1. **Muestra por pantalla, usando variables, la suma de los empleados y los clientes de la base de datos jardinería**

USE Jardineria

GO

DECLARE @NumeroClientes INT, @NumeroEmpleados INT

SELECT @NumeroEmpleados = SUM(Empleados.CodigoEmpleado) FROM empleados

SELECT @NumeroClientes = SUM(Clientes.CodigoCliente) FROM Clientes

SELECT @NumeroEmpleados, @NumeroClientes;

CORRECCION GEMMA

USE Jardineria

GO

DECLARE @NumeroClientes INT, @NumeroEmpleados INT

SELECT @NumeroEmpleados = COUNT(Empleados.CodigoEmpleado) FROM empleados

SELECT @NumeroClientes = COUNT(Clientes.CodigoCliente) FROM Clientes

SELECT @NumeroEmpleados AS EMPLEADOS , @NumeroClientes AS CLIENTES

O

USE Jardineria

GO

DECLARE @NumeroClientes INT, @NumeroEmpleados INT

SELECT @NumeroEmpleados = COUNT(Empleados.CodigoEmpleado) FROM empleados

SELECT @NumeroClientes = COUNT(Clientes.CodigoCliente) FROM Clientes

PRINT @NumeroEmpleados + @NumeroClientes;

1. **Mostrar por pantalla, usando variables, el nombre del cliente y su límite de crédito de aquel cliente cuyo representante de ventas sea Walter Santiago que tenga menor límite de crédito.**

USE Jardineria

GO

DECLARE @NombreCliente VARCHAR (50), @Limite NUMERIC (15,2)

SELECT @NombreCliente, @Limite FROM Clientes WHERE Clientes.CodigoEmpleadoRepVentas = (SELECT Empleados.CodigoEmpleado FROM Empleados WHERE

Empleados.Nombre LIKE 'Walter' AND Apellido1 LIKE 'Santiago')

SELECT @NombreCliente, @Limite

GROUP BY @NombreCliente

HAVING MIN(LimiteCredito);

CORRECCION GEMMA

USE Jardineria

GO

DECLARE @NombreCliente VARCHAR (50), @Limite NUMERIC (15,2), @LimiteMin NUMERIC (15,2)

SELECT @LimiteMin = MIN(LimiteCredito) FROM Clientes WHERE Clientes.CodigoEmpleadoRepVentas =

(SELECT Empleados.CodigoEmpleado FROM Empleados WHERE Empleados.Nombre LIKE 'Walter Santiago')

SELECT @NombreCliente = NombreCliente FROM Clientes WHERE Clientes.CodigoEmpleadoRepVentas =

(SELECT Empleados.CodigoEmpleado FROM Empleados WHERE Empleados.Nombre LIKE 'Walter Santiago')

SELECT @Limite = LimiteCredito FROM Clientes WHERE Clientes.CodigoEmpleadoRepVentas =

(SELECT Empleados.CodigoEmpleado FROM Empleados WHERE Empleados.Nombre LIKE 'Walter Santiago')

1. **Define una variable de tipo tabla y rellénala con los jugadores de la conferencia East. Guarda también el nombre del equipo. Muestra su contenido por pantalla.**

CORRECCION GEMMA

USE Nba

GO

DECLARE @Tabla\_Jugadores\_Nba TABLE(nombre VARCHAR (50),equipo VARCHAR (50))

INSERT INTO @Tabla\_Jugadores\_Nba

SELECT Nombre\_equipo, Nombre

FROM jugadores

WHERE

Nombre\_equipo in (SELECT Nombre FROM equipos WHERE Conferencia = 'East')

SELECT \* FROM @Tabla\_Jugadores\_Nba

GO

1. **Ejercicio 6.3 pág 246 del libro de texto. Diseña un código que lea el valor de dos variables y escriba en pantalla la mayor de ellas**

DECLARE @Numero1 INT,@Numero2 INT;

SET @Numero1 = 30;

SET @Numero2 = 20;

IF @Numero1 > @Numero2

PRINT 'El mayor es ' + cast (@Numero1 AS CHAR);

ELSE

BEGIN

IF @Numero1 < @Numero2

PRINT 'El mayor es ' + cast (@Numero2 AS CHAR);

ELSE

PRINT 'Error en la operacion';

END ;

GO

EJERICIOS DE COMPROBACION

USE nba

GO

IF (SELECT puntos\_local FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2 ) >

(SELECT puntos\_visitante FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2) PRINT 'Raptors ha ganado el partido'

ELSE

BEGIN

IF (SELECT puntos\_local FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2) <

(SELECT puntos\_visitante FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2) PRINT 'Raptors ha perdido el partido'

ELSE

PRINT 'Raptors ha empatado';

END ;

GO

USE nba

GO

DECLARE @Puntos INT

IF (SELECT puntos\_local FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2 ) >

(SELECT puntos\_visitante FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2) PRINT 'Raptors ha ganado el partido por'

ELSE

BEGIN

IF (SELECT puntos\_local FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2) <

(SELECT puntos\_visitante FROM partidos WHERE equipo\_local LIKE 'Raptors' AND codigo = 2) PRINT 'Raptors ha perdido el partido y el equipo contrario ha sacado'

ELSE

PRINT 'Raptors ha empatado';

END ;

GO

1. **Ejercicio 6.4 pág 246 del libro de texto**

USE nba

GO

SELECT codigo ,equipo\_local, equipo\_visitante, puntos\_local,puntos\_visitante, temporada,

CASE

WHEN puntos\_local > puntos\_visitante THEN '1'

WHEN puntos\_local < puntos\_visitante THEN '2'

ELSE 'x'

END

FROM partidos

ORDER BY temporada;

GO

Correcion gemma

DECLARE @casa INT

DECLARE @fuera INT

DECLARE @resultado VARCHAR(50)

SET @casa = 5

SET @fuera = 0

SELECT @resultado =

CASE

WHEN @casa > @fuera THEN 'GOLES CASA:' + CAST(@casa AS CHAR) + 'goles fuera' + CAST(@fuera AS CHAR) + 'signo quiniela 1'

WHEN @casa < @fuera THEN 'GOLES CASA ' + CAST(@casa AS CHAR) + ' goles fuera ' + CAST(@fuera AS CHAR) + 'signo quiniela 2'

ELSE 'GOLES CASA' + CAST(@casa AS CHAR) + 'goles fuera' + CAST(@fuera AS CHAR) + 'signo quiniela X'

END

PRINT @resultado

1. **Ejercicio 6.5 pág 246 del libro de texto**

**CON IF ELSE**

DECLARE @Numero1 INT, @Numero2 INT, @Numero3 INT

SET @Numero1 = 4;

SET @Numero2 = 3;

SET @Numero3 = 3;

IF @Numero3 = 1

BEGIN

PRINT @Numero1 + @Numero2;

END

ELSE IF @Numero3 = 2

BEGIN

PRINT @Numero1 - @Numero2;

END

ELSE IF @Numero3 = 3

BEGIN

PRINT @Numero1 \* @Numero2;

END

ELSE

BEGIN

PRINT 'Error ,operación no permitida';

END

GO

**CON CASE WHEN**

DECLARE @Numero1 INT, @Numero2 INT, @OPERACION INT,@RESULTADO VARCHAR (50)

SET @Numero1 = 4;

SET @Numero2 = 3;

SET @OPERACION = 2;

SELECT @RESULTADO =

CASE

WHEN @OPERACION = 1 THEN CAST(@Numero1 AS VARCHAR) + ' + ' + CAST(@Numero2 AS VARCHAR) + ' = ' + CAST((@Numero1 + @Numero2) AS VARCHAR(50))

WHEN @OPERACION = 2 THEN CAST(@Numero1 AS VARCHAR) + ' - ' + CAST(@Numero2 AS VARCHAR) + ' = ' + CAST((@Numero1 - @Numero2) AS VARCHAR(50))

WHEN @OPERACION = 3 THEN CAST(@Numero1 AS VARCHAR) + ' \* ' + CAST(@Numero2 AS VARCHAR) + ' = ' + CAST((@Numero1 \* @Numero2) AS VARCHAR(50))

ELSE 'Error ,operación no permitida'

END

PRINT @RESULTADO

1. **Implementar un programa que permita listar los 100 primeros números enteros, visualizando en cada caso si es par o impar.**

DECLARE @Numero INT

SET @Numero = 1

WHILE (@Numero<=100)

BEGIN

IF (@Numero%2!=0)

PRINT CAST(@Numero AS VARCHAR) + ' IMPAR'

ELSE IF (@Numero%2 = 0)

PRINT CAST(@Numero AS VARCHAR) + ' PAR'

SET @Numero=@Numero+1;

IF @Numero = 101

BREAK

END

GO

**O correcion gemma**

DECLARE @Numero INT

SET @Numero = 1

WHILE (@Numero<=100)

BEGIN

IF (@Numero%2!=0)

begin

PRINT CAST(@Numero AS VARCHAR) + ' IMPAR'

end

ELSE IF (@Numero%2 = 0)

begin

PRINT CAST(@Numero AS VARCHAR) + ' PAR'

end

END

GO

1. **Visualice un mensaje donde indique si un cliente de la bd Jardinería (ingrese su código, por ejemplo el 5) ha realizado pedidos.**

**Por pantalla debe aparecer el mensaje:**

* **'EL CLIENTE NO HA REALIZADO ALGUN PEDIDO' si no tiene pedidos**
* **'HA REGISTRADO 1 PEDIDO, CONTINUE TRABAJANDO' SI TIENE UN PEDIDO**
* **HA REGISTRADO X PEDIDOS, SIENDO X EL NÚMERO DE PEDIDOS REALIZADOS**

Usar estructura IF/ELSE

USE Jardineria

DECLARE @CodigoCliente INT;

SET @CodigoCliente = 5;

IF (SELECT COUNT(CodigoPedido) FROM Pedidos WHERE CodigoCliente = @CodigoCliente) = 1

BEGIN

PRINT 'HA REGISTRADO 1 PEDIDO, CONTINUE TRABAJANDO'

END

ELSE IF (SELECT COUNT(CodigoPedido) FROM Pedidos WHERE CodigoCliente = @CodigoCliente) > 1

BEGIN

PRINT 'HA REGISTRADO X PEDIDOS'

END

ELSE

BEGIN

PRINT 'EL CLIENTE NO HA REALIZADO ALGUN PEDIDO' ;

END

GO

O correccion gemma

USE Jardineria

DECLARE @CodigoCliente INT, @cantidad INT;

SET @CodigoCliente = 5;

SELECT @cantidad = count(\*) from pedidos where CodigoCliente = @CodigoCliente;

IF @cantidad = 1

BEGIN

PRINT 'HA REGISTRADO 1 PEDIDO, CONTINUE TRABAJANDO'

END

ELSE IF @cantidad > 1

BEGIN

PRINT 'HA REGISTRADO X PEDIDOS'

END

ELSE

BEGIN

PRINT 'EL CLIENTE NO HA REALIZADO ALGUN PEDIDO' ;

END

GO

1. **Declare una variable donde le asigne el número del mes, evalúe el valor de la variable y retorne el mes en letras por pantalla.**

DECLARE @Numero INT, @Mes1 VARCHAR (50),@Mes2 VARCHAR (50),@Mes3 VARCHAR (50),@Mes4 VARCHAR (50),@Mes5 VARCHAR (50),

@Mes6 VARCHAR (50),@Mes7 VARCHAR (50),@Mes8 VARCHAR (50),@Mes9 VARCHAR (50),@Mes10 VARCHAR (50),

@Mes11 VARCHAR (50),@Mes12 VARCHAR (50)

SET @Numero = 4;

SET @Mes1 = 'Enero';

SET @Mes2 = 'Febrero';

SET @Mes3 = 'Marzo';

SET @Mes4 = 'Abril';

SET @Mes5 = 'Mayo';

SET @Mes6 = 'Junio';

SET @Mes7 = 'Julio';

SET @Mes8 = 'Agosto';

SET @Mes9 = 'Septiembre';

SET @Mes10 = 'Octubre';

SET @Mes11 = 'Noviembre';

SET @Mes12 = 'Diciembre';

GO

SELECT

CASE

WHEN @Numero = 1 THEN @Mes1;

WHEN @Numero = 2 THEN @Mes2;

WHEN @Numero = 3 THEN @Mes3;

WHEN @Numero = 4 THEN @Mes4;

WHEN @Numero = 5 THEN @Mes5;

WHEN @Numero = 6 THEN @Mes6;

WHEN @Numero = 7 THEN @Mes7;

WHEN @Numero = 8 THEN @Mes8;

WHEN @Numero = 9 THEN @Mes9;

WHEN @Numero = 10 THEN @Mes10;

WHEN @Numero = 11 THEN @Mes11;

WHEN @Numero = 12 THEN @Mes12;

ELSE 'Error ,operación no permitida'

END

GO

DECLARE @unmero INT, @MES VARCHAR (50);

SET @numero = 4;

SET @MES=

(

CASE @numero

WHEN @numero = 1 THEN @MES;

WHEN @numero = 2 THEN @MES;

WHEN @numero = 3 THEN @MES;

WHEN @numero = 4 THEN @MES;

WHEN @numero = 5 THEN @MES;

WHEN @numero = 6 THEN @MES;

WHEN @numero = 7 THEN @MES;

WHEN @numero = 8 THEN @MES;

WHEN @numero = 9 THEN @MES;

WHEN @Numero = 10 THEN @MES;

WHEN @numero = 11 THEN @MES;

WHEN @Numero = 12 THEN @MES;

ELSE 'Error ,operación no permitida' )

END;

1. **Utilizar la estructura IF para evaluar la existencia de un registro en la tabla artículos de la bd de compras: si existe actualizamos los datos de la tabla; si no existe insertamos el registro. Los datos se los pasamos por medio de variables. Según la definición de los campos de la tabla ver que variables deberé establecer.**

USE compras

GO

DECLARE @codigart CHAR(6),@descrart CHAR(40), @preunart DECIMAL(9, 2),@stockart INT,@stockmin INT,@fecbaja DATE

SET @codigart = '0010';

SET @descrart = 'ARMARIO NORMAL';

SET @preunart = 22.56;

SET @stockart = 100;

SET @stockmin = 1;

SET @fecbaja = NULL;

IF EXISTS(SELECT \* FROM Articulos WHERE descrart = @descrart )

BEGIN

UPDATE Articulos SET stockart = 101 WHERE descrart = @descrart

PRINT 'ACTUALIZADA TABLA'

END

ELSE

BEGIN

INSERT INTO [dbo].[Articulos]

([codigart]

,[descrart]

,[preunart]

,[stockart]

,[stockmin]

,[fecbaja])

VALUES (@codigart,@descrart,@preunart,@stockart,@stockmin,@fecbaja)

PRINT 'INSERTADOS NUEVOS DATOS'

END

GO

O

USE compras

GO

DECLARE @codigart CHAR(6),@descrart CHAR(40), @preunart DECIMAL(9, 2),@stockart INT,@stockmin INT,@fecbaja DATE

SET @codigart = '0010';

SET @descrart = 'ARMARIO NORMAL';

SET @preunart = 22.56;

SET @stockart = 100;

SET @stockmin = 1;

SET @fecbaja = NULL;

IF EXISTS(SELECT \* FROM Articulos WHERE descrart = @descrart )

BEGIN

UPDATE Articulos SET stockart = @stockart, codigart= codigart,preunart = @preunart ,

stockart = @stockart, stockmin = @stockmin, fecbaja = @fecbaja WHERE descrart = @descrart

PRINT 'ACTUALIZADA TABLA'

END

ELSE

BEGIN

INSERT INTO [dbo].[Articulos]

([codigart]

,[descrart]

,[preunart]

,[stockart]

,[stockmin]

,[fecbaja])

VALUES (@codigart,@descrart,@preunart,@stockart,@stockmin,@fecbaja)

PRINT 'INSERTADOS NUEVOS DATOS'

END

GO

1. **Listar la descripción, precio unitario y número en stock de los productos de la tabla artículos de la bd compras. Definir una columna llamada ESTADO, el cual evaluará el stock de cada producto de la tabla mostrando un valor: Con Margen, Al límite, Haga una solicitud.**

USE Compras

GO

DECLARE @Descripcion CHAR (6), @PrecioUnitario decimal(9, 2), @StockArticulos INT, @ESTADO VARCHAR(50)

SELECT @Descripcion = descrart FROM Articulos

SELECT @PrecioUnitario = preunart FROM Articulos

SELECT @StockArticulos = stockart FROM Articulos

SELECT @ESTADO =

CASE

WHEN @StockArticulos > 25 THEN 'Con Margen'

WHEN @StockArticulos = 25 AND @StockArticulos > 7 THEN 'Al límite'

WHEN @StockArticulos < 25 THEN 'Haga una solicitud'

ELSE 'NO ESPECIFICADO'

END

SELECT @Descripcion AS DESCRIPCION , @PrecioUnitario AS PRECIO\_UNITARIO, @StockArticulos AS STOCK, @ESTADO AS Estado\_Stock

USE Compras

GO

DECLARE @stock INT;

set @stock = 25;

SELECT descrart , preunart, stockart FROM Articulos

SELECT @ESTADO =

CASE ESTADO =

WHEN StockArticulos > @stock THEN 'Con Margen'

WHEN StockArticulos = @stock AND @stock > 7 THEN 'Al límite'

WHEN StockArticulos < @stock THEN 'Haga una solicitud'

ELSE 'NO ESPECIFICADO'

END

1. **Actualizar las unidades de existencia (stockart) de los productos de la tabla artículos en la base de datos de compras asignándoles el valor de 1000 en aquellos productos cuyo stock sea cero. A la vez debe mostrar por pantalla qué artículos se han actualizado. Si no hay artículos a actualizar debe decirlo también.**



USE compras;

GO

DECLARE @STOCK INT, @Articulos CHAR(40)

SET @STOCK = 1000

SET @Articulos = (SELECT Articulos.descrart FROM Articulos WHERE stockart = 0)

IF EXISTS(SELECT Articulos.stockart FROM Articulos WHERE stockart = 0 )

BEGIN

UPDATE Articulos SET Articulos.stockart = @STOCK FROM Articulos WHERE Articulos.stockart = 0 ;

PRINT 'ACTUALIZADOS STOCKS TABLA'

PRINT 'PRODUCTO: ' + CAST(@Articulos AS CHAR) + ' SE ACTUALIZO EL STOCK'

END

ELSE IF EXISTS(SELECT Articulos.stockart FROM Articulos WHERE stockart != 0)

BEGIN

PRINT 'NO SE HA ACTUALIZADO ARTICULOS STOCKS'

END

ELSE

BEGIN

PRINT 'ERROR EN LA OPERACION'

END

GO

1. **Diseñar con código en lenguaje T-SQL un bloque de instrucciones que calcule la media de días de los pedidos de la base de datos compras, y nos muestre un mensaje por consola indicando si la media supera o no los 30 días y media de días.**

USE compras

GO

DECLARE @DiasMediaPedidos INT

SELECT @DiasMediaPedidos = AVG(DATEDIFF(DAY, fechaped, fentrped)) FROM Pedidos

DECLARE @Numero INT;

SET @Numero = 30;

IF @Numero < @DiasMediaPedidos

PRINT 'La media de dias es de '+cast (@DiasMediaPedidos AS VARCHAR(20))+'dias, por lo tanto supera los '+cast (@Numero AS VARCHAR(20))+'días.';

ELSE

BEGIN

IF @Numero > @DiasMediaPedidos

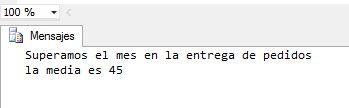
PRINT 'La media de dias es de ' + cast (@DiasMediaPedidos AS VARCHAR (20)) + ' dias, por lo tanto no supera los ' + cast (@Numero AS VARCHAR(20)) + ' días.';

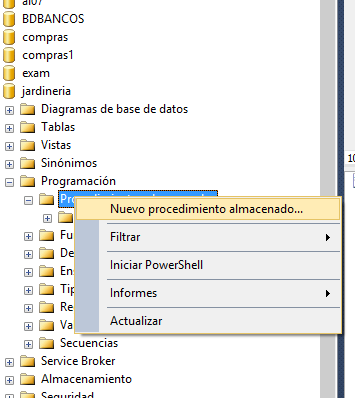
ELSE

PRINT 'Numero de dias iguales ' + cast (@DiasMediaPedidos AS VARCHAR(20));

END ;

GO





-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATE PROCEDURE USP\_CLIENTES

-- Add the parameters for the stored procedure here

--<@Param1, sysname, @p1> <Datatype\_For\_Param1, , int> = <Default\_Value\_For\_Param1, , 0>,

--<@Param2, sysname, @p2> <Datatype\_For\_Param2, , int> = <Default\_Value\_For\_Param2, , 0>

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

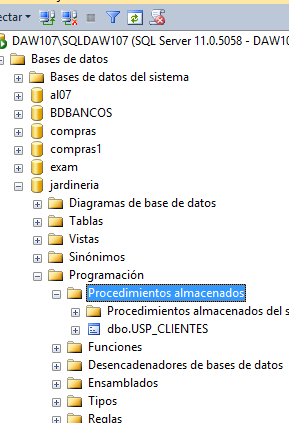
-- Insert statements for procedure here

SELECT codigoCliente AS CODIGO, NombreCliente AS CLIENTE, LineaDireccion1 as DIRECCION,telefono

FROM Jardineria.clientes

END

GO



EXEC dbo.USP\_CLIENTES;

ERROR

-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

ALTER PROCEDURE USP\_CLIENTES

-- Add the parameters for the stored procedure here

--<@Param1, sysname, @p1> <Datatype\_For\_Param1, , int> = <Default\_Value\_For\_Param1, , 0>,

--<@Param2, sysname, @p2> <Datatype\_For\_Param2, , int> = <Default\_Value\_For\_Param2, , 0>

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

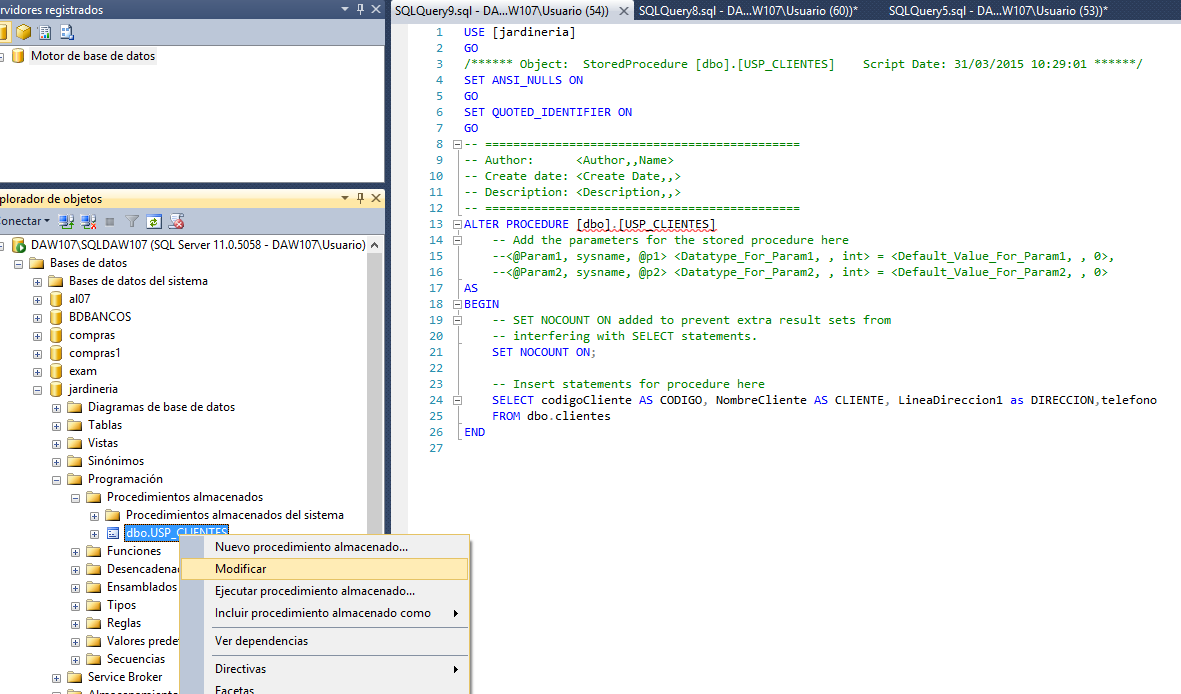
SELECT codigoCliente AS CODIGO, NombreCliente AS CLIENTE, LineaDireccion1 as DIRECCION,telefono

FROM dbo.clientes

END

GO

EXEC dbo.USP\_CLIENTES;



USE [jardineria]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_CLIENTES] Script Date: 31/03/2015 10:29:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

ALTER PROCEDURE [dbo].[USP\_CLIENTES] @Letra VARCHAR

-- Add the parameters for the stored procedure here

--<@Param1, sysname, @p1> <Datatype\_For\_Param1, , int> = <Default\_Value\_For\_Param1, , 0>,

--<@Param2, sysname, @p2> <Datatype\_For\_Param2, , int> = <Default\_Value\_For\_Param2, , 0>

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT codigoCliente AS CODIGO, NombreCliente AS CLIENTE, LineaDireccion1 as DIRECCION,telefono

FROM dbo.clientes WHERE NombreCliente like @Letra + '%'

END

GO

EXEC dbo.USP\_CLIENTES @Letra= 'A';

