--1.Hacer una funcion denominada "Suma", que reciba dos numeros y retorne la suma de ambos numeros

DROP FUNCTION Suma

CREATE FUNCTION Suma (@x decimal(10,2), @y decimal(10,2))

RETURNS DECIMAL(10,2)

AS

BEGIN

DECLARE @resultado decimal(10,2)

SET @resultado=@x + @y

RETURN @resultado

END

SELECT dbo.Suma(10,12)

--2.Hacer una funcion denominada "GetCiudad", que reciba como parámetro el código del proveedor y retorne la ciudad donde vive el proveedor.

DROP FUNCTION GetCiudad

CREATE FUNCTION GetCiudad (@cprv int)

RETURNS CHAR(2)

AS

BEGIN

DECLARE @ciudad CHAR(2)

SELECT @ciudad=ciud from prov where cprv=@cprv

RETURN @ciudad

END

SELECT dbo.GetCiudad(1)

--3.Hacer una funcion denominada "GetNombre", que reciba el codigo del proveedor y retorne su nombre

DROP FUNCTION GetNombre

CREATE FUNCTION GetNombre (@cprv int)

RETURNS CHAR(40)

AS

BEGIN

DECLARE @nombre CHAR(40)

SELECT @nombre=nomb from prov where cprv=@cprv

RETURN @nombre

END

SELECT dbo.GetNombre(1)

--4.Hacer una funcion denominada "CalcularPuntos", que reciba el codigo del proveedor y calcule los puntos de bonificacion en base a los siguientes criterios:

--Si el proveedor suministro entre 1 y 20 bs se le asigna 10 puntos.

--Si el proveedor suministro entre 11 y 50 bs se le asigna 15 puntos.

--Si el proveedor suministro mas de 51 bs se le asigna 20 puntos.

DROP FUNCTION CalcularPuntos

CREATE FUNCTION CalcularPuntos (@cprv int)

RETURNS INT

AS

BEGIN

DECLARE @impt DECIMAL

DECLARE @puntos INT

SELECT @impt=ISNULL (SUM(impt),0) FROM sumi

WHERE cprv=@cprv

SET @puntos=0

IF (@impt>=1 AND @impt<=20)

SET @puntos=10

IF(@impt>=11 AND @impt<=50)

SET @puntos=15

IF(@impt>50)

SET @puntos=20

RETURN @puntos

END

SELECT dbo.CalcularPuntos(2)

--5.Hacer una funcion denominada "GetStock", que devuelva el Stock existente de un producto que se encuentra en una ciudad en particular.

CREATE FUNCTION GetStock (@cprd int, @ciud char(2))

RETURNS DECIMAL(12,2)

AS

BEGIN

DECLARE @stock DECIMAL(12,2)

SELECT @stock =SUM(cant) FROM sumi,alma

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

RETURN @stock

END

SELECT dbo.GetStock(1,'LP')

--6.Hacer una funcion denominada "GetInven", que devuelva el Inventario Valorado de un producto.

CREATE FUNCTION GetInven (@cprd INT)

RETURNS DECIMAL(12,2)

AS

BEGIN

DECLARE @impt DECIMAL(12,2)

SELECT @impt = SUM(sumi.impt) FROM sumi,alma,prod

WHERE sumi.calm=alma.calm AND prod.cprd=sumi.cprd AND prod.cprd=@cprd

RETURN @impt

END

SELECT dbo.GetInven(1)

--7.Hacer una funcion denominada "GetProdxCiud", que devuelva en una tabla los Productos existentes en una ciudad en particular

CREATE FUNCTION GetProdxCiud (@ciudad CHAR(2))

RETURNS TABLE

AS

RETURN(

SELECT prod.\*

FROM prod

WHERE cprd in (SELECT cprd FROM sumi,alma WHERE alma.calm=sumi.calm AND alma.ciud=@ciudad)

);

SELECT \*FROM dbo.GetProdxCiud('LP')

--8.Hacer una funcion denominada "GetProvxProd", que devuelva en una tabla los Proveedores que suministraron algún Producto

CREATE FUNCTION GetProvxProd ()

RETURNS TABLE

AS

RETURN(

select \*

from prov

where cprv in (select cprv from sumi,alma where prov.cprv=sumi.cprv)

);

SELECT \*FROM dbo.GetProvxProd()

--9.Hacer una funcion denominada "GetProvNoSumi", que devuelva en una tabla los Proveedores que todavía no suministraron productos.

CREATE FUNCTION GetProvNoSumi ()

RETURNS TABLE

AS

RETURN(

SELECT \*

FROM prov

WHERE NOT EXISTS (SELECT cprv FROM sumi WHERE prov.cprv=sumi.cprv)

);

SELECT \*FROM dbo.GetProvNoSumi()

--10.Hacer una funcion denominada "GetProvSumi", que devuelva en una tabla los nombres de los proveedores que suministraron algún producto color rojo

DROP FUNCTION GetProvSumi

CREATE FUNCTION GetProvSumi ()

RETURNS TABLE

AS

RETURN(

SELECT prov.nomb

FROM prov

WHERE EXISTS (SELECT cprv from sumi,prod WHERE prov.cprv=sumi.cprv and colo='ROJO')

);

SELECT \*FROM dbo.GetProvSumi()

--11.Hacer una funcion denominada "GetProdxProv", que devuelva en una tabla productos existente en un almacen y que fueron suministrado por un proveedor en particular

DROP FUNCTION GetProdxProv

CREATE FUNCTION GetProdxProv (@calma INT, @cprv INT)

RETURNS TABLE

AS

RETURN(

SELECT DISTINCT prod.\*

FROM prod

WHERE EXISTS (SELECT cprd FROM sumi,alma,prov WHERE prod.cprd=sumi.cprd and prov.cprv=sumi.cprv and alma.calm=@calma and prov.cprv=@cprv)

);

SELECT \*FROM dbo.GetProdxProv(2,2)

--12.Hacer una funcion denominada "GetProdxColor", que devuelva en una tabla productos de color rojo suministrados por un proveedor

CREATE FUNCTION GetProdxColor(@cprv INT)

RETURNS TABLE

AS

RETURN(

SELECT prod.\*

FROM prod

WHERE cprd in (SELECT cprd FROM sumi WHERE sumi.cprv=@cprv)

and colo='ROJO'

);

SELECT \*FROM dbo.GetProdxColor(1)

--13.Hacer una funcion denominada "GetProvTodo", que devuelva en una tabla los nombres de los proveedores que suministraron todos los productos.

CREATE FUNCTION GetProvTodo()

RETURNS TABLE

RETURN(

SELECT prov.nomb FROM prov

WHERE (SELECT COUNT(DISTINCT cprd) FROM sumi WHERE sumi.cprv=prov.cprv)=(SELECT COUNT(\*) FROM prod)

);

SELECT \*FROM dbo.GetProvTodo()

--14.Hacer una funcion denominada "GetProvTres", que devuelva en una tabla los nombres de los proveedores que suministraron por lo menos tres productos diferentes

DROP FUNCTION GetProvTres

CREATE FUNCTION GetProvTres()

RETURNS TABLE

AS

RETURN(

SELECT prov.nomb FROM prov

WHERE (SELECT COUNT(DISTINCT cprd) FROM sumi WHERE sumi.cprv=prov.cprv)>=3

);

SELECT \*FROM dbo.GetProvTres()

--15.Hacer una funcion denominada "GetProvOutCiud", que devuelva en una tabla los nombres de los proveedores que suministraron algún producto fuera de su ciudad.

CREATE FUNCTION GetProvOutCiud()

RETURNS TABLE

AS

RETURN(

SELECT prov.nomb

FROM prov

WHERE cprv IN (SELECT cprv from sumi,alma WHERE prov.cprv=sumi.cprv and alma.ciud<>prov.ciud)

);

SELECT \*FROM dbo.GetProvOutCiud()

--16.Hacer una funcion denominada "GetMaxCantxCiud", que devuelva la cantidad más alta suministrada de un producto en una ciudad en particular.

CREATE FUNCTION GetMaxCantxCiud (@cprd INT,@ciud CHAR(2))

RETURNS INT

AS

BEGIN

DECLARE @cantAlta INT

SELECT @cantAlta= MAX(sumi.cant) FROM sumi,alma,prod

WHERE prod.cprd=sumi.cprd and alma.calm=sumi.calm and prod.cprd=@cprd and alma.ciud=@ciud

RETURN @cantAlta

END

SELECT dbo.GetMaxCantxCiud(1,'SC') AS CANTIDAD\_ALTA

--17.Hacer una funcion denominada "GetUltFecxProv", que devuelva la ultima fecha que se suministró un producto por un proveedor en particular.

DROP FUNCTION GetUltFecxProv

CREATE FUNCTION GetUltFecxProv (@cprd INT, @nomb CHAR(40))

RETURNS DATE

AS

BEGIN

DECLARE @ftra DATE

SELECT TOP 1 @ftra = sumi.ftra FROM sumi,prod,prov

WHERE sumi.cprd=prod.cprd and prov.cprv=sumi.cprv AND prod.cprd=@cprd AND prov.nomb=@nomb

ORDER BY sumi.ftra DESC

RETURN @ftra

END

SELECT dbo.GetUltFecxProv(3,'PROV3')

--18.Hacer una funcion denominada "GetPrimFecxColor", que devuelva la fecha que por primera vez suministró algún producto de color Rojo.

DROP FUNCTION GetPrimFecxColor

CREATE FUNCTION GetPrimFecxColor (@colo CHAR(15))

RETURNS DATE

AS

BEGIN

DECLARE @ftra DATE

SELECT TOP 1 @ftra = sumi.ftra FROM sumi,prod,prov

WHERE sumi.cprd=prod.cprd and prov.cprv=sumi.cprv AND prod.colo=@colo

ORDER BY sumi.ftra

RETURN @ftra

END

SELECT dbo.GetPrimFecxColor('ROJO')

--19.Hacer una funcion denominada "GetPromxProv", que devuelva el importe promedio de productos suministrados por un proveedor.

DROP FUNCTION GetPromxProv

CREATE FUNCTION GetPromxProv (@cprv INT)

RETURNS DECIMAL(12,2)

AS

BEGIN

DECLARE @impt DECIMAL(12,2)

SELECT @impt=ISNULL(AVG(impt),0)FROM sumi,alma

WHERE alma.calm=sumi.calm AND cprv=@cprv

RETURN @impt

END

SELECT dbo.GetPromxProv(2) AS IMPORTE\_PROMEDIO

--20.Hacer una funcion denominada "HayStock", que devuelva 1 si un producto tiene stock disponibles en un determinado almacen,

-- de lo contrario que devuelva 0

DROP FUNCTION HayStock

CREATE FUNCTION HayStock(@cprd INT,@calma INT)

RETURNS INT

AS

BEGIN

DECLARE @stock DECIMAL,@flasg INT

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

WHERE cprd=@cprd AND calm=@calma

IF(@stock)>0

SET @flasg=1

ELSE

SET @flasg=0

RETURN @flasg

END

SELECT dbo.HayStock(1,1) AS VERIFICAR\_STOCK