1. Muestra el primer registro de la tabla artículos en la BD Compras. Usa cursores

USE compras;

GO

--CURSORES

DECLARE MI\_CURSOR CURSOR FOR

SELECT TOP 1 \* FROM dbo.Articulos

--ABRIR

OPEN MI\_CURSOR

--IMPRIMIR EL PRIMER REGISTRO

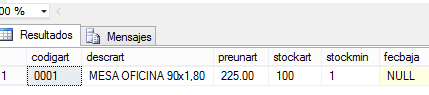
FETCH NEXT FROM MI\_CURSOR

--CERRAR

CLOSE MI\_CURSOR

--LIBERAR

DEALLOCATE MI\_CURSOR



1. Cursor que permita mostrar por pantalla, en la forma que se indica, el código del artículo, descripción y precio para todos los artículos de la BD compras.

Use compras;

Go

declare cursorP cursor

for select codigart, descrart, preunart from Articulos

open cursorP

declare @cod int, @pro varchar(50), @pre money

fetch next from cursorP into @cod, @pro, @pre

while @@fetch\_status=0

begin

print 'Codigo :' + cast(@cod as varchar(10))

print 'producto :' + @pro

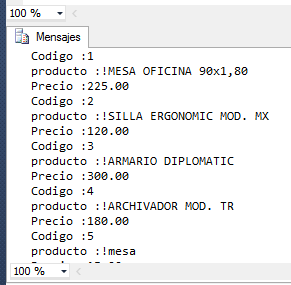
print 'Precio :' + cast(@pre as varchar(10))

fetch next from cursorP into @cod,@pro,@pre

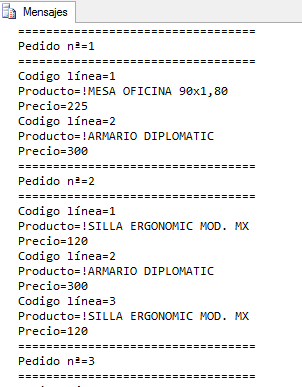
end

close cursorP

deallocate cursorP



1. Cursor que permita mostrar las líneas de pedido de cada pedido en la BD Compras de la siguiente manera.



Use compras;

go

declare cursorPedido cursor for select a.numped, b.numlin,c.descrart, c.preunart from Pedidos a,Lineas b,Articulos c

where a.numped=b.numped and b.codigart=c.codigart

declare @codped int,@codped2 int

declare @codlin int, @producto varchar(50), @precio decimal

open cursorPedido

fetch next from cursorpedido into @codped, @codlin, @producto, @precio

while @@FETCH\_STATUS = 0

begin

print '=================================='

print 'Pedido nª=' + cast(@codped as varchar(10))

print '=================================='

print 'Codigo línea=' + cast(@codlin as varchar(5))

print 'Producto=' + @producto

print 'Precio=' + cast(@precio as varchar(10))

SET @codped2=@codped

fetch next from cursorpedido into @codped, @codlin, @producto, @precio

While @codped = @codped2 and @@FETCH\_STATUS = 0

begin

print 'Codigo línea=' + cast(@codlin as varchar(5))

print 'Producto=' + @producto

print 'Precio=' + cast(@precio as varchar(10))

SET @codped2=@codped

fetch next from cursorpedido into @codped, @codlin, @producto, @precio

end

set @codped2 = 0

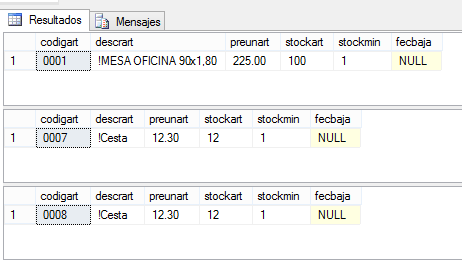
end

close cursorPedido

deallocate cursorpedido

go

1. Defina un cursor dinámico que permita visualizar: el primer registro, el registro en la posición 6 y el último registro de la tabla artículos de la BD Compras



USE compras

--CURSORES

DECLARE MI\_CURSOR CURSOR SCROLL FOR SELECT \* FROM Articulos

-- ABRIR

OPEN MI\_CURSOR

-- IMPRIMIR LOS REGISTROS

FETCH FIRST FROM MI\_CURSOR

FETCH ABSOLUTE 6 FROM MI\_CURSOR

FETCH LAST FROM MI\_CURSOR

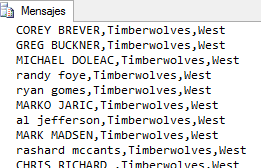
-- CERRAR

CLOSE MI\_CURSOR

-- LIBERAR

DEALLOCATE MI\_CURSOR

1. Trabajando con un cursor, listar nombre, nombre del equipo y Conferencia de los jugadores de la nba registrados en la base de datos.



USE nba;

go

-- DECLARO VARIABLES DE TRABAJO

DECLARE @NOMJUG VARCHAR(30), @NOMEQUIPO VARCHAR(20), @CONF VARCHAR(4)

-- DECLARO EL CURSOR

DECLARE MI\_CURSOR CURSOR FOR SELECT C.Nombre,

c.Nombre\_equipo, P.Conferencia

FROM jugadores C

JOIN equipos P ON C.Nombre\_equipo=P.Nombre

-- ABRIR

OPEN MI\_CURSOR

-- LEER EL PRIMER REGISTRO

FETCH MI\_CURSOR INTO @NOMJUG, @NOMEQUIPO, @CONF

-- MIENTRAS PUEDA LEER EL REGISTRO

WHILE @@FETCH\_STATUS=0

BEGIN

--IMPRIMIR EL REGISTRO

PRINT @NOMJUG + ','+@NOMEQUIPO+','+@CONF

--LEER EL REGISTRO SIGUIENTE

FETCH MI\_CURSOR INTO @NOMJUG, @NOMEQUIPO, @CONF

END

-- CERRAR

CLOSE MI\_CURSOR

-- LIBERAR

DEALLOCATE MI\_CURSOR;

GO

1. Sobre la BD Jardinería y trabajando con un cursor, listar la relación de los clientes hayan registrado pedidos o no. En dicho proceso, debemos imprimir el nombre del cliente, la cantidad de pedidos registrados a su nombre. Si no ha realizado pedidos debe aparecer 0. Al finalizar totalizar el proceso (suma total de todos los pedidos realizados por todos los clientes).



USE JARDINERIA;

Go

-- DECLARO VARIABLES DE TRABAJO

DECLARE @NOMBRE VARCHAR(50), @Q INT, @TOTAL INT

SET @TOTAL=0

-- DECLARO EL CURSOR

-- si count(\*) LOS NULOS LOS CUENTA

DECLARE MI\_CURSOR CURSOR FOR SELECT C.NombreCliente,

ISNULL(COUNT(p.CodigoCliente),0) FROM clientes C left

JOIN Pedidos P ON c.CodigoCliente=P.CodigoCliente

GROUP BY c.NombreCliente

-- ABRIR

OPEN MI\_CURSOR

-- LEER EL PRIMER REGISTRO

FETCH MI\_CURSOR INTO @NOMBRE, @Q

-- MIENTRAS PUEDA LEER EL REGISTRO

WHILE @@FETCH\_STATUS=0

BEGIN

-- IMPRIMIR EL REGISTRO

PRINT @NOMBRE+','+CAST(@Q AS VARCHAR)

-- ACUMULAR

SET @TOTAL += @Q

-- LEER EL REGISTRO SIGUIENTE

FETCH MI\_CURSOR INTO @NOMBRE, @Q

END

-- CERRAR

CLOSE MI\_CURSOR

-- LIBERAR

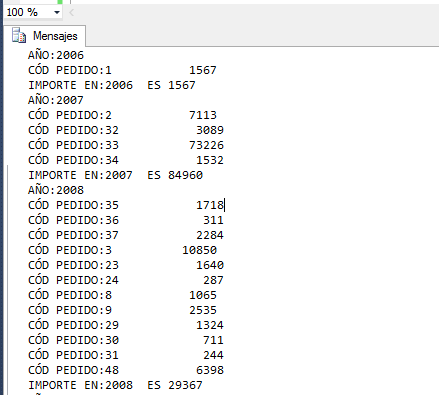
DEALLOCATE MI\_CURSOR;

-- IMPRIMIR

PRINT 'TOTAL DE PEDIDOS:' + CAST(@TOTAL AS VARCHAR)

GO

1. Sobre la BD Jardinería y trabajando con un cursor, saca por pantalla un listado de los pedidos realizados por cada año. La forma de presentación es : listar por cada año los pedidos registrados y totalizar dichos pedidos por dicho año.



USE JARDINERIA;

go

DECLARE @Y INT, @Y1 INT, @PEDIDO INT, @MONTANTE DECIMAL, @TOTAL DECIMAL

SET @TOTAL=0

-- DECLARO EL CURSOR

DECLARE MI\_CURSOR CURSOR FOR SELECT YEAR(FechaPedido),

C.CodigoPedido,

SUM(Cantidad\*PrecioUnidad) FROM dbo.DetallePedidos C

JOIN dbo.Pedidos D ON C.CodigoPedido=D.CodigoPedido

GROUP BY YEAR(FECHAPEDIDO), C.CodigoPedido

ORDER BY 1 -- ordeno por el primer campo que selecciono.

-- ABRIR EL CURSOR

OPEN MI\_CURSOR

-- LEER EL PRIMER REGISTRO

FETCH MI\_CURSOR INTO @Y, @PEDIDO, @MONTANTE

-- ASIGNAR A LA VARIABLE @Y1 EL VALOR INICIAL DE @Y

SET @Y1 = @Y

-- IMPRIMIR EL PRIMER AÑO

PRINT 'AÑO:' + CAST(@Y1 AS VARCHAR)

-- MIENTRAS PUEDA LEER EL REGISTRO

WHILE @@FETCH\_STATUS=0

BEGIN

-- SI COINCIDEN LOS VALORES ACUMULAR EL TOTAL

IF(@Y = @Y1)

SET @TOTAL += @MONTANTE

ELSE

-- SI NO COINCIDEN IMPRIMIR EL TOTAL, INICIALIZAR VARIABLES

BEGIN

PRINT 'IMPORTE EN:' +CAST(@Y1 AS VARCHAR) + SPACE(2)+ 'ES ' +CAST(@TOTAL AS VARCHAR)

PRINT 'AÑO:' + CAST(@Y AS VARCHAR)

SET @Y1=@Y

SET @TOTAL=@MONTANTE

END

-- IMPRIMIR EL REGISTRO

PRINT 'CÓD PEDIDO:' +CAST(@PEDIDO AS VARCHAR) + SPACE(5)+STR(@MONTANTE)

-- LEER EL REGISTRO SIGUIENTE

FETCH MI\_CURSOR INTO @Y, @PEDIDO, @MONTANTE

END

-- CERRAR

CLOSE MI\_CURSOR

-- LIBERAR

DEALLOCATE MI\_CURSOR;

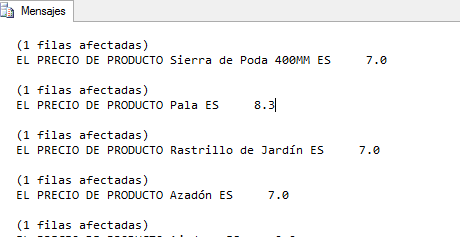
-- IMPRIMIR LOS TOTALES FINALES

PRINT 'IMPORTE EN:'+CAST(@Y1 AS VARCHAR)+SPACE(2)+'ES'+STR(@TOTAL)

-- SPACE(x) devuelve una cadena de x-espacios

--STR() convierte datos numéricos en una cadena de caracteres.

1. En la bd Jardinería, actualizar el precio de los productos: si su stock es mayor o igual a 100, se descuenta al precio del Proveedor el 50%, sino se descuenta al precio del Proveedor el 20%.



USE JARDINERIA;

go

DECLARE @ID VARCHAR(15), @NOMBRE VARCHAR(255), @PRECIO NUMERIC(15,2),@ST smallint

-- DECLARACIÓN DEL CURSOR DE ACTUALIZACION

DECLARE CPRODUCTO CURSOR FOR

SELECT CodigoProducto, NOMBRE, PrecioProveedor, CantidadEnStock

FROM dbo.Productos FOR UPDATE

-- APERTURA DEL CURSOR

OPEN CPRODUCTO

-- LECTURA DE LA PRIMERA FILA DEL CURSOR

FETCH CPRODUCTO INTO @ID, @NOMBRE, @PRECIO, @ST

-- MIENTRAS PUEDA LEER EL REGISTRO

WHILE (@@FETCH\_STATUS = 0 )

BEGIN

IF(@ST>=100)

SET @PRECIO = 0.5\*@PRECIO

ELSE

SET @PRECIO = 0.80\*@PRECIO

UPDATE dbo.Productos SET PrecioProveedor = @PRECIO WHERE CURRENT OF CPRODUCTO

--IMPRIMIR

PRINT 'EL PRECIO DE PRODUCTO '+ @NOMBRE+ ' ES ' + STR(@PRECIO,7,1)

-- LECTURA DE LA SIGUIENTE FILA DEL CURSOR

FETCH CPRODUCTO INTO @ID, @NOMBRE, @PRECIO, @ST

END

-- CIERRE DEL CURSOR

CLOSE CPRODUCTO

-- LIBERAR LOS RECURSOS

DEALLOCATE CPRODUCTO

1. Sobre la bd Banco, implementar, usando cursores, un procedimiento que muestre los datos del cliente, la cuenta y el saldo de los clientes con saldo negativo en alguna de sus cuentas.

USE banco;

Go

CREATE PROCEDURE saldo\_neg as

DECLARE @vsaldo int, @vcod\_cliente int,@vcuenta int

DECLARE MI\_CURSOR CURSOR FOR SELECT c.saldo,b.codigo\_cliente,c.cod\_cuenta from

cuenta c, tiene t, cliente b where c.cod\_cuenta=t.cod\_cuenta and t.dni=b.dni

-- APERTURA DEL CURSOR

OPEN MI\_CURSOR

-- LECTURA DE LA PRIMERA FILA DEL CURSOR

FETCH MI\_CURSOR INTO @vsaldo, @vcod\_cliente, @vcuenta

-- MIENTRAS PUEDA LEER EL REGISTRO

WHILE (@@FETCH\_STATUS = 0 )

BEGIN

IF @vsaldo<0

SELECT \*, @vcuenta as NUM\_CUENTA,@vsaldo as SALDO from cliente where cliente.codigo\_cliente=@vcod\_cliente

FETCH MI\_CURSOR INTO @vsaldo, @vcod\_cliente, @vcuenta

END

-- CIERRE DEL CURSOR

CLOSE MI\_CURSOR

-- LIBERAR LOS RECURSOS

DEALLOCATE MI\_CURSOR

GO

-------------------------

exec saldo\_neg

1. Sobre la bd Banco, calcular con un procedimiento la suma de todos los ingresos y cargos (por separado) juntando todas las cuentas.

USE banco;

Go

CREATE PROCEDURE total\_mov as

DECLARE @vcantidad int, @tcargos int,@tingresos int

SET @tcargos=0

SET @tingresos=0

DECLARE MI\_CURSOR CURSOR FOR SELECT cantidad from movimiento

-- APERTURA DEL CURSOR

OPEN MI\_CURSOR

-- LECTURA DE LA PRIMERA FILA DEL CURSOR

FETCH MI\_CURSOR INTO @vcantidad

-- MIENTRAS PUEDA LEER EL REGISTRO

WHILE (@@FETCH\_STATUS = 0 )

BEGIN

IF @vcantidad<0

SET @tcargos+=@vcantidad

else

SET @tingresos+=@vcantidad

FETCH MI\_CURSOR INTO @vcantidad

END

-- CIERRE DEL CURSOR

CLOSE MI\_CURSOR

-- LIBERAR LOS RECURSOS

DEALLOCATE MI\_CURSOR

SELECT @tingresos as TOTAL\_INGRESOS, @tcargos as TOTAL\_CARGOS

GO

-------------------------

exec total\_mov

1. Modifica el procedimiento anterior para que sobre la bd Banco, calcule la suma de todos los ingresos y cargos (por separado) para una cuenta que se le pasa como parámetro.

USE banco;

Go

CREATE PROCEDURE total\_mov\_cuenta @id\_cuenta int as

DECLARE @vid\_cuenta int, @vcantidad int, @tcargos int,@tingresos int

SET @tcargos=0

SET @tingresos=0

DECLARE MI\_CURSOR CURSOR FOR SELECT cod\_cuenta, cantidad from movimiento

-- APERTURA DEL CURSOR

OPEN MI\_CURSOR

-- LECTURA DE LA PRIMERA FILA DEL CURSOR

FETCH MI\_CURSOR INTO @vid\_cuenta,@vcantidad

WHILE (@@FETCH\_STATUS = 0 )

BEGIN

IF @vid\_cuenta=@id\_cuenta

begin

IF @vcantidad<0

SET @tcargos+=@vcantidad

else

SET @tingresos+=@vcantidad

end

FETCH MI\_CURSOR INTO @vid\_cuenta,@vcantidad

END

-- CIERRE DEL CURSOR

CLOSE MI\_CURSOR

-- LIBERAR LOS RECURSOS

DEALLOCATE MI\_CURSOR

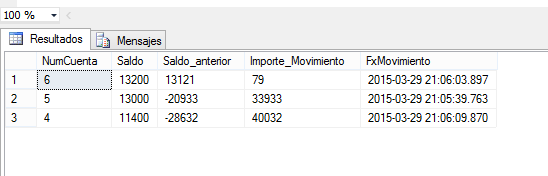
SELECT @tingresos as TOTAL\_INGRESOS, @tcargos as TOTAL\_CARGOS

GO

-------------------------

exec total\_mov\_cuenta @id\_cuenta=4

1. Implementa sobre la bd Banco una función que busca la tres cuentas con mayor saldo y obtiene para cada una de ellas: el saldo actual, saldo anterior, la fecha del último movimiento y el importe del movimiento. Usa cursores



USE banco;

go

CREATE FUNCTION fn\_CuentaMovimietos()

RETURNS @datos TABLE

( -- Estructura de la tabla que devuelve la funcion.

NumCuenta int,

Saldo int,

Saldo\_anterior int,

--Saldo\_posterior int,

Importe\_Movimiento int,

FxMovimiento datetime

)

AS

BEGIN

-- Variables necesarias para la lógica de la funcion.

DECLARE

@numcuenta int,

@saldo int

-- Cursor con las 3 cuentas de mayor saldo

DECLARE MI\_CURSOR CURSOR FOR

SELECT TOP 3 cod\_cuenta, saldo

FROM cuenta

ORDER BY saldo DESC

OPEN MI\_CURSOR

FETCH MI\_CURSOR INTO @numcuenta, @saldo

-- Recorremos el cursor

WHILE (@@FETCH\_STATUS = 0)

BEGIN

-- Insertamos el último movimiento de la cuenta

INSERT INTO @datos (NumCuenta, Saldo, Saldo\_anterior, Importe\_Movimiento, FxMovimiento )

SELECT TOP 1 @numcuenta, @saldo,

@saldo-cantidad, cantidad, fecha

FROM movimiento

WHERE cod\_cuenta = @numcuenta

ORDER BY fecha DESC

-- Vamos a la siguiente cuenta

FETCH MI\_CURSOR INTO @numcuenta, @saldo

END

CLOSE MI\_CURSOR;

DEALLOCATE MI\_CURSOR;

RETURN

END

go

-------------

select \* from fn\_CuentaMovimietos()

1. Crea un script para realizar un backup de todas las bases de datos del sql server (excepto de la master ,model, msdb, tempdb).

NOTA: en Master.dbo.sysdatabases se encuentra información de todas las bases de datos de sql server. Habría que declarar las siguientes variables:

DECLARE @name VARCHAR(50) -- nombre de la bd

DECLARE @path VARCHAR(256) -- Ruta para los ficheros de backup

DECLARE @fileName VARCHAR(256) -- nombre del fichero para el backup

DECLARE @fileDate VARCHAR(20) – fecha para el nombre del fichero

--------------------------------------------------

DECLARE @name VARCHAR(50) -- nombre de la bd

DECLARE @path VARCHAR(256) -- Ruta para los ficheros de backup

DECLARE @fileName VARCHAR(256) -- nombre del fichero para el backup

DECLARE @fileDate VARCHAR(20) – fecha para el nombre del fichero

SET @path = 'C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\Backup\'

SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)

DECLARE MI\_cursor CURSOR FOR

SELECT name

FROM MASTER.dbo.sysdatabases

WHERE name NOT IN ('master','model','msdb','tempdb')

OPEN MI\_cursor

FETCH NEXT FROM MI\_cursor INTO @name

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @fileName = @path + @name + '\_' + @fileDate + '.BAK'

BACKUP DATABASE @name TO DISK = @fileName

FETCH NEXT FROM MI\_cursor INTO @name

END

CLOSE MI\_cursor

DEALLOCATE MI\_cursor