use preventas

go

--1. Hacer un PA denominado PA\_Setear, donde declare variables de tipo Fecha, Entero, Real y Cadena, inicializarlas y mostrar su valor

DROP PROC PA\_Setear

CREATE PROC PA\_Setear

AS

RETURN

DECLARE @fecha DATE,@entero INT,@real DECIMAL(12,2),@cadena CHAR(10)

SET @fecha=GETDATE()

SET @entero=10

SET @real=10.23

SET @cadena='BASE II'

EXEC PA\_Setear

PRINT @fecha

PRINT @entero

PRINT @real

PRINT @cadena

--2. Hacer un PA denominado PA\_Igual, en la que se le pasan 2 números y retorne 1 si son iguales, 0 si no lo son

DROP PROC PA\_Igual

CREATE PROC PA\_Igual (@X DECIMAL(12,2), @Y DECIMAL(12,2),@RESULTADO DECIMAL(12,2) OUTPUT)

AS

IF(@X=@Y)

SET @RESULTADO=1

ELSE

SET @RESULTADO=0

RETURN

DECLARE @resul INT

EXEC PA\_Igual 5,5,@RESULTADO=@resul OUTPUT

PRINT @resul

--3. Hacer un PA denominado PA\_Nombre, que reciba como parámetro su <nombre> y muestre un mensaje "Hola <nombre>"

DROP PROC PA\_Nombre

CREATE PROC PA\_Nombre (@nombre CHAR(40),@mensaje CHAR(40) OUTPUT)

AS

SET @mensaje=@nombre

RETURN

DECLARE @mensajeF CHAR(40)

EXEC PA\_Nombre 'ANTONIO',@mensaje=@mensajeF OUTPUT

PRINT 'HOLA '+@mensajeF

--4. Hacer un PA denominado PA\_GetProv, que reciba como parámetro el código del proveedor y retorne la ciudad donde vive el proveedor.

CREATE PROC PA\_GetProv (@cprv INT,@ciud CHAR(2) OUTPUT)

AS

SELECT @ciud=ciud FROM prov WHERE cprv=@cprv

RETURN

DECLARE @CIUDAD CHAR(2)

EXEC PA\_GetProv 2,@ciud=@CIUDAD OUTPUT

PRINT 'LA CIUDAD ES: ' + @CIUDAD

--5. Hacer un PA denominado PA\_ExisteProducto, que reciba como parámetro el código del producto y retorne 1 si existe el producto de lo contrario retorne 0.

DROP PROC PA\_ExisteProducto

CREATE PROC PA\_ExisteProducto (@cprd INT, @mensaje INT OUTPUT)

AS

IF(SELECT COUNT(\*) FROM prod WHERE cprd in (SELECT cprd FROM prod WHERE cprd=@cprd))=1

SET @mensaje=1

ELSE

SET @mensaje=0

RETURN

DECLARE @mes INT

EXEC PA\_ExisteProducto 5,@mensaje=@mes OUTPUT

PRINT @mes

--6. Hacer un PA denominado PA\_LeerProducto, que reciba como parámetro el código del producto y retorne nombre y el color del producto si el código del producto introducido es valido, de lo contrario debe retornar nulo para ambos casos.

DROP PROC PA\_LeerProducto

CREATE PROC PA\_LeerProducto (@cprd INT, @mensaje CHAR(55) OUTPUT)

AS

DECLARE @colo CHAR(15), @nomp CHAR(40)

IF(SELECT COUNT(\*) FROM prod WHERE cprd in (SELECT cprd FROM prod WHERE cprd=@cprd))=1

BEGIN

SET @colo = (SELECT colo FROM prod WHERE cprd=@cprd)

SET @nomp = (SELECT nomp FROM prod WHERE cprd=@cprd)

SET @mensaje = @nomp + @colo

END

ELSE

SET @mensaje='EL PRODUCTO NO EXISTE'

RETURN

DECLARE @mens CHAR(55)

EXEC PA\_LeerProducto 5,@mensaje=@mens OUTPUT

PRINT @mens

--7. Hacer un PA denominado PA\_TotalStock que reciba como parámetro el código del producto y retorne el TotalStock existente del producto. (El stock es la sumatoria de cantidades suministrada del producto).

CREATE PROC PA\_TotalStock (@cprd INT, @TotalStock DECIMAL(12,2) OUTPUT)

AS

SELECT @TotalStock=isnull(SUM(cant),0) FROM sumi

WHERE cprd=@cprd

RETURN

DECLARE @stockF DECIMAL(12,2)

EXEC PA\_TotalStock 4,@TotalStock=@stockF OUTPUT

PRINT @stockF

--8. Hacer un PA denominado PA\_HayStock que reciba como parámetro el código del producto y que retorne 1 si hay Stock disponible, de lo contrario que retorne 0.

CREATE PROC PA\_HayStock (@cprd INT,@resul INT OUTPUT)

AS

DECLARE @Xstock DECIMAL

SET @resul=0

EXEC PA\_TotalStock @cprd,@Xstock OUTPUT

IF (@Xstock>0)

SET @resul=1

RETURN

DECLARE @result INT

EXEC PA\_HayStock 3,@resul=@result OUTPUT

PRINT @result

--9. Hacer un PA denominado PA\_StockxCiudad que reciba el código del producto y la ciudad del almacen, y que retorne el Stock existente del producto en la ciudad. (El stock es la sumatoria de cantidades suministrada del producto en una determinada ciudad).

CREATE PROC PA\_StockxCiudad (@cprd INT, @ciuda CHAR(2), @StockExis DECIMAL(12,2) OUTPUT)

AS

SELECT @StockExis=isnull(SUM(cant),0) FROM sumi,alma

WHERE alma.calm=sumi.calm AND cprd=@cprd AND alma.ciud=@ciuda

RETURN

DECLARE @result DECIMAL(12,2)

EXEC PA\_StockxCiudad 3,'SC',@StockExis=@result OUTPUT

PRINT @result

--10. HAcer un PA denominado PA\_HayStockxAlmacen que reciba como parámetro el código del producto y el código del almacén, y que retorne 1 si hay Stock disponible del producto en ese almacen, de lo contrario que retorne 0. (usar PA\_StockxAlmacen)

CREATE PROC PA\_TotalStockxAlmacen (@cprd INT, @calma INT, @stock DECIMAL(12,2) OUTPUT)

AS

SELECT @stock=isnull(SUM(cant),0) FROM sumi

WHERE cprd=@cprd AND calm=@calma

RETURN

CREATE PROC PA\_HayStockxAlmacen (@xcprd INT, @xcalma INT, @resul DECIMAL(12,2) OUTPUT)

AS

DECLARE @xstock INT

SET @resul=0

EXEC PA\_TotalStockxAlmacen @xcprd, @xcalma, @xstock OUTPUT

IF(@xstock)>0

SET @resul=1

RETURN

DECLARE @result DECIMAL(12,2)

EXEC PA\_HayStockxAlmacen 1,3,@resul=@result OUTPUT

PRINT @result

--11. Hacer un denominado PA\_ValidarPreVenta, que reciba el Numero de Venta y que retorne 1 si todos los productos de la Pre Venta cuenta con Stock suficiente, de lo contrario retorna 0. (PA\_StockxAlmacen). Se debe insertar datos a la tabla pventas y dventas para probar.

CREATE PROC PA\_ValidarPreVenta (@numventa INT, @resul INT OUTPUT)

AS

DECLARE @calma INT = (SELECT pventas.calm FROM pventas WHERE pventas.nvta=@numventa)

PRINT @calma

DECLARE @cprd INT

DECLARE @cant INT

DECLARE C\_CURSOR CURSOR FOR SELECT dventas.cprd, dventas.cant FROM dventas WHERE dventas.nvta=@numventa

OPEN C\_CURSOR

FETCH NEXT FROM C\_CURSOR INTO @cprd, @cant

SET @resul=1

WHILE (@@FETCH\_STATUS=0)

BEGIN

DECLARE @stock DECIMAL(12,2)

EXEC PA\_HayStockxAlmacen @cprd, @calma, @stock OUTPUT

PRINT 'Stock prod' + cast(@cprd as char(10)) + ' cant ' + cast(@cant as char(10)) + ' stock ' + cast(@stock as char(20))

IF(@cant>=@stock)

SET @resul=0

FETCH NEXT FROM C\_CURSOR INTO @cprd, @cant

END

CLOSE C\_CURSOR

DEALLOCATE C\_CURSOR

RETURN

DECLARE @x INT

EXEC PA\_ValidarPreVenta 1, @x OUTPUT

PRINT @x

--12. Hacer un PA denominado PA\_TotalPreVenta, que reciba el Numero de Venta y que retorne el Importe Total de la Pre Venta.

DROP PROC PA\_TotalPreVenta

CREATE PROC PA\_TotalPreVenta (@nventa INT, @imptot DECIMAL OUTPUT)

AS

SELECT @imptot=isnull(SUM(impt),0) FROM dventas

WHERE nvta=@nventa

RETURN

DECLARE @Impot DECIMAL

EXEC PA\_TotalPreVenta 1,@imptot=@Impot OUTPUT

PRINT @Impot

--13. Hacer un PA denominado PA\_DescPreVenta, que reciba como parámetro Numero de Venta y retorne el importe de descuento de la Pre Ventas, el descuento es dado bajo los siguientes criterios:

-- Si el importe total de la pre venta esta entre 10 y 20 bs se aplica un descuento de 10%

-- Si el importe total de la pre venta esta entre 11 y 50 bs se aplica un descuento de 15%

-- Si el importe total de la pre venta es mayor a 50 bs se aplica un descuento de 20%

CREATE PROCEDURE PA\_DescPreVenta(@Nvta int ,@impDesc decimal(12,2) output)

AS

declare @impt decimal(12,2)

select @impt=isnull(sum(impt),0) from dventas where nvta=@Nvta

set @impDesc= (case

when (@impt>=10 and @impt<=20) then @impt\*0.1

when (@impt>=11 and @impt<=50) then @impt\*0.15

when (@impt>50) then @impt\*0.2

else 0

end)

RETURN

DECLARE @imptdescuento decimal(12,2)

EXECUTE PA\_DescPreVenta 1,@imptdescuento OUTPUT

PRINT @imptdescuento

--14. Hacer un PA denominado PA\_PromPreVenta, que reciba como parámetro Numero de Venta y retorne el importe promedio de las cantidades los producto de la Pre Ventas.

DROP PROCEDURE PA\_PromPreVenta

CREATE PROCEDURE PA\_PromPreVenta(@Nvta int ,@impProTot decimal output)

AS

DECLARE @n1 int,@impt decimal

SET @n1= (select COUNT(cprd) from dventas

where nvta=@Nvta)

select @impt=isnull(sum(cant),0) from dventas

where nvta=@Nvta

set @impProTot=@impt/@n1

RETURN

DECLARE @promedio decimal

EXECUTE PA\_PromPreVenta 1,@promedio OUTPUT

PRINT @promedio

--15. Hacer un PA denominado PA\_DelDetPreVentas, que reciba como parámetro Numero de Ventas y elimina las tuplas de las tablas dventas.

CREATE PROC PA\_DelDetPreVentas (@nventa INT)

AS

DELETE FROM dventas WHERE nvta=@nventa

RETURN

EXEC PA\_DelDetPreVentas 1

--16. hacer un PA denominado PA\_ProvSumistra, que reciba el código del proveedor y retorne 1 si el proveedor ha suministrado algún Producto, de lo contrario retorna 0.

CREATE PROC PA\_ProvSumistra (@cprv INT, @result INT OUTPUT)

AS

IF (SELECT count(\*) FROM sumi WHERE cprv=@cprv)>1

SET @result=1

ELSE

SET @result=0

RETURN

DECLARE @resulta INT

EXEC PA\_ProvSumistra 1,@result=@resulta OUTPUT

PRINT @resulta

--17. Hacer un PA denominado PA\_ListaProvSumistra, que muestre la lista de los Proveedores que Suministraron algún Producto (usar PA\_ProvSumistra).

DROP PROC PA\_ListaProvSumistra

CREATE PROC PA\_ListaProvSumistra

AS

DECLARE @count INT, @result INT

SET @count =1

WHILE (@count<=(SELECT COUNT(\*) FROM prov))

BEGIN

EXEC PA\_ProvSumistra @count, @result OUTPUT

IF (@result)=1

select \* from prov where cprv=@count

SET @count =@count + 1

END

RETURN

EXEC PA\_ListaProvSumistra