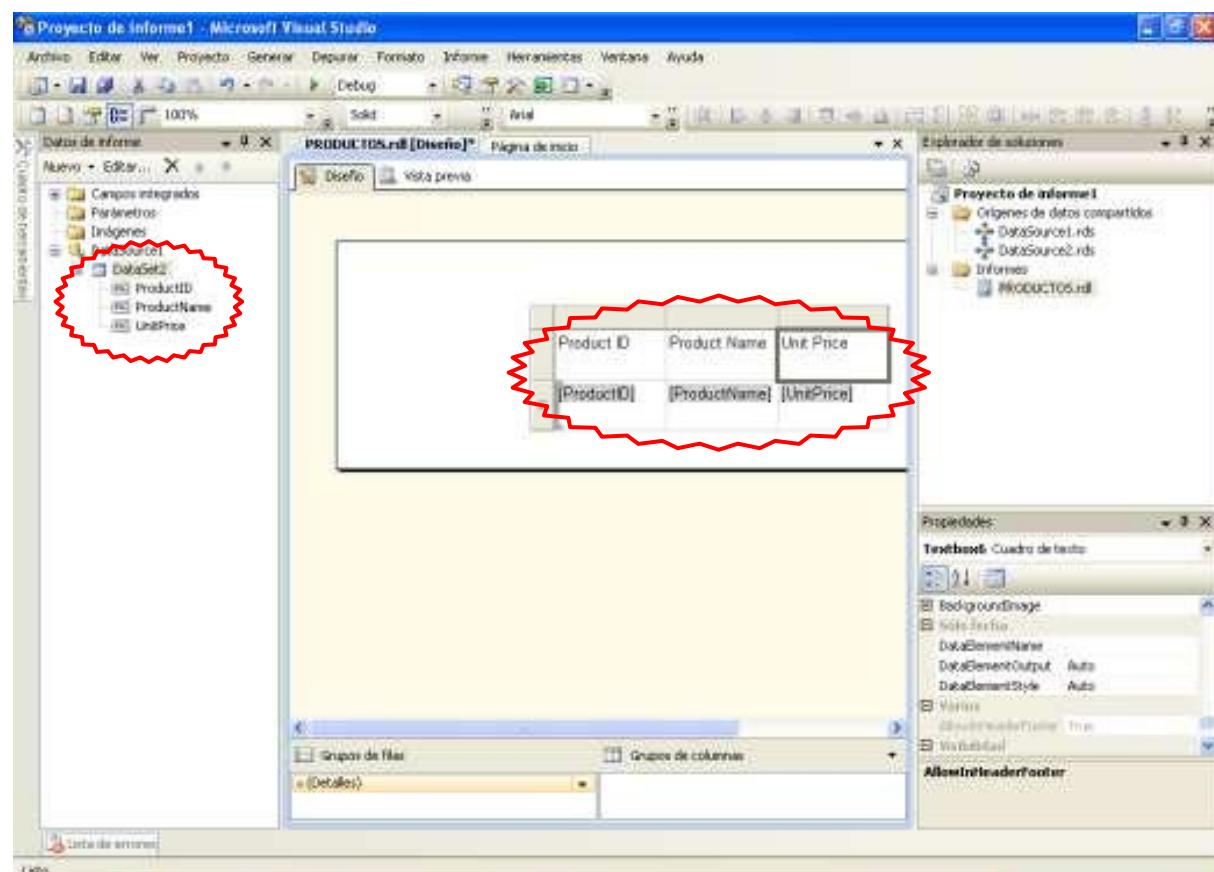
Al momento de seleccionar el procedimiento almacenado y aceptar, se mostraran las ventanas los campos que se optinen de dicho procedimiento almacenado.



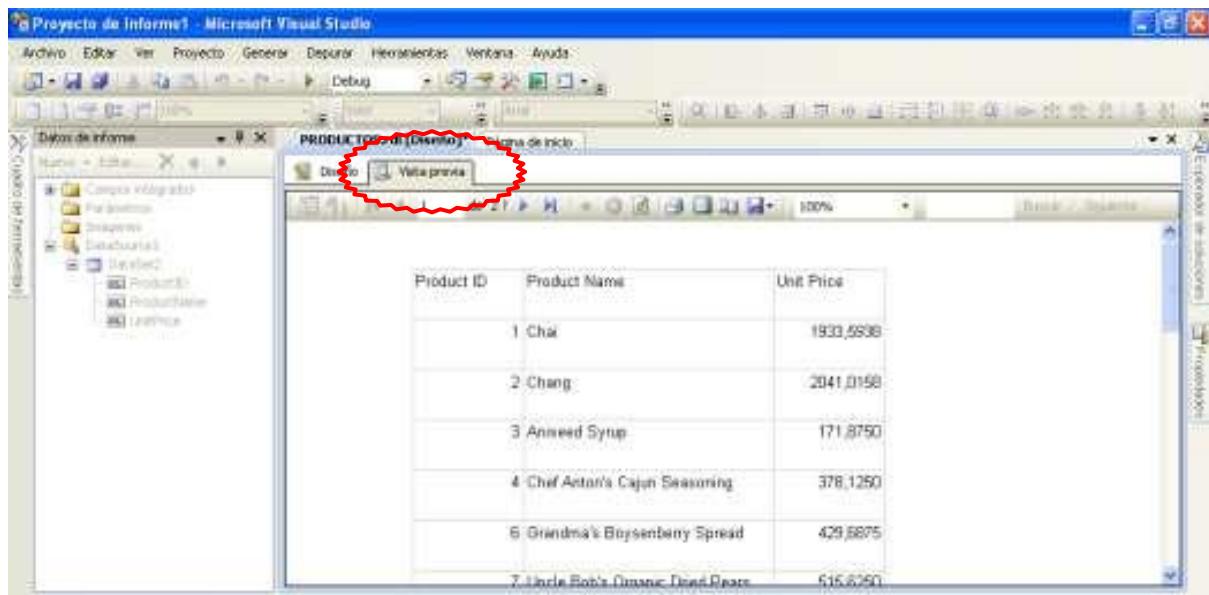
Luego podemos crear tablas u otros objetos para asignar los datos:



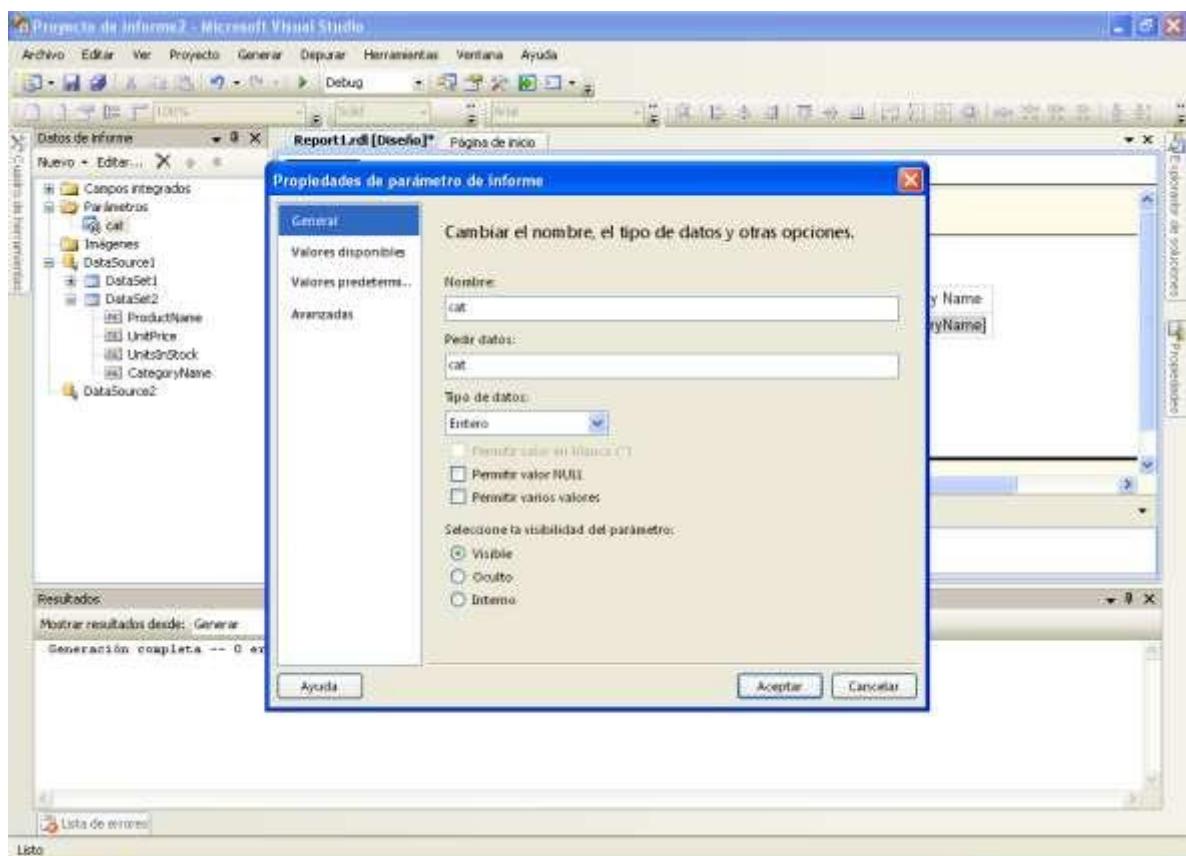
Se asigna los campos a las tablas arrastrándolos

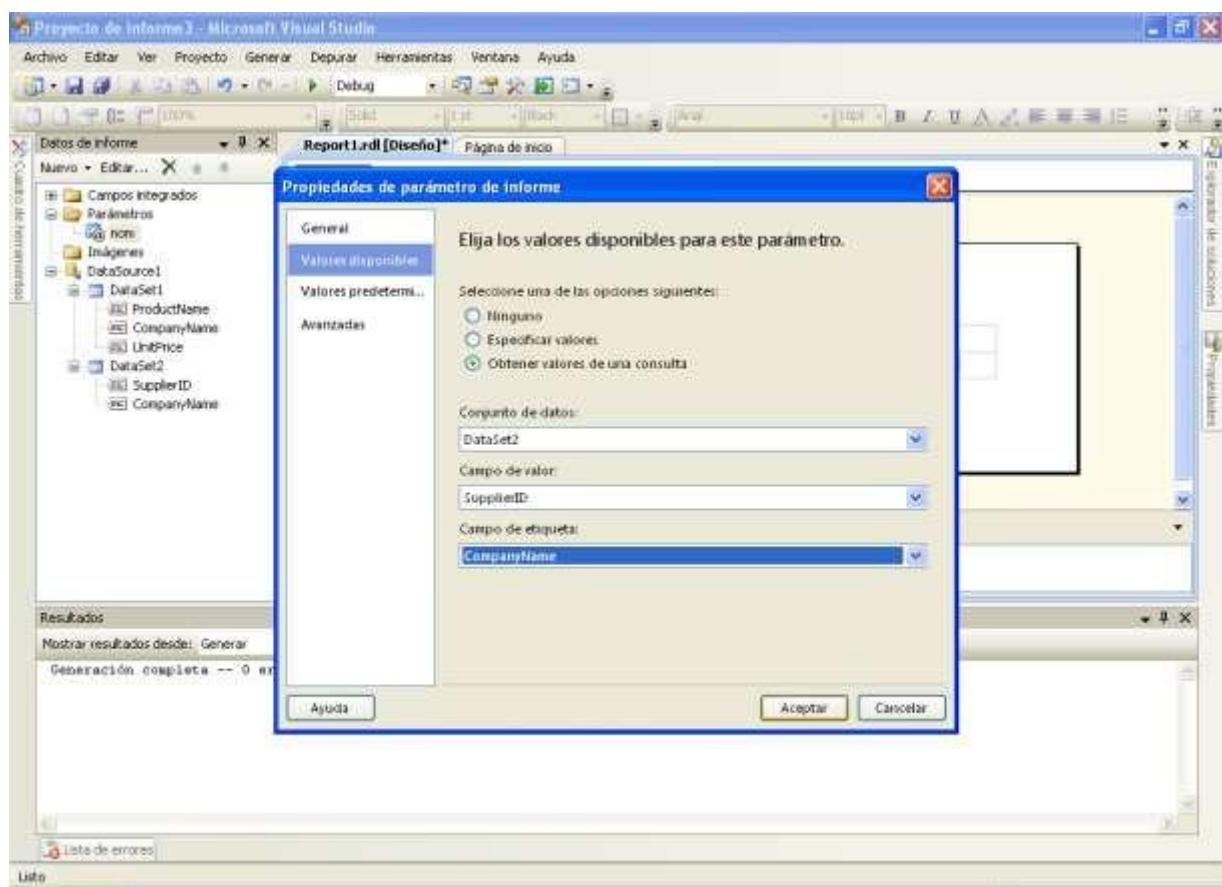


Para probar los informes se hace clic en la pestaña vista previa



Se pueden asignar valores para los parámetros





EJERCICIOS:

CREAR LOS SIGUIENTES INFORMES EN SQL SERVER

Ejercicio 1

mostrar en un objeto table, el nombre del cliente, el nro de orden, el nombre del producto, precio y cantidad comprada, ademas el nombre del proveedor.

Mostrar el nombre del cliente en un combobox o drop down para filtrar la información (cliente, orden, produ, detalle, proveedor)

Solución

--PROCEDIMIENTOS UTILIZADOS--

```
create proc ej1
@cod char(5)
as
select c.CompanyName as cliente, o.OrderID, p.ProductName,
p.UnitPrice, od.Quantity, s.CompanyName as proveedor
from Customers as c
inner join Orders as o on o.CustomerID=c.CustomerID
inner join [Order Details] as od on od.OrderID=o.OrderID
inner join Products as p on p.ProductID=od.ProductID
inner join Suppliers as s on s.SupplierID=p.SupplierID
where c.CustomerID=@cod
-----
select c.CustomerID,c.CompanyName
from Customers as c
```

cliente	Order ID	Product Name	Unit Price	Quant
Alfreds Futterkiste	10643	Rössle Sauerkraut	45,8000	
Alfreds Futterkiste	10643	Chartreuse verte	18,0000	
Alfreds Futterkiste	10643	Spegesild	12,0000	
Alfreds Futterkiste	10692	Vegie-spread	43,9000	
Alfreds Futterkiste	10702	Aniseed Syrup	10,0000	
Alfreds Futterkiste	10702	Lakkalikööri	18,0000	
Alfreds Futterkiste	10835	Raclette Courdavault	55,0000	
Alfreds Futterkiste	10835	Original Frankfurter grüne Soße	13,0000	
Alfreds Futterkiste	10952	Grandma's Boysenberry Spread	25,0000	
Alfreds Futterkiste	10952	Rössle Sauerkraut	45,8000	
Alfreds Futterkiste	11011	Escargots de Bourgogne	13,2500	
Alfreds Futterkiste	11011	Flotemysost	21,5000	

Ejercicio 2

precio y XXX

Productname	Unitprice	Categoryname	companyname

(producto,categori,prove)

Solución

--PROCEDIMIENTOS UTILIZADOS--

```
create proc ej2
@min int,
@max int
as
select p.ProductName, p.UnitPrice, ca.CategoryName, s.CompanyName
from Products as p
inner join Categories as ca on ca.CategoryID=p.CategoryID
inner join Suppliers as s on s.SupplierID=p.SupplierID
where p.UnitPrice>@min and p.UnitPrice<@max
```

Product Name	Unit Price	Category Name	Company Name
Chef Anton's Cajun Seasoning	22,0000	Condiments	New Orleans Cajun Delights
Chef Anton's Gumbo Mix	21,3500	Condiments	New Orleans Cajun Delights
Grandma's Boysenberry Spread	25,0000	Condiments	Grandma Kelly's Homestead
Queso Cabrales	21,0000	Dairy Products	Cooperativa de Quesos 'Las Cabras'
Tofu	23,2500	Produce	Mayumi's
Gustaf's Knäckebrot	21,0000	Grains/Cereals	PB Knäckebrot AB
Nord-Ost Matjeshering	25,8900	Seafood	Nord-Ost-Fisch Handelsgesellschaft mbH
Gravad lax	26,0000	Seafood	Svensk Sjöföda AB
Pâté chinois	24,0000	Meat/Poultry	Ma Maison
Sirop d'éable	28,5000	Condiments	Forêts d'éables
Louisiana Fiery Hot Pepper Sauce	21,0500	Condiments	New Orleans Cajun Delights
Flotemysost	21,5000	Dairy Products	Norske Meierier

Ejercicio 3

Proveedores (mostrar companyname)

Categoría (muestra categoryname)

Productname	Categoryname	Companyname	unitsinstock

Solución

--PROCEDIMIENTOS UTILIZADOS--

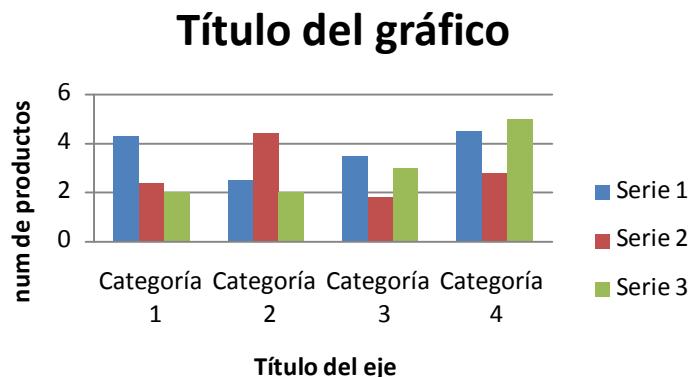
```
create proc ej3
@a int, --proveedor
@b int --categoria
as
select p.ProductName,ca.CategoryName,s.CompanyName,p.UnitPrice
from Products as p
inner join Categories as ca on ca.CategoryID=p.CategoryID
inner join Suppliers as s on s.SupplierID=p.SupplierID
where s.SupplierID=@a and ca.CategoryID=@b
```

The screenshot shows the Microsoft Visual Studio interface with the title bar "ivan - Microsoft Visual Studio". The menu bar includes Archivo, Editar, Ver, Proyecto, Generar, Depurar, Formato, Informe, Herramientas, Ventana, Ayuda. The toolbar has various icons for file operations like New, Open, Save, Print, etc. The status bar at the bottom shows "100%" and other settings. On the left, there's a "Cuadro de herramientas" (Toolbox) and a "Datos de informe" (Report Data) window showing fields like @a, @b, and parameters. The main workspace displays the report "ejercicio3.rdl [Diseño]". The "Diseño" tab is selected, showing a preview of the report with a table containing the following data:

Product Name	Category Name	Company Name	Unit Price
Chai	Beverages	Exotic Liquids	18,000
Chang	Beverages	Exotic Liquids	19,000

Ejercicio 4

Mostar en un gráfico y en una tabla el número de productos y categorías



Solución

-- PROCEDIMIENTOS UTILIZADOS --

```
create proc ej4
as
select count (p.ProductName)as cantidad,ca.CategoryName
from Products as p
inner join Categories as ca on ca.CategoryID=p.CategoryID
group by ca.CategoryName
```

A screenshot of Microsoft Visual Studio showing the report designer interface for "ejercicio4.rdl [Diseño]". The menu bar includes Archivo, Editar, Ver, Proyecto, Generar, Depurar, Formato, Informe, Herramientas, Ventana, Ayuda. The toolbar includes various icons for file operations and design tools. The left sidebar shows the "Cuadro de herramientas" (Toolbox) and the "Datos de informe" (Report Data) pane, which lists "Campos integrados", "Parámetros", "Imagenes", "DataSource1", and "DataSet1" containing "cantidad" and "CategoryName". The main workspace displays a 3D bar chart titled "Número de productos y categorías". The chart has "Cantidad" on the vertical axis (0 to 15) and categories on the horizontal axis: Beverages, Condiments, Confections, Dairy Products, Grains/Cereals, Meat/Poultry, Produce, and Seafood. A legend indicates that blue bars represent "cantidad".

Ejercicio 5

Año - (consulta de años de la bd) (distint)(matriz) (employ,orde)
Meses

	1	2	3	4	...	12
Juan león						
Pedro león						
Ana Díaz						
María Huamán						

Empleados

Nombre y apellidos

Número de órdenes realizadas en un mes del año seleccionado

Solución

--PROCEDIMIENTOS UTILIZADOS--

```
alter proc ej5
@a int
as
select distinct e.FirstName + ' ' + e.LastName as nombre,
COUNT(o.OrderID) as ordenes,
MONTH(o.OrderDate) as mes, YEAR(o.OrderDate)
from Employees as e
inner join Orders as o on o.EmployeeID=e.EmployeeID
where YEAR(o.OrderDate) = @a
group by o.OrderDate, FirstName, LastName, o.OrderID
-----
select distinct YEAR(orderdate)
from Orders
```

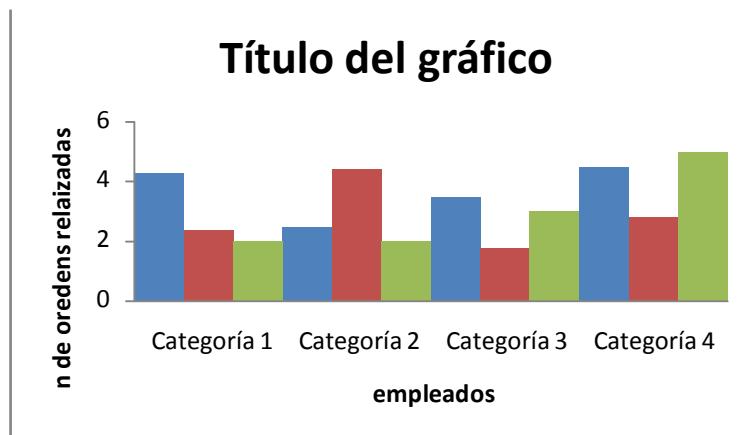
The screenshot shows the Microsoft Visual Studio interface with the title bar "ivan - Microsoft Visual Studio". The menu bar includes Archivo, Editar, Ver, Proyecto, Generar, Depurar, Formato, Informe, Herramientas, Ventana, Ayuda. The toolbar has various icons for file operations like New, Open, Save, Print, etc. The status bar at the bottom shows "100%" and other settings.

The main area shows the "Diseño" tab for "ejercicio5.rdl [Diseño]". On the left, the "Cuadro de herramientas" (Toolbox) is visible. In the center, there's a preview window showing a matrix report. The matrix has "nombre" in the first column, "mes" (month) in the second column, and employee names (Andrew Fuller, Anne Dodsworth, Janet Leverling, Laura Callahan, Margaret Peacock, Michael Suyama, Nancy Davolio, Robert King, Steven Buchanan) in the rows. The matrix cells contain the value "1", indicating one order per employee per month.

nombre	1	2	3	4	5
Andrew Fuller	1	1	1	1	1
Anne Dodsworth	1	1	1	1	
Janet Leverling	1	1	1	1	
Laura Callahan	1	1	1	1	1
Margaret Peacock	1	1	1	1	1
Michael Suyama	1	1	1	1	
Nancy Davolio	1	1	1	1	1
Robert King	1	1	1	1	1
Steven Buchanan	1	1	1	1	

Ejercicio 6

Mes Año



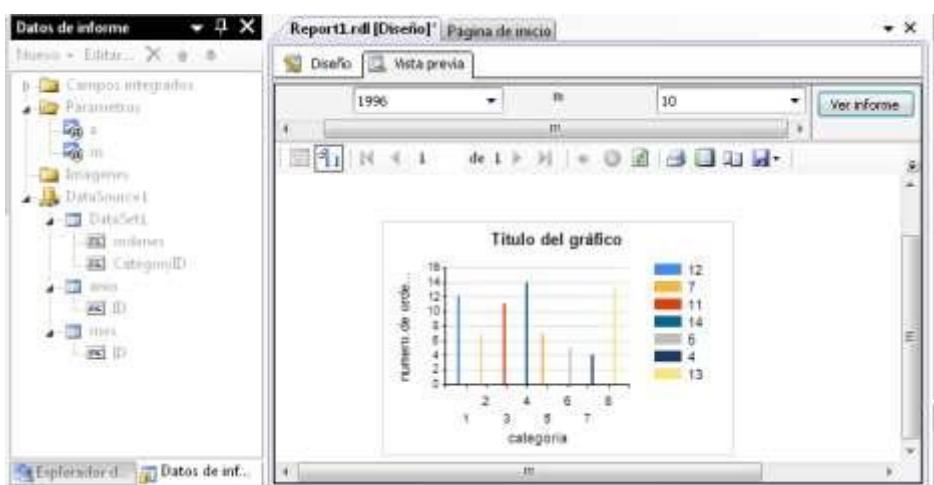
Solución

--PROCEDIMIENTOS UTILIZADOS--

```
alter proc ej6
@a int,
@m int
as
select c.CategoryID, count(o.OrderID) as ordenes
from employees as e
inner join Orders as o on o.EmployeeID=e.EmployeeID
inner join [Order Details] as od on od.OrderID=o.OrderID
inner join Products as p on p.ProductID=od.ProductID
inner join Categories as c on c.CategoryID=p.CategoryID
where YEAR(o.OrderDate)=@a and MONTH( o.OrderDate)=@m
group by c.CategoryID

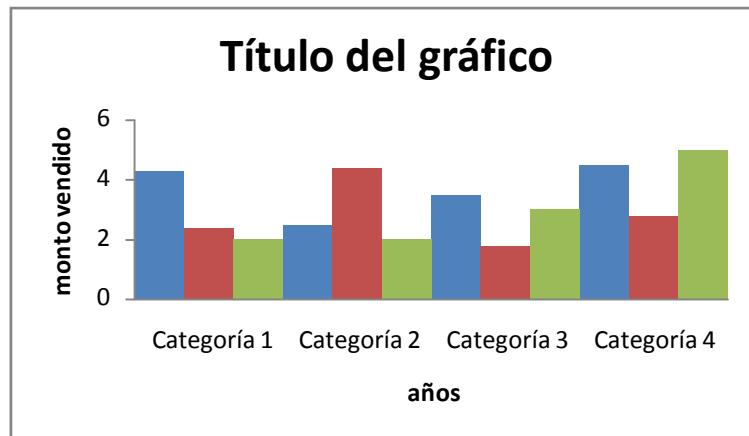
select distinct MONTH( o.OrderDate)
from Orders as o

select distinct year( o.OrderDate)
from Orders as o
```



Ejercicio 7

Empleados - (nombres y apellidos del empleado)



Solución

--PROCEDIMIENTOS UTILIZADOS--

```
create proc ej7
@nombre varchar(50)
as
select c.CategoryID, sum(o.OrderID) as monto
from employees as e
inner join Orders as o on o.EmployeeID=e.EmployeeID
inner join [Order Details] as od on od.OrderID=o.OrderID
inner join Products as p on p.ProductID=od.ProductID
inner join Categories as c on c.CategoryID=p.CategoryID
where @nombre= e.FirstName + ' ' + e.LastName
group by c.CategoryID

select e.FirstName + ' ' + e.LastName
from Employees as e
```

The screenshot shows the Report Designer interface with the following details:

- Datos de informe (Data):** Shows the data source structure:
 - Campos integrados (Integrated Fields)
 - Parámetros (Parameters): `@nombre`
 - Imágenes (Images)
 - DataSource1
 - DataSet1
 - CategoryID
 - monto
 - listaempleados
 - ID
- Report2.rdl [Diseño]:** The main report design window.
 - Nombre (Name):** Andrew Fuller
 - Ver informe (Preview):** A preview pane showing a bar chart.
- ReportLrdl [Diseño]:** A secondary report design window.
- Página de inicio (Home Page):** A link to the home page.

Ejercicio 8

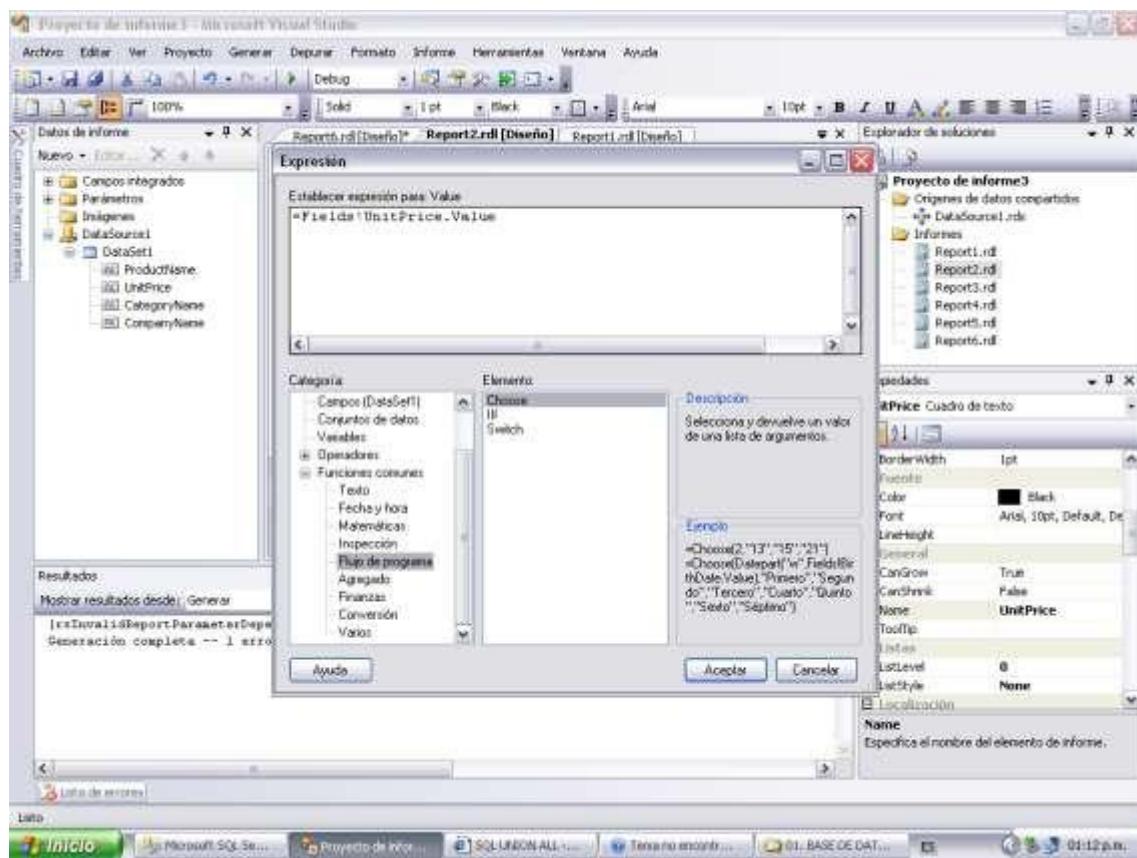
productname	Compani mane	Unitprice	unitsintock

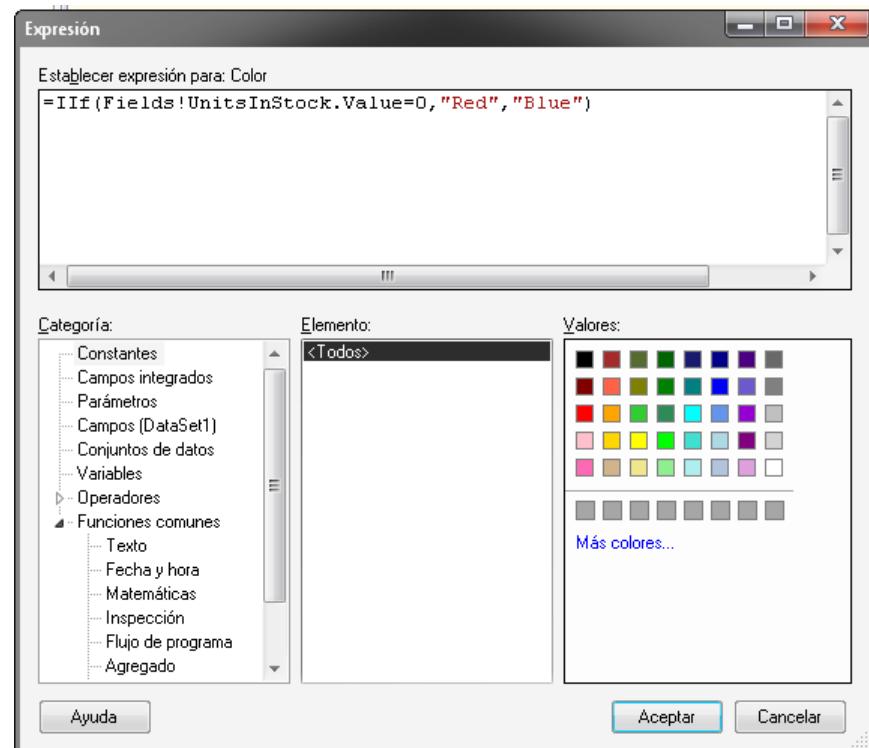
Stock cero color rojo y stock mayor k cero de color azul

Solución

--PROCEDIMIENTOS UTILIZADOS--

```
create proc ejer7
as
select p.ProductName, p.SupplierID,
p.CategoryID, p.UnitsInStock
from Products as p
```





Ejercicio 9

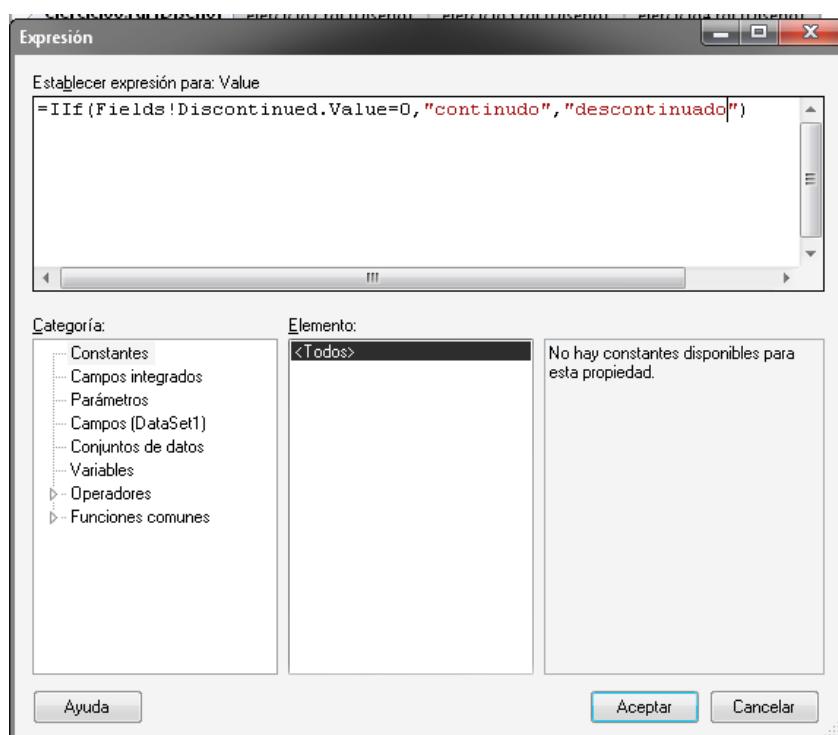
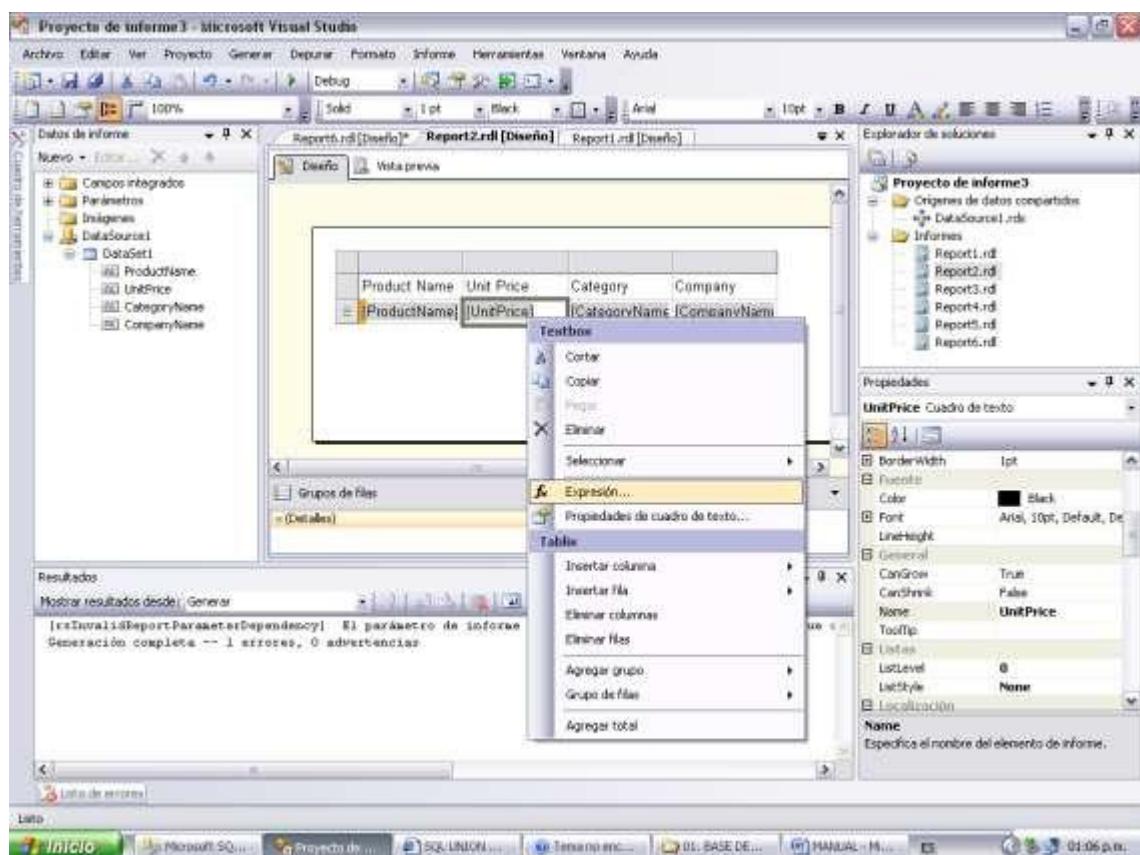
Productname	Unitprice	discontine

Mostrar discontuado o vigente

Solución

--PROCEDIMIENTOS UTILIZADOS--

```
create proc ejer8
as
select p.ProductName, p.UnitPrice, p.Discontinued
from Products as p
```



ivan - Microsoft Visual Studio

Archivo Editar Ver Proyecto Generar Depurar Formato Informe Herramientas Ventana Ayuda

100% Solid 1 pt Black Arial 10pt B I U A

Datos de informe ejercicio8.rdl [Diseño] ejercicio7.rdl [Diseño]* ejercicio5.rdl [Diseño]* ejercicio4.rdl [Diseño]*

Nuevo Editar... Diseño Vista previa

Campos integrados Parámetros Imágenes Data Source1 DataSet1 ProductName UnitPrice Discontinued

Product Name	Unit Price	Discontinued
Chai	18,0000	continuado
Chang	19,0000	continuado
Aniseed Syrup	10,0000	continuado
Chef Anton's Cajun Seasoning	22,0000	continuado
Chef Anton's Gumbo Mix	21,3500	descontinuado
Grandma's Boysenberry Spread	25,0000	continuado
Uncle Bob's Organic Dried Pears	30,0000	continuado
Northwoods Cranberry Sauce	40,0000	continuado
Mishi Kobe Niku	97,0000	descontinuado
Ikura	31,0000	continuado
Queso Cabrales	21,0000	continuado
Queso Manchego La Pastora	38,0000	continuado
Konbu	6,0000	continuado
Tofu	23,2500	continuado
Genen Shouyu	15,5000	continuado
Pavlova	17,4500	continuado
Alice Mutton	39,0000	descontinuado

PRACTICA CALIFICAD DE REPORTING SERVICES SQL SERVER 2008

Implementar los siguientes informes en reporting services, usando stored procedure.

1. Mostrar los empleados (nombres y apellidos) en una lista desplegable, para luego mostrar nro de la orden, la fecha de la orden, código del producto, nombre del producto, nombre de la categoría y el nombre de la compañía proveedora.
2. Crear un grafico que muestre el número de productos pedidos en cada categoría por cada año. Adicionalmente al costado del grafico debe salir una tabla con los mismos datos.
3. Crear un grafico que muestre los años, los empleados y el monto total ingresado pro los pedidos. Filtrar por años que se mostrara desde una lista desplegable.
4. Crear un gráfico que muestre los años, los meses y el número de productos pedidos de un cliente (solo considerar los productos vigentes). Escogiendo el cliente de una lista desplegable, donde se mostrara el nombre de la compañía del cliente.
5. En una matriz mostrar los clientes (filas), los meses (columnas) y en la intersección del numero de productos pedidos, solo considerar los productos vigentes.
6. En una matriz mostrar los años (filas), los meses (columnas) y en la intersección el monto vendió. Filtrar por empleado, mostrando en una lista desplegable el nombre y apellido de los empleados.
7. Mostrar el nombre del producto, nombre de la categoría y nombre de la empresa proveedora. Mostrar toda la fila de un registro en color amarillo si son de las categorías 1 y 2, en color verde las categorías 3 y 4, en color azul las categorías 5 y 6, y en color rojo el resto de categorías.
8. Mostrar por grupos con el icono de expandir y contraer (+), el número de las órdenes de compra y al expandir mostrar, el numero de la orden, código del producto, nombre del producto, precio, cantidad comprada, sub total (precio*cantidad) y en el pie del grupo el monto total pagado por orden de compra.

Ejercicio 1

```
alter proc ejer_1
@nombre varchar(50)
as
select o.OrderID , o.OrderDate ,
p.ProductID,p.ProductName , c.CategoryName , s.CompanyName
from Employees as e
inner join Orders as o on o.EmployeeID=e.EmployeeID
inner join [Order Details]as od on od.OrderID=o.OrderID
inner join Products as p on p.ProductID=od.ProductID
inner join Categories as c on c.CategoryID =p.CategoryID
inner join Suppliers as s on p.SupplierID=s.SupplierID
where FirstName +' '+ LastName =@nombre
-----
select FirstName +' '+ LastName
from Employees
```

The screenshot shows the Microsoft Visual Studio interface with the title bar "PracticaCalificada - Microsoft Visual Studio". The menu bar includes Archivo, Editar, Ver, Proyecto, Generar, Depurar, Herramientas, Ventana, Ayuda. The toolbar has various icons for file operations. The main window displays the Report Designer for "Report1.rdl [Diseño]*". On the left, the "Cuadro de herramientas" (Toolbox) is visible. The "Datos de informe" (Report Data) pane shows a hierarchy: Campos integrados, Parámetros, and a DataSource1 node containing a DataSet with columns: OrderID, OrderDate, ProductID, ProductName, CategoryName, and CompanyName. A parameter node "nombre" is expanded, showing its value is set to "Andrew Fuller". The report preview pane shows a table with the following data:

Order ID	Order Date	Product ID	Product Name	Category Name	Company Name
10265	25/07/1996 0:00:00	17	Alice Mutton	Meat/Poultry	Pavlova
10265	25/07/1996 0:00:00	70	Outback Lager	Beverages	Pavlova
10277	09/08/1996 0:00:00	28	Rössle Sauerkraut	Produce	Plutzer Lebens oßmärk
10277	09/08/1996 0:00:00	62	Tarte au sucre	Confections	Forêts d'éable
10280	14/08/1996 0:00:00	24	Guaraná Fantástica	Beverages	Refresc Americ LTDA
10280	14/08/1996 0:00:00	55	Pâté chinois	Meat/Poultry	Ma Ma
10280	14/08/1996 0:00:00	75	Rhönbräu Klosterbier	Beverages	Plutzer Lebens oßmärk
10295	02/09/1996 0:00:00	56	Gnocchi di nonna Alice	Grains/Cereals	Pasta E s.r.l.
10300	09/09/1996 0:00:00	66	Louisiana Hot Spiced Okra	Condiments	New Or Cajun D

Ejercicio 2

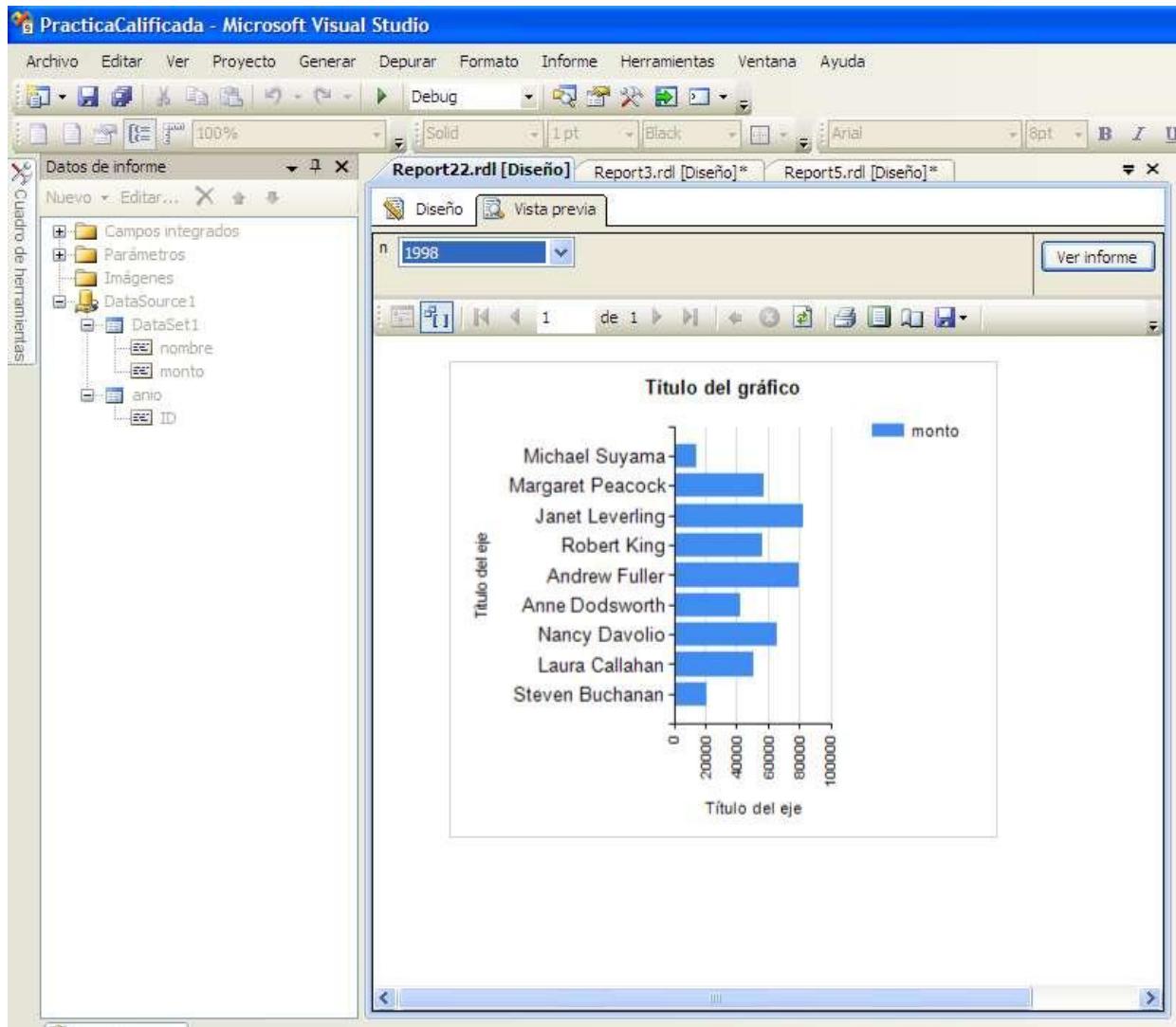
```
create proc ejer_2
@a int
as
select c.CategoryID, SUM(od.Quantity)
from Orders as o
inner join [Order Details]as od on od.OrderID=o.OrderID
inner join Products as p on p.ProductID=od.ProductID
inner join Categories as c on c.CategoryID =p.CategoryID
inner join Suppliers as s on p.SupplierID=s.SupplierID
where YEAR ( o.OrderDate) =@a
group by c.CategoryID
-----
select distinct YEAR (orderdate)
from Orders
```

The screenshot shows the Microsoft Visual Studio interface for report development. The title bar reads "PracticaCalificada - Microsoft Visual Studio". The menu bar includes Archivo, Editar, Ver, Proyecto, Generar, Depurar, Formato, Informe, Herramientas, Ventana, Ayuda. The toolbar has various icons for file operations. The main window displays the "Report2.rdl [Diseño]" tab. On the left, the "Datos de informe" pane shows a hierarchy: Campos integrados, Parámetros (with a parameter named 'a' set to 1996), Imágenes, and DataSource1 (DataSet1 with fields CategoryID, ID, anio, ID). The central pane shows the report design with a bar chart titled "Titulo del gráfico" and a table. The bar chart has categories labeled N, M, I, S, A, O and a legend entry "ID". The table on the right has columns "Category ID" and "ID" and contains the following data:

Category ID	ID
1	1842
2	962
3	1357
4	2086
5	549
6	950
7	549
8	1286

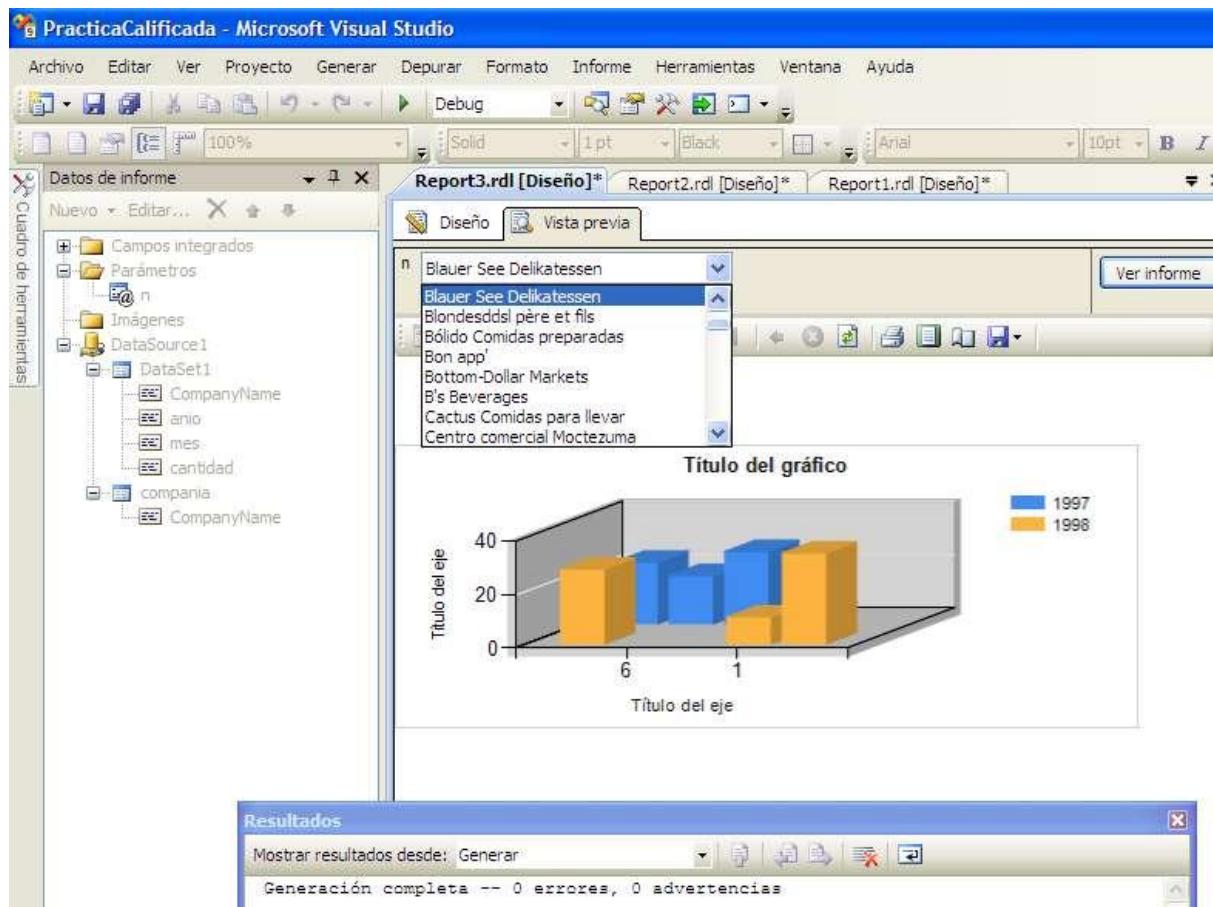
Ejercicio 3

```
alter proc ejer_22
@n int
as
select FirstName +' '+ LastName as nombre, SUM(od.Quantity*od.UnitPrice) as
monto
from Employees as e
inner join Orders as o on o.EmployeeID=e.EmployeeID
inner join [Order Details] as od on od.OrderID=o.OrderID
where YEAR ( o.OrderDate)=@n
group by FirstName, LastName
```



Ejercicio 4

```
create proc ejer_3
@n char(50)
as
select c.CompanyName, YEAR ( o.OrderDate) as anio, MONTH(o.OrderDate) as
mes ,
SUM (od.Quantity) as cantidad
from Customers as c
inner join Orders as o on o.CustomerID =c.CustomerID
inner join [Order Details] as od on od.OrderID=o.OrderID
where od.Discount=0 and @n=c.CompanyName
group by c.CompanyName,o.OrderDate
order by c.CompanyName
-----
select distinct CompanyName
from Customers
```



Ejercicio 5

```
alter proc ejer_4
as
select c.ContactName, MONTH (o.OrderDate) as mes , SUM(od.Quantity) as
cantidad
from Customers as c
inner join Orders as o on o.CustomerID =c.CustomerID
inner join [Order Details] as od on od.OrderID =o.OrderID
where Discount =0
group by c.ContactName, o.OrderDate
order by MONTH (o.OrderDate)
```

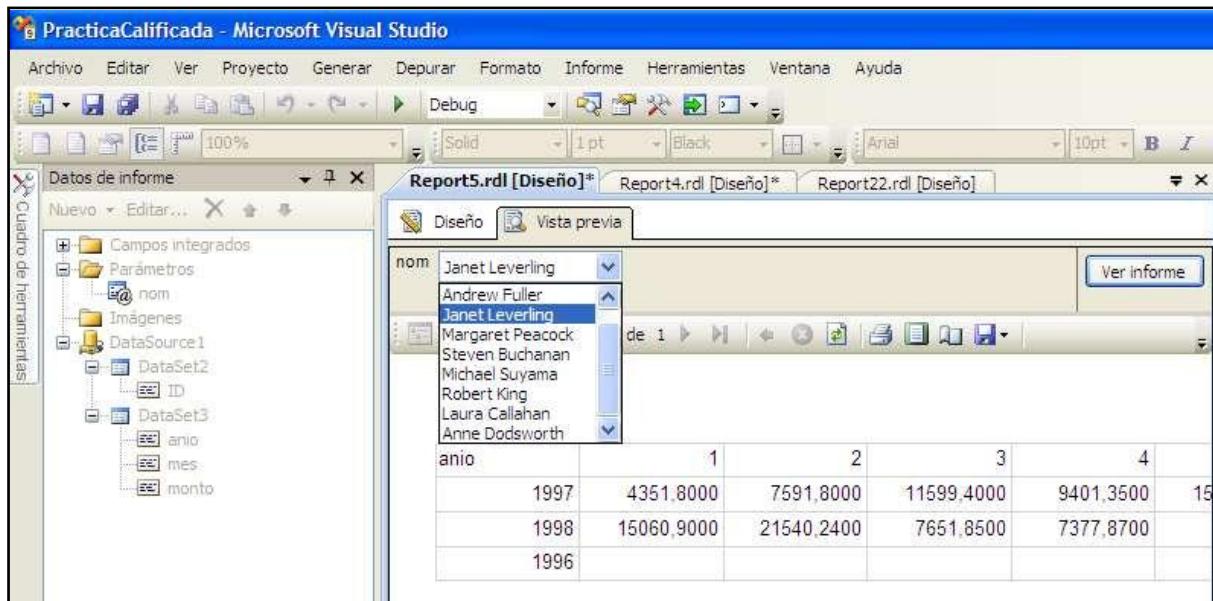
The screenshot shows the Microsoft Visual Studio interface for report design. The title bar reads "PracticaCalificada - Microsoft Visual Studio". The menu bar includes Archivo, Editar, Ver, Proyecto, Generar, Depurar, Formato, Informe, Herramientas, Ventana, Ayuda. The toolbar has various icons for file operations. The ribbon tabs include Solid, 1pt, Black, Arial, 10pt, B, I, U. The left sidebar shows the "Cuadro de herramientas" (Toolbox) and the "Datos de informe" (Report Data) window, which lists fields: Campos integrados, Parámetros, Imágenes, and a DataSource1 node containing a DataSet1 with fields ContactName, mes, and cantidad. The main workspace displays the "Report4.rdl [Diseño]" tab, showing a table preview with three columns. The table data is as follows:

Contact Name	1	2	3
Ann Devon	86		86
Paula Wilson	269	80	208
Roland Mendel	377	516	271
Felipe Izquierdo	113		98
Henriette Pfalzheim	60		
Lúcia Carvalho	10	90	95
Martine Rancé	51		42
Yvonne Moncada	24	5	101
Elizabeth Lincoln	65		181
Annette Roulet	81		
Aria Cruz	50		
Yoshi Latimer	22		
Jytte Petersen	85		
Pirkko Koskitalo	50	140	
Horst Kloss	194		195
Bernardo Batista	30		59
Paolo Accorti	2		
André Fonseca	34		

Ejercicio 5

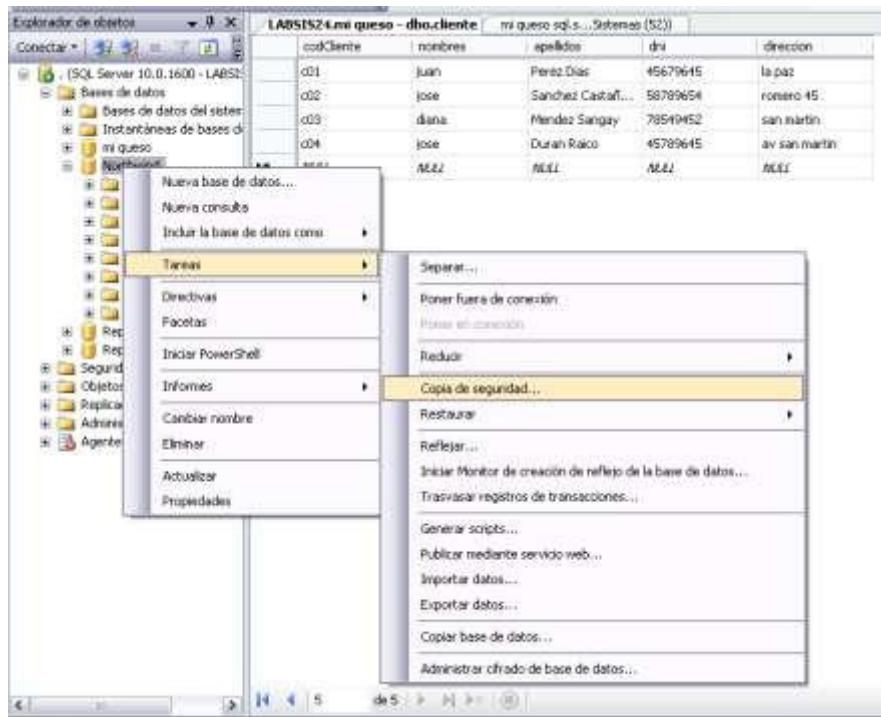
```
alter proc ejer_5
@nom varchar(50)
as
select year(o.orderdate) as anio, MONTH (o.OrderDate) as mes ,
SUM(od.Quantity*od.UnitPrice) as monto
from employees as e
inner join Orders as o on o.EmployeeID=e.EmployeeID
inner join [Order Details] as od on od.OrderID=o.OrderID
where Discount =0 and e.FirstName +' '+ e.LastName =@nom
group by e.FirstName,e.LastName, o.OrderDate
order by MONTH (o.OrderDate)

-----
select e.FirstName +' '+ e.LastName
from Employees as e
```

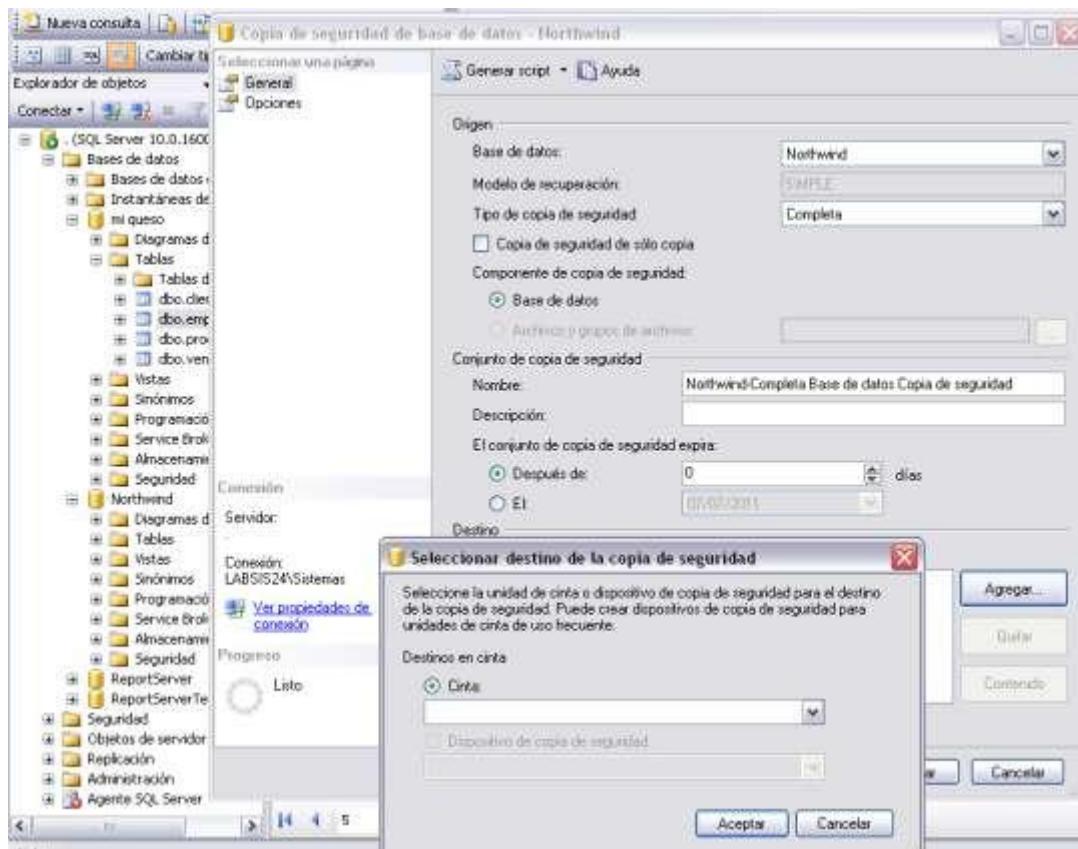


CREACION DE BACKUP

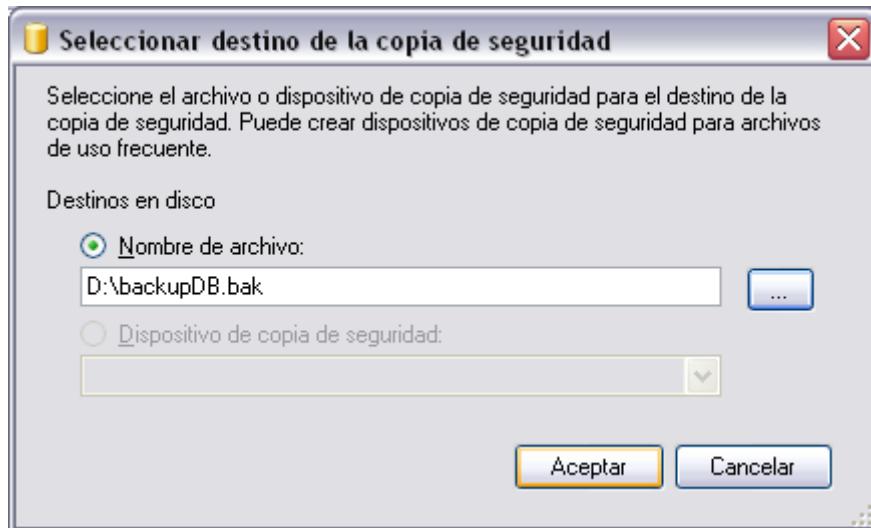
Escogemos la base de datos que queremos hacer la copia de seguridad y seguimos los pasos:



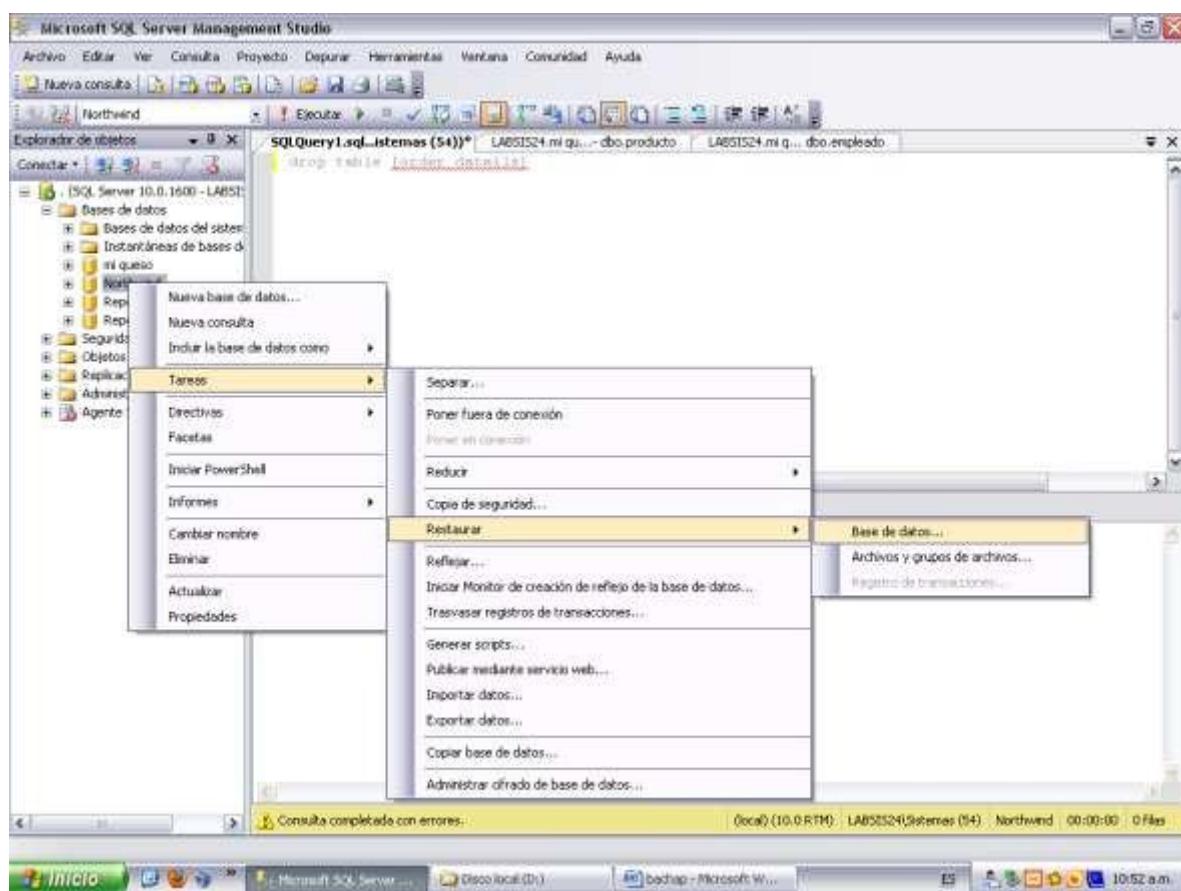
Luego especificamos en el medio que se hará la copia de seguridad.



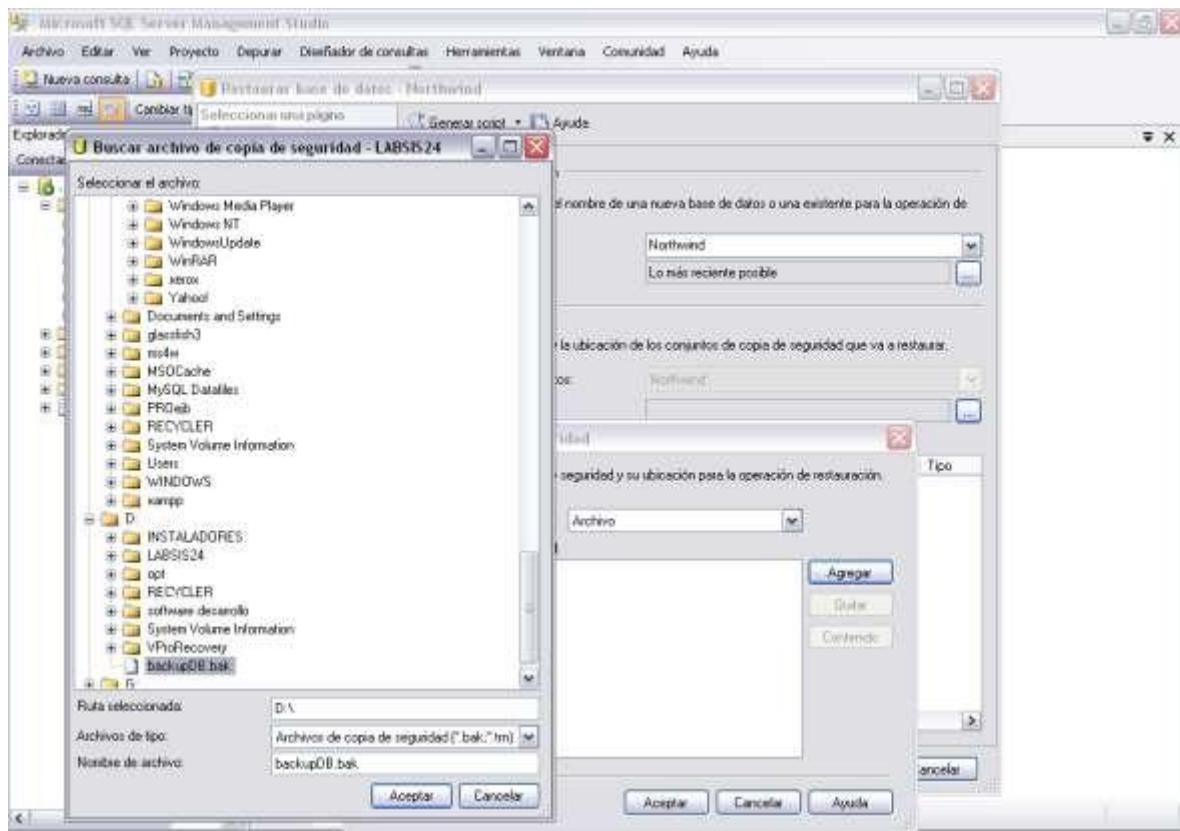
Cuando especificamos en disco se tiene que especificar la ruta en donde se guardara:



PARA RESTAURA UNA COPIA DE SEGURIDAD SE REALIZAN LOS SIGUIENTES PASOS:



Seleccionamos la ubicación de la base de datos.

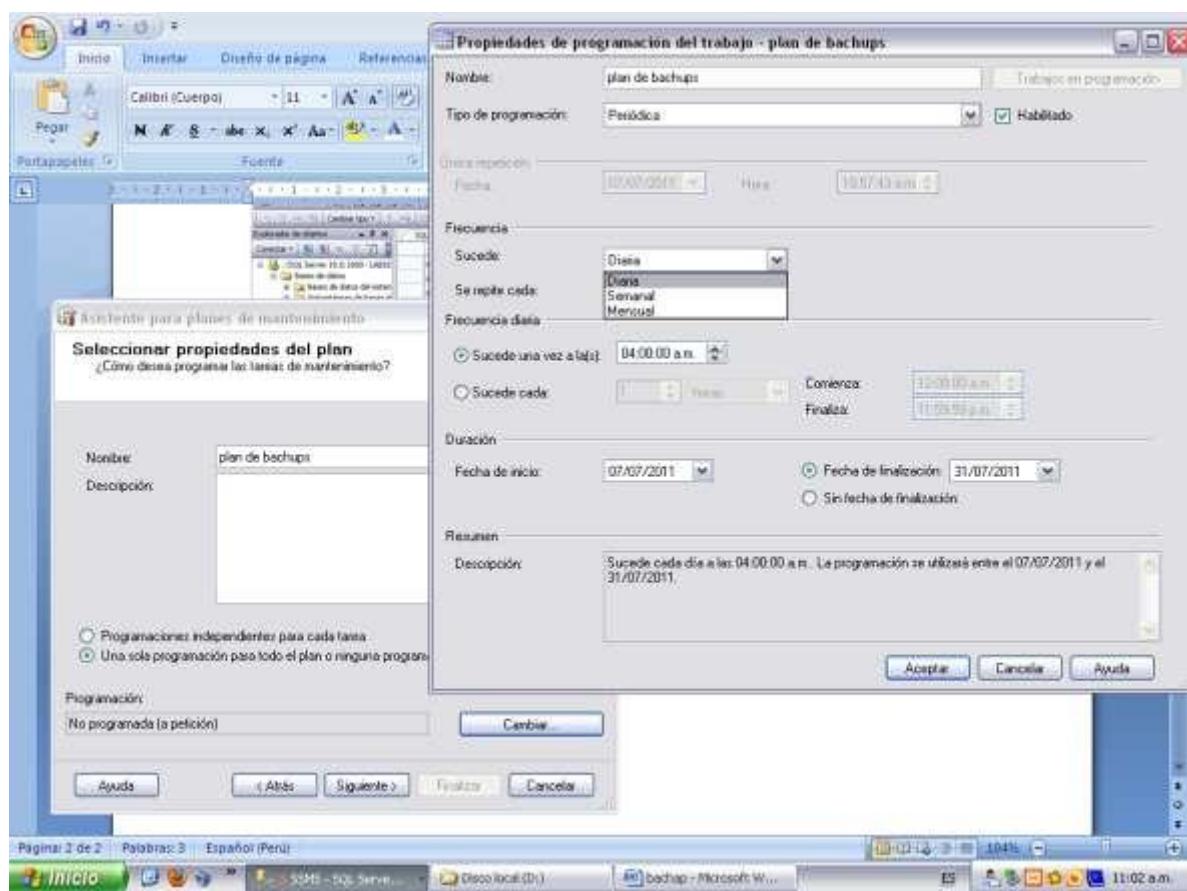
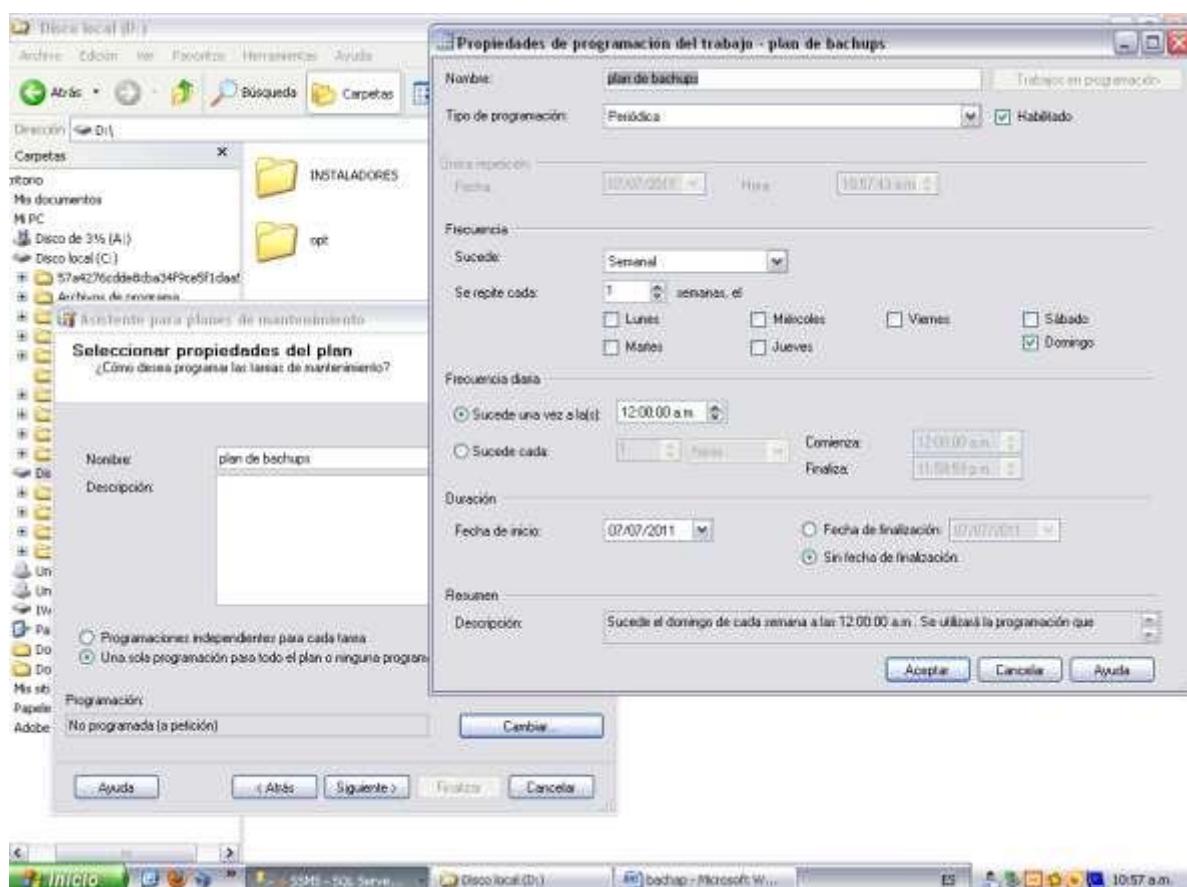


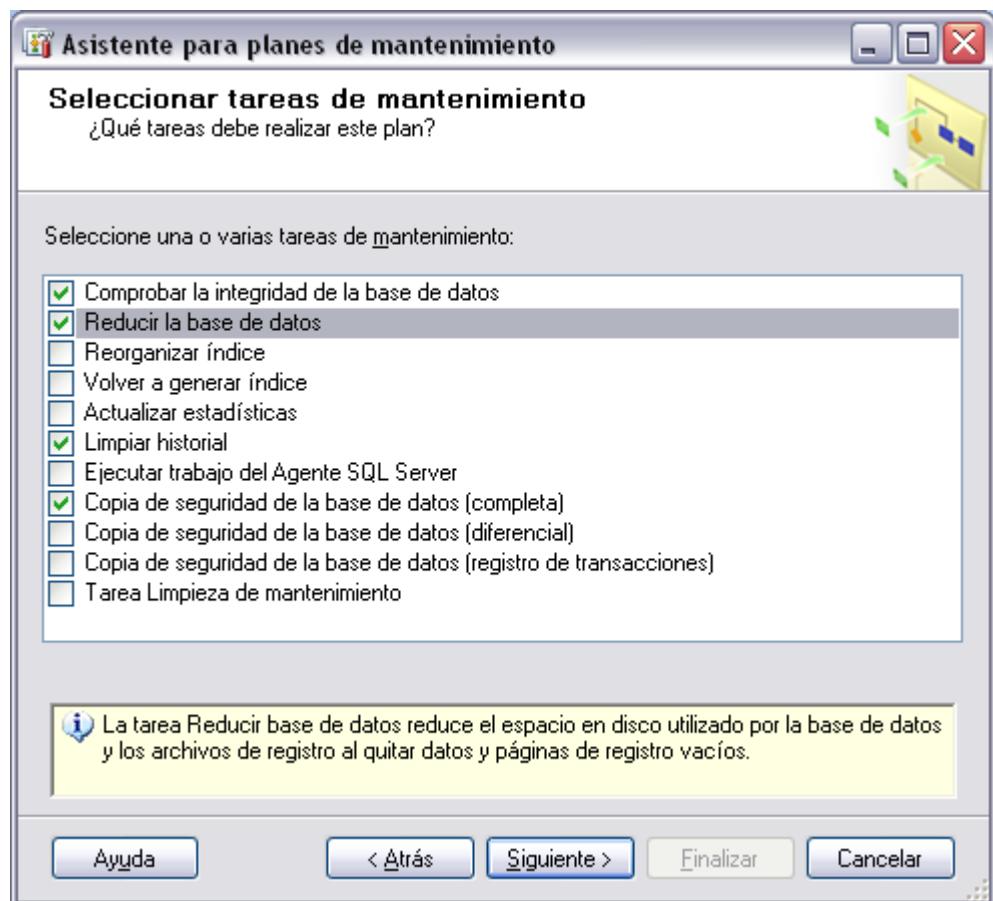
PROGRAMACIÓN DE REALIZADO DE COPIAS DE SEGURIDAD AUTOMÁTICAMENTE POR EL SISTEMA

The screenshot shows the Object Explorer on the left with a tree view of databases, tables, and other objects. In the center, a table named 'LABSIS24.mib...- dbo.p' is displayed with the following data:

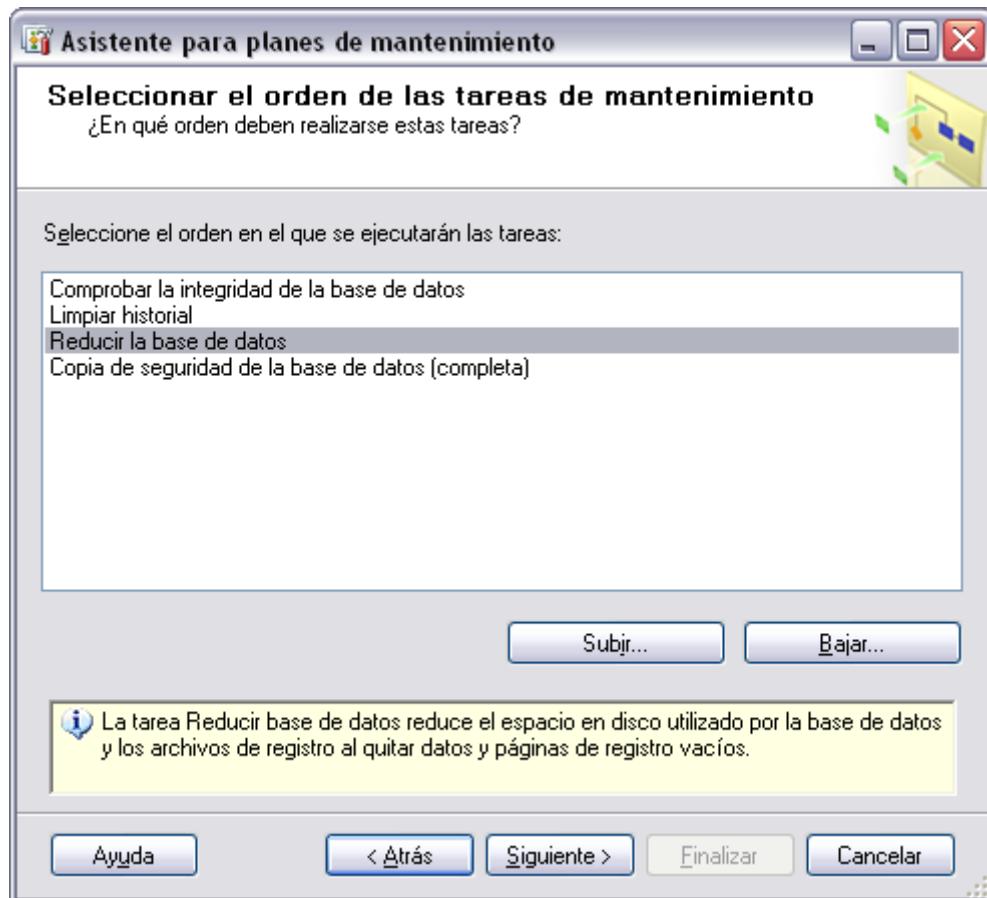
codProducto	nombre	categoria
p0001	queso	lacteos
p0002	mantequilla	lacteos
p0003	rosquitas	panaderia
p0004	chocolate	chocolates
NULL	NULL	NULL

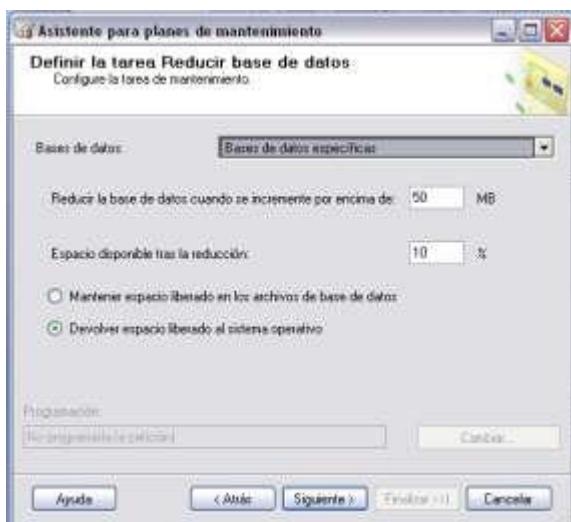
A context menu is open over the table, with the 'Nuevo plan de mantenimiento...' option highlighted. Other options in the menu include 'Asistente para planes de mantenimiento', 'Ver historial', 'Informes', and 'Actualizar'.



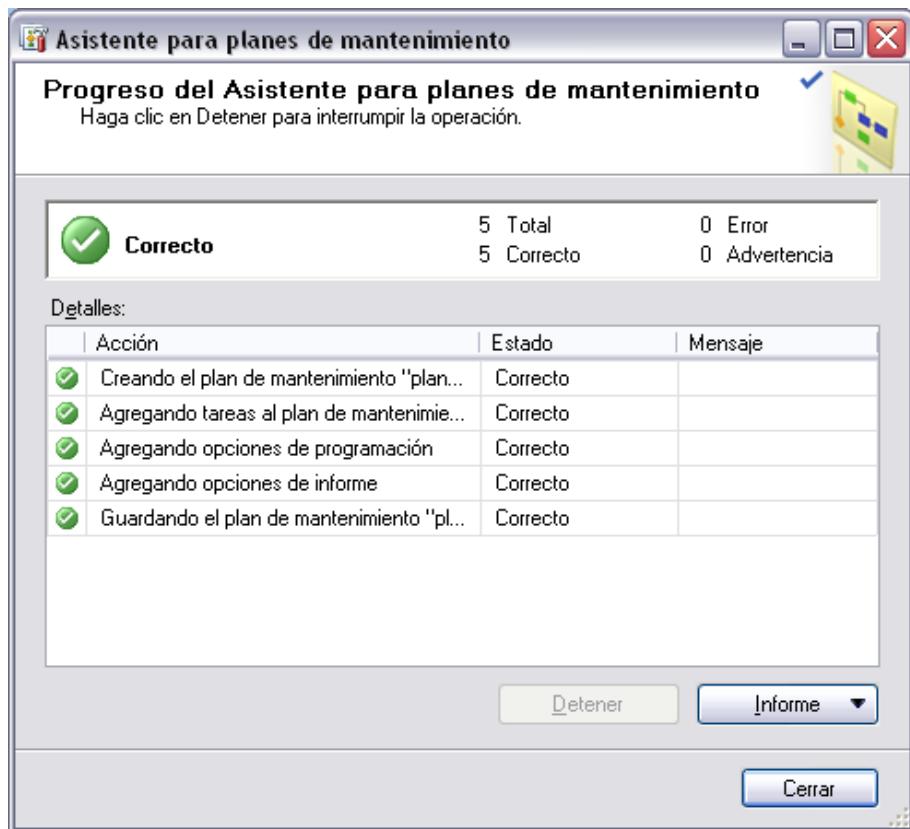


Programando el orden de las tareas









ADMINISTRACIÓN DE USUARIOS

1. PUEDO CREAR USUARIOS EN EL SQL SERVER 2008

Existen tres formas de crear usuarios en SQL las que veremos a continuación:

Primera forma: Para crear un usuario de base de datos mediante SQL Server Management Studio (para esto necesitamos tener creado previamente un login o inicio de sesión, sin ningún usuario asignado a este. Ejemplo

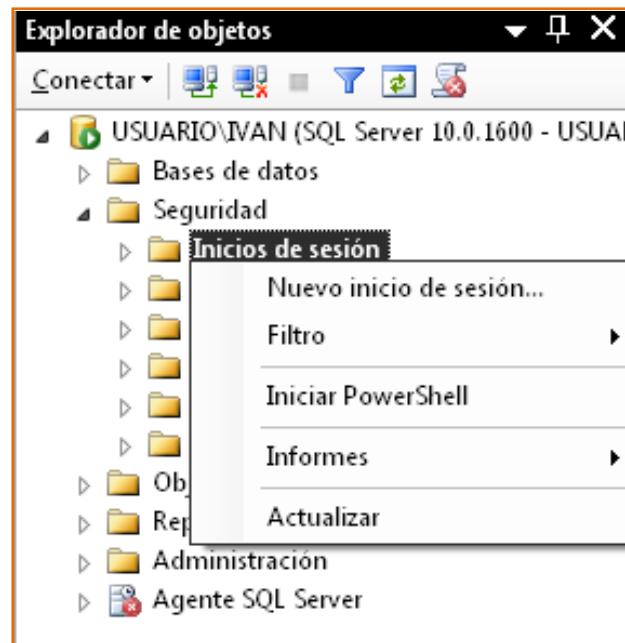
```
Create login login02
```

```
with password='123456' ).
```

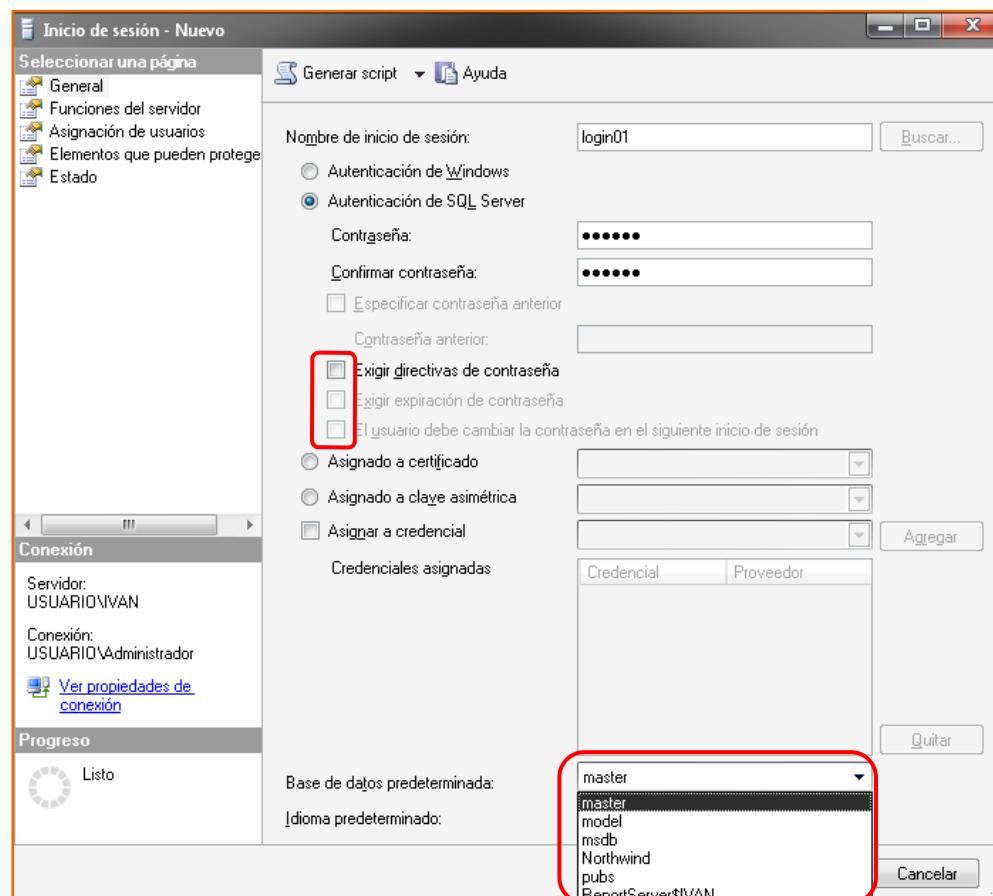
- 1) En SQL Server Management Studio, abra el Explorador de objetos y expanda la carpeta Bases de datos.
- 2) Expanda la base de datos en la que se va a crear el usuario de la misma.
- 3) Haga clic con el botón secundario en la carpeta Seguridad, seleccione Nuevo y, a continuación, haga clic en Usuario.
- 4) En la página General, escriba un nombre para el usuario en el cuadro Nombre de usuario.
- 5) En el cuadro Nombre de inicio de sesión, escriba el nombre de un inicio de sesión de SQL Server para asignarlo al usuario de la base de datos.
- 6) Haga clic en Aceptar.

1.1. Segunda forma: Crear un usuario por la pestaña seguridad (en el explorador de objetos)

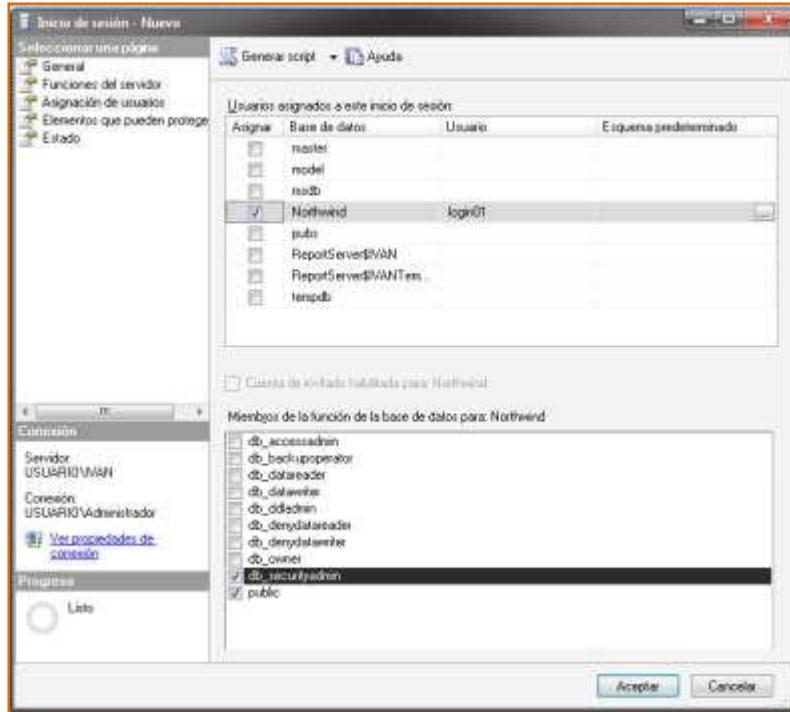
- 1) Dentro de nuestro servidor de base de datos en la pestaña **Seguridad**, hacemos clic derecho en el **inicios de sesión**, (Login), escoger **Nuevo inicio de sesión**:



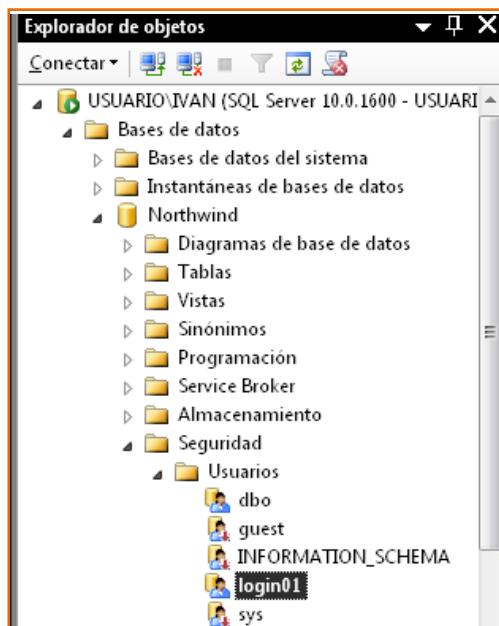
- 2) Asignamos un **Nombre de Inicio de sesión** (loginname), por seguridad de modo **Autenticación de SQL Server**, la base de datos en la cual operará.



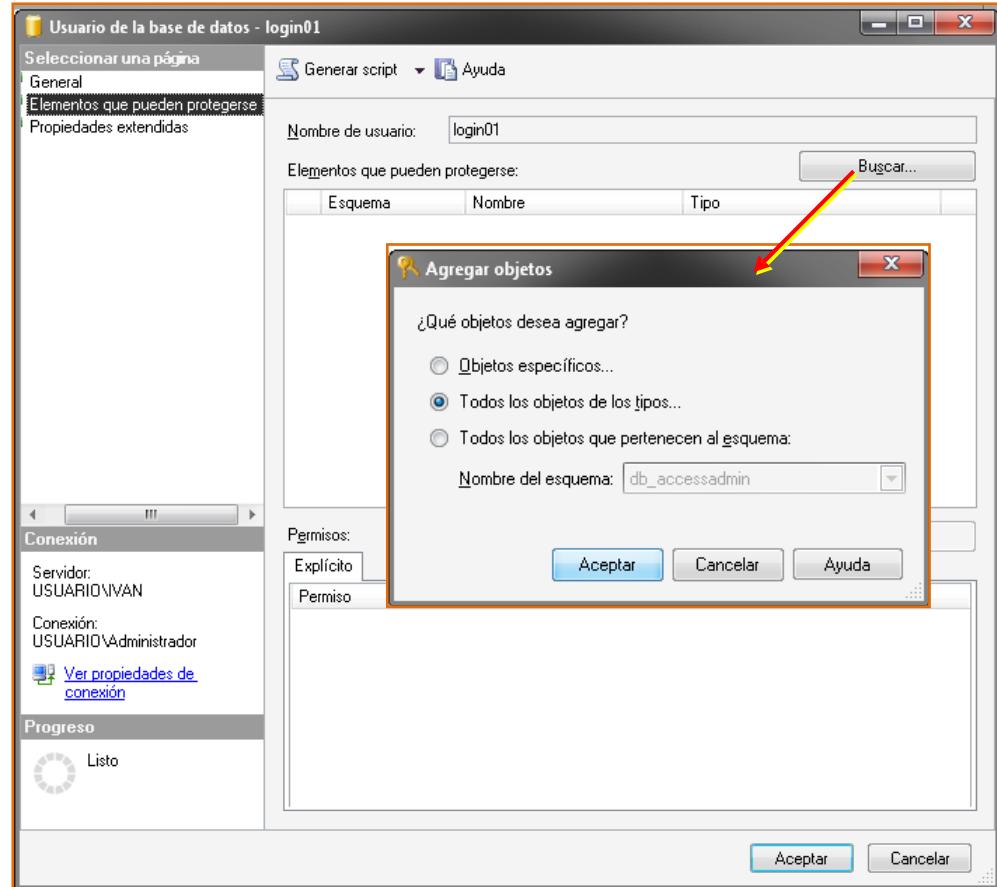
- 3) En la parte de Asignación de usuarios(UserMapping), apunta a la base de datos y al esquema en el que se trabajará.



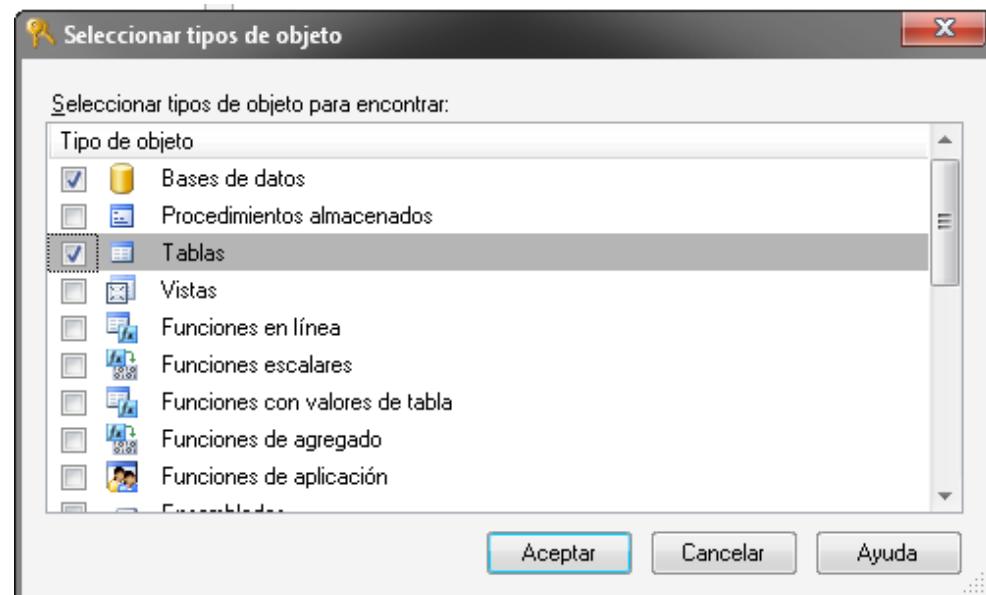
- 4) El siguiente paso es ir a la **base de datos** en la pestaña de **seguridad** y después en **Usuarios** luego chequeamos que exista el usuario que hayamos creado.



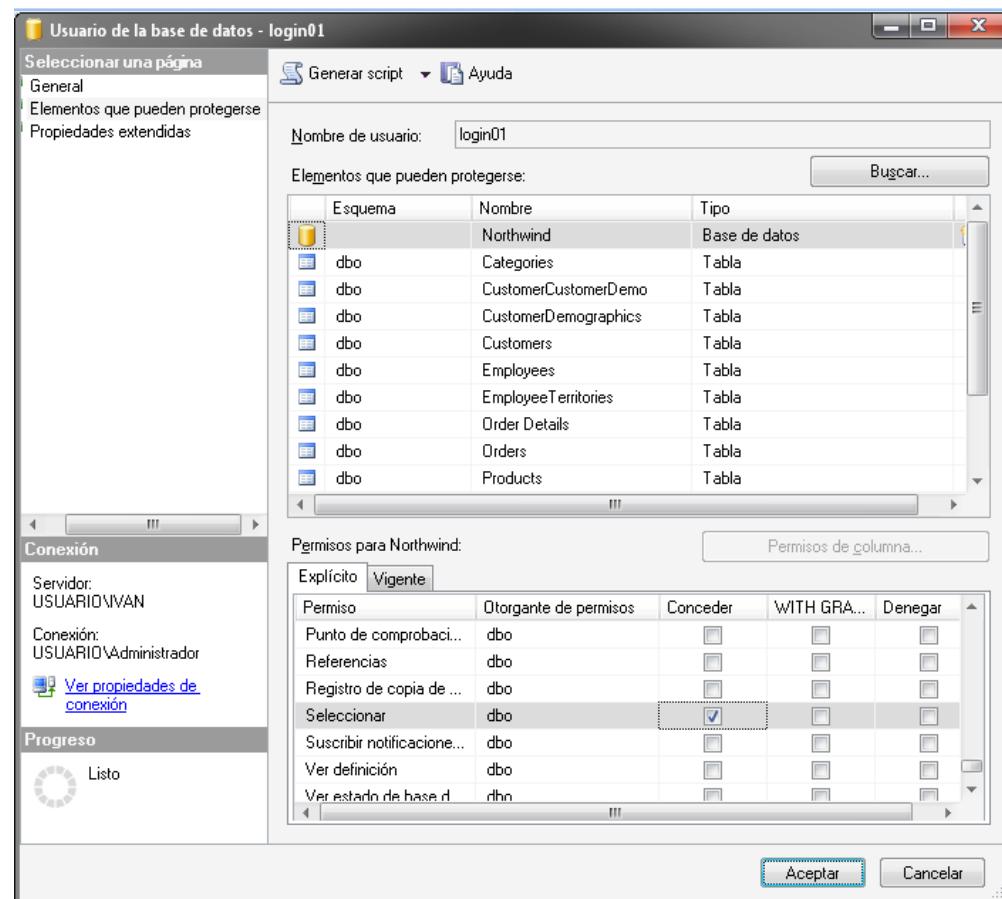
- 5) Despues de haber verificado seleccionamos el usuario y damos clic derecho-propiedades. Luego hacemos clic en buscar:



6) Escogemos la opción de **todos los objetos de los tipos**



7) En este caso solo quiero que se conecte a la base de datos y que pueda ejecutar sentencias de tipo SELECT sobre dichas tablas. Checamos que tenga los permisos de Connect y SELECT.

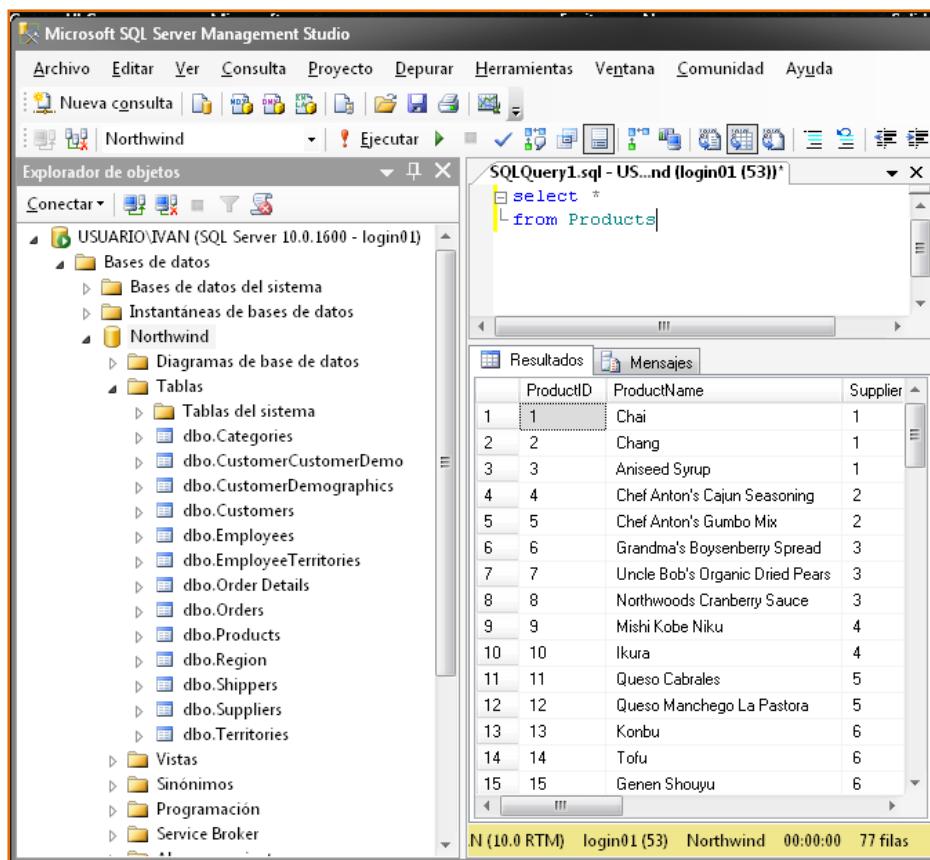


- 8) Cerramos la conexión



- 9) Iniciamos con el usuario creado.





1.2. Para crear un usuario de base de datos mediante Transact-SQL

- 1) En el Editor de consultas, conéctese a la base de datos en la que se va a crear el usuario de la base de datos; para ello, ejecute el siguiente comando Transact-SQL: USE <databasename>
- 2) Cree el usuario ejecutando el siguiente comando de Transact-SQL: CREATEUSER<new username>FORLOGIN<loginname>;

NOTA:

user_name: Especifica el nombre por el que se identifica al usuario en esta base de datos. user_name es de tipo sysname. Puede tener una longitud máxima de 128 caracteres.

LOGIN login_name: Especifica el inicio de sesión de SQL Server del usuario de base de datos que se va a crear. login_name debe ser un inicio de sesión válido en el servidor. Cuando este inicio de sesión de SQL Server se introduzca en la base de datos adquirirá el nombre y el identificador del usuario de la base de datos que se va a crear.

```
--EJEMPLO:
--Creamos un login
Create login login02
With password='123456'

--especifiquemos la base de datos
Use Northwind
go

--creamos al usuario
Create user user01
For login login02
go
```

2. PUEDO ELIMINAR USUARIOS EN EL SQL SERVER 2008

Para eliminar un usuario en Transact SQL se utiliza:sp_dropuser, este código quita un usuario de la base de datos actual.

Sintaxis:

```
sp_dropuser[ @name_in_db = ]'user'
```

Donde:

```
[ @name_in_db = ] 'user'
```

Es el nombre del usuario que se va a quitar. user es de tipo sysname y no tiene valor predeterminado. user debe existir en la base de datos actual. Cuando especifique un inicio de sesión de Windows, utilice el nombre por el que se conoce ese inicio de sesión en la base de datos.

NOTAS:

sp_dropuser ejecuta sp_revokedbaccess para quitar el usuario de la base de datos actual.

Puede utilizar sp_helpuser para obtener una lista de los nombres de usuario que pueden quitarse de la base de datos actual.

Cuando se quita un usuario de base de datos, también se quitan los alias que el usuario tuviera. Si el usuario posee un esquema vacío con su mismo nombre, se eliminará también el esquema. Si el usuario posee otros elementos que pueden protegerse en la base de datos, no se eliminará el usuario. La propiedad de los objetos debe transferirse primero a otra entidad de seguridad. Para obtener más información, vea ALTER AUTHORIZATION (Transact-SQL). Al quitar un usuario de la base de datos se quitan automáticamente los permisos que tiene asociados y, también, se quita el usuario de todas las funciones de las que es miembro.

sp_dropuser no puede utilizarse para quitar el propietario de la base de datos (dbo), usuarios de INFORMATION_SCHEMA ni el usuario guest de las bases de master o tempdb. En el caso de bases de datos que no son del sistema, EXECsp_dropuser 'guest' revocará el permiso CONNECT al usuario guest. Pero no se quitará al usuario.

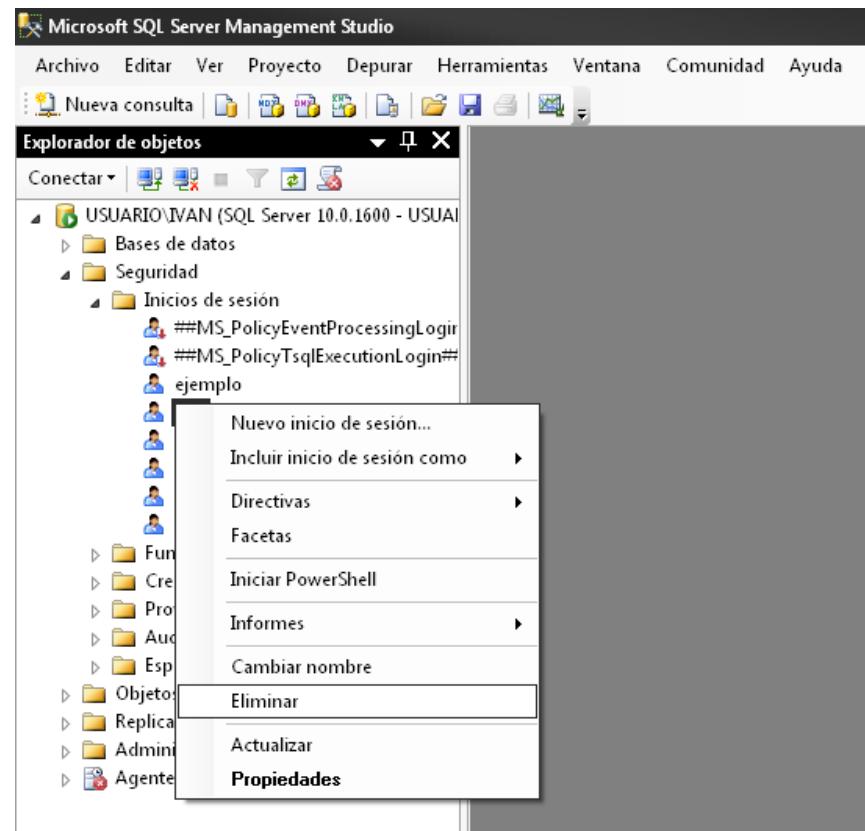
sp_dropuser no puede ejecutarse en una transacción definida por el usuario.

EJEMPLO:

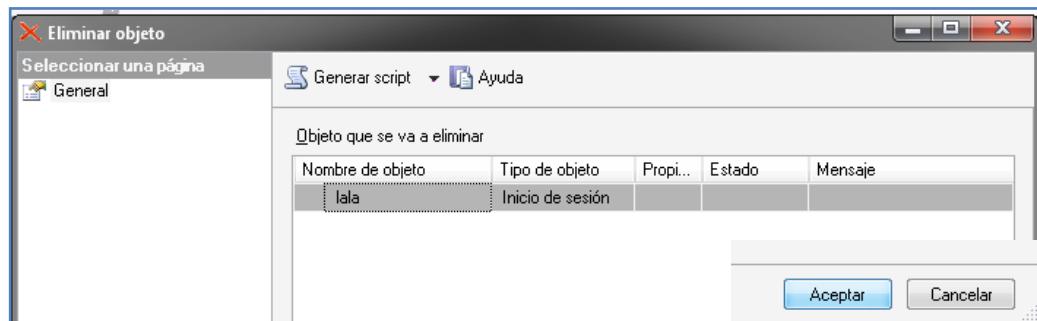
```
useNorthwind  
execsp_dropuser[user01]
```

```
USE [Northwind]  
GO  
DROP USER [lilianaduran]  
GO
```

Otra forma de eliminar un usuario es a través del explorador de objetos, se puede seguir los siguientes pasos:



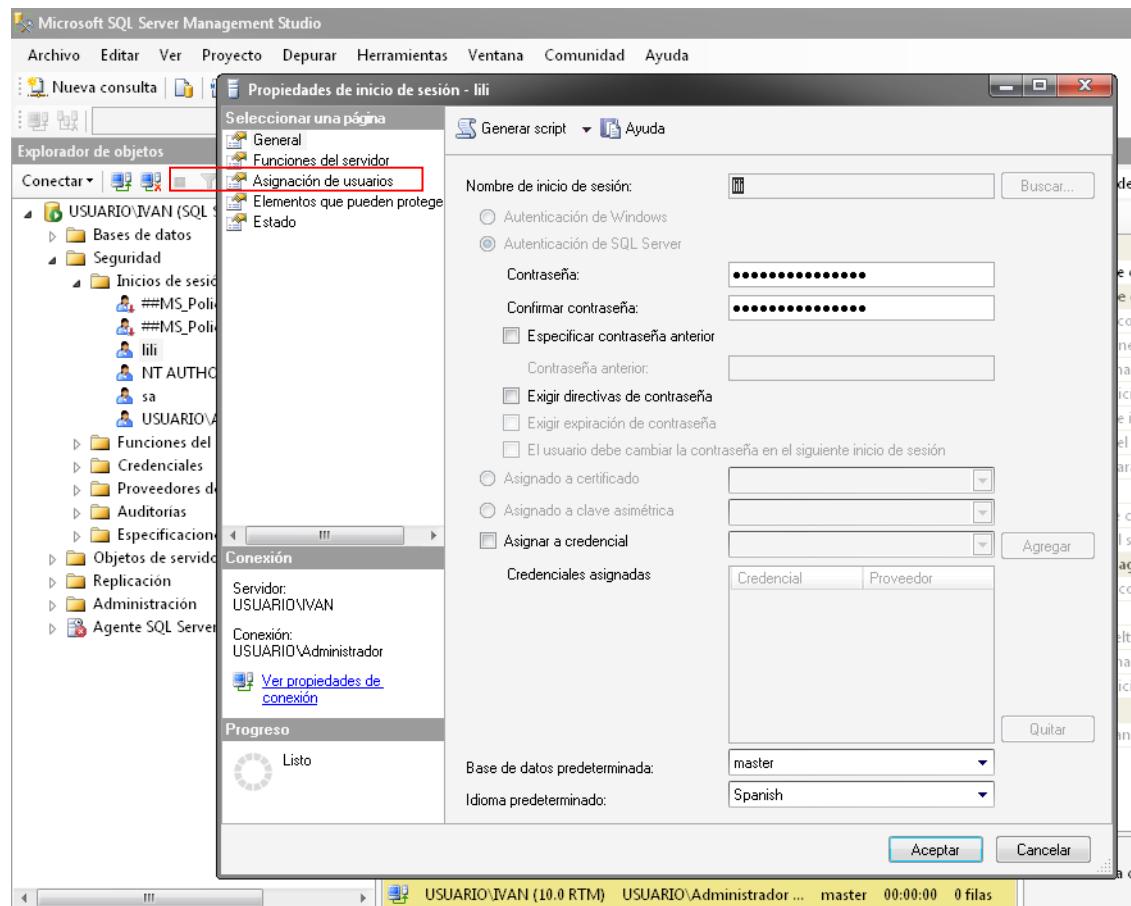
Al eliminar de esta manera solo se eliminar el login (inicio de sesión) por lo que se tiene que eliminar también el usuario asignado en cada base de datos.



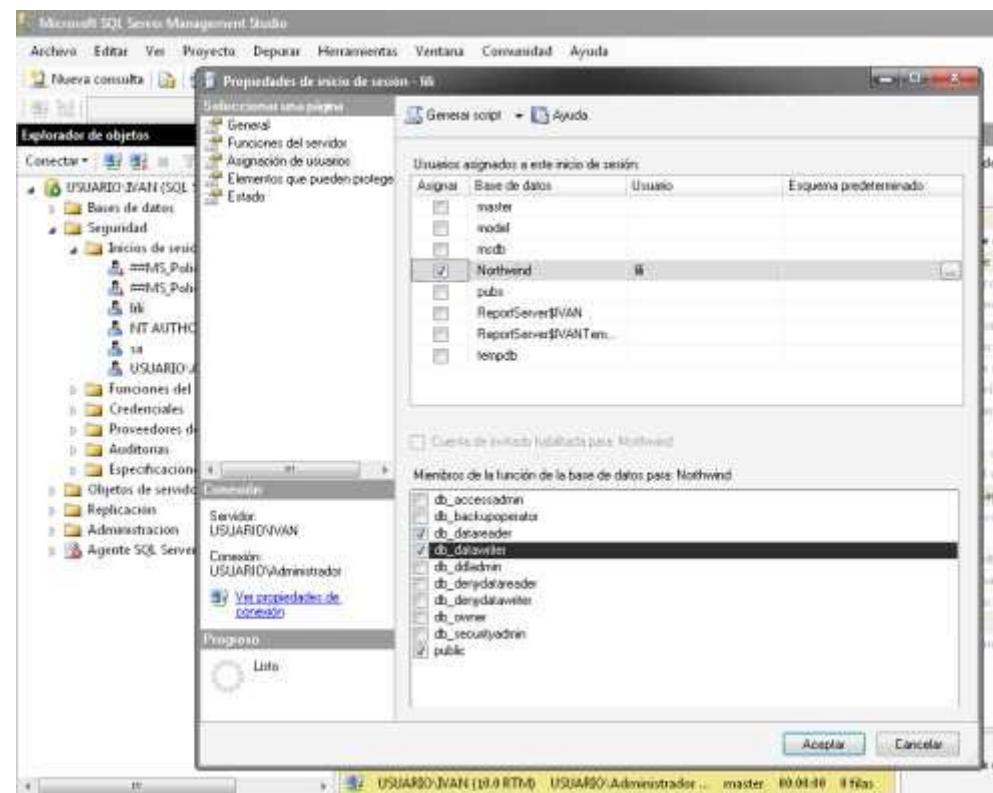
3. A UN USUARIO PUEDO RESTRINGIR O CONFIGURAR A CIERTAS TABLAS

-si puedo restringir ciertas tablas:

Primeramente accedo a un usuario existente creado:



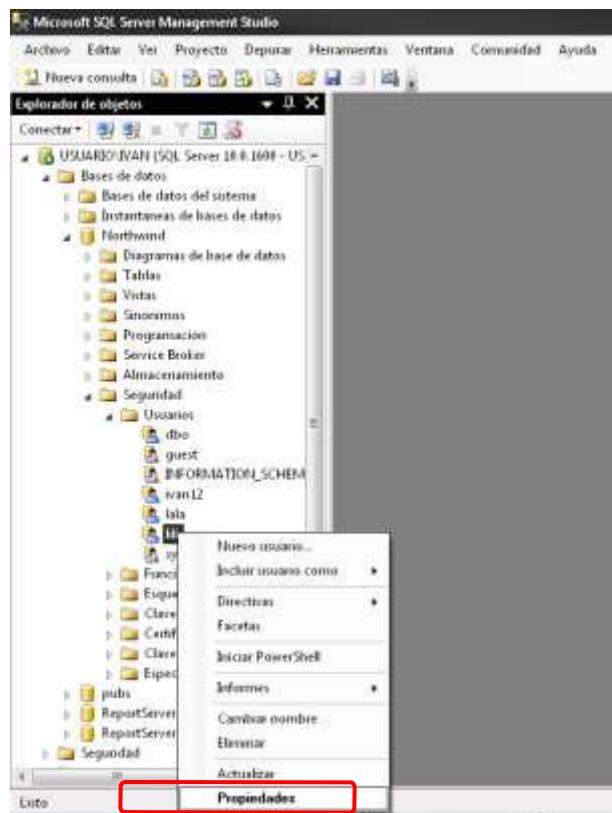
Luego en la opción de asignación de usuarios selecciono la base de datos, y seleccionamos las opciones de: db_datawriter y db_datareader



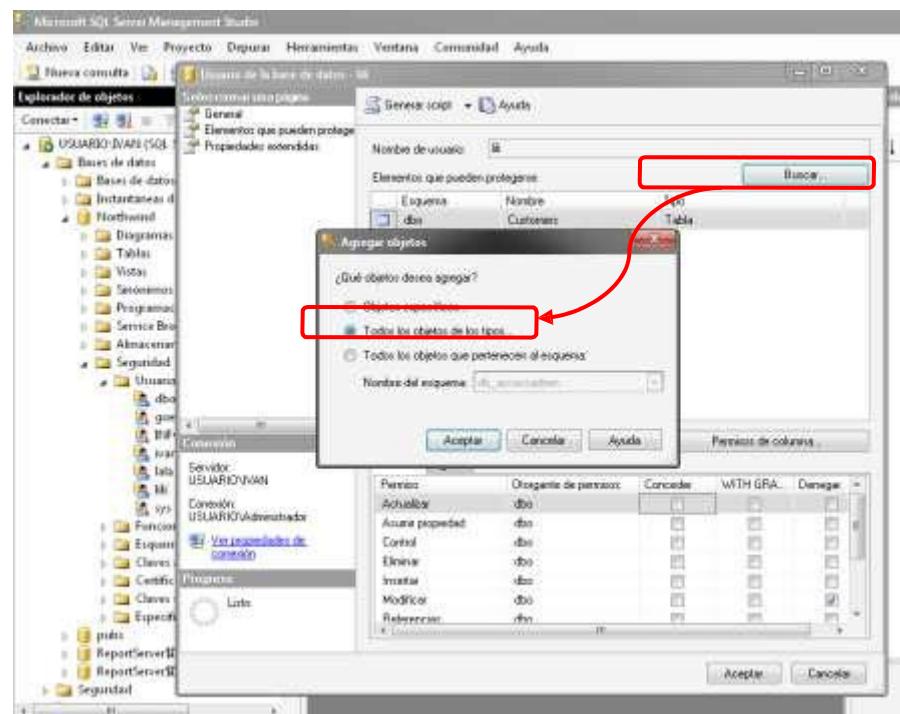
En la siguiente tabla comprobamos que si se pueden realizar consultas, sin ninguna restricción:

CustomerID	CompanyName	ContactName	ContactTitle
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Rep/Ex.
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner
AROUT	Around the Horn	Thomas Hardy	Sales Rep/Ex.
BERGS	Berglunds snabbköp	Christina Berglund	Order Admin/Rep
BLAUS	Blaauw See Delicatesen	Hanna Moos	Sales Rep/Ex.
BLONP	Blondel pâté et filé	Frédérique Céraux	Marketing Mgr
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner
BONAP	Bon app'	Laurence Lebihan	Owner
BOTTM	Bolton-Dollar Markets	Elizabeth Lincoln	Accounting Mgr
SCARL	SCARLET SISTEMAS	Viktoria Andrade	Owner

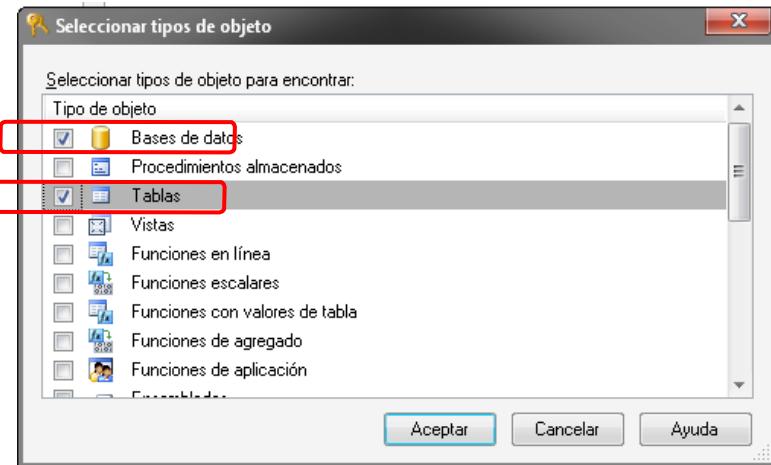
Luego vuelvo a iniciar el administrador de usuarios para realizar las restricciones respectivas. Para lo cual ingresamos a las propiedades del usuario al que vamos a restringir:



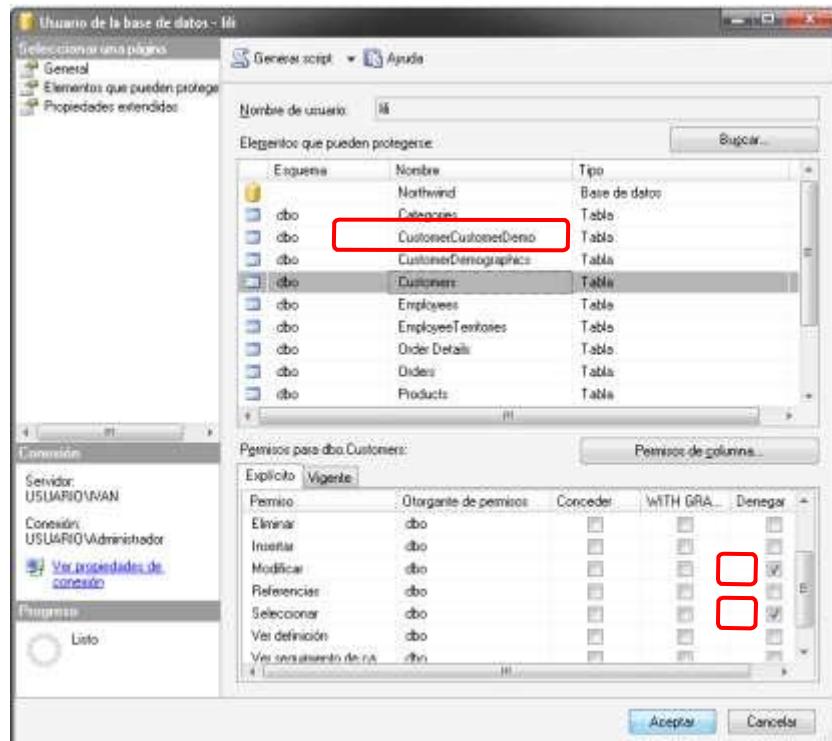
En la opción elementos que pueden protegerse, buscamos todos los objetos...



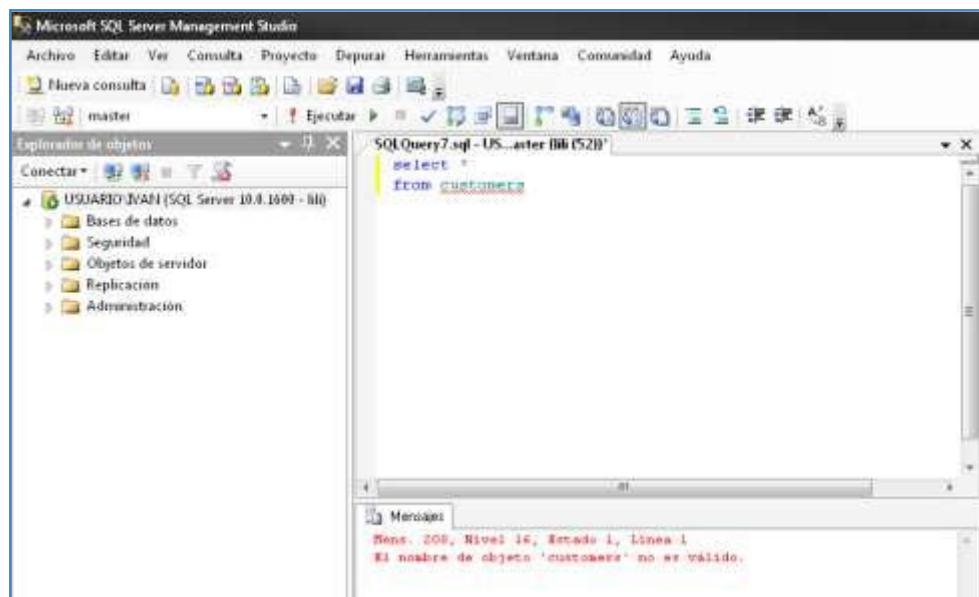
En este caso seleccionamos las opciones de siguientes



Seleccionamos las tablas con las que deseamos trabajar y denegamos permisos:

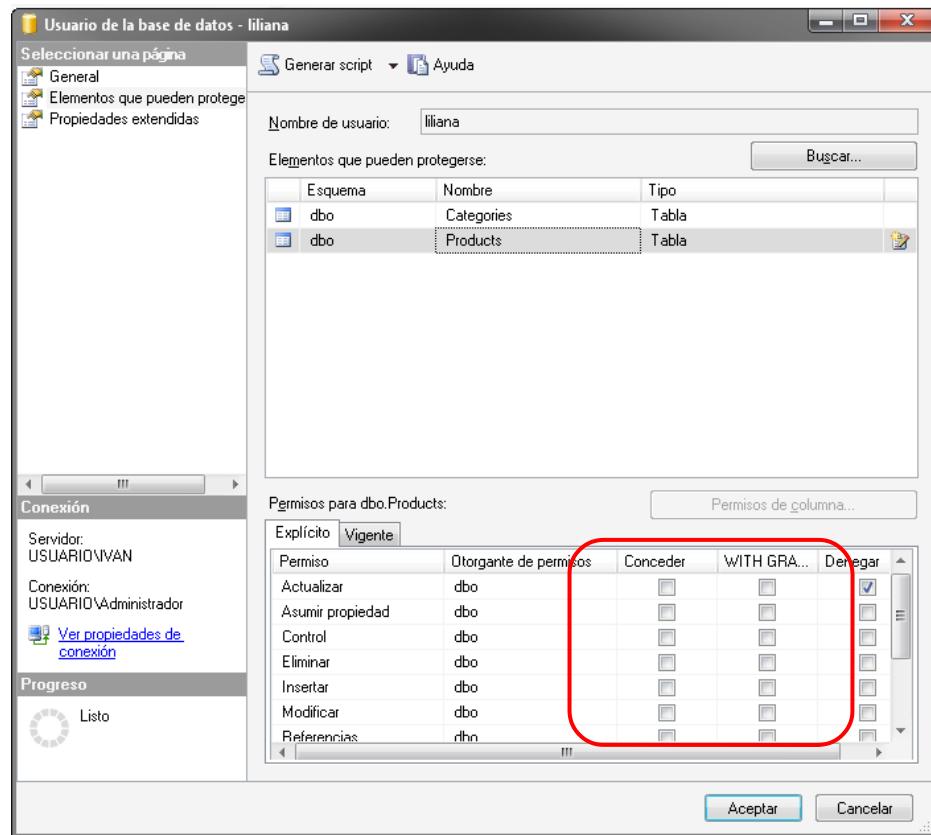


Comprobamos que no se pueden realizar ninguna consulta de selección, a las tablas anteriormente restringidas:



4. PUEDO CONFIGURAR O ACTUALIZAR CIERTAS TABLAS

Podemos ver que se pueden restringir incluso las vistas de las tablas de la base de datos. Para esto se repite el procedimiento anterior, con la diferencia de que ahora se va a denegar la opción actualizar o la opción que se requiera



Comprobamos el procedimiento antes explicado, que ya no se puede realizar consultas. Ni actualizaciones en la tabla restringida

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a query window titled 'SQLQuery9.sql - OSU.mdf (Última (55))' containing the following T-SQL code:

```
select * from products

select UnitPrice, CategoryID
from Products
order by UnitPrice
update Products
set UnitPrice=UnitPrice*1.1
```

Below the query window is a 'Messages' window displaying error messages in red:

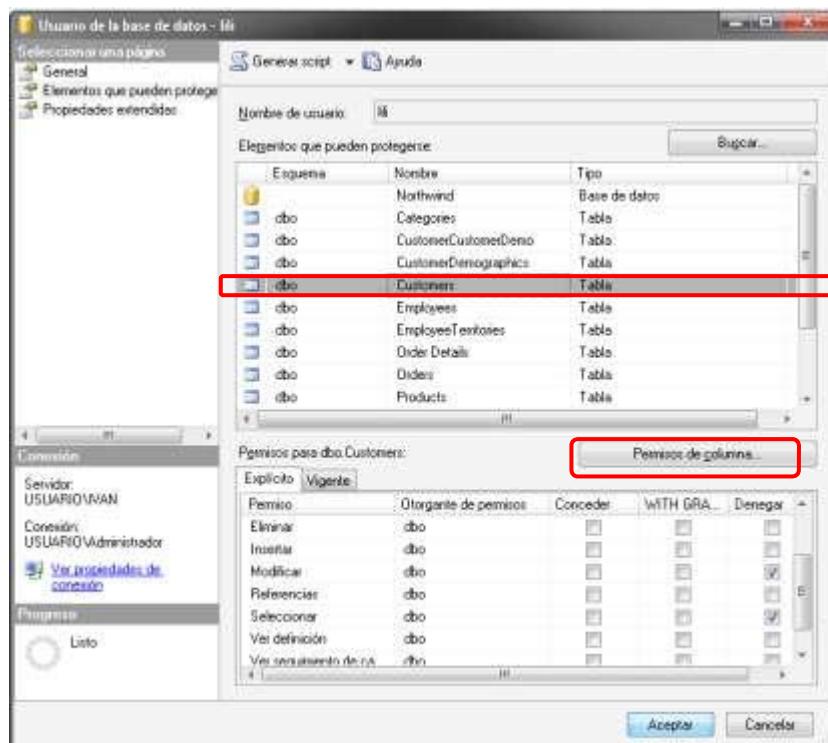
```
Bens...229, Nivel 14, Estado 1, Línea 1
Se denegó el permiso SELECT en el objeto 'Products', base de datos 'Northwind'.
Bens...229, Nivel 14, Estado 1, Línea 4
Se denegó el permiso SELECT en el objeto 'Products', base de datos 'Northwind'.
Bens...229, Nivel 14, Estado 1, Línea 4
Se denegó el permiso UPDATE en el objeto 'Products', base de datos 'Northwind'.
```

At the bottom of the screen, a status bar indicates: 'Consulta completada... USUARIO\IVAN [10.0 BTd] Última (55) Northwind - 00:09:04 0 filas'.

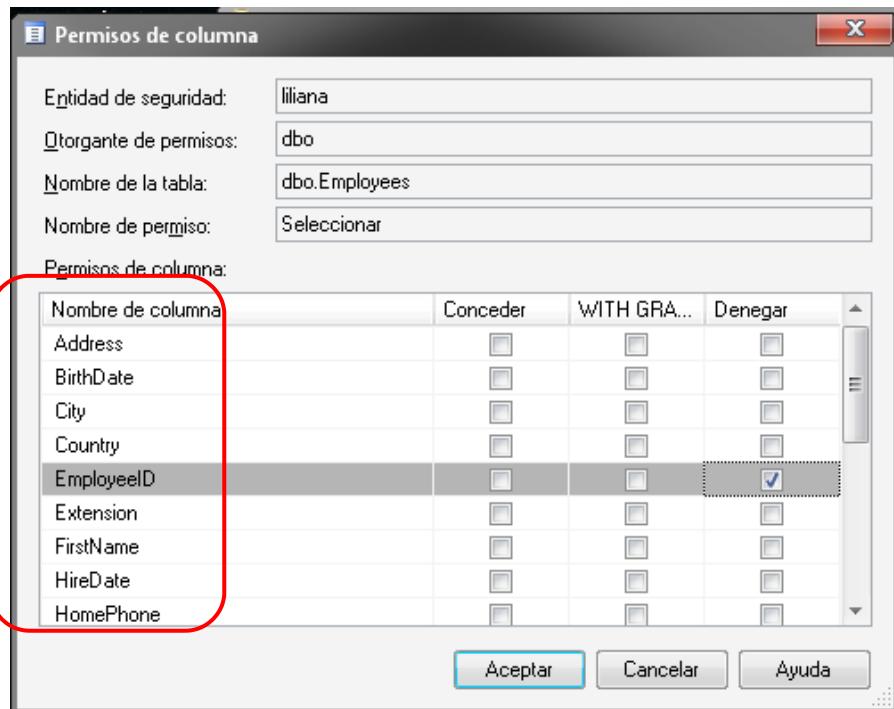
5. RESTRINGIR AL USUARIO A NIVEL DE CAMPOS

Para restringir columnas se realiza los siguientes procedimientos:

En las propiedades del usuario de la base de datos en la que deseamos trabajar, seleccionamos la tabla que contiene los campos a restringir



Se eligen los campos a restringir:



Comprobamos la restricción hecha:

The screenshot shows the following interface elements:

- Object Explorer:** Shows the database structure under 'Northwind'. The 'Seguridad' (Security) node is expanded, showing 'Usuarios' (Users) with 'dbo' selected.
- SQL Query Window:** Contains the following SQL code:

```
select EmployeeID
from Employees
```
- Message Window:** Displays the error message:

```
Ha sido denegado el permiso: SELECT en la columna 'EmployeeID' del objeto 'Employees'
```
- Status Bar:** Shows 'Consulta completada...', 'USUARIO:IVAN (10.0.1809 - US...)', 'Máxima (53)', 'Northwind', '00:00:00', '0 filas'.

6. SE PUEDE ASIGNAR ROLES A UN GRUPO DE USUARIOS

Para poder utilizar Administrador de autorización de forma efectiva para controlar el acceso a los recursos, primero debe definir roles, tareas y operaciones.

- 6.1. **Rol:**es un conjunto de permisos de los que debe disponer un usuario para realizar un trabajo. Los roles bien diseñados deberían corresponderse con una categoría o responsabilidad profesional (por ejemplo, recepcionista, director de contratación o archivista) y ser nombradas en consecuencia. En Administrador de autorización, puede agregar usuarios a un rol a fin de autorizarles para un trabajo.
- 6.2. **Tarea:**es un conjunto de operaciones y, en ocasiones, de otras tareas. Las tareas bien diseñadas son lo suficientemente inclusivas para representar elementos de trabajo que sean reconocibles (por ejemplo, "cambiar contraseña" o "presentar gastos").
- 6.3. **Operación:**es un conjunto de permisos que se asocian a procedimientos de seguridad del nivel del sistema o de la API, como Write Attributes o Read Attributes (atributos de escritura o atributos de lectura). Las operaciones se usan como bloques de creación para las tareas.

Los roles, las tareas y las operaciones sólo se pueden definir en el modo de programador, pero no en el modo de administrador. Para establecer el modo de programador..

6.4. Asignar un usuario o grupo de Windows a un rol

A fin de utilizar de forma efectiva Administrador de autorización para controlar el acceso a los recursos, debe definir qué grupos de usuarios están asociados a cada rol. Para asignar un usuario o un grupo de Windows a un rol, utilice el siguiente procedimiento.

Para realizar este procedimiento, debe tener asignado el rol de usuario **Administrador** del Administrador de autorización. De manera predeterminada, este rol debe pertenecer como mínimo al grupo **Administradores** de Windows.

- 1) Si es necesario, abra Administrador de autorización.
- 2) Si es necesario, cree o abra un almacén de autorización.
- 3) En el árbol de consola, haga clic con el botón secundario en Asignaciones de roles, en una aplicación o un ámbito, y haga clic en Nueva asignación de roles. La carpeta Asignaciones de roles se utiliza como contenedor para vincular grupos a roles. No todos los roles tienen grupos asociados, ya que los roles pueden combinarse con roles de un tamaño superior.
- 4) Para seleccionar el rol al que desea asignar grupos, active la casilla situada junto al nombre de la definición de rol correspondiente y haga clic en Aceptar. Se puede agregar una misma definición de rol al contenedor Asignaciones de roles más de una vez. Esto permite administrar las asignaciones de forma flexible.
- 5) Si desea cambiar el nombre para mostrar de la asignación de roles, haga clic en él con el botón secundario en la lista de asignaciones de roles, haga clic en Propiedades y escriba el nuevo nombre para mostrar.

- 6) En la lista de asignaciones de roles, haga clic con el botón secundario en la asignación de roles de los pasos anteriores, seleccione Asignar usuarios y grupos y, a continuación, haga clic en De Windows y Active Directory.
- 7) En el cuadro Escriba los nombres de objeto que desea seleccionar, escriba los nombres de usuario de los miembros que desee. O bien, haga clic en Opciones avanzadas para buscar en los Servicios de dominio de Active Directory (AD DS).
- 8) Haga clic en Aceptar.

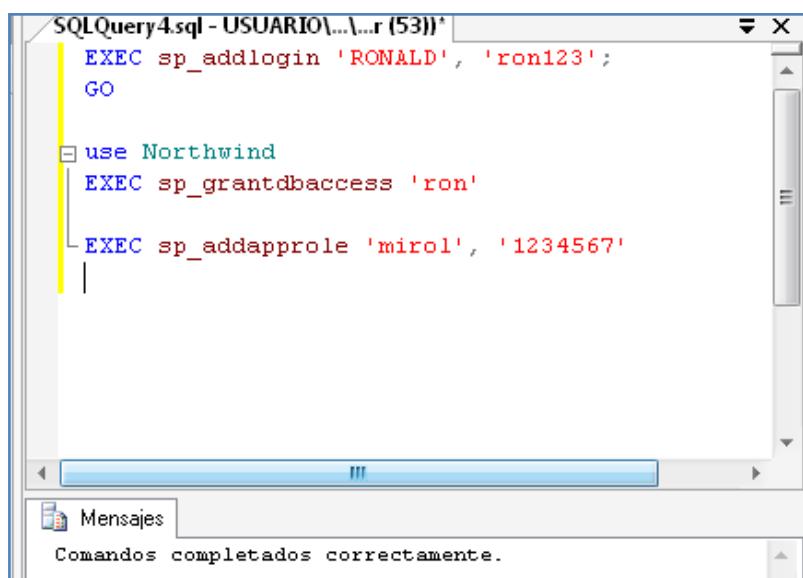
Consideraciones adicionales

Para llevar a cabo este procedimiento, debe tener acceso a un almacén de autorización. De manera predeterminada, los miembros del grupo **Administradores** tienen el acceso necesario, pero el Administrador de autorización permite delegar la responsabilidad. Para obtener más información, consulte la sección "Referencias adicionales" de este tema.

Creación de roles de aplicaciones en sql server 2008

Primero tenemos que verificar que nuestro servidor permita autenticación mixta (SQL server y windows)

1. Creamos nuestro inicio de sesión (en una consulta).
2. Damos acceso a nuestro inicio de sesión en nuestra base de datos
3. Creamos en el rol de aplicación



```

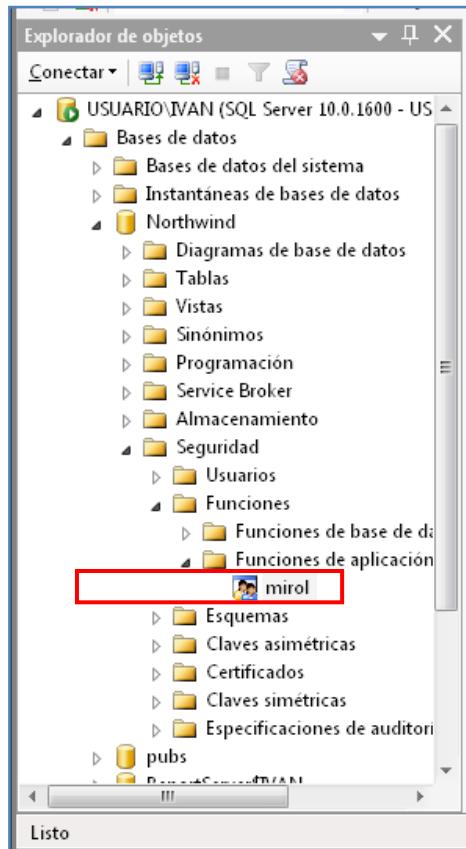
SQLQuery4.sql - USUARIO\...r (53))
EXEC sp_addlogin 'RONALD', 'ron123'
GO

use Northwind
EXEC sp_grantdbaccess 'ron'

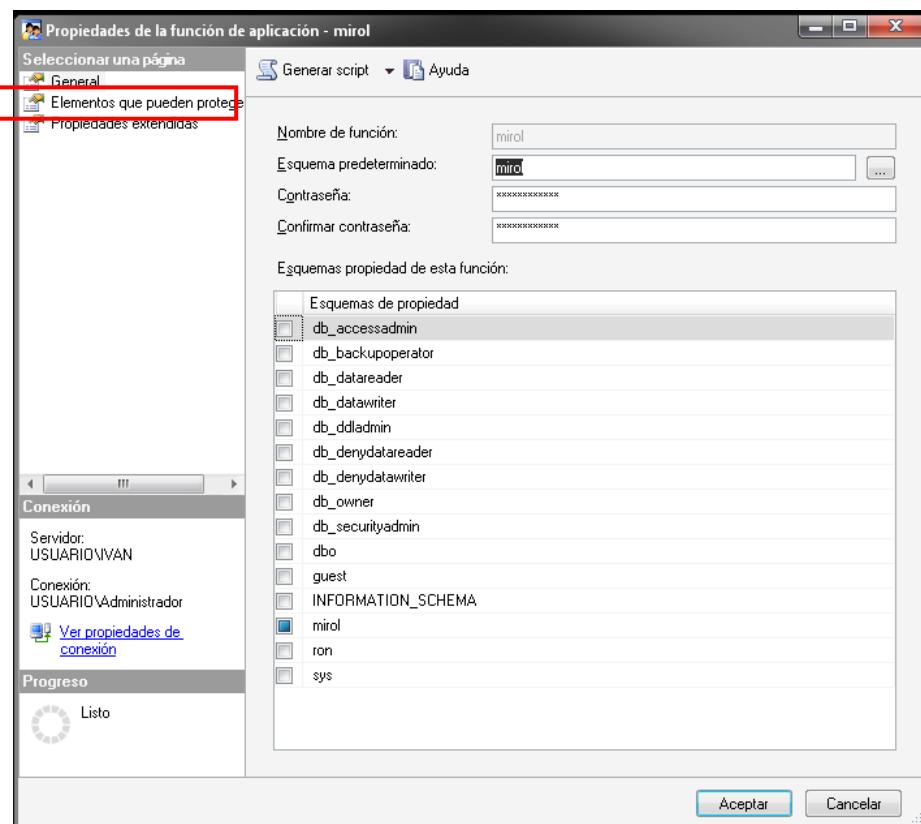
EXEC sp_addapprole 'mirol', '1234567'
  
```

The screenshot shows a SQL query window titled 'SQLQuery4.sql - USUARIO\...r (53))'. It contains three T-SQL commands: creating a login ('RONALD') with password 'ron123', granting database access to the user 'ron' from the 'Northwind' database, and adding a user 'mirol' to the application role with password '1234567'. The status bar at the bottom right indicates 'Comandos completados correctamente.'

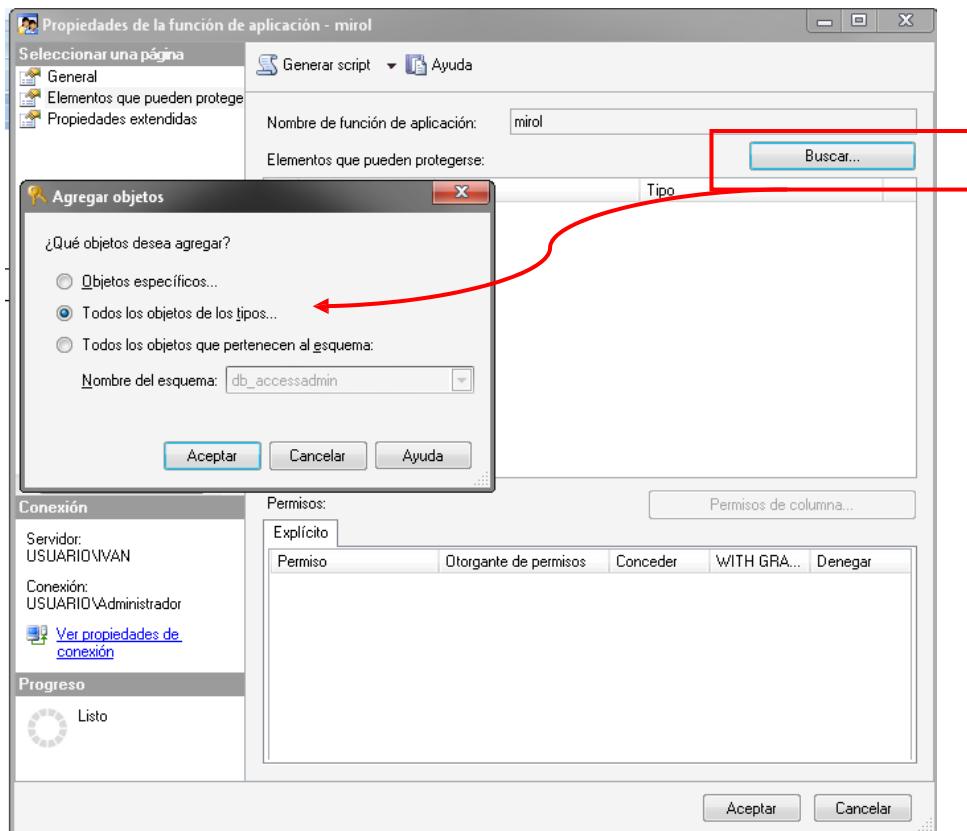
Luego entramos a la base de datos y ubicamos el rol creado, para asignarle sus respectivas funciones.



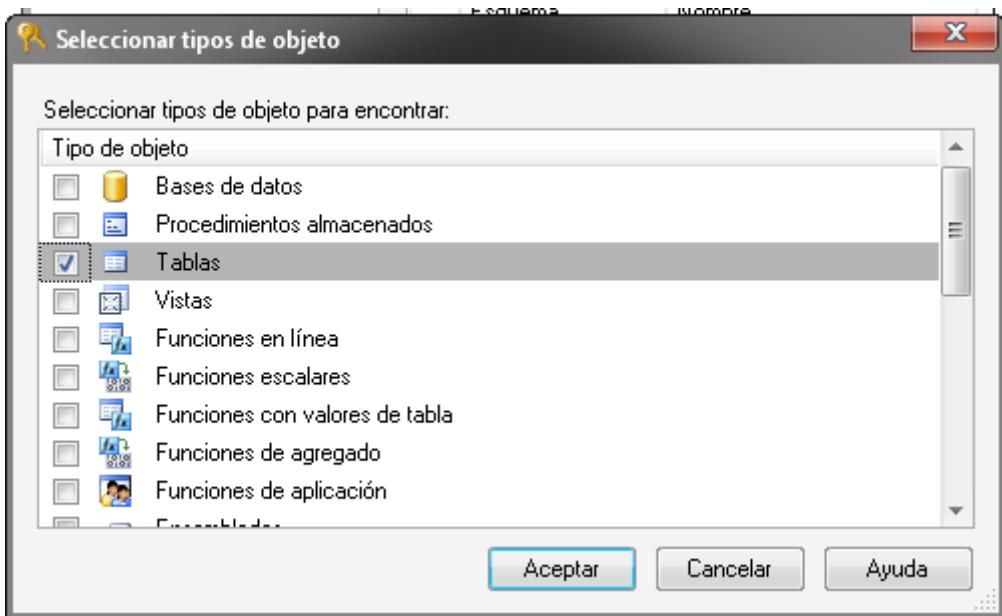
Ingresamos a la opción de elementos que pueden protegerse:



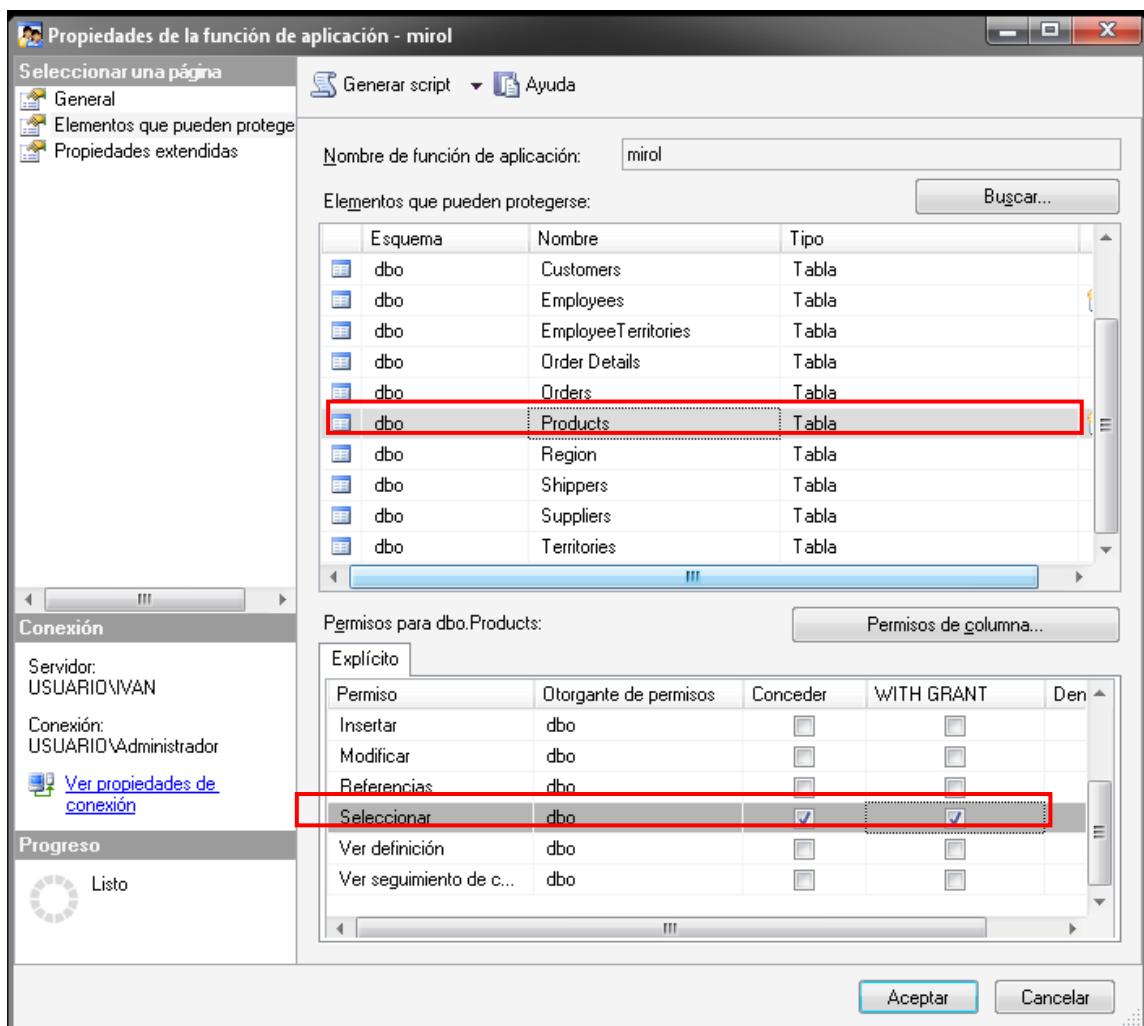
En la ventana abierta accedemos a buscar 'todos los objetos de los tipos...'



Checamos la opción tablas:



Escogemos la tabla y damos los permisos respectivos



Hacemos una consulta select y observamos que no se puede ejecutar dicho comando.

```
Microsoft SQL Server Management Studio
Archivo Editar Ver Consulta Proyecto Depurar Herramientas Vistas Comunidad Ayuda
Nueva consulta Ejecutar
Explorador de objetos
USUARIO\JIVAN (SQL Server 18.0.1690 - ronald)
Bases de datos
Bases de datos del sistema
Instantáneas de bases de datos
Northwind
Diagramas de base de datos
Tablas
Tablas del sistema
Vistas
Síntomas
Programación
Service Broker
Almacenamiento
Seguridad
pubs
ReportServer\JIVAN
ReportServer\JIVAN\TempDB
Seguridad
Objetos de servidor
Replicación
Administración
SQLQuery5.sql - U... and ronald (53)
select * from products
Mensajes
Mensaje 129, Nivel 14, Estado 5, Línea 1
Se denegó el permiso SELECT en el objeto 'Products', base de datos 'Northwind'.
Consulta completada con éxito. USUARIO\JIVAN (18.0 RTM) ronald (53) Northwind 0:00:00 0 filas
Lin 1 Col 23 Car 23 INN
```

Activamos el rol de aplicación para esta sesión, y vemos que ya se ejecuto la consulta.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, a connection to 'USUARIO\IVAN (SQL Server 10.0.1600 - ronald)' is selected, with 'Bases de datos' expanded to show 'Northwind'. In the center, a query window titled 'SQLQuery5.sql - U...ind ronald (53)' contains the following T-SQL code:

```
sp_setapprole 'mirol', '1234567'

select * from products
```

The results pane below displays the output of the 'products' query, showing 8 rows of product information:

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice
1	Chai	1	1	10 boxes x 20 bags	18.00
2	Chang	1	1	24 - 12-oz bottles	19.00
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10.00
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22.00
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25.00
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30.00
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40.00

The status bar at the bottom indicates 'Consulta ejecutada correct...' and '77 filas'.

El rol de aplicación solo se puede utilizar o está activo, cuando la sesión donde este se encuentra esta activo.

7. Es igual USER con LOGIN?

7.1. USER

Un usuario es un individuo que utiliza una computadora, sistema operativo, servicio o cualquier sistema informático este, generalmente se identifica frente al sistema o servicio utilizando un nombre de usuario (nick) y a veces una contraseña, este tipo es llamado usuario registrado. Por lo general un usuario se asocia a una única cuenta de usuario, en cambio, una persona puede llegar a tener múltiples cuentas en un mismo sistema o servicio (si eso está permitido).

Un usuario registrado accede a un servicio a través de un login luego de su autentificación.

Un sistema puede soportar múltiples usuarios (multiusuario) o un único usuario (monousuario).

7.2. LOGIN

El login es el momento de autentificación al ingresar a un servicio o sistema.

En el momento que se inicia el login, el usuario entra en una sesión, empleando usualmente un nombre de usuario y contraseña.

Suele usarse como verbo y conjugarse al españolizarse, por ejemplo: "loguearse". En inglés la acción de "loguearse" es "logging in".

Un término más apropiado para "loguearse" sería "Iniciar sesión" o "Autenticarse". La acción contraria es cerrar sesión o desidentificarse (loggingout).

Loginname, nombre de usuario. Es el nombre que adquiere el usuario para acceder a un determinado servicio.

7.3. DIFERENCIAS ENTRE USER Y LOGIN

La diferencia entre Inicio de Sesión y Usuario de Base de Datos. Al contrario que en otros motores de base de datos, SQL Server tiene dos niveles de profundidad en la definición de sus Usuarios.

Por un lado está el Inicio de Sesión (el usuario con el que nos conectamos, el de la password) y por otro lado está el Usuario de Base de Datos (se le asigna al Inicio de Sesión) que es sobre el que se asignan los permisos de acceso a los objetos de base de datos. Esta es una duda típica en quienes empiezan con SQL Server.

Un Inicio de Sesión (Login) representa la conexión a la Instancia de SQL Server. Dicha conexión debe ser validada por algún tipo de servidor de autenticación, de tal modo, que podemos encontrar Inicios de Sesión de Windows (la validación la realiza el Sistema Operativo, y representa al usuario contextual con el que hemos iniciado sesión en Windows) e Inicios de Sesión de SQL Server (la validación la realiza SQL Server, luego es el motor de base de datos quién debe almacenar la contraseña o su hash y quién debe validar al usuario).

En cualquier caso, un Inicio de Sesión definido en una Instancia puede pertenecer a determinadas Funciones de Servidor (Server Roles) de dicha Instancia, y cuya pertenencia suele conceder determinados privilegios en dicha Instancia de SQL Server (ej: pertenecer a BULKADMIN permite poder realizar cargas masivas en cualquier base de datos de la Instancia, siempre que además se tengan permisos sobre la tabla de destino, claro).

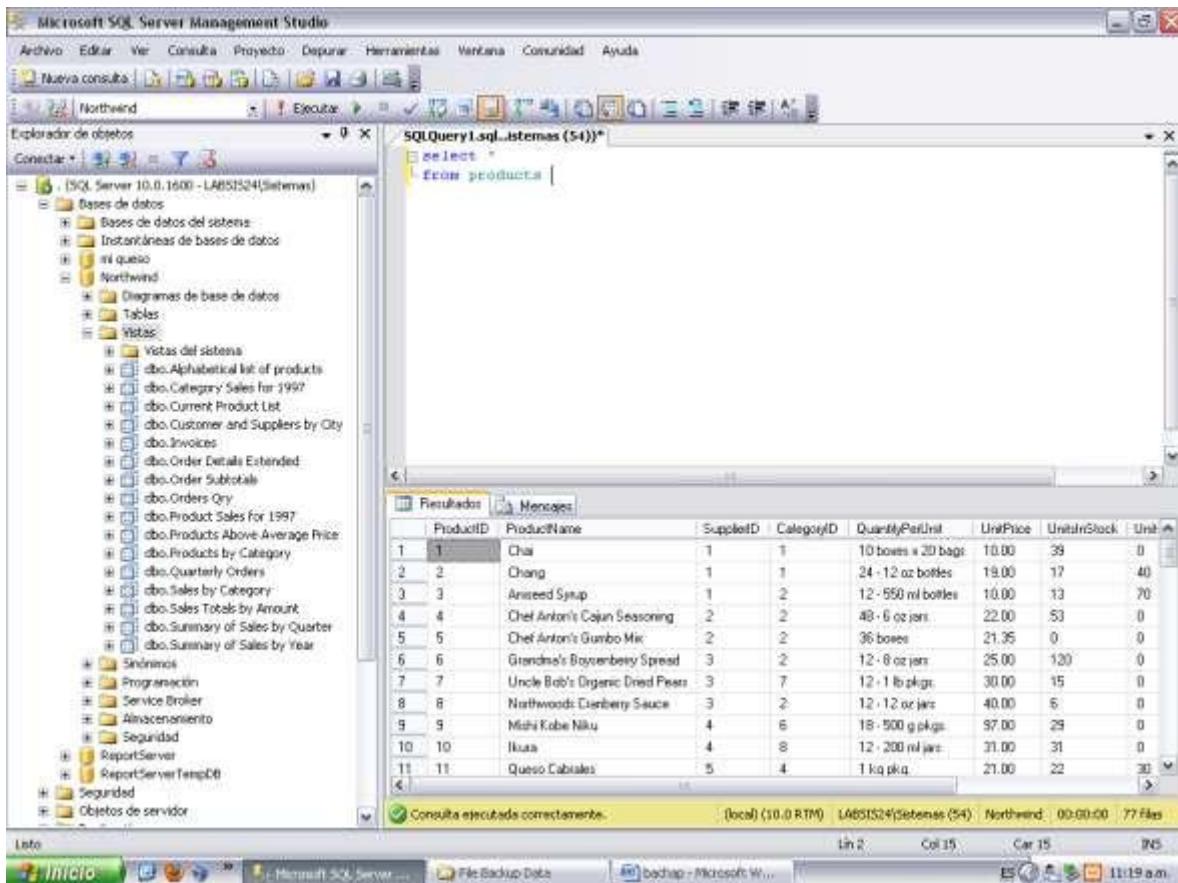
Por otro lado, ocurre que los objetos a los cuales habitualmente deseamos acceder (procedimientos almacenados, tablas, etc.), se encuentran en una u otra base de datos de la Instancia, es decir, no se encuentra definidos en la Instancia como tal. Por ello, en cada Base de Datos a la que tengamos que acceder deberemos de tener un Usuario. Aquí está la principal diferencia entre ambos conceptos: un Inicio de Sesión (Login) se define a nivel de Instancia mientras que un Usuario se define a nivel de Base de Datos.

Habitualmente, se crea un Inicio de Sesión para una persona (o aplicación) que necesite conectarse a SQL Server, y seguidamente se crea un Usuario para ese Inicio de Sesión sobre la base de datos a la que se desea conceder acceso, de tal modo, que si dicho Inicio de Sesión debe de poder acceder a varias bases de datos, deberá tener un Usuario en cada base de datos.

VISTAS

Las vistas se crean para simplificar la programación

Es una consulta que se convierte en una tabla temporal (por lo que no hay sentencias como into, insert, order by, etc)



```
create view v_todosproduc --view para crear vistas
as -- luego de estos se hace la consulta
select *
from products
```

EJECUTAMOS LAS VISTA

```
select *
from v_todosproduc
```

MODIFICANDO LA VISTA

```
alter view v_todosproduc --view para crear vistas
as -- luego de estos se hace la consulta
select ProductID , ProductName
from products
```

AGREGAMOS TOP Y ORDER BY

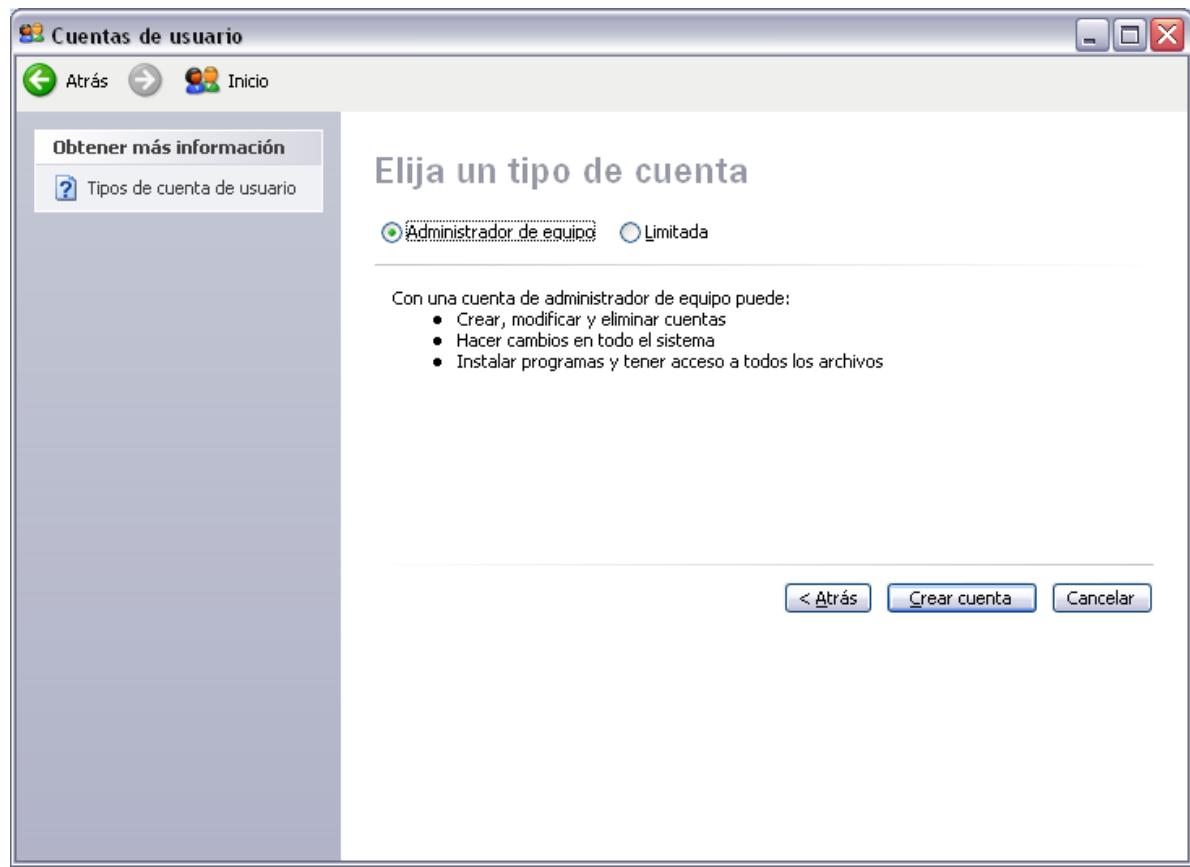
```
alter view v_todosproduc --view para crear vistas
as -- luego de estos se hace la consulta
select top 200 ProductID , ProductName
from products
order by ProductName
```

BORRADO DE VISTAS

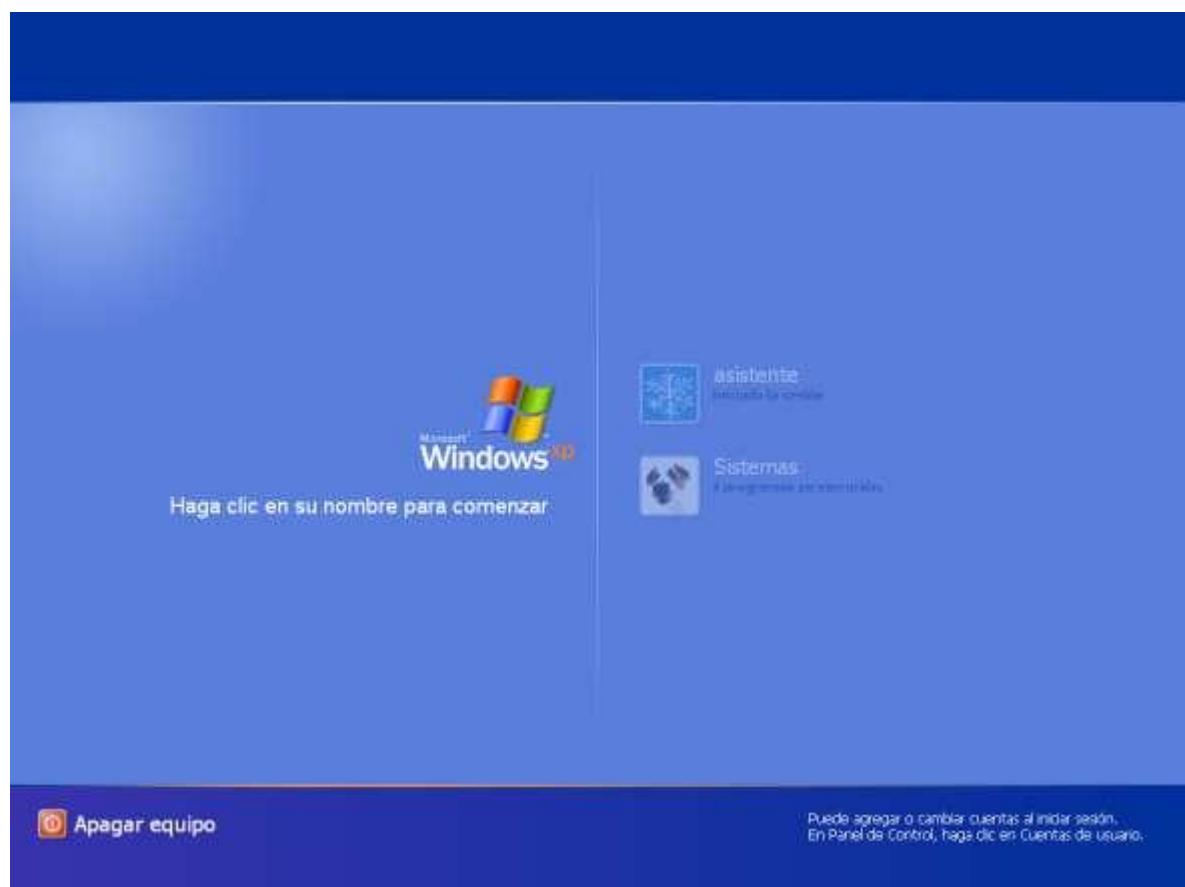
drop view [nombre de la vista] -- para borrar vistas

Creando Usuarios

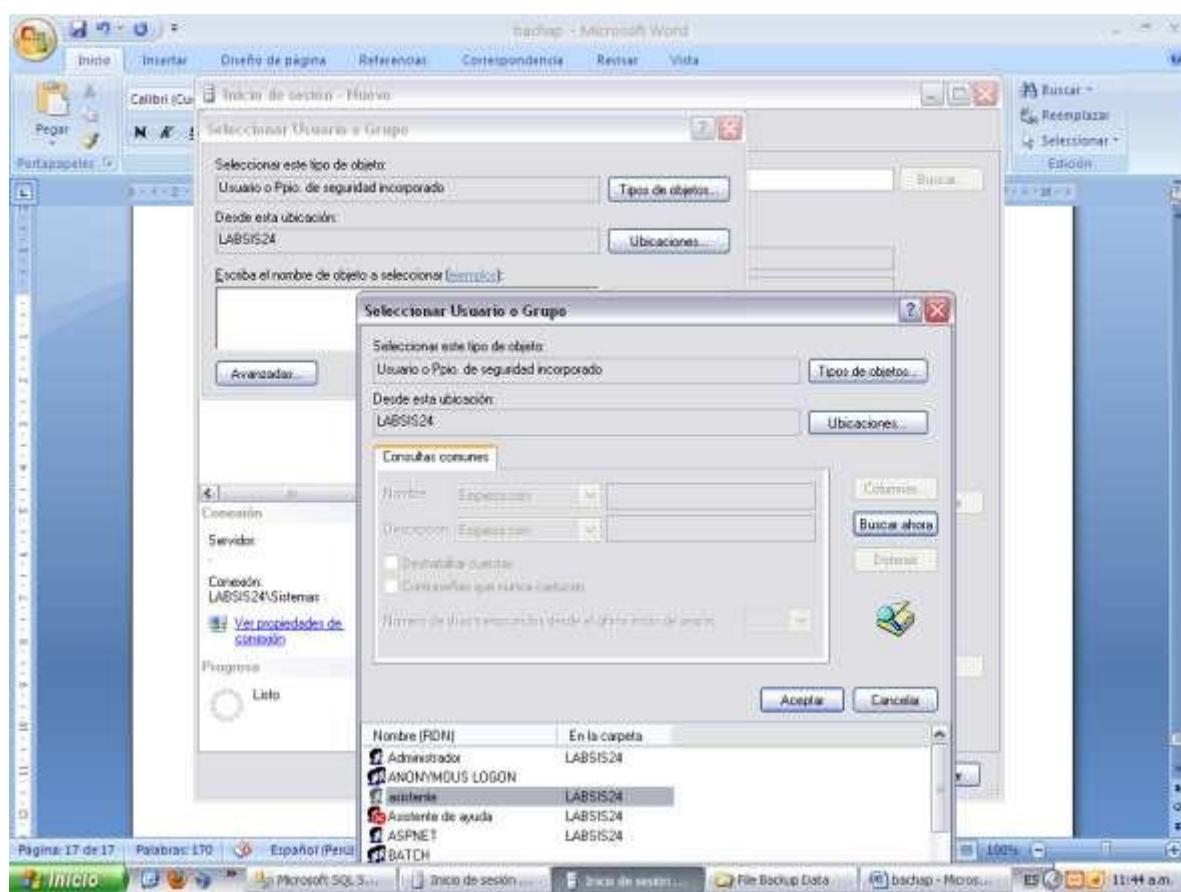
Creamos en el panel de control un nuevo usuario con el nombre de asistente

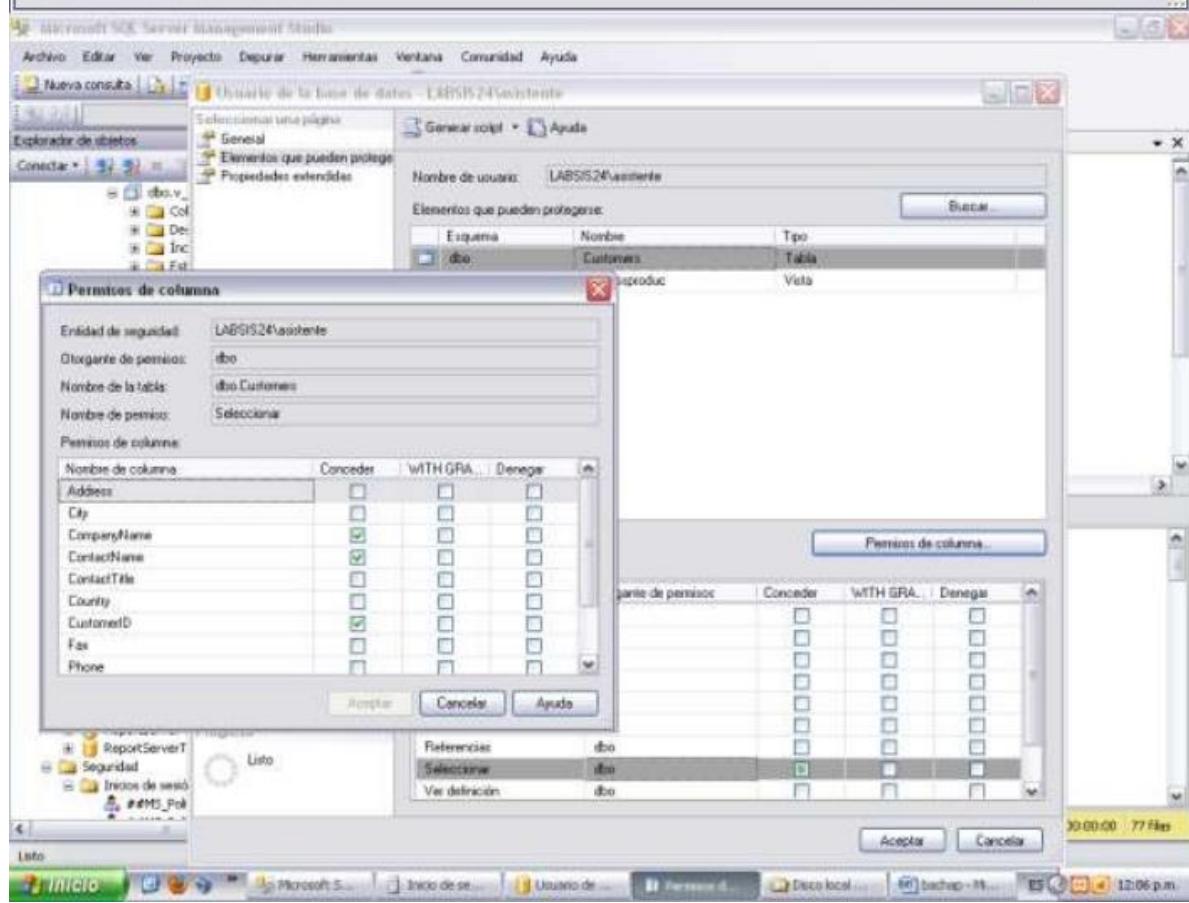
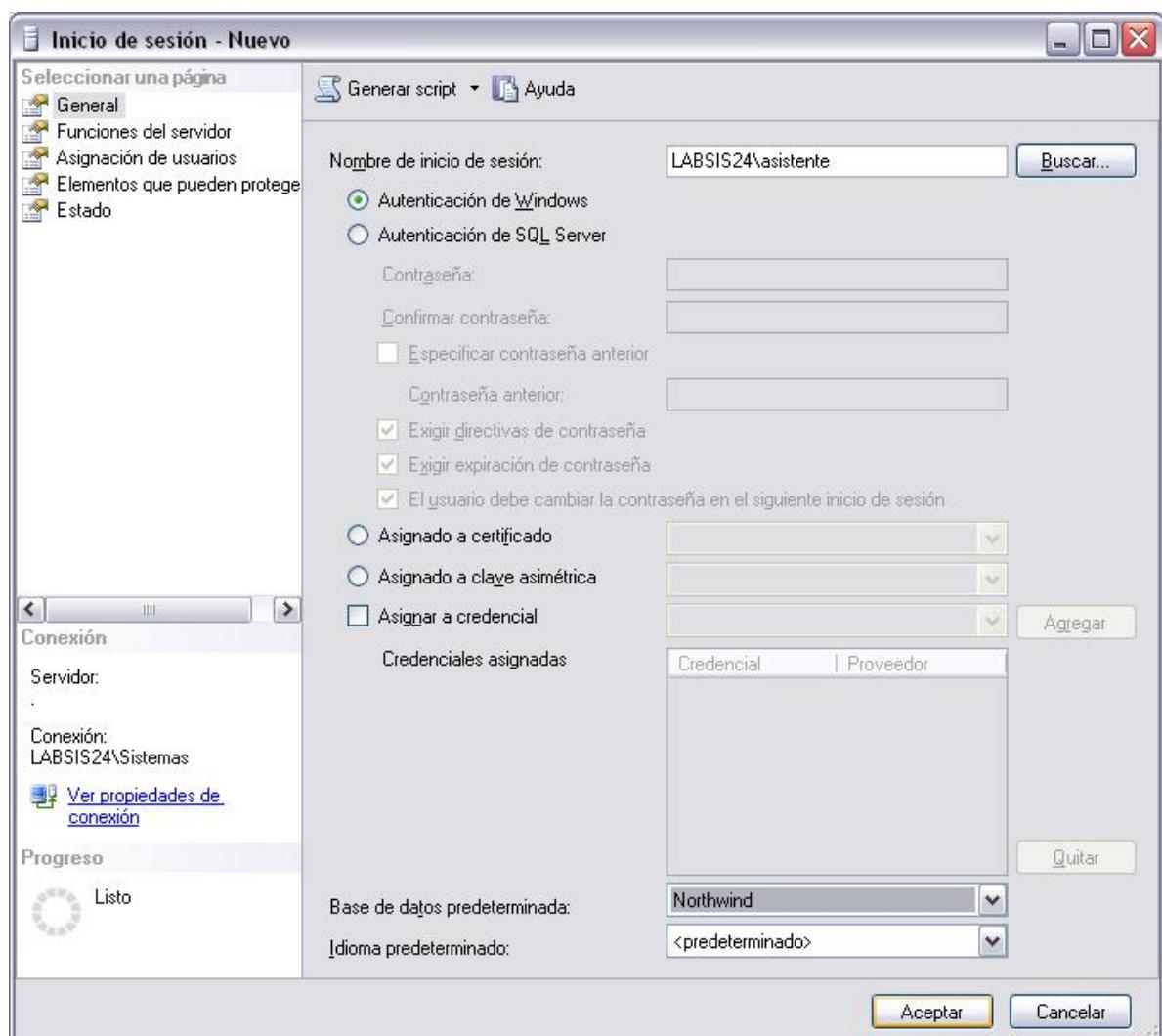


Crear un usuario de windows (en windows)

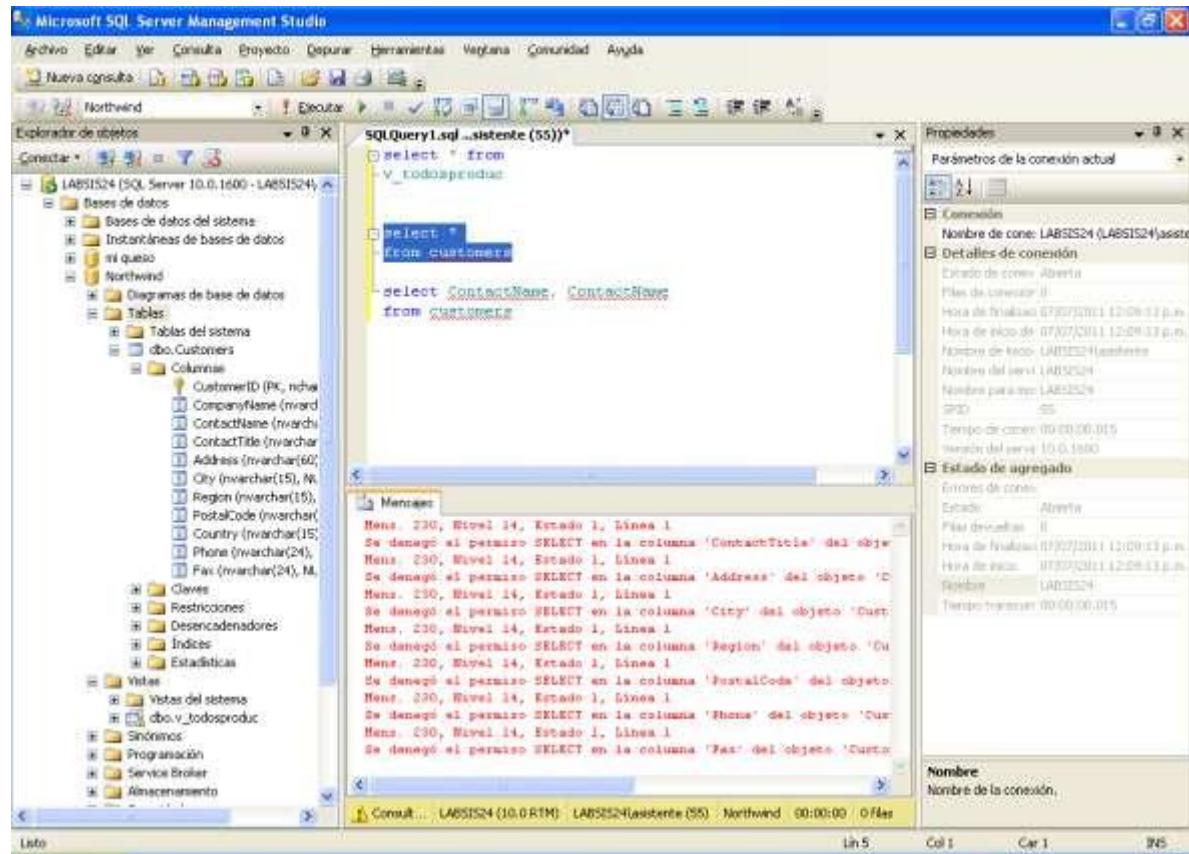


Dar permisos para ver la vista

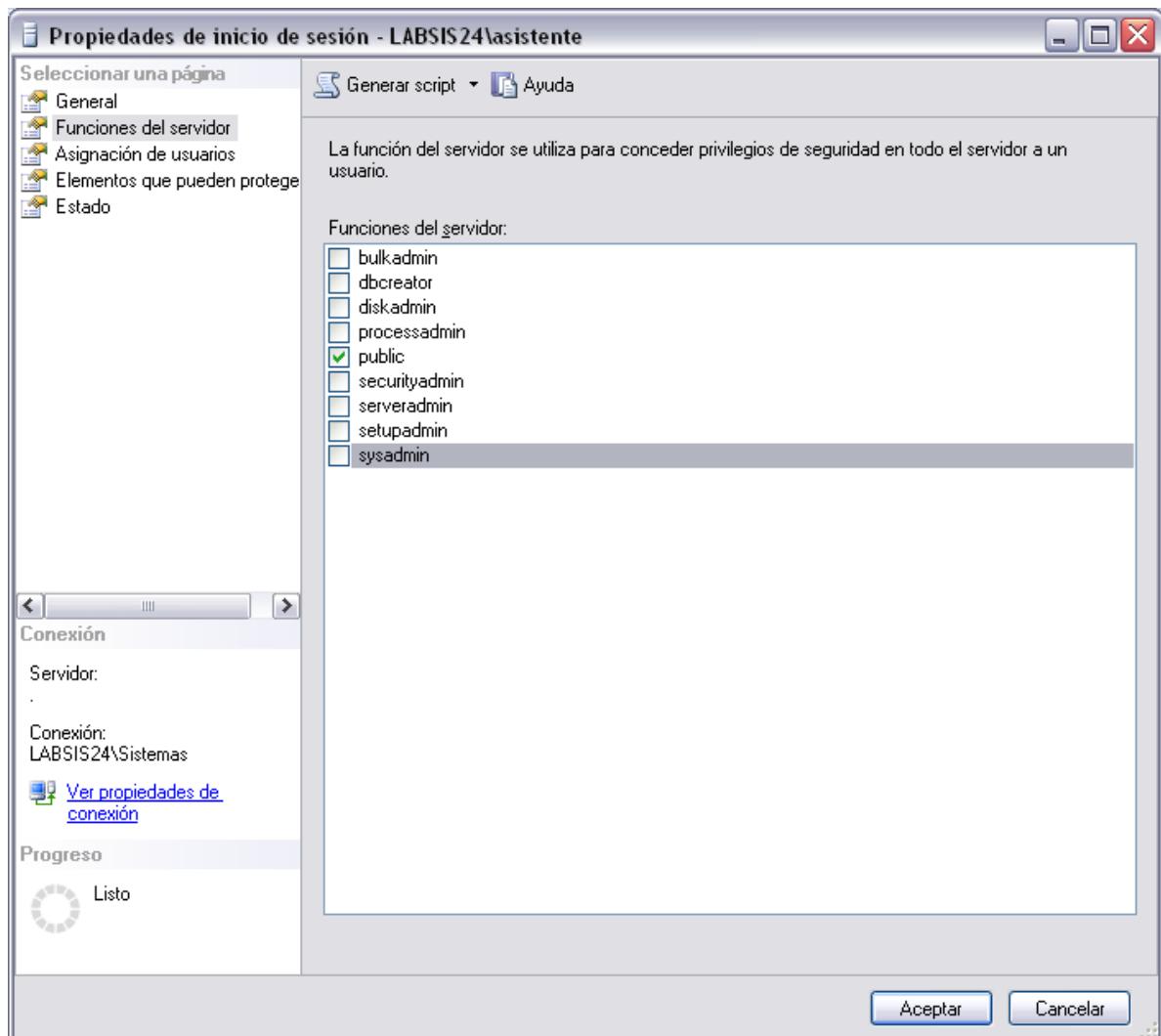




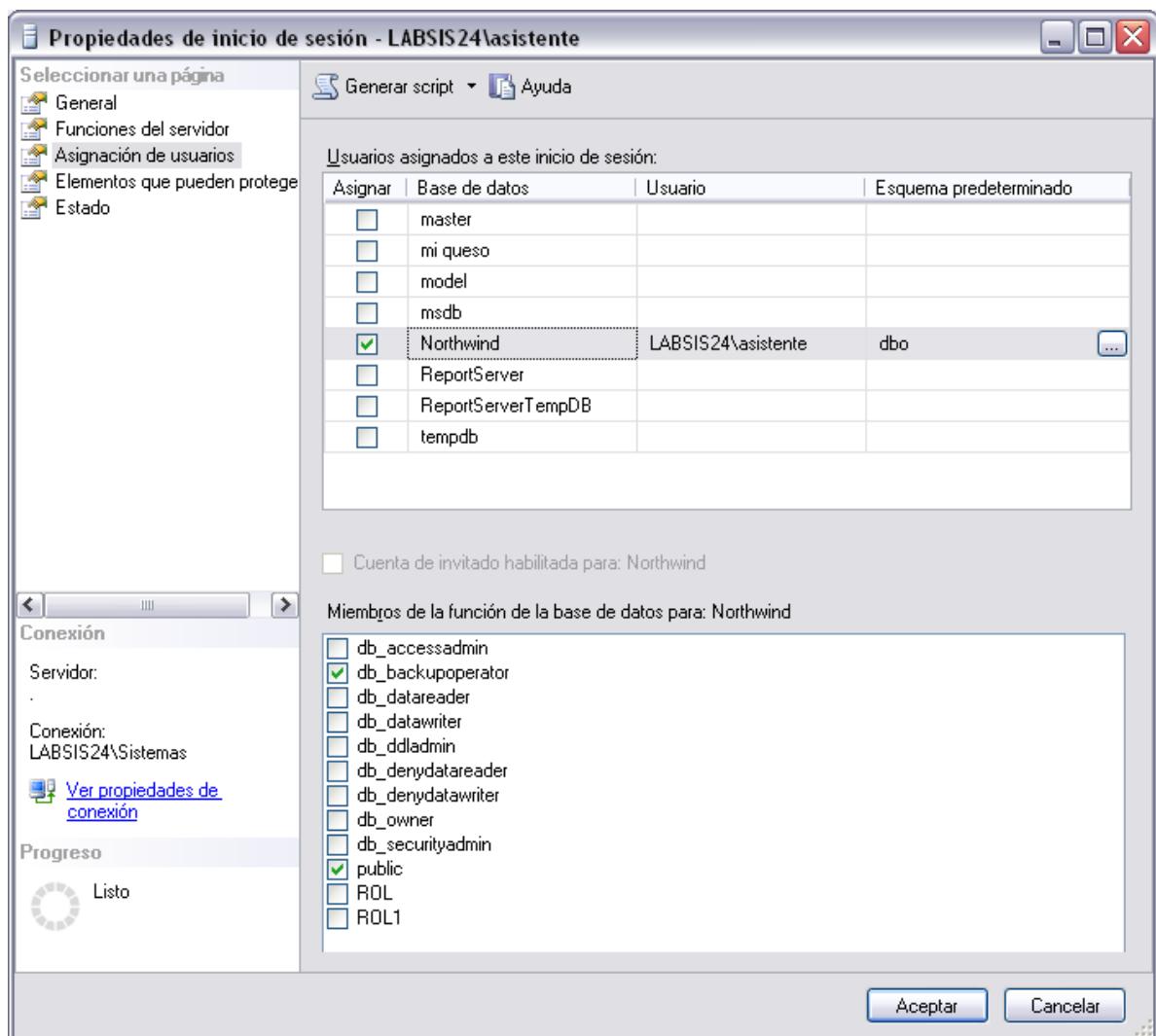
Ventanas del usuario

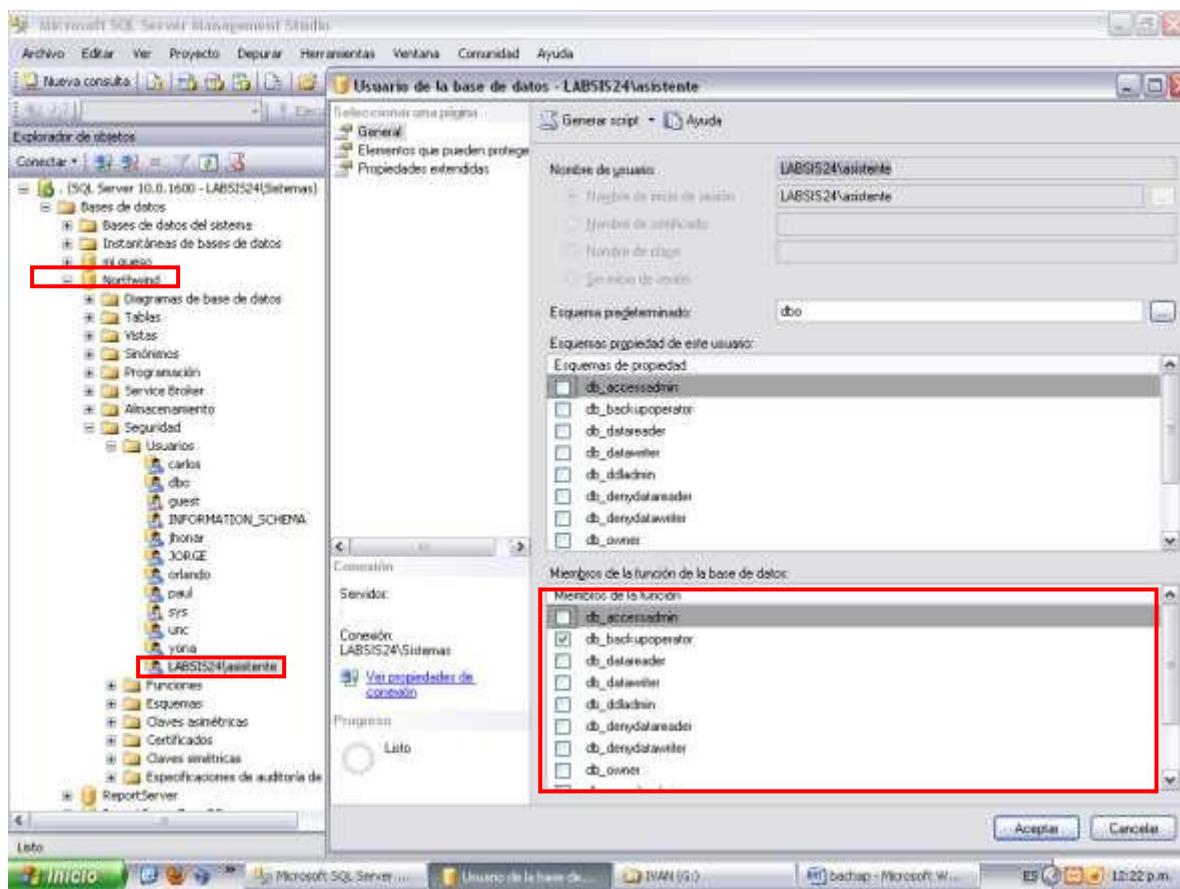


Restringimos solo para backup



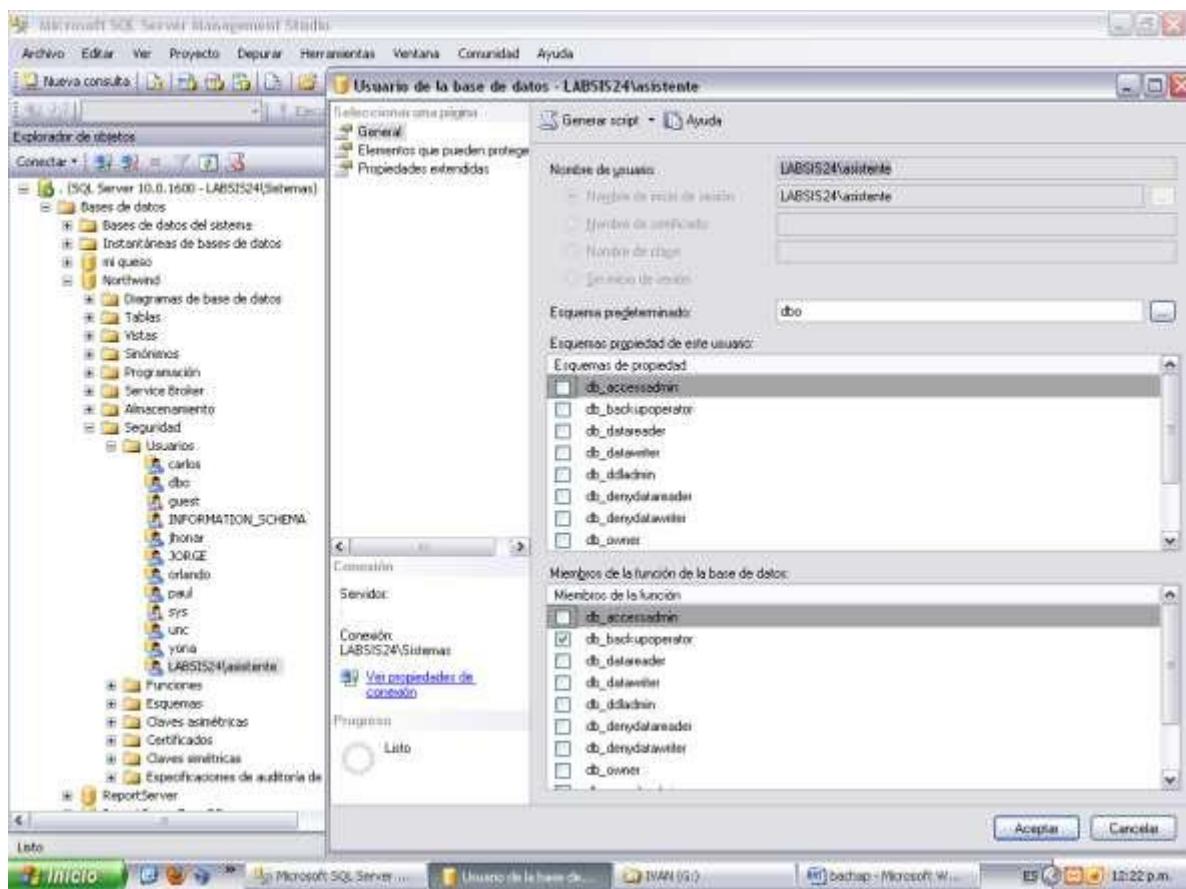
En esta opción se da el permiso de administrador





En esta ventaja se dan los permisos en la base de datos

ContactName	ContactName
Maria Anders	Maria Anders
Ana Trujillo	Ana Trujillo
Antonio Moreno	Antonio Moreno
Thomas Hardy	Thomas Hardy
Christina Berglund	Christina Berglund
Hanna Moos	Hanna Moos
Frédérique Céleste	Frédérique Céleste
Martin Sommer	Martin Sommer
Laurence Lebihan	Laurence Lebihan
Elizabeth Lincoln	Elizabeth Lincoln
Victoria Ashworth	Victoria Ashworth
Patricia SImpson	Patricia Simpso



Damos el permiso de administrador al usuario

Db_owner: tiene el permiso solo al objeto

Db_accessadmin: tiene el permiso de administrador

```

/* crear un procedimiento almacenado para crear login a partir de los
nombres de */
create proc usuarios
as

declare @c as int
declare @n as varchar(20)
declare @a as varchar(20)

set @c =1

while (@c<=9)
begin

    select @n=rtrim(ltrim.FirstName)) ,@a=rtrim(ltrim.LastName))
    from Employees
    where EmployeeID =@c

    exec sp_addlogin @n, @a
    set @c =@c+1

end

```

Eliminación de login

```

create proc eliminar
as
    declare @c as int
    declare @n as varchar(10)
    set @c=1

    select firstname from Employees
    while (@c<=9)
        begin
            select @n=rtrim(ltrim.FirstName))
            from Employees
            where EmployeeID =@c
            set @c=@c+1

            exec sp_droplogin @n

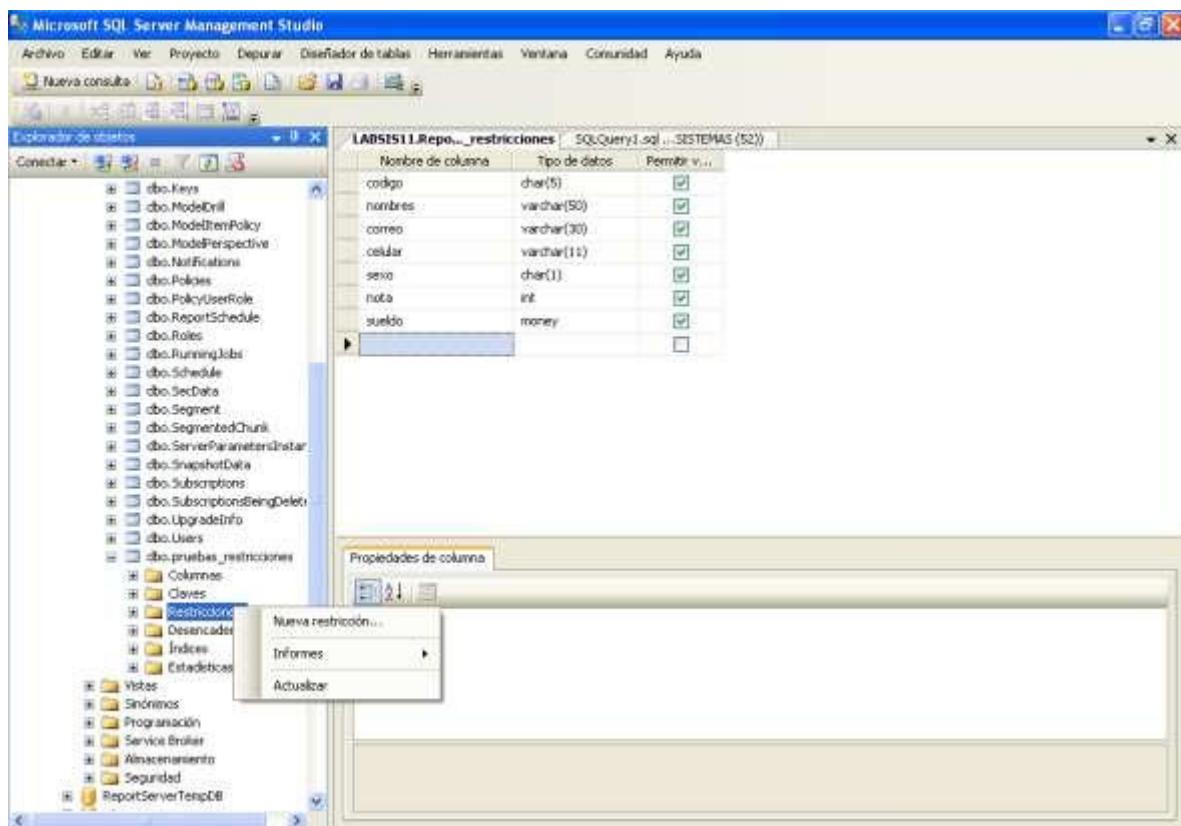
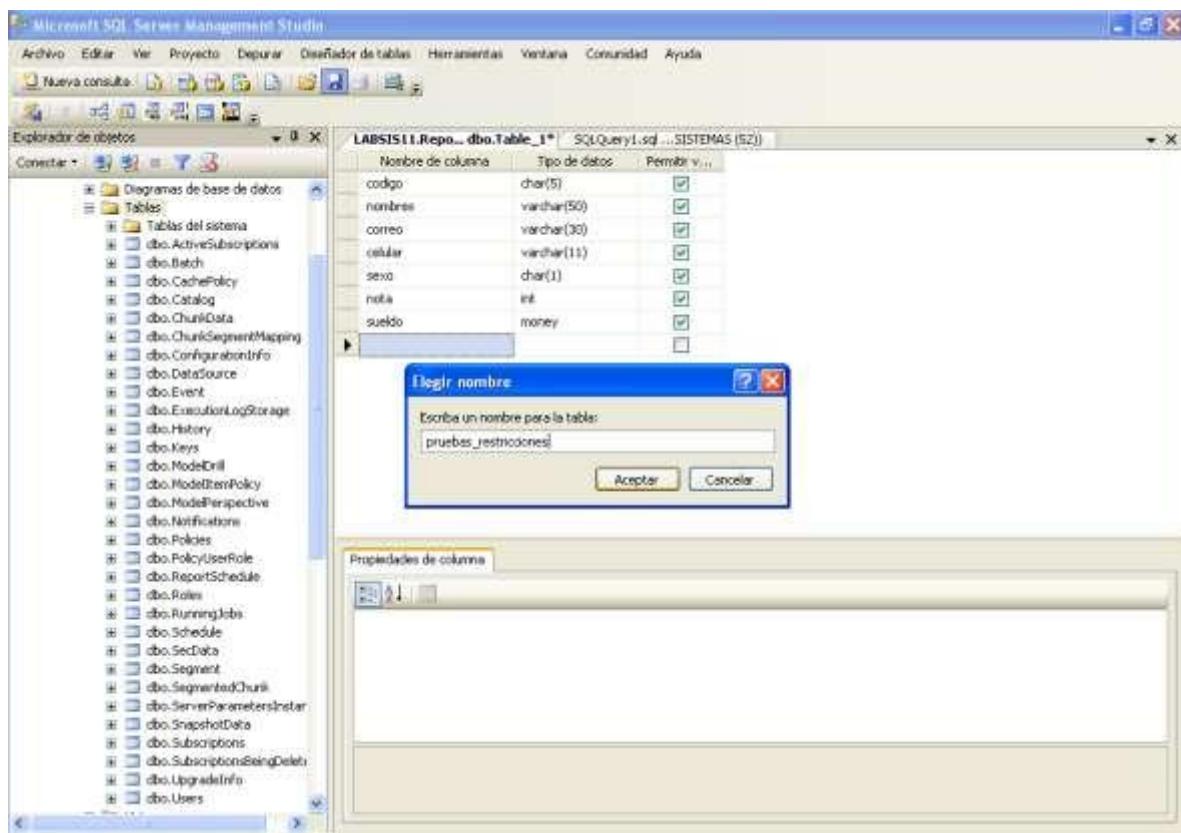
        end

```

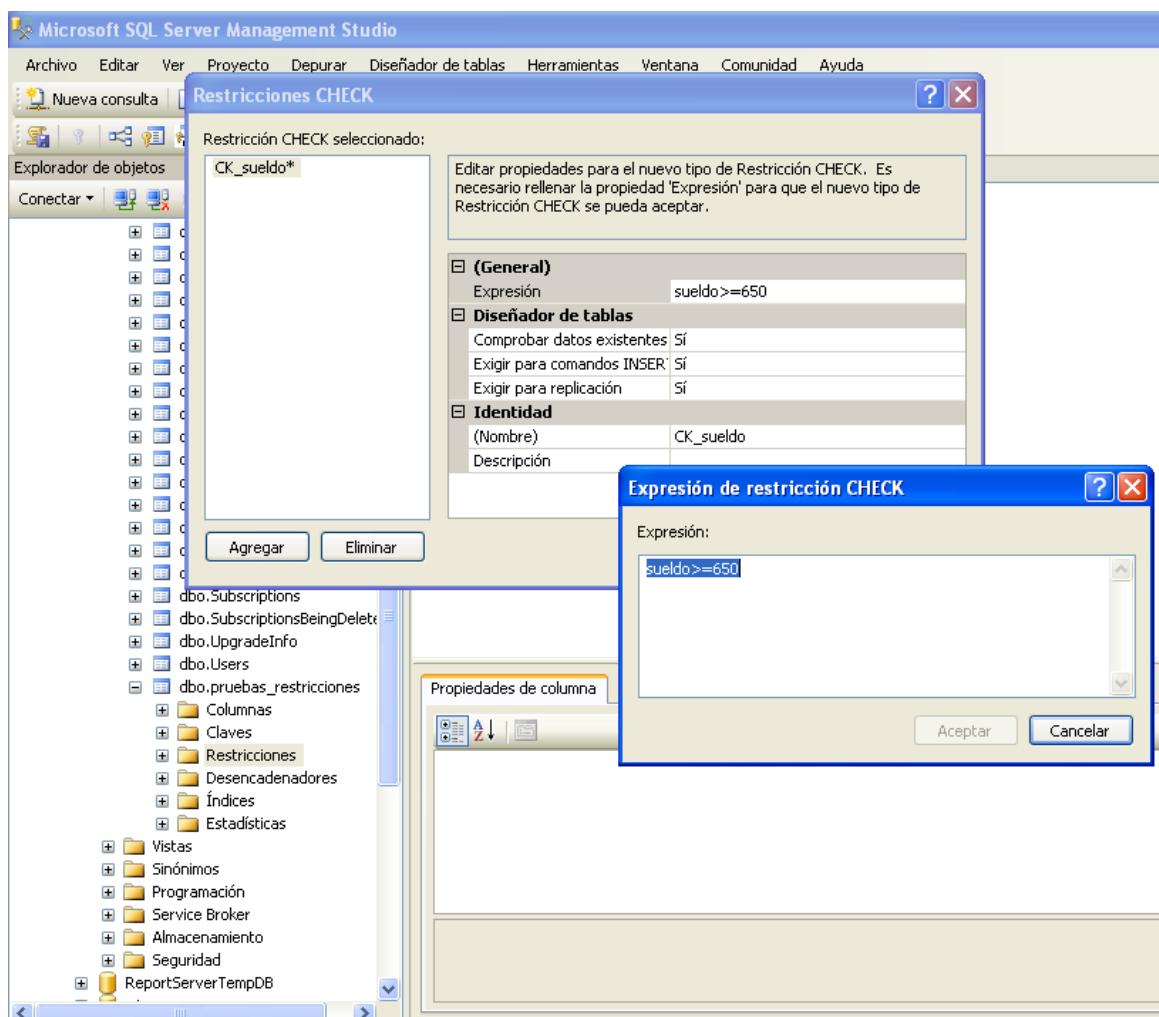
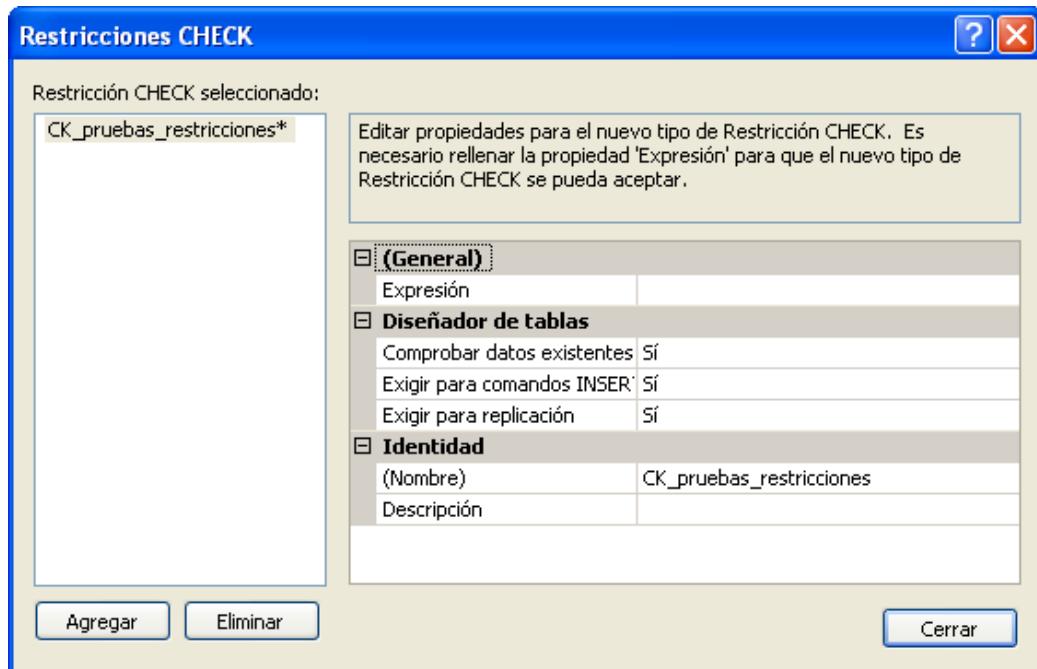
RESTRICCIONES

Cajamarca 13 de julio del 2011

Creamos una tabla



Restricciones: es un filtro al momento de ingreso de un dato

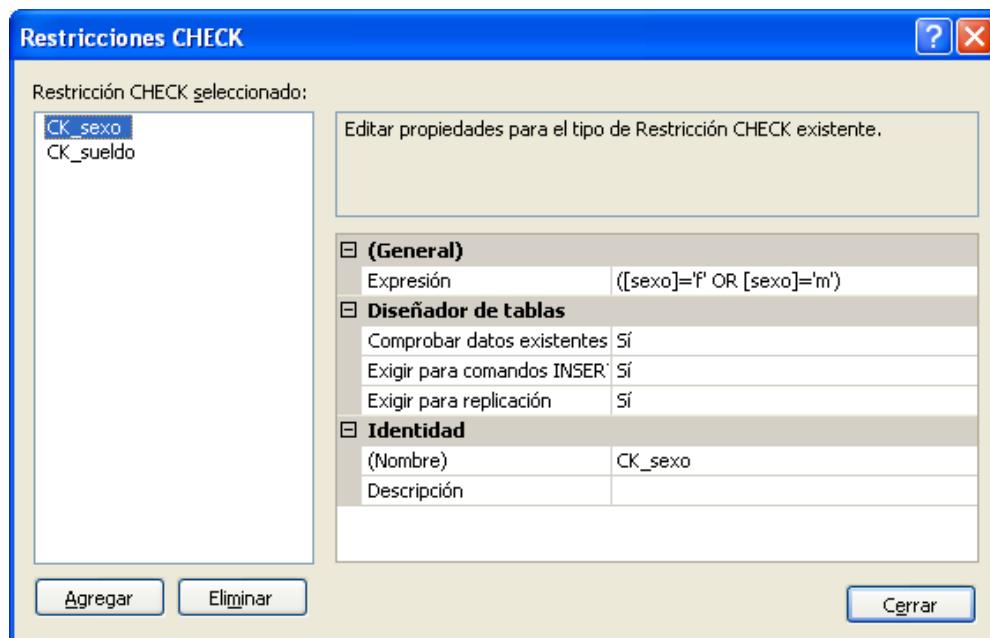


Luego guardar ya que no se actualiza automáticamente

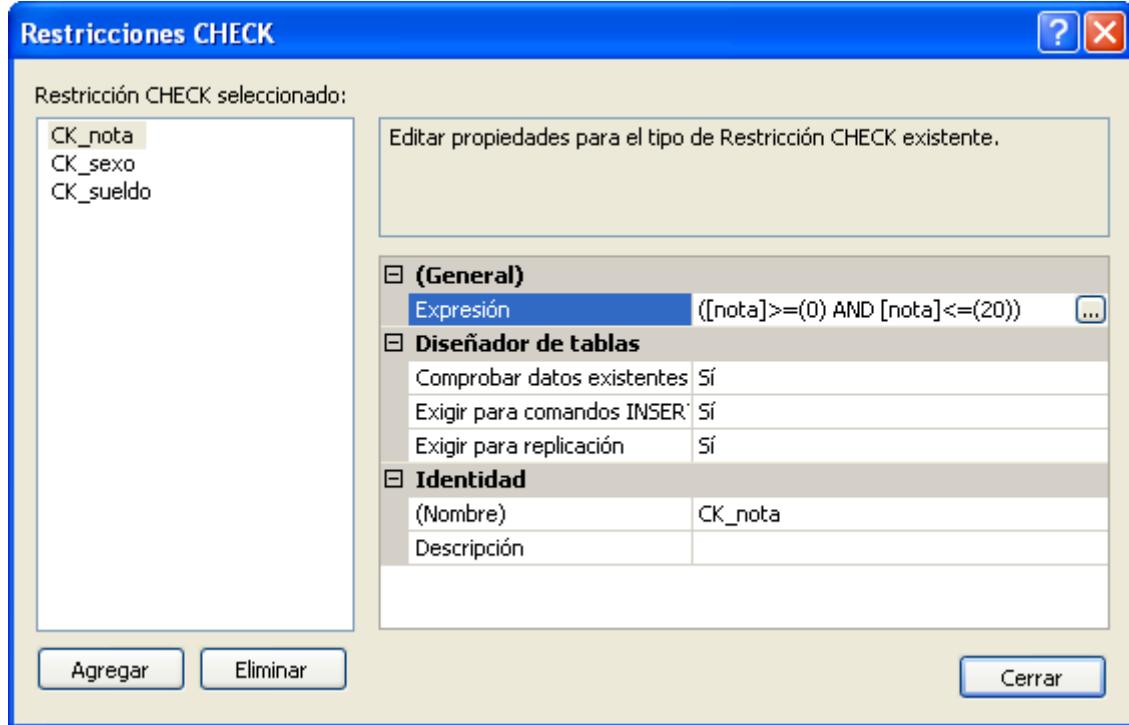
Luego le probamos llenando datos en la tabla

codigo	nombres	correo	celular	sexo	nota	sueldo
00001	ronald	roaldo	97632084	'm'	12	500
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Agregar restricción para el sexo



Sexo in ('F','M')



Creado para el celular a través del código

```
ALTER TABLE dbo.pruebas_restricciones  
ADD CONSTRAINT CK_CELULAR  
CHECK (celular like '519%')
```



Condicionando el ingreso del numero telefónico

```
ALTER TABLE dbo.pruebas_restricciones  
ADD CONSTRAINT CKCELULAR2  
CHECK (len(rtrim(celular))=11)
```

Para restringir solo a numeros

```
alter table pruebas_restricciones  
add constraint ck_tel_num  
check (celular like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]0-9][0-9]')
```

FUNCIONES

Es un sub programa que sirve para devolver un solo valor.

TIPOS

Funciones escalares

- o Similar a una función integrada

Funciones con valores de tabla de varias instrucciones (procedimientos almacenado)

- o Contenido como un procedimiento almacenado
- o Se hace referencia como una vista

Funciones con valores de tabla en línea (solo va un select)

- o Similar a una vista con parámetros
- o Devuelve una tabla como el resultado de una instrucción SELECT única

SINTAXIS DE CREACION DE UNA FUNCION

```
USE Northwind  
GO  
CREATE FUNCTION fn_NewRegion  
    (@myinput nvarchar(30))  
RETURNS nvarchar(30)  
BEGIN  
    IF @myinput IS NULL  
        SET @myinput = 'Not Applicable'  
    RETURN @myinput  
END
```

ELIMINACION DE UNA FUNCION

```
DROP FUNCTION dbo.fn_NewRegion
```

Ejemplos de funciones del sistema (rand, count, ltrim, rtrim, sen, etc)

FUNCIONES DEL SISTEMA

Definición de algunas funciones:

- ✓ Convertimos a mayúsculas con la función Upper

```
SELECT UPPER (CompanyName)  
FROM Suppliers
```

- ✓ Convertimos a minúsculas con la función Lower

```
SELECT LOWER (Suppliers.CompanyName)  
FROM Suppliers
```

- ✓ Podemos Borrar Los Espacios De Izquierda O Derecha

```
SELECT LTRIM('      RONALD')
```

```
SELECT RTRIM('RONALD      ')
```

- ✓ Podemos Medir La Longitud De Una Cadena

```
SELECT LEN(LastName), LastName FROM Employees
```

- ✓ Poder sustraer unos caracteres a partir de un carácter, ejemplo a partir del 4, más 3 caracteres.

```
SELECT SUBSTRING ('UNIVERSIDAD', 4, 3)
```

- ✓ Ahora buscamos la letra V en la palabra UNIVERSIDAD a partir del carácter 1.

```
SELECT CHARINDEX ('V', 'UNIVERSIDAD', 1)
```

- ✓ Sacar la fecha del sistema

```
SELECT GETDATE()
```

- ✓ Sumamos un año a la fecha del sistema

```
SELECT DATEADD(YY, 1, GETDATE())
```

- ✓ Sacamos una parte de la fecha

```
SELECT DATEPART(YY, GETDATE())
```

- ✓ Sacar la diferencia de dos fechas

```
SELECT DATEDIFF(YY, GETDATE(), DATEADD(YY, 1, GETDATE()))
```

- ✓ Sacar el nombre del mes de la fecha del sistema.

```
SELECT DATENAME(MM, GETDATE())
```

- ✓ Para sacar el nombre del día de una fecha.

```
SELECT DATENAME(DW, GETDATE())
```

- ✓ Preguntar si es una fecha.(devuelve 1 ó 0)

```
SELECT ISDATE(Orders.OrderDate)  
FROM Orders
```

- ✓ PREGUNTAR SI ES NÚMERO.

```
SELECT ISNUMERIC (521)
```

- ✓ Cuantos registros se vieron afectados en la última instrucción SQL.

```
SELECT @@ROWCOUNT
```

Las variables que empiezan con @@ son variables del servidor.

- ✓ Para ver el nombre del servidor.

```
SELECT @@Servername
```

FUNCIONES DEFINIDAS POR EL USUARIO

❖ FUNCIONES ESCALARES

Solo se utiliza para un sólo valor.

- ✓ Para calcular el igv

```
CREATE FUNCTION FN_IGV
(@MONTO MONEY)
RETURNS MONEY
AS
BEGIN
    DECLARE @IGV MONEY
    SET @IGV = @MONTO * 0.19
    RETURN @IGV
END
```

--PARA EJECUTAR LA FUNCION

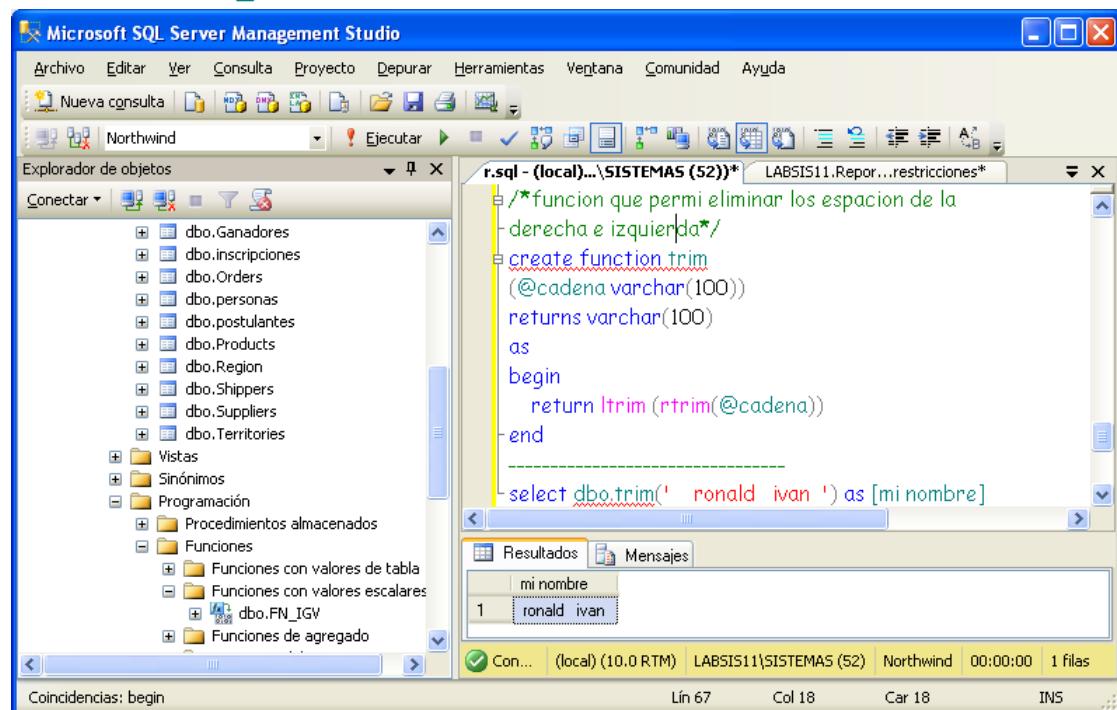
```
SELECT UnitPrice ,DBO.FN_IGV(UnitPrice) AS 'IGV'
FROM Products
```

- ✓ Borrar los espacios de derecha e izquierda

```
CREATE FUNCTION FN_ESPACIOS
(@TEXTO VARCHAR(50))
RETURNS VARCHAR(30)
AS
BEGIN
    RETURN LTRIM(RTRIM(@TEXTO))
END
```

--PARA EJECUTAR LA FUNCION

```
SELECT dbo.FN_ESPACIOS('      RONALD IVAN      ')
```



- ✓ CREAR FUNCION PARA CAMBIAR DE SOLES A DOLARES

```
CREATE FUNCTION FN_PRECIO  
(@MD MONEY, @TC MONEY)  
RETURNS MONEY  
AS  
BEGIN  
DECLARE @MS MONEY  
SET @MS=@MD * @TC  
RETURN @MS  
END
```

```
--PARA EJECUTAR LA FUNCION
```

```
SELECT (UnitPrice), DBO.FN_PRECIO(UnitPrice, 2.84)  
FROM Products
```

✓ CREAR UNA FUNCION CUANDO EL USUARIO Y LA CLAVE ES CORRECTA.

```
CREATE FUNCTION FN_VERIUSU
(@U VARCHAR(15), @C VARCHAR(20))
RETURNS INT
AS
BEGIN
    DECLARE @NR INT
    SELECT @NR=COUNT(*)
    FROM dbo.USUARIO_DB
    WHERE USUARIO=@U AND CLAVE=@C

    RETURN @NR
END

--PARA EJECUTAR LA FUNCION

SELECT dbo.FN_VERIUSU ('IVAN', '12345')
```

✓ PARA CAMBIAR LA CLAVE

```
CREATE PROC FN_CAMBIOCLAV
(@U VARCHAR(15), @C VARCHAR(20), @C_N VARCHAR(20))
AS
BEGIN
    UPDATE dbo.USUARIO_DB
    SET CLAVE =@C_N
    WHERE USUARIO=@U AND CLAVE=@C

    IF (@@ROWCOUNT = 1)
        RAISERROR('SE CAMBIO LA CLAVE',10,1)
    ELSE
        RAISERROR('USUARIO O CLAVE INCORRECTO',10,1)
END

FN_CAMBIOCLAV 'IVAN', '12345', '0521'
```

❖ FUNCIÓN CON VALORES DE TABLA DE VARIAS INSTRUCCIONES

1. Crear una función que ingrese “C” (corto) si se desea visualizar solo el ID y apellido del Empleado, y “L” (Largo) si se desea visualizar id, nombre y apellido del Empleado.

```
CREATE FUNCTION fn_EJER1
(@tipo char(1))
RETURNS @fn_Empleadados TABLE
    (ID int PRIMARY KEY NOT NULL,
     nombre nvarchar(61) NOT NULL)
AS
BEGIN
    IF upper(@tipo) = 'C'
        INSERT @fn_Empleadados SELECT EmployeeID, LastName
        FROM Employees
    ELSE IF upper(@tipo) = 'L'
        INSERT @fn_Empleadados SELECT EmployeeID,
        (FirstName + ' ' + LastName)
        FROM Employees
```

```

    RETURN
END
-----
```

```
select * from dbo.fn_EJER1('c')
```

	ID	nombre
1	1	Davolio
2	2	Fuller
3	3	Leverling
4	4	Peacock
5	5	Buchanan
6	6	Suyama
7	7	King
8	8	Callahan
9	9	Dodsworth

2. Crear una función que ingrese "C" (Clientes), "P" (Proveedores) y "E" (Empleados) y muestre el Nombre Completo.

```

CREATE FUNCTION fn_EJER2 (@tipo char(1))
RETURNS @fn_personas TABLE
(nombre varchar(61) NOT NULL)
AS
BEGIN
IF upper(@tipo) = 'C'
    INSERT @fn_personas SELECT Companyname
    FROM Customers
ELSE IF upper(@tipo) = 'P'
    INSERT @fn_personas SELECT Companyname
    FROM SUPPLIERS
ELSE IF UPPER(@TIPO) = 'E'
    INSERT @fn_personas SELECT FIRSTNAME + ' ' + LASTNAME
    FROM EMPLOYEES

RETURN
END
```

```
select * from dbo.fn_EJER2('C')
```

	nombre
1	Alfreds Futterkiste
2	Ana Trujillo Emparedados y helados
3	Antonio Moreno Taquería
4	Around the Horn
5	Berglunds snabbköp
6	Blauer See Delikatessen
7	Blondesddsl père et fils
8	Bólido Comidas preparadas
9	Bon app'
10	Bottom-Dollar Markets
11	B's Beverages

	nombre
1	Nancy Davolio
2	Andrew Fuller
3	Janet Leverling
4	Margaret Peacock
5	Steven Buchanan
6	Michael Suyama
7	Robert King
8	Laura Callahan
9	Anne Dodsworth

3. CREAR UNA FUNCION QUE SELECCIONE ALEATORIAMENTE UN PROVEEDOR (función escalar)

```
--crear una vista que almacene el numero aleatorio

create view v_rand
as
select rand() as number

select rand()

-- FUNCION ALEATORIO

create function fn_aleatorio
()
returns varchar(20)
as
begin
    declare @nreg int,@reg varchar(20),@rnd float

    select @nreg=count(*) from suppliers --# de registros

    set @rnd=(select number from v_rand) --# aleatorio de la vista

    select @reg=companyname from suppliers
    where supplierid=cast((@rnd*@nreg+1) as int)

    return @reg

end

select dbo.fn_aleatorio()
```

4. Crear una función que nos permita seleccionar aleatoriamente --“N” registro de la tabla Products.

```
--crear una vista que almacene el numero aleatorio

create view v_r
as
select rand() as number

select rand()

--*****FUNCION*****

create function FN_ALEAN
(@num int)
RETURNS @prod TABLE
(codigo int,
nombre nvarchar(40)
)
AS
BEGIN
declare @reg_azar int, @i int, @j int, @nr int
set @i=0
```

```

        select @nr=COUNT(*) from products
        while @i<@num
            begin
                SET @reg_azar=CAST((select number from v_r)*@nr+1 as
int)
                    SELECT @j= COUNT (*) FROM @prod
                    WHERE codigo = @reg_azar
                    IF @j=0
                    BEGIN
                        SET @i=@i+1
                        INSERT @prod SELECT productid,productname
                        FROM products
                        WHERE productid=@reg_azar
                    END
                END
            RETURN
        END

    PARA EJECUTAR

    SELECT *
    FROM FN_ALEAN(5)

```

```

select *
from Customers

select @@ROWCOUNT (devuelve el numero de filas que se mostraron con el select)

```

Variables de servidores son los que empiezan con dos @@

@@rowcount

@@..... Buscar que funcione permite ver al usuario logueado

```

SQLQuery1.sql - USUARIO\...l...r (53) *
select *
from Customers

select @@ROWCOUNT

```

	CustomerID	CompanyName	ContactName	ContactTitle	Address
80	TORTU	Tortuga Restaurante	Miguel Angel Pa...	Owner	Avda.
81	TRADH	Tradição Hipermercados	Anabela Doming...	Sales Representative	Av. In
82	TRAIH	Trail's Head Gourmet Provisioners	Helvetius Nagy	Sales Associate	722 D
83	VAFFE	Vaffeljernet	Palle Ibsen	Sales Manager	Smags
84	VICTE	Victuailles en stock	Mary Saveley	Sales Agent	2, rue
85	VINET	Vins et alcools Chevalier	Paul Henriot	Accounting Manager	59 rue
86	WANDK	Die Wandernde Kuh	Rita Müller	Sales Representative	Adena
87	WARTH	Wartian Herkku	Pirkko Koskitalo	Accounting Manager	Torika
88	WELLI	Wellington Importadora	Paula Parente	Sales Manager	Rua d
89	WHITC	White Clover Markets	Karl Jablonski	Owner	305 -
90	WILMK	Wilman Kala	Matti Karttunen	Owner/Marketing A...	Kesku
91	WOLZA	Wolski Zajazd	Zbyszek Piestrze...	Owner	ul. Filtr

	(Sin nombre de columna)
1	91

TRIGGER

Un trigger (o disparador) en una Base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE). Algunas bases de datos pueden ejecutar triggers al crear, borrar o editar usuarios, tablas, bases de datos u otros objetos.

ejemplos

1. PERMITIR ELIMINAR UN SOLO REGISTRO DE LA TABLA ORDER DETAILS.

```
CREATE TRIGGER BorrarOrd ON [order details]
FOR DELETE
AS
IF (SELECT COUNT(*) FROM Deleted) > 1
BEGIN
    RAISERROR('Solo puede eliminar un registro a la vez', 16, 1)
    ROLLBACK TRANSACTION
END
-----
delete from [Order Details]
where OrderID = 10529
```

The screenshot shows a SQL Server Management Studio window titled "SQLQuery1.sql - USUARIO\...r (51)". It contains the following T-SQL code:

```
alter TRIGGER BorrarOrd ON [order details]
FOR DELETE
AS
IF (SELECT COUNT(*) FROM Deleted) > 1
BEGIN
    RAISERROR('Solo puede eliminar un registro a la vez', 16, 1)
    ROLLBACK TRANSACTION
END

delete from [Order Details]
where OrderID = 10529
```

Below the code, there is a "Mensajes" (Messages) pane with the following output:

```
Solo puede eliminar un registro a la vez
Mens. 3609, Nivel 16, Estado 1, Línea 1
La transacción terminó en el desencadenador. Se anuló el lote.
```

2. CUANDO SE INSERTE UN REGISTRO EN LA TABLA ORDER DETAILS, LA CANTIDAD PEDIDA SE DEBE RESTAR EN EL CAMPO UNITSINSTOCK DE LA TABLA PRODUCTS

```
CREATE TRIGGER A9
ON [ORDER DETAILS]
FOR INSERT
AS
UPDATE P
SET UNITSINSTOCK = (P.UNITSINSTOCK - I.QUANTITY)
```

```
FROM PRODUCTS AS P
INNER JOIN INSERTED AS I
    ON P.PRODUCTID = I.PRODUCTID
```

3. CUANDO SE BORRE UN DETALLE DE LA ORDEN, EL PRODUCTO QUE ESTABA PEDIDO EN LA ORDEN SE ACTUALIZARA A DESCONTINUADO EN LA TABLA PRODUCTS.

```
CREATE TRIGGER A3
ON [ORDER DETAILS]
FOR DELETE
AS
    UPDATE PRODUCTS
    SET DISCONTINUED = 1
    FROM PRODUCTS AS P
    INNER JOIN DELETED AS D
    ON P.PRODUCTID = D.PRODUCTID
```

```
-----  
DELETE FROM [ORDER DETAILS]  
WHERE ORDERID = 10333 AND PRODUCTID = 1
```

The screenshot shows the SSMS interface with a query window titled 'SQLQuery1.sql - USUARIO\...r (51)'. The window contains two parts of T-SQL code. The first part is a trigger definition:

```
CREATE TRIGGER A3
ON [ORDER DETAILS]
FOR DELETE
AS
    UPDATE PRODUCTS
    SET DISCONTINUED = 1
    FROM PRODUCTS AS P
    INNER JOIN DELETED AS D
    ON P.PRODUCTID = D.PRODUCTID
```

The second part is a delete statement:

```
-----  
DELETE FROM [ORDER DETAILS]  
WHERE ORDERID = 10333 AND PRODUCTID = 1
```

Below the code, the 'Mensajes' (Messages) pane displays two rows of output:

```
(0 filas afectadas)  
(0 filas afectadas)
```

4. CUANDO SE QUIERA ACTUALIZAR EL NOMBRE DE LA COMPAÑIA DEL PROVEEDOR, NO DEBE PERMITIRLO

```
CREATE TRIGGER A5
ON SUPPLIERS
FOR UPDATE
AS
    IF UPDATE (COMPANYNAME)
        BEGIN
            RAISERROR ('NO PUEDE CAMBIAR EL NOMBRE DE LA
COMPAÑIA',10,1)
            ROLLBACK TRANSACTION
```

```
END
```

```
UPDATE SUPPLIERS  
SET COMPANYNAME = 'JUAN'  
WHERE SUPPLIERID = 2
```

5. CREAR UNA TABLA HISTORICO QUE VAYA GUARDANDO EL NOMBRE DE LA COMPAÑIA (ACTUAL NOMBRE Y NUEVO NOMBRE) DE LOS PROVEEDORES. TABLA: HISTORICO (ID, ACTUAL, NUEVO) EL ID ES IDENTITY.

```
CREATE TRIGGER A6  
ON SUPPLIERS  
FOR UPDATE  
AS  
IF UPDATE (COMPANYNAME)  
BEGIN  
DECLARE @C_ACT VARCHAR(50)  
DECLARE @C_NEW VARCHAR(50)  
  
SELECT @C_ACT=COMPANYNAME FROM DELETED  
SELECT @C_NEW=COMPANYNAME FROM INSERTED  
  
INSERT INTO HISTORICO (ACTUAL, NUEVO)  
VALUES (@C_ACT, @C_NEW)  
END
```

```
UPDATE SUPPLIERS  
SET COMPANYNAME = 'JUAN'  
WHERE SUPPLIERID = 2
```