1. **Crear sobre la bd prueba una función que devuelva el valor de la hipotenusa de un triángulo a partir de los valores de sus lados.**

USE Prueba

GO

CREATE FUNCTION dbo.hipotenusa1() RETURNS FLOAT AS

BEGIN

DECLARE @ladoA FLOAT ,@ladoB FLOAT , @hipotenusa FLOAT

SET @ladoB = 3;

SET @ladoA = 4;

SELECT @hipotenusa = ((@ladoB\*@ladoB) + (@ladoA\*@ladoA))/(@ladoA + @ladoB)

RETURN @hipotenusa

END

GO

-- MOSTRAR EL RESULTADO

PRINT dbo.hipotenusa1();

o

1. **Crear sobre la bd prueba una función que devuelva 1 o 0 si un número es divisible por otro. Utiliza la función y que se vea escrito por pantalla si son divisibles o no.**

--Crear sobre la bd prueba una función que devuelva 1 o 0 si un número es divisible por otro. Utiliza la función y que se vea escrito

--por pantalla si son divisibles o no.

USE Prueba

GO

CREATE FUNCTION dbo.numeroDivisible()

RETURNS VARCHAR AS

BEGIN

DECLARE @numero1 INT ,@numero2 INT , @resultado VARCHAR(50)

SET @numero1 = 3;

SET @numero2 = 4;

IF (@numero1/@numero2) = 0 SET @resultado = '1'

ELSE SET @resultado = '0'

RETURN @resultado

END

GO

-- MOSTRAR EL RESULTADO--

PRINT dbo.numeroDivisible();

1. **Crea una función IVA que incremente en un 16% el valor de la cantidad que se le pasa.**

**Utilízala para mostrar por pantalla los productos de la BD compras aplicando ese IVA. Utiliza la función para mostrar por pantalla la descripción del artículo, el precio unitario, el precio con el IVA aplicado.**

USE compras

GO

CREATE FUNCTION dbo.funcionIVA()

RETURNS TABLE

AS

RETURN

(SELECT descrart AS DESCRIPCION, preunart AS PRECIO, ((preunart) + (preunart/100)\*16) AS PRECIOIVA

FROM dbo.Articulos)

GO

-- MOSTRAR EL RESULTADO

SELECT \* FROM dbo.funcionIVA()

GO

1. Utilizando funciones lista la descripción del artículo, su precio, el precio medio(función a crear) y la diferencia con el precio medio(función a crear), para la tabla artículos en la bd de compras. NOTA. Se pueden usar funciones dentro de una función.

USE compras

GO

--funcion para calcular precio medio

CREATE FUNCTION dbo.PRECIO\_1()

RETURNS INT

AS

BEGIN

DECLARE @PRECIO\_MEDIO INT = 0

SELECT @PRECIO\_MEDIO = (SELECT AVG(preunart) AS PRECIO\_MEDIO FROM dbo.Articulos)

RETURN @PRECIO\_MEDIO

END

GO

--funcion para calcular diferencia con el precio medio

CREATE FUNCTION dbo.PRECIO\_2()

RETURNS INT

AS

BEGIN

DECLARE @PRECIO\_MEDIO\_DIFERENCIA INT = 0

SELECT @PRECIO\_MEDIO\_DIFERENCIA = (SELECT (preunart - AVG(preunart)) AS PRECIO\_MEDIO FROM dbo.Articulos)

RETURN @PRECIO\_MEDIO\_DIFERENCIA

END

GO

--FUNCION LISTA ARTICULOS

CREATE FUNCTION dbo.listaArticulos()

RETURNS TABLE

AS

RETURN

(SELECT descrart AS DESCRIPCION, preunart as PRECIO,

(SELECT dbo.PRECIO\_1()) AS PRECIOMEDIO,

(SELECT dbo.PRECIO\_2()) AS PRECIOMEDIODIFERENCIA

FROM dbo.Articulos GROUP BY descrart, preunart)

GO

-- MOSTRAR EL RESULTADO FUNCION LISTA ARTICULOS

SELECT \* FROM dbo.listaArticulos()

GO

1. **Crea una función en la bd de jardinería que devuelva el listado de nombre del cliente , y el nombre\_contacto unido con una coma al apellido\_contacto en la tabla clientes.**

**Usa la función para listar sólo aquellos registros cuyos contactos empecen por A.**

--Crea una función en la bd de jardinería que devuelva el listado de nombre del cliente ,

--y el nombre\_contacto unido con una coma al apellido\_contacto en la tabla clientes.

-- Usa la función para listar sólo aquellos registros cuyos contactos empecen por A.

USE jardineria

GO

CREATE FUNCTION dbo.listadoNombreCliente\_contactos()

RETURNS TABLE

AS

RETURN

(SELECT NombreCliente, CONCAT (NombreContacto,',',ApellidoContacto) AS Nombre\_apellidos\_contactos

FROM dbo.Clientes WHERE NombreContacto LIKE 'A%')

GO

-- MOSTRAR EL RESULTADO

SELECT \* FROM dbo.listadoNombreCliente\_contactos()

GO

1. **Modifica la función anterior tal que se le pueda pasar como parámetro por qué letra/s debe empezar el nombre. Utiliza la función con un ejemplo. (BIEN)**

USE jardineria

GO

CREATE FUNCTION dbo.LETRANombreCliente\_contactos(@LETRA VARCHAR(2))

RETURNS TABLE

AS

RETURN

(SELECT NombreCliente, CONCAT (NombreContacto,',',ApellidoContacto) AS Nombre\_apellidos\_contactos

FROM dbo.Clientes WHERE NombreContacto LIKE @LETRA)

GO

-- MOSTRAR EL RESULTADO

SELECT \* FROM dbo.LETRANombreCliente\_contactos('A%')

GO

1. **Definir sobre la BD Jardinería una función que liste los registros de los pedidos para un determinado año, incluyendo código del pedido, el nombre del producto, el precio que fue vendido y la cantidad vendida**

--Definir sobre la BD Jardinería una función que liste los registros de los pedidos para un determinado año,

-- incluyendo código del pedido, el nombre del producto, el precio que fue vendido y la cantidad vendida

USE jardineria

GO

CREATE FUNCTION dbo.LISTADO\_REGISTROS(@año VARCHAR(4))

RETURNS TABLE

AS

RETURN

(SELECT DISTINCT Pedidos.CodigoPedido, Productos.Nombre, YEAR(Pedidos.FechaPedido) AS Año\_Pedido, DetallePedidos.PrecioUnidad AS PRECIO,

DetallePedidos.Cantidad

FROM Pedidos ,DetallePedidos,Productos

WHERE YEAR(Pedidos.FechaPedido) = @año AND Pedidos.CodigoPedido IN (SELECT DetallePedidos.CodigoPedido FROM DetallePedidos

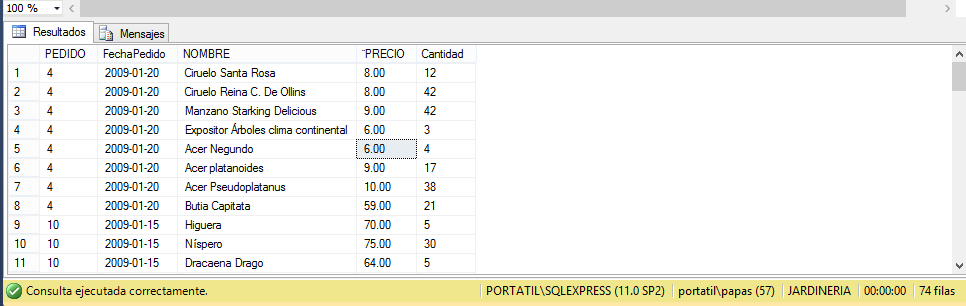
WHERE DetallePedidos.CodigoProducto IN (SELECT Productos.CodigoProducto FROM Productos )))

GO

-- MOSTRAR EL RESULTADO

SELECT \* FROM dbo.LISTADO\_REGISTROS('2009')

GO



1. En la bd Jardinería implementar una función que devuelva el código de pedidos, el número de líneas de ese pedido y el coste total del pedido para un determinado cliente y en un determinado año.

--En la bd Jardinería implementar una función que devuelva el código de pedidos, el número de líneas de ese pedido y el coste total

-- del pedido para un determinado cliente y en un determinado año.

USE jardineria

GO

CREATE FUNCTION dbo.LISTADO\_REGISTROS33(@año VARCHAR(4),@cliente VARCHAR(50))

RETURNS TABLE

AS

RETURN

(SELECT Pedidos.CodigoPedido, COUNT(DISTINCT DetallePedidos.NumeroLinea) AS Numero\_Lineas\_Pedidos,

SUM(DetallePedidos.Cantidad \* Productos.PrecioProveedor) AS COSTE\_TOTAL\_CLIENTE

FROM Pedidos ,DetallePedidos,Productos, Clientes

WHERE YEAR(Pedidos.FechaPedido) = @año AND Clientes.nombreCliente LIKE @cliente

AND Pedidos.CodigoCliente IN (SELECT Clientes.CodigoCliente FROM Clientes)

AND Pedidos.CodigoPedido IN (SELECT DetallePedidos.CodigoPedido FROM DetallePedidos

WHERE DetallePedidos.CodigoProducto IN (SELECT Productos.CodigoProducto FROM Productos ))

GROUP BY Pedidos.CodigoPedido )

GO

-- MOSTRAR EL RESULTADO

SELECT \* FROM dbo.LISTADO\_REGISTROS33('2009', 'Tendo Garden')

GO

1. **Sobre la base de datos banco crea una función que incremente el saldo de una cuenta (se pasa a la función qué cuenta), al multiplicarlo por un valor que también se pasa.**

USE banco

GO

CREATE FUNCTION dbo.incrementoSaldo1(@Incremento INT, @NumeroCuenta INT)

RETURNS TABLE

AS

RETURN

(SELECT cuenta.tipo, cuenta.fecha\_creacion,cuenta.saldo , cuenta.cod\_cuenta, (cuenta.saldo\*@Incremento) AS Nuevo\_Saldo

FROM dbo.cuenta

WHERE cuenta.cod\_cuenta = @NumeroCuenta)

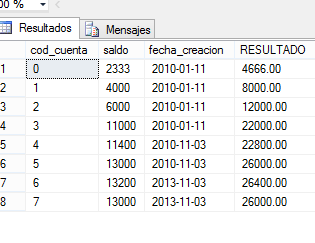
GO

-- EJECUTANDO LA FUNCION

SELECT \* FROM incrementoSaldo1(2,2)

GO

* **Usa la función creada para listar como quedaría para cada cuenta su saldo si se le aplicase un incremento de 2 ptos. Ejemplo de uso de la función en una sentencia de T-sql**



USE banco

GO

CREATE FUNCTION dbo.incrementoSaldo2(@Incremento INT)

RETURNS TABLE

AS

RETURN

(SELECT cuenta.tipo, cuenta.fecha\_creacion,cuenta.saldo , cuenta.cod\_cuenta, (cuenta.saldo\*@Incremento) AS Nuevo\_Saldo

FROM dbo.cuenta)

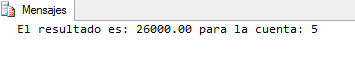
GO

-- EJECUTANDO LA FUNCION

SELECT \* FROM incrementoSaldo2(2)

GO

* **Usa la función creada en un script tal que declarando variables, para un código de cuenta determinado, imprima por pantalla como quedaría el saldo. El incremento=2.**



USE banco

GO

CREATE FUNCTION dbo.incrementoSaldo3() RETURNS INT AS

BEGIN

DECLARE @Incremento INT ,@NumeroCuenta INT , @Resultado INT

SET @Incremento = 2;

SET @NumeroCuenta = 5;

SELECT @Resultado = (SELECT cuenta.saldo\*@Incremento FROM dbo.cuenta

WHERE cuenta.cod\_cuenta = @NumeroCuenta)

RETURN @Resultado

END

GO

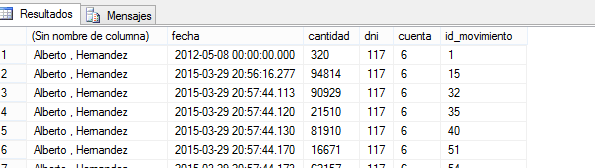
-- EJECUTANDO LA FUNCION

PRINT 'El resultado es ' + cast(dbo.incrementoSaldo3() AS VARCHAR) + ' para la cuenta 5'

GO

1. Sobre la BD banco crear una función que me devuelva todos los movimientos realizados en una determinada cuenta que se le pasa

* Usa la función anterior para ver los movimientos de la cuenta con código 6
* Realiza una consulta tal que, utilizando la función anterior, muestres al lado del movimiento el nombre del propietario de la cuenta con código 6



-- Sobre la BD banco crear una función que me devuelva todos los movimientos realizados en una determinada cuenta que se le pasa

--Usa la función anterior para ver los movimientos de la cuenta con código 6

--Realiza una consulta tal que, utilizando la función anterior, muestres al lado del movimiento el nombre del propietario de la cuenta

--con código 6

USE banco1

GO

CREATE FUNCTION dbo.movimientos\_cuenta(@codigo\_cuenta INT,@movimiento INT)

RETURNS TABLE

AS

RETURN

SELECT CONCAT(cliente.nombre,' ',CONCAT(cliente.apellido1,' ',cliente.apellido2)) AS Nombre\_Cliente

, cliente.dni, movimiento.cantidad, movimiento.cod\_cuenta, movimiento.id\_movimiento

FROM cliente, movimiento, cuenta, tiene

WHERE cliente.dni IN (SELECT movimiento.dni FROM movimiento WHERE movimiento.cod\_cuenta IN

(SELECT cuenta.cod\_cuenta FROM cuenta) )

GO

-- MOSTRAR EL RESULTADO

SELECT \* FROM dbo.movimientos\_cuenta(6, 5)

GO

1. Sobre la bd nba crear una función, tal que cuando se le pase el nombre de una división saque todos los datos de todos los jugadores pertenecientes a equipos que jueguen en la conferencia a la que pertenece esa división.

1ºforma

--Sobre la bd nba crear una función, tal que cuando se le pase el nombre de una división saque todos los datos de todos los

-- jugadores pertenecientes a equipos que jueguen en la conferencia a la que pertenece esa división.

USE nba

GO

CREATE FUNCTION dbo.jugadores\_division(@division VARCHAR(50))

RETURNS TABLE

AS

RETURN

(SELECT jugadores.Nombre FROM dbo.jugadores, equipos

WHERE jugadores.Nombre IN (SELECT equipos.Nombre FROM equipos WHERE equipos.Division = @division))

GO

-- MOSTRAR EL RESULTADO

SELECT \* FROM dbo.jugadores\_division('Atlantic')

GO

2ºforma

--Sobre la bd nba crear una función, tal que cuando se le pase el nombre de una división saque todos los datos de todos los

-- jugadores pertenecientes a equipos que jueguen en la conferencia a la que pertenece esa división.

USE nba

GO

CREATE FUNCTION dbo.jugadores\_division(@division VARCHAR(50)) RETURNS VARCHAR AS

BEGIN

DECLARE @Jugadores VARCHAR

SELECT @Jugadores = jugadores.Nombre FROM dbo.jugadores, equipos

WHERE jugadores.Nombre IN (SELECT equipos.Nombre FROM equipos WHERE equipos.Division = @division)

RETURN @Jugadores

END

GO

-- MOSTRAR EL RESULTADO

PRINT dbo.jugadores\_division('Atlantic')

3ºforma

USE nba

GO

CREATE FUNCTION dbo.jugadores\_division91(@division VARCHAR) RETURNS @TABLA\_jugadores\_Atlantic TABLE(jugadores VARCHAR(50))

AS

BEGIN

INSERT INTO @TABLA\_jugadores\_Atlantic

SELECT jugadores.Nombre FROM dbo.jugadores

WHERE jugadores.Nombre = (SELECT equipos.Nombre FROM equipos WHERE equipos.Division = @division)

RETURN

END

GO

-- EJECUTANDO LA FUNCION

SELECT \* FROM dbo.jugadores\_division91('Central') GO

4ºforma

USE nba

GO

CREATE FUNCTION dbo.jugadores\_division9() RETURNS @TABLA\_jugadores\_Atlantic TABLE(jugadores VARCHAR(50))

AS

BEGIN

DECLARE @division VARCHAR

SET @division = 'Central'

INSERT INTO @TABLA\_jugadores\_Atlantic (jugadores)

SELECT jugadores.Nombre FROM dbo.jugadores

WHERE jugadores.Nombre = (SELECT equipos.Nombre FROM equipos WHERE equipos.Division = @division)

RETURN

END

GO

-- EJECUTANDO LA FUNCION

SELECT \* FROM dbo.jugadores\_division9() GO

USE nba

GO

CREATE FUNCTION dbo.jugadores\_division11111(@division VARCHAR(1000)) RETURNS VARCHAR AS

BEGIN

DECLARE @Jugadores VARCHAR

SELECT @Jugadores = jugadores.Nombre FROM dbo.jugadores

JOIN equipos ON Nombre\_equipo = equipos.Nombre WHERE equipos.Division = @division

RETURN @Jugadores

END

GO

-- MOSTRAR EL RESULTADO

PRINT dbo.jugadores\_division11111('Atlantic')