LABORATORIO No. 7 SQL SERVER TEMA: PROCEDIMIENTOS ALMACENADOS Y VISTAS

Ejercicios utilizando la base de datos Northwind.

1. Realizar un procedimiento almacenado que devuelva los clientes (Customers) según el país (Country).

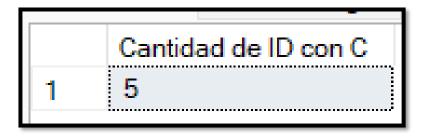
```
create proc tarea_1 as
select *
from Customers
order by Country

exec tarea_1
```



 Crear un procedimiento que determine la cantidad de clientes cuyo identificador (o sea el código) inicia con un determinado carácter.

```
create proc tarea_2 as
    select count(CustomerID) as 'Cantidad de ID con C'
    from Customers
    where CustomerID like 'C%'
exec tarea_2
```



3. Realizar un procedimiento que determine la cantidad de órdenes que tiene registradas un cliente (tabla ORDERS.)

```
-- 3
create proc tarea_3 @id_cliente nchar(5)
as
select count(OrderID) as 'Cantidad de ordenes'
from Orders
where CustomerID = @id_cliente

_exec tarea_3 'ALFKI'
```

4. Realizar una base de datos similar a Northwind, pero que los nombres de tablas, campos y registros estén en español (si ud. La tiene en español, hágalos en ingles)

```
-- 4
create database North_Español

CREATE TABLE "Categorias" (
    "CategoriaID" "int" IDENTITY (1, 1) NOT NULL ,
    "CategoriaNombre" nvarchar (15) NOT NULL ,
    "Descripcion" "ntext" NULL ,
    "Imagen" "image" NULL ,
    CONSTRAINT "PK_Categories" PRIMARY KEY CLUSTERED (
    "CategoriaID"
)
```

```
Producto_ID int IDENTITY(1,1) NOT NULL CONSTRAINT PK_Products PRIMARY KEY,

Nombre_Producto nvarchar(40) NOT NULL,

ID_Proveedor int NULL,

Categoria_ID int NULL,

Cantidad_por_unidad nvarchar(20) NULL,

Precio_por_Unidad money NULL CONSTRAINT DF_Products_UnitPrice DEFAULT (0),

UnidadesEnStock smallint NULL CONSTRAINT DF_Products_UnitsInStock DEFAULT (0),

Unidades_en_pedido smallint NULL CONSTRAINT DF_Products_UnitsOnOrder DEFAULT (0),

Reordenar_nivel smallint NULL CONSTRAINT DF_Products_ReorderLevel DEFAULT (0),

Descontinuado bit NOT NULL CONSTRAINT DF_Products_Discontinued DEFAULT (0),

CONSTRAINT FK_Products_Categoria FOREIGN KEY (Categoria_ID) REFERENCES Categorias(CategoriaID)
```

```
"ClientesID" nchar (5) NOT NULL,

"NombreCompañia" nvarchar (40) NOT NULL,

"NombreContacto" nvarchar (30) NULL,

"TituloContacto" nvarchar (30) NULL,

"Direccion" nvarchar (60) NULL,

"Ciudad" nvarchar (15) NULL,

"Region" nvarchar (15) NULL,

"CodigoPostal" nvarchar (10) NULL,

"Pais" nvarchar (15) NULL,

"Telefono" nvarchar (24) NULL,

"Fax" nvarchar (24) NULL,

CONSTRAINT "PK_Customers" PRIMARY KEY CLUSTERED

(

"ClientesID"
)
```

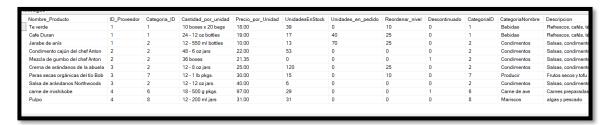
```
REATE TABLE "Ordenes" (
   "OrdenID" "int" IDENTITY (1, 1) NOT NULL ,
   "ClienteID" nchar (5) NULL ,
   "EmploadoID" "int" NULL ,
   "FechaOrden" "datetime" NULL ,
   "FechaRequerida" "datetime" NULL ,
   "FechaEnvio" "datetime" NULL ,
   "EnviarPor" "int" NULL ,
   "Cargo" "money" NULL CONSTRAINT "DF_Orders_Freight" DEFAULT (0),
   "NombreDelBarco" nvarchar (40) NULL ,
   "DirecciónDeEnvío" nvarchar (60) NULL .
   "CiudadBarco" nvarchar (15) NULL
   "RegiónDeEnvio" nvarchar (15) NULL ,
   "CódigoPostalDeEnvío" nvarchar (10) NULL ,
   "PaísDeEnvio" nvarchar (15) NULL
  CONSTRAINT "PK_Orders" PRIMARY KEY CLUSTERED
       "OrdenID"
  CONSTRAINT "FK_Orders_Customers" FOREIGN KEY
       "ClienteID"
   ) REFERENCES "dbo"."Clientes" (
      "ClientesID"
```

```
INSERT "Categorias" VALUES('Bebidas', 'Refrescos, cafés, tés, cervezas y cervezas',0x151C2F0000 INSERT "Categorias" VALUES('Condimentos', 'Salsas, condimentos, productos para untar y condimentos in the condiment of the condition of the condiment of the condiment of the condition of the conditio
```

```
INSERT "Productos" VALUES('Chai',1,1,'10 boxes x 20 bags',18,39,0,10,0)
INSERT "Productos" VALUES('Chang',1,1,'24 - 12 oz bottles',19,17,40,25,0)
INSERT "Productos" VALUES('Aniseed Syrup',1,2,'12 - 550 ml bottles',10,13,70,25,0)
INSERT "Productos" VALUES('Chef Anton''s Cajun Seasoning',2,2,'48 - 6 oz jars',22,53,0,0,0)
INSERT "Productos" VALUES('Chef Anton''s Gumbo Mix',2,2,'36 boxes',21.35,0,0,0,1)
INSERT "Productos" VALUES('Grandma''s Boysenberry Spread',3,2,'12 - 8 oz jars',25,120,0,25,0)
INSERT "Productos" VALUES('Uncle Bob''s Organic Dried Pears',3,7,'12 - 1 lb pkgs.',30,15,0,10,0)
INSERT "Productos" VALUES('Northwoods Cranberry Sauce',3,2,'12 - 12 oz jars',40,6,0,0)
INSERT "Productos" VALUES('Mishi Kobe Niku',4,6,'18 - 500 g pkgs.',97,29,0,0,1)
INSERT "Productos" VALUES('Ikura',4,8,'12 - 200 ml jars',31,31,0,0,0)
```

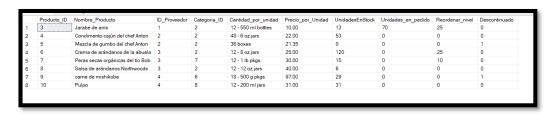
5. Realizar una Vista que contenga todos los registros de tabla de Productos y Categorias a la vez.

```
-- 5
create view tarea_5
as
select * from Productos
inner join
Categorias on Productos.Categoria_ID = Categorias.CategoriaID
select * from tarea_5
```



6. Realizar un procedimiento que seleccione todos los productos que no corresponde a la categoría bebidas (1)

```
-- 6
create view tarea_6
as
select * from Productos
where Productos.Categoria_ID ≠ 1
select * from tarea_6
```



7. Realizar un procedimiento que obtenga la cantidad de registros que no corresponde a condimentos

```
-- 7
create view tarea_7
as
select * from Productos
where Productos.Categoria_ID ≠ 2
select * from tarea_7
```

```
        Producto_ID
        Nombre_Producto
        ID_Proveedor
        Categoria_ID
        Cantidad_por_unidad
        Precio_por_Unidad
        UnidadesEnStock
        Unidades_en_pedido
        Reordenar_nivel
        Descontinuado

        1
        1
        Te verde
        1
        1
        10 boxes x 20 bags
        18.00
        39
        0
        10
        0

        2
        2
        Cafe Duran
        1
        1
        24 - 12 oz bottles
        19.00
        17
        40
        25
        0

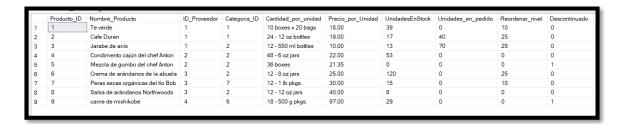
        3
        7
        Peras secas orgánicas del tío Bob
        3
        7
        12 - 1 lb pkgs.
        30.00
        15
        0
        10
        0

        4
        9
        came de mishikobe
        4
        6
        18 - 500 g pkgs.
        97.00
        29
        0
        0
        1

        5
        10
        Pulpo
        4
        8
        12 - 200 ml jars
        31.00
        31
        0
        0
        0
```

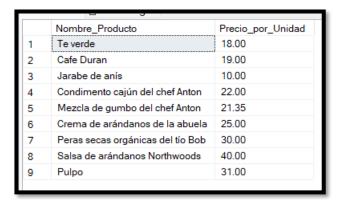
8. Realizar un procedimiento que seleccione todos los campos de los registros que no corresponden a categoría mariscos de la tabla productos.

```
-- 8
create proc tarea_8 as
select * from Productos
where Productos.Categoria_ID ≠ 8
exec tarea_8
```



 Realizar un procedimiento que seleccionar los campos nombre del producto y precio (únicamente) de los productos diferentes a cárnicos.

```
-- 9
create proc tarea_9 as
select Nombre_Producto, Precio_por_Unidad
from Productos
where Productos.Categoria_ID \neq 6
```



10. Realizar un procedimiento que obtenga la cantidad de productos granos y cereales.

```
-- 10
Screate proc tarea_10 as
select count(Nombre_Producto) as 'Cantidad de Productos Grano Cereales'
from Productos
where Productos.Categoria_ID = 5

exec tarea_10
```

```
Cantidad de Productos Grano Cereales
1 0
```

11. Realizar un procedimiento que seleccione los campos nombre del producto y precio (únicamente) de los quesos y productos cárnicos.

```
-- 11
create proc tarea_11 as
select Nombre_Producto, Precio_por_Unidad
from Productos
where Productos.Categoria_ID = 12 or Productos.Categoria_ID = 6
exec tarea_11
```



12. Realizar un procedimiento que seleccione los campos nombre del producto, precio y stock (únicamente) de las frutas secas y mariscos.

```
-- 12
create proc tarea_12 as
select
    Nombre_Producto,
    Precio_por_Unidad,
    UnidadesEnStock
from Productos
where
    Productos.Producto_ID = 7 or
    Productos.Producto_ID = 8

exec tarea_12
```

```
Nombre_Producto Precio_por_Unidad UnidadesEnStock

1 Peras secas orgánicas del tío Bob 30.00 15

2 Salsa de arándanos Northwoods 40.00 6
```

13. Realizar un procedimiento que seleccionar el promedio de los precios de los confites.

```
-- 13
CREATE PROCEDURE tarea_13
    @categoria int

AS

    declare @promedio float
    SELECT @promedio = AVG(Precio_por_Unidad)
        FROM Productos
        WHERE Categoria_ID = @categoria;
    RETURN @promedio

declare @promedio1 int
Execute @promedio1 = tarea_13 2;
select @promedio1 AS 'PROMEDIO DE CONFITES'
```



14. Realice un procedimiento que permita obtener cual fue el producto mas vendido en un año particular suministrado como entrada y cual fue el menos vendido.

15. Realice un procedimiento que encuentre los productos que no se llegaron a vender en un año particular, suministrado como valor de entrada.



16. Escriba un procedimiento que reciba como parámetro de entrada el nombre de un producto o parte de este, como una frase y sea capaz de listar todos los productos que contengan en su nombre dicha frase,

```
-- 16
create proc tarea_16
    @nombrep varchar(20)
as
select ProductName
from Products
where ProductName like '%'+@nombrep+'%'
exec tarea_16 'be';
```



17. Escriba un procedimiento que recibiendo un número de orden, encuentre los días que han transcurrido desde que se realizó dicha órden. Maneje este dato como un parámetro de salida.

```
--17

create proc tarea_17
    @idorden int,
    @dias int OUTPUT

as

select @dias = DATEDIFF(DAY, RequiredDate, GETDATE())
from Orders
where OrderID = @idorden

declare @dias1 INT
exec tarea_17 10271, @dias1 output
select @dias1 AS 'CANTIDAD DE DIAS'

CANTIDAD DE DIAS
1 9931
```

18. Escriba un procedimiento que valide si tenemos registrado a un cliente. Enviar mensajes indicando si existe o no el cliente. Debe recibir el código del cliente como dato de entrada.

```
create proc tarea_18
    @idcliente varchar(10)
as
begin
    if exists (select * from Customers where CustomerID = @idcliente)
    begin
        select 'EL CLIENTE INGRESADO EXISTE' as RESULTADO
    end
    else
        select 'EL CLIENTE INGRESADO NO EXISTE' AS RESULTADO
    return
end

Execute tarea_18 WARTP;
```

RESULTADO

1 EL CLIENTE INGRESADO NO EXISTE