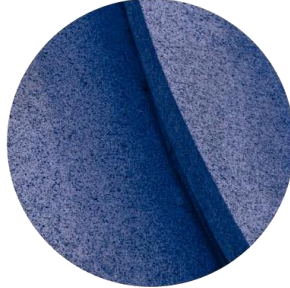
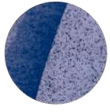


# Base de datos Avanzada



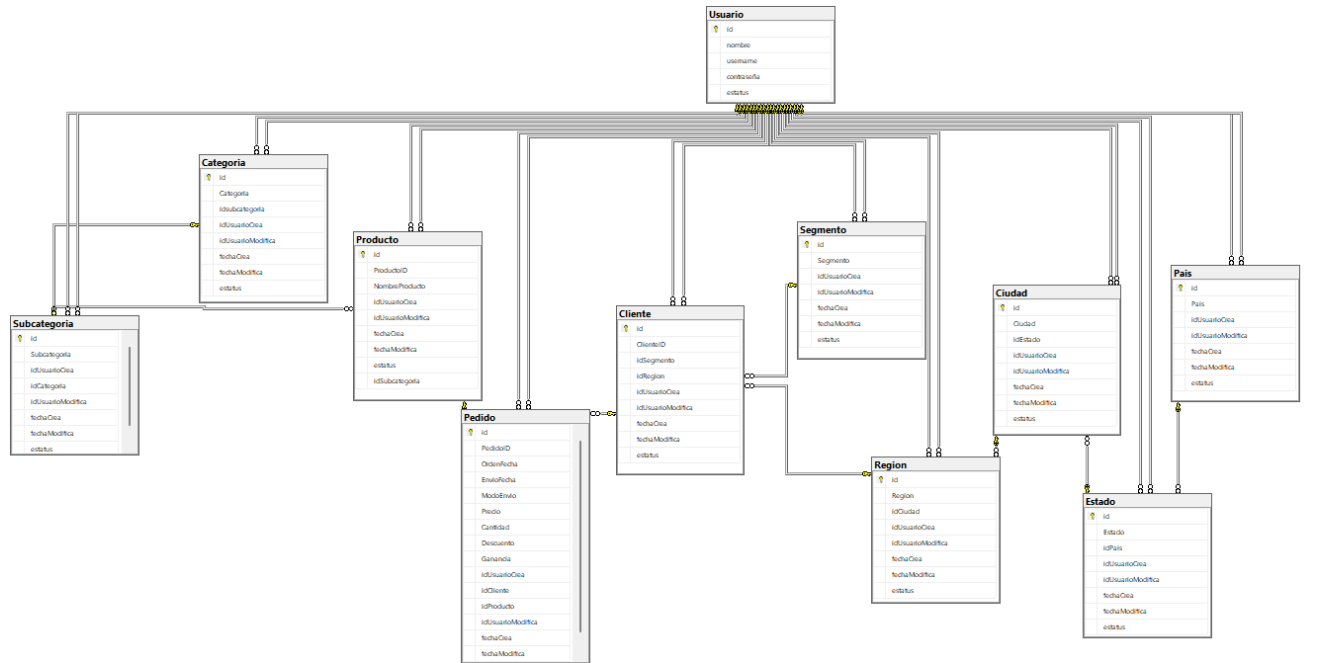
***DATA: Supermarket***

Karen Guadalupe Valdes Torres

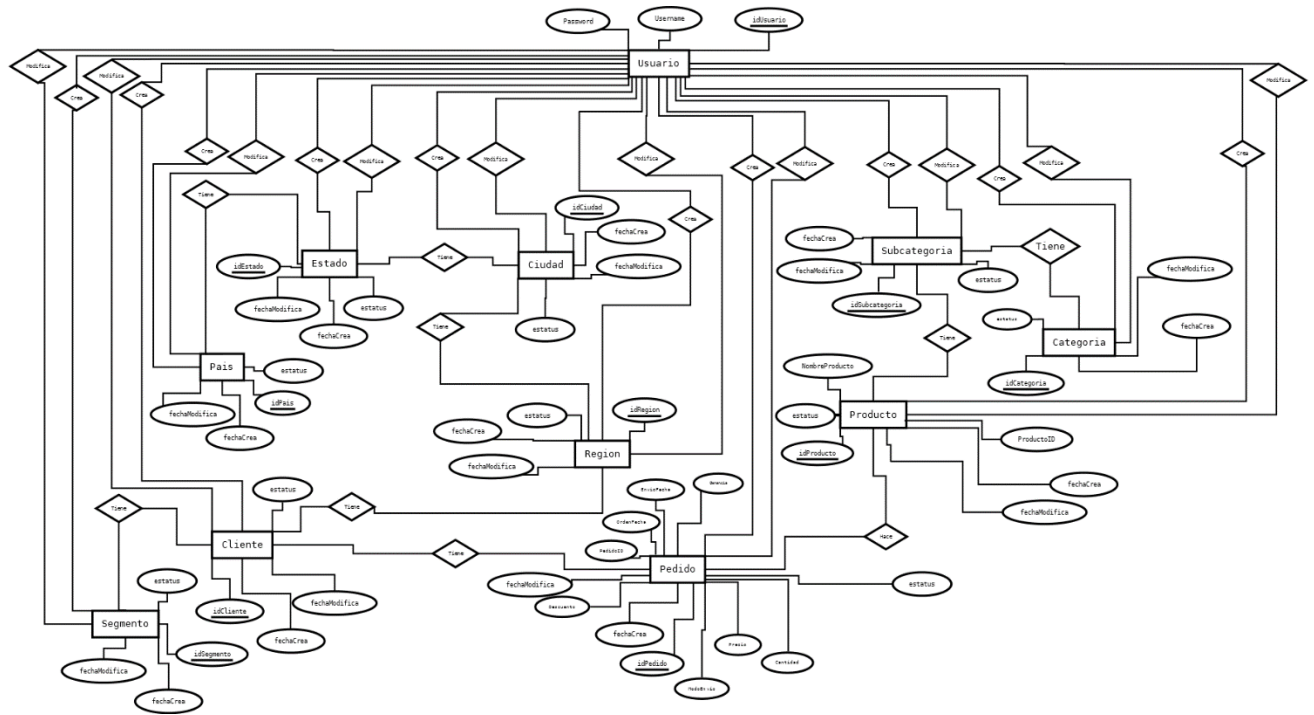
ISC 8A TV

BASE DE DATOS AVANZADA

# Diagrama Relación



## Diagrama E-R



# SQL SERVER QUERY

```
use SuperMarketPrueba2
```

```
select * from [dbo].[Sample_ Superstore]
```

```
create table Producto(  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    ProductoID varchar (30),  
    NombreProducto VARCHAR(255),  
    idUsuarioCrea INT,  
    idUsuarioModifica INT DEFAULT NULL,  
    fechaCrea DATETIME DEFAULT GETDATE(),  
    fechaModifica DATETIME DEFAULT NULL,  
    estatus BIT DEFAULT 1  
)
```

```
alter table Producto  
add idSubcategoria int
```

```
create table Categoria(  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    Categoria varchar(255),  
    idsubcategoria int,  
    idUsuarioCrea INT,  
    idUsuarioModifica INT DEFAULT NULL,  
    fechaCrea DATETIME DEFAULT GETDATE(),  
    fechaModifica DATETIME DEFAULT NULL,  
    estatus BIT DEFAULT 1  
)
```

```
create table Subcategoria(  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    Subcategoria varchar(40),
```

```
idUsuarioCrea INT,  
idCategoria INT,  
idUsuarioModifica INT DEFAULT NULL,  
fechaCrea DATETIME DEFAULT GETDATE(),  
fechaModifica DATETIME DEFAULT NULL,  
estatus BIT DEFAULT 1  
)  
-- alter table Subcategoria add idCategoria int
```

```
create table Cliente(  
id INT IDENTITY(1,1) PRIMARY KEY,  
    ClienteID varchar (30),  
    idSegmento int,  
    idRegion int,  
    idUsuarioCrea INT,  
idUsuarioModifica INT DEFAULT NULL,  
fechaCrea DATETIME DEFAULT GETDATE(),  
fechaModifica DATETIME DEFAULT NULL,  
estatus BIT DEFAULT 1  
)
```

```
--alter table Cliente add idSegmento int
```

```
select * from Cliente
```

```
create table Segmento(  
id INT IDENTITY(1,1) PRIMARY KEY,  
Segmento varchar (50),  
idUsuarioCrea INT,  
idUsuarioModifica INT DEFAULT NULL,  
fechaCrea DATETIME DEFAULT GETDATE(),  
fechaModifica DATETIME DEFAULT NULL,
```

```
estatus BIT DEFAULT 1
```

```
)
```

```
create table Ciudad(
```

```
id INT IDENTITY(1,1) PRIMARY KEY,
```

```
Ciudad varchar (max),
```

```
idEstado int,
```

```
idUsuarioCrea INT,
```

```
idUsuarioModifica INT DEFAULT NULL,
```

```
fechaCrea DATETIME DEFAULT GETDATE(),
```

```
fechaModifica DATETIME DEFAULT NULL,
```

```
estatus BIT DEFAULT 1
```

```
)
```

```
--alter table Ciudad add idEstado int
```

```
create table Pais(
```

```
id INT IDENTITY(1,1) PRIMARY KEY,
```

```
Pais varchar (max),
```

```
idUsuarioCrea INT,
```

```
idUsuarioModifica INT DEFAULT NULL,
```

```
fechaCrea DATETIME DEFAULT GETDATE(),
```

```
fechaModifica DATETIME DEFAULT NULL,
```

```
estatus BIT DEFAULT 1
```

```
)
```

```
create table Estado(
```

```
id INT IDENTITY(1,1) PRIMARY KEY,
```

```
Estado varchar (max),
```

```
idPais int,
```

```
idUsuarioCrea INT,
```

```
idUsuarioModifica INT DEFAULT NULL,
```

```
fechaCrea DATETIME DEFAULT GETDATE(),
fechaModifica DATETIME DEFAULT NULL,
estatus BIT DEFAULT 1
)
--alter table Estado add idPais int
```

```
create table Region(
    id INT IDENTITY(1,1) PRIMARY KEY,
    Region varchar (max),
    idCiudad int,
    idUsuarioCrea INT,
    idUsuarioModifica INT DEFAULT NULL,
    fechaCrea DATETIME DEFAULT GETDATE(),
    fechaModifica DATETIME DEFAULT NULL,
    estatus BIT DEFAULT 1
)
--alter table Region add idCiudad int
select * from Region
```

```
create table Pedido(
    id INT IDENTITY(1,1) PRIMARY KEY,
    PedidoID varchar (30),
    OrdenFecha nvarchar (30),
    EnvioFecha nvarchar (30),
    ModoEnvio nvarchar (30),
    Precio nvarchar (30),--venta
    Cantidad nvarchar (30),
    Descuento nvarchar (30),
    Ganancia nvarchar (30) ,
    idUsuarioCrea INT,
    idCliente int,
    idProducto int,
    idUsuarioModifica INT DEFAULT NULL,
```

```
fechaCrea DATETIME DEFAULT GETDATE(),  
fechaModifica DATETIME DEFAULT NULL,  
estatus BIT DEFAULT 1,  
  
)
```

```
CREATE TABLE Usuario(  
id INT IDENTITY(1,1) PRIMARY KEY,  
nombre VARCHAR(50) NOT NULL,  
username VARCHAR(50) NOT NULL,  
contraseña VARCHAR(50),  
estatus BIT DEFAULT 1,  
)
```

```
ALTER TABLE Pedido  
ADD CONSTRAINT fk_cliente  
FOREIGN KEY (idCliente)  
REFERENCES Cliente(id);
```

```
ALTER TABLE Pedido  
ADD CONSTRAINT fk_producto  
FOREIGN KEY (idProducto)  
REFERENCES Producto(id);
```

```
-- Relación entre país y estado  
ALTER TABLE Estado  
ADD CONSTRAINT fk_estado_pais  
FOREIGN KEY (idPais)  
REFERENCES Pais(id);
```

```
-- Relación entre estado y ciudad  
ALTER TABLE Ciudad  
ADD CONSTRAINT fk_ciudad_estado
```



```
FOREIGN KEY (idEstado)
REFERENCES Estado(id);
```

-- Relación entre ciudad y región

```
ALTER TABLE Region
ADD CONSTRAINT fk_region_ciudad
FOREIGN KEY (idCiudad)
REFERENCES Ciudad(id);
```

-- Relación entre región y cliente

```
ALTER TABLE Cliente
ADD CONSTRAINT fk_cliente_region
FOREIGN KEY (idRegion)
REFERENCES Region(id);
```

--

```
ALTER TABLE Producto
ADD CONSTRAINT fk_subcategoria
FOREIGN KEY (idSubcategoria)
REFERENCES Subcategoria(id);
```

```
ALTER TABLE subcategoria
ADD CONSTRAINT fk_categoria
FOREIGN KEY (idCategoria)
REFERENCES Categoria(id);
```

```
ALTER TABLE Cliente
ADD CONSTRAINT fk_segmento
FOREIGN KEY (idSegmento)
REFERENCES Segmento(id);
```

-- CONECTAR TABLAS CON USUARIO

ALTER TABLE Cliente

ADD CONSTRAINT FK\_ClienteUsuarioCrea

FOREIGN KEY (idUserioCrea) REFERENCES Usuario(id)

ALTER TABLE Cliente

ADD CONSTRAINT FK\_ClienteUsuarioModifica

FOREIGN KEY (idUserioModifica) REFERENCES Usuario(id)

ALTER TABLE Pedido

ADD CONSTRAINT FK\_PedidoUsuarioCrea

FOREIGN KEY (idUserioCrea) REFERENCES Usuario(id)

ALTER TABLE Pedido

ADD CONSTRAINT FK\_PedidoUsuarioModifica

FOREIGN KEY (idUserioModifica) REFERENCES Usuario(id)

ALTER TABLE Producto

ADD CONSTRAINT FK\_ProductoUsuarioCrea

FOREIGN KEY (idUserioCrea) REFERENCES Usuario(id)

ALTER TABLE Producto

ADD CONSTRAINT FK\_ProductoUsuarioModifica

FOREIGN KEY (idUserioModifica) REFERENCES Usuario(id)

ALTER TABLE Categoria

ADD CONSTRAINT FK\_CategoriaUsuarioCrea

FOREIGN KEY (idUserioCrea) REFERENCES Usuario(id)

ALTER TABLE Categoria

```
ADD CONSTRAINT FK_CategoriaUsuarioModifica  
FOREIGN KEY (idUsuarioModifica) REFERENCES Usuario(id)
```

```
ALTER TABLE Subcategoria  
ADD CONSTRAINT FK_SubcategoriaUsuarioCrea  
FOREIGN KEY (idUsuarioCrea) REFERENCES Usuario(id)
```

```
ALTER TABLE Subcategoria  
ADD CONSTRAINT FK_SubcategoriaUsuarioModifica  
FOREIGN KEY (idUsuarioModifica) REFERENCES Usuario(id)
```

```
ALTER TABLE Segmento  
ADD CONSTRAINT FK_SegmentoUsuarioCrea  
FOREIGN KEY (idUsuarioCrea) REFERENCES Usuario(id)
```

```
ALTER TABLE Segmento  
ADD CONSTRAINT FK_SegmentoUsuarioModifica  
FOREIGN KEY (idUsuarioModifica) REFERENCES Usuario(id)
```

```
ALTER TABLE Pais  
ADD CONSTRAINT FK_PaisUsuarioCrea  
FOREIGN KEY (idUsuarioCrea) REFERENCES Usuario(id)
```

```
ALTER TABLE Pais  
ADD CONSTRAINT FK_PaisUsuarioModifica  
FOREIGN KEY (idUsuarioModifica) REFERENCES Usuario(id)
```

```
ALTER TABLE Ciudad  
ADD CONSTRAINT FK_CiudadUsuarioCrea  
FOREIGN KEY (idUsuarioCrea) REFERENCES Usuario(id)
```

```
ALTER TABLE Ciudad  
ADD CONSTRAINT FK_CiudadUsuarioModifica
```

FOREIGN KEY (idUserioModifica) REFERENCES Usuario(id)

ALTER TABLE Estado

ADD CONSTRAINT FK\_EstadoUsuarioCrea

FOREIGN KEY (idUserioCrea) REFERENCES Usuario(id)

ALTER TABLE Estado

ADD CONSTRAINT FK\_EstadoUsuarioModifica

FOREIGN KEY (idUserioModifica) REFERENCES Usuario(id)

ALTER TABLE Region

ADD CONSTRAINT FK\_RegionUsuarioCrea

FOREIGN KEY (idUserioCrea) REFERENCES Usuario(id)

ALTER TABLE Region

ADD CONSTRAINT FK\_RegionUsuarioModifica

FOREIGN KEY (idUserioModifica) REFERENCES Usuario(id)

-- INDICES

CREATE INDEX IX\_Usuario ON Usuario(id)

CREATE INDEX IX\_Pedido ON Pedido(id)

CREATE INDEX IX\_Cliente ON Cliente(id)

CREATE INDEX IX\_Producto ON Producto(id)

CREATE INDEX IX\_Categoria ON Categoria(id)

CREATE INDEX IX\_Subcategoria ON Subcategoria(id)

CREATE INDEX IX\_Segmento ON Segmento(id)

```
CREATE INDEX IX_Region ON Region(id)
```

```
CREATE INDEX IX_Ciudad ON Ciudad(id)
```

```
CREATE INDEX IX_Estado ON Estado(id)
```

```
CREATE INDEX IX_Pais ON Pais(id)
```

```
--POBLAR
```

```
INSERT INTO usuario (nombre, username, contraseña)
```

```
VALUES ('Admin','admin',CONVERT(NVARCHAR(50), HASHBYTES('SHA1','admin'),2))
```

```
select * from Subcategoria
```

```
--
```

```
insert into Categoria(Categoria,idUsuarioCrea)
```

```
select distinct Category,1
```

```
from [dbo].[Sample_ Superstore]
```

```
--
```

```
insert into Pais(Pais,idUsuarioCrea)
```

```
select distinct Country,1
```

```
from [dbo].[Sample_ Superstore]
```

```
--
```

```
insert into Segmento(Segmento,idUsuarioCrea)
```

```
select distinct Segment,1
```

```
from [dbo].[Sample_ Superstore]
```

```
insert into Estado(Estado,idPais ,idUsuarioCrea)
```

```
select distinct State,
```

```
(SELECT (id) FROM Pais WHERE Pais = [dbo].[Sample_ Superstore].[Country]) as idPais
```

```
,1
```

```
from [dbo].[Sample_ Superstore]
```

```
insert into Ciudad(Ciudad,idUsuarioCrea,idEstado)
```

```
select distinct City,1,(SELECT TOP 1(id) FROM Estado WHERE Estado = [dbo].[Sample_
Superstore].[State]) as idEstado
```

```
from [dbo].[Sample_ Superstore]
```

```
insert into Region(Region,idCiudad ,idUsuarioCrea)
```

```
select distinct Region,
```

```
(SELECT TOP 1(id) FROM Ciudad WHERE Ciudad = [dbo].[Sample_ Superstore].[City]) as idCiudad,1
```

```
from [dbo].[Sample_ Superstore]
```

```
insert into Subcategoria(Subcategoria,idUsuarioCrea,idCategoria)
```

```
select distinct Sub_Category,1,(SELECT (id) FROM Categoria WHERE Categoria = [dbo].[Sample_
Superstore].[Category]) as idCategoria
```

```
from [dbo].[Sample_ Superstore]
```

```
insert into Producto(ProductoID,NombreProducto,idUsuarioCrea,idSubcategoria)
```

```
select distinct Product_ID,Product_Name,1,(SELECT (id) FROM Subcategoria WHERE Subcategoria =
[dbo].[Sample_ Superstore].[Sub_Category]) as idSubcategoria
```

```
from [dbo].[Sample_ Superstore]
```

```
select* from Producto
```

```
INSERT INTO Cliente (ClienteID, idRegion, idSegmento, idUsuarioCrea)
```

```
SELECT DISTINCT
```

```
Customer_ID,
```

```
(SELECT top 1 (id) FROM Region WHERE Region = [dbo].[Sample_ Superstore].[Region]) AS idRegion,
```

```
(SELECT(id) FROM Segmento WHERE Segmento =[dbo].[Sample_ Superstore].[Segment]) AS
idSegmento,
```

```
1 AS idUsuarioCrea
```

```
FROM [dbo].[Sample_ Superstore];
```

```
select * from Pedido
```

```
insert into
```

```
Pedido(PedidoID,Precio,Cantidad,Descuento,Ganancia,OrdenFecha,EnvioFecha,ModoEnvio,idCliente,idProducto ,idUsuarioCrea)
```

```
select distinct Order_ID,Sales,Quantity,Discount,Profit,Order_Date,Ship_Date,Ship_Mode,
```

```
(SELECT TOP 1(id) FROM Cliente WHERE ClienteID = [dbo].[Sample_ Superstore].[Customer_ID]) as  
idCliente,
```

```
(SELECT TOP 1(id) FROM Producto WHERE ProductID = [dbo].[Sample_ Superstore].[Product_ID]) as  
idProducto,1
```

```
from [dbo].[Sample_ Superstore]
```

```
CREATE VIEW Vw_DatosCliente AS
```

```
SELECT c.id, c.ClienteID AS Cliente_Identificador,
```

```
p.Pais AS NombrePais,
```

```
e.Estado AS NombreEstado,
```

```
ci.Ciudad AS NombreCiudad,
```

```
r.Region AS NombreRegion,
```

```
s.Segmento AS NombreSegmento
```

```
FROM Cliente c
```

```
inner JOIN Region r ON c.idRegion = r.id
```

```
inner JOIN Ciudad ci ON r.idCiudad = ci.id
```

```
inner JOIN Estado e ON ci.idEstado = e.id
```

```
inner JOIN Pais p ON e.idPais = p.id
```

```
inner JOIN Segmento s ON c.idSegmento = s.id;
```

```
create view VW_DatosSuperMarket as
```

```
SELECT c.id, c.ClienteID AS Cliente_Identificador,
```

```
p.Pais AS NombrePais,
```

```
e.Estado AS NombreEstado,
```

```
ci.Ciudad AS NombreCiudad,
```

```
r.Region AS NombreRegion,
```

```
s.Segmento AS NombreSegmento,
```

```

pro.NombreProducto, sub.Subcategoria, Cat.Categoria
FROM Pedido ped
inner join Producto pro on ped.idProducto= pro.id
inner join Subcategoria sub on pro.idSubcategoria =sub.id
inner join Categoria cat on sub.idCategoria = cat.id
inner join Cliente c on Ped.idCliente =c.id
inner JOIN Region r ON c.idRegion = r.id
inner JOIN Ciudad ci ON r.idCiudad = ci.id
inner JOIN Estado e ON ci.idEstado = e.id
inner JOIN Pais p ON e.idPais = p.id
inner JOIN Segmento s ON c.idSegmento = s.id

```

```

create view VW_DatosSuperMarket1 as
SELECT c.id, c.ClienteID AS Cliente_Identificador,
p.Pais AS NombrePais,
e.Estado AS NombreEstado,
ci.Ciudad AS NombreCiudad,
r.Region AS NombreRegion,
s.Segmento AS NombreSegmento,
pro.NombreProducto, ped.PedidoID, ped.Precio, ped.Cantidad, ped.Descuento, ped.Ganancia, ped.ModoEnvio,
ped.OrdenFecha, ped.EnvioFecha,
sub.Subcategoria, Cat.Categoria,
pro.ProductoID
FROM Pedido ped
inner join Producto pro on ped.idProducto= pro.id
inner join Subcategoria sub on pro.idSubcategoria =sub.id
inner join Categoria cat on sub.idCategoria = cat.id
inner join Cliente c on Ped.idCliente =c.id
inner JOIN Region r ON c.idRegion = r.id
inner JOIN Ciudad ci ON r.idCiudad = ci.id
inner JOIN Estado e ON ci.idEstado = e.id
inner JOIN Pais p ON e.idPais = p.id
inner JOIN Segmento s ON c.idSegmento = s.id

```



```
select distinct Cliente_Identificador,NombreSegmento,NombreRegion, NombreCiudad, NombreEstado ,
NombrePais
```

```
from VW_DatosSuperMarket
```

```
where NombreEstado='Oregon'
```

```
select id, ProductoID,NombreProducto,Categoria,Subcategoria
```

```
from VW_DatosSuperMarket1
```

```
select * from Producto where ProductoID ='lola'
```

```
select * from Cliente
```

```
alter procedure LlenarCbRegion
```

```
@id varchar (255)
```

```
as
```

```
begin
```

```
    select id, Region as Nombre
```

```
    from Region
```

```
    where idCiudad=@id
```

```
end
```

```
exec LlenarCbRegion
```

```
create procedure LlenarCbCiudad
```

```
as
```

```
begin
```

```
    select id, Ciudad as Nombre
```

```
    from Ciudad
```

```
end
```

```
create procedure LlenarCbPais
```

```
as
```

```
begin
```

```
    select id, Pais as Nombre
```

```
        from Pais
end
```

```
create procedure LlenarCbEstado
as
begin
    select id, Estado as Nombre
    from Estado
end
```

```
alter procedure llenarDtCliente
@idpais varchar (255),
@idEstado varchar (255),
@idCiudad varchar (255),
@idRegion varchar (255),
@texto varchar (255)
as
begin
    select c.id, c.ClienteID AS Cliente_Identificador,
    p.Pais AS NombrePais,
    e.Estado AS NombreEstado,
    ci.Ciudad AS NombreCiudad,
    r.Region AS NombreRegion
    from Cliente c
    inner JOIN Region r ON c.idRegion = r.id
    inner JOIN Ciudad ci ON r.idCiudad = ci.id
    inner JOIN Estado e ON ci.idEstado = e.id
    inner JOIN Pais p ON e.idPais = p.id
    where r.id like '%'+ @idRegion+'%' and ci.id like '%'+ @idCiudad +'%'+and e.id like '%'+
    @idEstado+'%'and p.id like '%'+ @idpais+'%' and c.ClienteID like '%'+@texto+'%'
end
```

```
exec llenarDtCliente '1','1','1','1','1'
```

```
select * from region
```

```
select * from Estado where Estado='Pennsylvania'
```

```
select * from Ciudad order by Ciudad
```

## SP QUERY

--SP CLIENTE

```
CREATE PROCEDURE [dbo].[SPAgregarCliente]
```

```
    @ClienteID NVARCHAR(100),
```

```
    @IdSegmento INT,
```

```
    @idRegion INT,
```

```
    @IdCiudad INT,
```

```
    @IdEstado INT,
```

```
    @IdPais INT
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    BEGIN TRY
```

```
        BEGIN TRANSACTION;
```

```
        -- Insertar el nuevo cliente
```

```
        INSERT INTO Cliente (ClienteID, idSegmento, idRegion)
```

```
        VALUES (@ClienteID, @IdSegmento, @IdRegion);
```

```
        -- Insertar nueva región si no existe
```

```
            INSERT INTO Region (id, Region, idCiudad)
```

```
            VALUES (@IdRegion, 'Nombre de la Región',@IdCiudad); -- Aquí puedes especificar  
el nombre de la región
```

```
        -- Insertar nueva ciudad si no existe
```

```
            INSERT INTO Ciudad (id, Ciudad,idEstado)
```

```
            VALUES (@IdCiudad, 'Nombre de la Ciudad',@IdEstado); -- Aquí puedes especificar  
el nombre de la ciudad
```

```
        -- Insertar nuevo estado si no existe
```

```
            INSERT INTO Estado (id, Estado,idPais)
```

```
            VALUES (@IdEstado, 'Nombre del Estado',@IdPais); -- Aquí puedes especificar el  
nombre del estado
```

```
        -- Insertar nuevo país si no existe
```

```
            INSERT INTO Pais (id, Pais)
```

```
            VALUES (@IdPais, 'Nombre del País'); -- Aquí puedes especificar el nombre del  
país
```

```
        COMMIT TRANSACTION;
```

```
    END TRY
```

```

        BEGIN CATCH
            ROLLBACK TRANSACTION;
            -- Aquí puedes manejar cualquier error que ocurra durante la transacción
        END CATCH;
    END;

```

```

CREATE PROCEDURE SP_EditarCliente
(
    @_id INT,
    @_ClienteID VARCHAR(30),
    @NuevoIdUsuarioModifica INT
)
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION;

        -- Actualizar el nombre del producto y el usuario que lo modificó
        UPDATE cliente
        SET ClienteID = @_ClienteID,
            idUsuarioModifica = @NuevoIdUsuarioModifica,
            fechaModifica = GETDATE()
        WHERE id = @_id;

        COMMIT TRANSACTION;
        SELECT 'El producto ha sido actualizado correctamente.' AS SuccessMessage;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
        SELECT 'Ocurrió un error al intentar actualizar el cliente.' AS ErrorMessage;
    END CATCH;
END

```

--ELIMINAR

```

CREATE PROCEDURE SP_EliminarCliente
(
    @ClienteID INT
)
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION;

        -- Eliminar los registros relacionados en la tabla Segmento
        DELETE FROM segmento WHERE id = (SELECT id FROM cliente WHERE id = @ClienteID);

        -- Eliminar los registros relacionados en la tabla Region
        DELETE FROM region WHERE id = (SELECT id FROM cliente WHERE id = @ClienteID);

        -- Eliminar los registros relacionados en la tabla Estado
        DELETE FROM estado WHERE id IN (SELECT estado.id FROM ciudad INNER JOIN cliente ON
cliente.id = ciudad.id WHERE cliente.id = @ClienteID);

        -- Eliminar los registros relacionados en la tabla Ciudad
        DELETE FROM ciudad WHERE id = (SELECT id FROM cliente WHERE id = @ClienteID);

        -- Eliminar los registros relacionados en la tabla Pais
        DELETE FROM pais WHERE id IN (SELECT pais.id FROM estado INNER JOIN ciudad ON
estado.id = ciudad.id INNER JOIN cliente ON cliente.id = ciudad.id WHERE cliente.id =

```

```

@ClienteID);

-- Eliminar el registro de la tabla Cliente
DELETE FROM cliente WHERE id = @ClienteID;

COMMIT TRANSACTION

```

## --SP PRODUCTO

```

Create PROCEDURE [dbo].[SP_AgregarProducto]
    @ProductoID VARCHAR(30),
    @NombreProducto VARCHAR(255),
    @IdUsuarioCrea INT
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        -- Insertar el nuevo producto
        INSERT INTO Producto (ProductoID, NombreProducto, idUsuarioCrea)
        VALUES (@ProductoID, @NombreProducto, @IdUsuarioCrea);

        COMMIT TRANSACTION;
        PRINT 'El producto se ha agregado correctamente.';
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;

        PRINT 'Ocurrió un error al intentar agregar el producto.';
        THROW; -- Lanza el error para su manejo externo
    END CATCH;
END;

---

CREATE PROCEDURE [dbo].[SP_EditarProducto]
(
    @ProductoID VARCHAR(30),
    @NuevoNombreProducto VARCHAR(255),
    @NuevoIdUsuarioModifica INT,
    @fechaModifica date
)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        -- Actualizar el nombre del producto y el usuario que lo modificó
        UPDATE Producto
        SET NombreProducto = @NuevoNombreProducto,
            idUsuarioModifica = @NuevoIdUsuarioModifica,
            fechaModifica = GETDATE()
        WHERE ProductoID = @ProductoID;

        COMMIT;
        SELECT 'El producto ha sido actualizado correctamente.' AS SuccessMessage;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK;
    END CATCH;
END;

```

```

        SELECT 'Ocurrió un error al intentar actualizar el producto.' AS ErrorMessage;
    END CATCH;
END;

```

```

CREATE PROCEDURE SP_EditarProducto
(
    @_ProductoID VARCHAR(30),
    @NuevoNombreProducto VARCHAR(255),
    @NuevoIdUsuarioModifica INT
)
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION;

        -- Actualizar el nombre del producto y el usuario que lo modificó
        UPDATE producto
        SET NombreProducto = @NuevoNombreProducto,
            idUsuarioModifica = @NuevoIdUsuarioModifica,
            fechaModifica = GETDATE()
        WHERE ProductoID = @_ProductoID;

        COMMIT TRANSACTION;
        SELECT 'El producto ha sido actualizado correctamente.' AS SuccessMessage;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
        SELECT 'Ocurrió un error al intentar actualizar el producto.' AS ErrorMessage;
    END CATCH;
END

```

--

```

Create PROCEDURE [dbo].[SP_EliminarProducto]
    @ProductoID VARCHAR(30)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        -- Actualizar el estado del producto a inactivo
        UPDATE Producto
        SET estatus = 0,
            fechaModifica = GETDATE()
        WHERE ProductoID = @ProductoID;

        COMMIT TRANSACTION;
        PRINT 'El producto ha sido eliminado correctamente.';
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;

        PRINT 'Ocurrió un error al intentar eliminar el producto.';
    END CATCH;
END;

```

--SP PEDIDO

CREATE PROCEDURE SP\_AgregarPedido

(

    @PedidoID VARCHAR(30),

    @Precio DECIMAL(18, 2),

    @Cantidad INT,

    @Descuento DECIMAL(18, 2),

    @Ganancia DECIMAL(18, 2),

    @OrdenFecha DATETIME,

    @EnvioFecha DATETIME,

    @ModoEnvio NVARCHAR(100)

)

AS

BEGIN

    SET NOCOUNT ON;

    -- Insertar el nuevo pedido

    INSERT INTO Pedido (PedidoID, Precio, Cantidad, Descuento, Ganancia, OrdenFecha, EnvioFecha, ModoEnvio)

    VALUES (@PedidoID, @Precio, @Cantidad, @Descuento, @Ganancia, @OrdenFecha, @EnvioFecha, @ModoEnvio);

    SELECT 'El nuevo pedido ha sido agregado correctamente.' AS SuccessMessage;

END

--

CREATE PROCEDURE SP\_EditarPedido

(

    @\_PedidoID VARCHAR(30),

    @\_Precio DECIMAL(18, 2),

    @\_Cantidad INT,

    @\_Descuento DECIMAL(18, 2),

    @\_Ganancia DECIMAL(18, 2),

    @\_OrdenFecha DATETIME,

    @\_EnvioFecha DATETIME,

    @\_ModoEnvio VARCHAR(100),

    @\_idUserModifica INT

)

AS

BEGIN

```

SET NOCOUNT ON;

-- Actualizar el pedido existente
UPDATE Pedido
SET Precio = @_Precio,
    Cantidad = @_Cantidad,
    Descuento = @_Descuento,
    Ganancia = @_Ganancia,
    OrdenFecha = @_OrdenFecha,
    EnvioFecha = @_EnvioFecha,
    ModoEnvio = @_ModoEnvio,
    idUsuarioModifica = @_idUsuarioModifica,
    fechaModifica = GETDATE()
WHERE PedidoID = @_PedidoID;

SELECT 'El pedido ha sido actualizado correctamente.' AS SuccessMessage;
END

--

CREATE PROCEDURE SP_EliminarPedido
(
    @_PedidoID VARCHAR(255)
)
AS
BEGIN
    SET NOCOUNT ON;

    -- Actualizar el pedido existente
    UPDATE pedido
    SET estatus = 0,
        idUsuarioModifica = 1, -- Actualiza el ID del usuario que lo modificó
        fechaModifica = GETDATE() -- Actualiza la fecha de modificación
    WHERE PedidoID = @_PedidoID;

    SELECT 'El pedido ha sido eliminado correctamente.' AS SuccessMessage;
END

```



## Vistas

```
CREATE VIEW Vw_DatosCliente AS
SELECT c.id, c.ClienteID AS Cliente_Identificador,
p.Pais AS NombrePais,
e.Estado AS NombreEstado,
ci.Ciudad AS NombreCiudad,
r.Region AS NombreRegion,
s.Segmento AS NombreSegmento
FROM Cliente c
inner JOIN Region r ON c.idRegion = r.id
inner JOIN Ciudad ci ON r.idCiudad = ci.id
inner JOIN Estado e ON ci.idEstado = e.id
inner JOIN Pais p ON e.idPais = p.id
inner JOIN Segmento s ON c.idSegmento = s.id;
```

```
create view VW_DatosSuperMarket1 as
SELECT c.id, c.ClienteID AS Cliente_Identificador,
p.Pais AS NombrePais,
e.Estado AS NombreEstado,
ci.Ciudad AS NombreCiudad,
r.Region AS NombreRegion,
s.Segmento AS NombreSegmento,
pro.NombreProducto, ped.PedidoID, ped.Precio, ped.Cantidad, ped.Descuento, ped.Ganancia, ped.Modo
Envio, ped.OrdenFecha, ped.EnvioFecha,
sub.Subcategoria, Cat.Categoria,
pro.ProductoID
FROM Pedido ped
inner join Producto pro on ped.idProducto= pro.id
inner join Subcategoria sub on pro.idSubcategoria =sub.id
inner join Categoria cat on sub.idCategoria = cat.id
inner join Cliente c on Ped.idCliente =c.id
inner JOIN Region r ON c.idRegion = r.id
inner JOIN Ciudad ci ON r.idCiudad = ci.id
inner JOIN Estado e ON ci.idEstado = e.id
inner JOIN Pais p ON e.idPais = p.id
inner JOIN Segmento s ON c.idSegmento = s.id
```

# Diccionario de datos

## PRODUCTO

.	Columna	Tipo de dato	Descripción	Null
PK	id	int	Identificación unica del producto	
	ProductoID	varchar (30)	Código unico del producto	
	NombreProducto	VARCHAR(255)	Nombre del producto	
FK	idSubcategoría	Int	Id Subcategoría en la que se encuentra	
FK	idUsuarioCrea	int	Identificación del usuario que crea registro	
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro	
	fechaCrea	DATETIME	Fecha en la que se creo el registro	
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro	

## Cliente

.	Columna	Tipo de dato	Descripción	Null
PK	id	int	Identificación unica del cliente	
	ClienteID	varchar (30)	Código unico del cliente (nombre)	
FK	idSegmento	int	Id de Es su posición laboral en la empresa, el como trabaja	

FK	idRegión	int	Id Región de donde se encuentra
FK	idUsuarioCrea	int	Identificación del usuario que crea registro
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro
	fechaCrea	DATETIME	Fecha en la que se creo el registro
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro

Pedidos				
.	Columna	Tipo de dato	Descripción	Null
PK	id	int	Identificación unica del pedido	
	PedidoID	varchar (30)	Código unico del pedido	
	OrdenFecha	DATETIME	Fecha en el que se ordenó el pedido	
	EnvioFecha	DATETIME	Fecha en el que se envió el pedido	
	ModoEnvio	nvarchar (max)	Metodo empleado para enviar el pedido	
	Precio	nvarchar (max)	precio	
	Cantidad	Decimal(10,8)	Cantidad de pedidos	
	Descuento	Decimal(10,8)	Cantidad restada	
	Ganancia	Decimal(10,8)	Ganancias del pedido	
FK	idUsuarioCrea	int	Identificación del usuario que crea registro	
FK	idCliente	int	Id del cliente	
FK	idProducto	int	Id del cliente	

FK	idUsuarioModifica	int	identificación del usuario que Modifica registro
	fechaCrea	DATETIME	Fecha en la que se creo el registro
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro
	Estatus	bit	Activo/inactivo

	Columna	Tipo de dato	Region Descripción	Null

PK	id	int	Identificación unica del Region
	Region	VARCHAR(255)	Nombre del Region
FK	idCiudad	Int	Id ciudad en la que se encuentra
FK	idUsuarioCrea	int	Identificación del usuario que crea registro
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro
FK	fechaCrea	DATETIME	Fecha en la que se creo el registro
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro
	Estatus	bit	Activo/inactivo

	Columna	Tipo de dato	Ciudad Descripción	Null

PK	id	int	Identificación unica del Region
	Ciudad	VARCHAR(255)	Nombre del ciudad

FK	idEstado	Int	Id Estado en la que se encuentra
FK	idUsuarioCrea	int	Identificación del usuario que crea registro
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro
FK	fechaCrea	DATETIME	Fecha en la que se creo el registro
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro
	Estatus	bit	Activo/inactivo

	Columna	Tipo de dato	Estado Descripción	Null
.				

PK	id	int	Identificación unica del Region
	Estado	VARCHAR(255)	Nombre del Estado
FK	idPais	Int	Id Pais en la que se encuentra
FK	idUsuarioCrea	int	Identificación del usuario que crea registro
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro
FK	fechaCrea	DATETIME	Fecha en la que se creo el registro
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro
	Estatus	bit	Activo/inactivo

	Columna	Tipo de dato	Pais Descripción	Null
.				

PK	id	int	Identificación unica del Region
	Pais	VARCHAR(255)	Nombre del Pais
FK	idUsuarioCrea	int	Identificación del usuario que crea registro
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro
FK	fechaCrea	DATETIME	Fecha en la que se creo el registro
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro
	Estatus	bit	Activo/inactivo

Segmento			
.	Columna	Tipo de dato	Descripción
PK	id	int	Identificación unica del Region
	Segmento	VARCHAR(255)	Nombre del Segmento
FK	idUsuarioCrea	int	Identificación del usuario que crea registro
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro
FK	fechaCrea	DATETIME	Fecha en la que se creo el registro
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro
	Estatus	bit	Activo/inactivo

Categoria				
.	Columna	Tipo de dato	Descripción	Null
PK	id	int	Identificación unica del Region	
	<b>Categoria</b>	VARCHAR(255)	Nombre del <b>Categoria</b>	
FK	idUsuarioCrea	int	Identificación del usuario que crea registro	
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro	
	fechaCrea	DATETIME	Fecha en la que se creo el registro	
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro	
	Estatus	bit	Activo/inactivo	

Subcategoria				
.	Columna	Tipo de dato	Descripción	Null
PK	id	int	Identificación unica del Region	
	<b>Subcategoria</b>	VARCHAR(255)	Nombre del <b>Subcategoria</b>	
FK	idCategoria	Int	Id Categoriaen la que se encuentra	
FK	idUsuarioCrea	int	Identificación del usuario que crea registro	
FK	idUsuarioModifica	int	identificación del usuario que Modifica registro	
	fechaCrea	DATETIME	Fecha en la que se creo el registro	
	fechaModifica	DATETIME	Fecha en la que se Modificó el registro	
	Estatus	bit	Activo/inactivo	

