



Proyecto Final - BASE DE Datos

Base de Datos (Universidad Tecnológica del Perú)

UNIVERSIDAD TECNOLÓGICA DEL PERÚ



FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS E INFORMÁTICA

PROYECTO FINAL

CREACIÓN DE UNA BASE DE DATOS PARA UNA TIENDA

ASIGNATURA:

BASE DE DATOS

DOCENTE:

ING. MARIANA CHAVARRY CHANCAY

INTEGRANTES

- CURO BERNILLA DANIEL AGAPITO CÓDIGO: U20226361
- SALAZAR SOPLAPUCO FRANCO ALEXIS CÓDIGO: U20236813
- SANCHEZ MEDINA MARK STEPHANO CÓDIGO: U21214443
- TEQUEN OBLITAS KEVIN ENRIQUE CÓDIGO: U20225217

CHICLAYO, 18 DE JULIO 2022

Tabla De Contenido

Introducción	1
1. La Organización.....	3
1.1. Antecedentes Generales	3
1.2. Misión.....	3
1.3. Visión	3
1.4. Objetivos Empresariales.....	3
1.5. Valores	4
1.6. Organigrama De La Organización	5
1.7. Principales Procesos De La Organización	5
2. Aspectos De La Problemática	7
2.1. Identificación Del Problema.....	7
2.2. Alcance Propuesto.....	7
2.3. Propuesta de Solución	7
3. Diseño De La Base De Datos.....	9
3.1. Caso Del Modelamiento.....	9
3.2. Diagrama Entidad Relación - Conceptual.....	10
3.3. Diagrama Entidad Relación - Lógico.....	11
3.4. Diagrama Entidad Relación - Físico	12
3.5. Diccionario de la base de datos.....	13
4. Importación Del Diseño En SQL Server	16
4.1. Creación de la base de datos	16
4.2. Creación de las tablas.....	16
4.3. Diagrama De Base De Datos.....	23
5. Registro De Datos En Las Tablas	26
6. Consultas De Tablas	32
6.1. Consultas Básicas.....	32
6.2. Consultas De Múltiples Tablas	37
6.3. Subconsultas.....	40
7. Funciones	44
7.1. Funciones De Agregación	44
7.2. Funciones Escalares	45

8.	Procedimientos Almacenados	49
8.1.	Procedimiento Almacenado Para Registrar.....	49
8.2.	Procedimiento Almacenado Para Eliminar	51
8.3.	Procedimiento Almacenado Para Buscar	52
8.4.	Procedimiento Almacenado Para Actualizar.....	53
8.5.	Procedimiento Almacenado para listar productos.....	54
8.6.	Procedimiento Almacenado Para Listar Cargos.....	55
8.7.	Procedimiento Almacenado Para Listar Ventas.....	55
8.8.	Procedimiento Almacenado Cantidad De Productos Por Cliente	56
9.	Vistas.....	58
9.1.	Vistas Creadas	58
9.1.1.	Vista – Mostrar Venta	58
9.1.2.	Vista - Productos Mayores.....	59
9.1.3.	Vista – Mostrar Empleado Cargo	60
9.1.4.	Vista - Producto Proveedor.....	60
9.2.	Vistas Encriptadas.....	61

Introducción

El siguiente tema de investigación, nos basaremos en la creación de una base de datos para una tienda o bodega, y para ello, analizaremos el procedimiento para lograr cumplir el objetivo que se quiere llevar a cabo. Tenemos en cuenta que hoy en día los negocios se esfuerzan por tener una mayor participación en el mercado, y esto a generado a que muchos de estos implementen un control de registros para poder llevar el crecimiento de su negocio y saber a su vez las preferencias de los consumidores en general. Estos controles proporcionan un mejor manejo administrativo que permite tener un seguimiento preciso de todas las transacciones que se realicen dentro de un negocio en tiempo real, proporcionando reportes detallados de ventas que les permitirá a los administradores y dueños tener un orden sobre la cantidad de productos en el momento adecuado y así mejorar el servicio de atención al cliente, reduciendo el tiempo necesario para terminar una transacción.

A grandes rasgos podemos conocer el desarrollo de la base de datos para las ventas, con el fin de que estos pueden ser personalizados para poder cumplir con las necesidades específicas de que todo negocio requiere, por ejemplo, las organizaciones de venta al menudeo pueden localizar rápidamente los precios de venta y costos actuales de todos los productos, una tienda de ropa puede incluir todos los elementos que ofrece y organizarlos ya sea por marca o por precio, etc.

Por lo tanto, el objetivo general que nos proponemos es poder desarrollar un control que nos permita como pequeña empresa poder contar fácilmente con los productos de manera inmediata y satisfacer al cliente de una manera confortable con productos de buena calidad.

Finalmente, con el desarrollo de esta implementación, se quiere optimizar el control de las ventas y a su vez generar seguridad, confianza, calidad y garantía por parte de todos nuestros colaboradores y productos en general.

CAPÍTULO 1: LA ORGANIZACIÓN

1. La Organización

1.1. Antecedentes Generales

La empresa en la cual estamos interesados es una tienda fundada hace 17 años en la Ciudad de Ferreñafe, Departamento de Lambayeque, con el nombre tienda “Señor Del Gran Poder”, la cual se dedica básicamente a la venta de abarrotes en general, con el transcurrir de los años la tienda a implementando nuevos tipos de ventas cómo muebles, higiene personal y cuidados para brindarle calidad y garantía en sus productos que sean vendidos. Actualmente ha tenido una buena acogida por parte de la población y por el buen servicio que brinda.

1.2. Misión

Buscamos ser la tienda líder y primera opción de nuestros clientes al momento de adquirir nuestros productos referentes al hogar y a la salud; permitiéndonos crear relaciones duraderas basadas en la confianza, el respeto y la honestidad que nos permita crecer conjuntamente todos los días.

1.3. Visión

Somos una empresa encargada de brindar y ofrecer productos de excelente calidad; a través de un buen servicio, el mejor precio y un trato amable por parte de quienes conformamos esta empresa.

1.4. Objetivos Empresariales

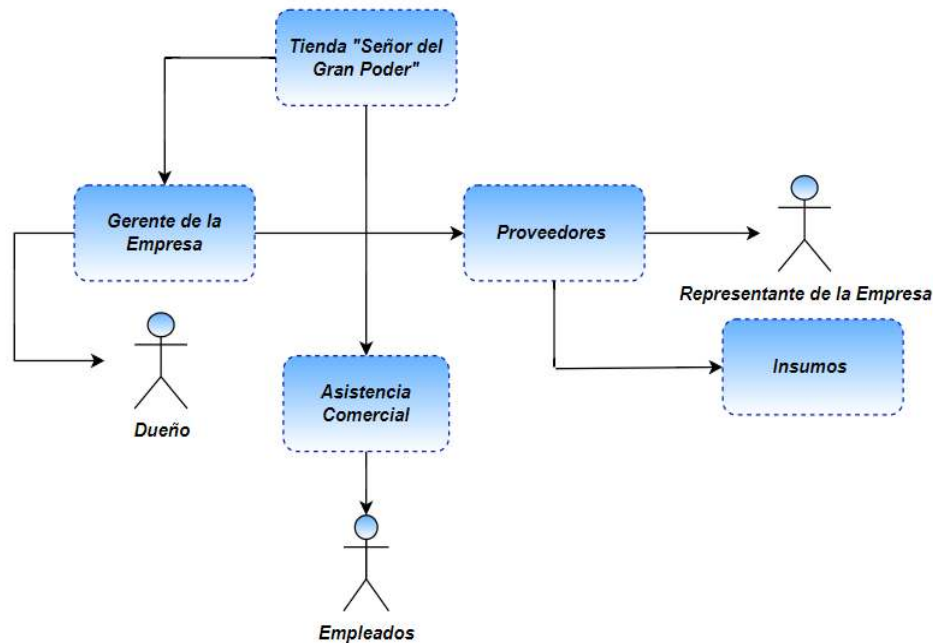
- Obtener más ingresos económicos en las ventas, realizando propuesta de ofertas que permitan a los clientes satisfacer sus necesidades y demostrar la calidad de productos innovadores para su uso.

- Implementar nuevos productos de calidad que cumplan todos los estándares indicados para el buen cuidado del hogar y la salud que todos nuestros clientes necesitan para satisfacer sus necesidades.
- Ofrecer una atención de alta calidad, manteniendo un orden para mayor comodidad y acogida al cliente.
- Generar satisfacción y participación de los empleados, buscando que los trabajadores y consumidores creen un ambiente de confianza y crezcas la productiva en la empresa.
- Tener un control de calidad en los inmuebles, para que los productos circulen de manera eficiente dentro de la empresa y cuando se le entregue al cliente llegue en condiciones óptimas.

1.5.Valores

- **Compromiso:** Tener una firme implicación de la organización con todos nuestros grupos de interés (Clientes, Personas, Aliados y Sociedad).
- **Calidad:** Nuestros productos cumplen con todos los parámetros establecidos y prometidos, que sólo pueden ser cercanos a la perfección.
- **Honestidad:** Actuamos con justicia, trabajando de forma integra con transparencia y justicia.

1.6. Organigrama De La Organización



1.7. Principales Procesos De La Organización

- **Proceso De Las Compras De Insumos**

Primero el dueño de la bodega contacta al proveedor y le emite una orden con los productos que solicita para abastecer su negocio.

- **Proceso De Ventas:** El empleado es el encargado de atender al cliente, con el fin de poder atender sus necesidades con los productos que va a seleccionar, luego se realiza un detalle de todos los productos que va a llevar y realiza el registro de los artículos que le hizo al cliente y finalmente le entrega su boleta de venta.

- **Proceso Administrativo:**

El dueño realiza un informe de todas las ventas realizadas semanales por cada empleado, luego realiza un control de los impuestos que tiene que pagar por su funcionamiento y finalmente saca el presupuesto del pago hacia a los empleados.

CAPÍTULO 2: ASPECTOS DE LA PROBLEMÁTICA

2. Aspectos De La Problemática

2.1. Identificación Del Problema

El problema que presenta la tienda “SEÑOR DEL GRAN PODER” es el control de las ventas. La tienda se dedica a la venta de abarrotes en general y muebles, por lo que, no cuenta con un registro de las ventas realizadas día a día y no presenta un informe detallado adecuado.

2.2. Alcance Propuesto

Esta base de datos estará al alcance del gerente de la empresa que le va a permitir observar cómo van las ventas realizadas al día y determinar si hay alta o baja ganancia.

2.3. Propuesta de Solución

Implementar una base de datos que le permita obtener el control de sus ventas realizadas de los productos en general.

.

.

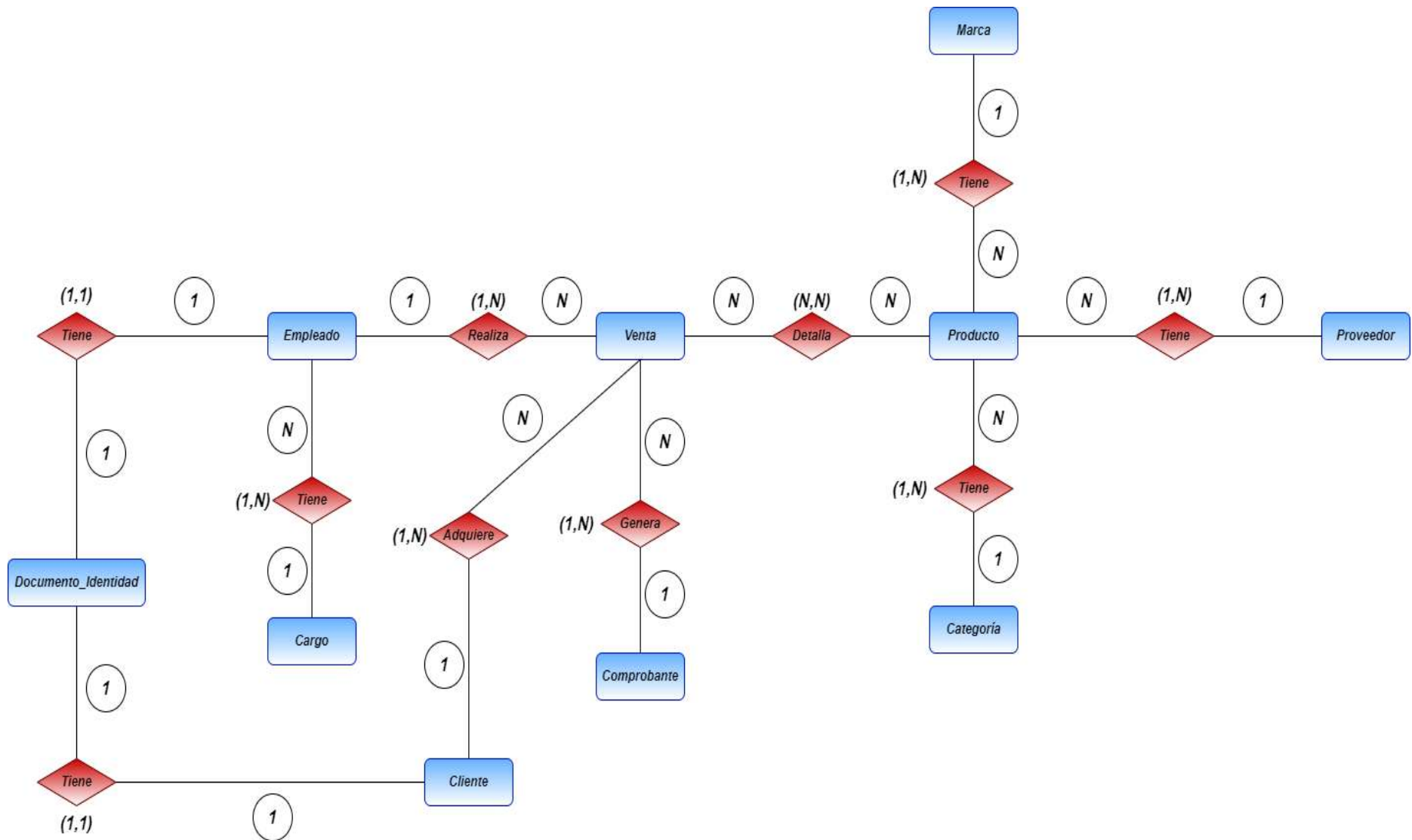
CAPÍTULO 3: DISEÑO DE LA BASE DE DATOS

3. Diseño De La Base De Datos

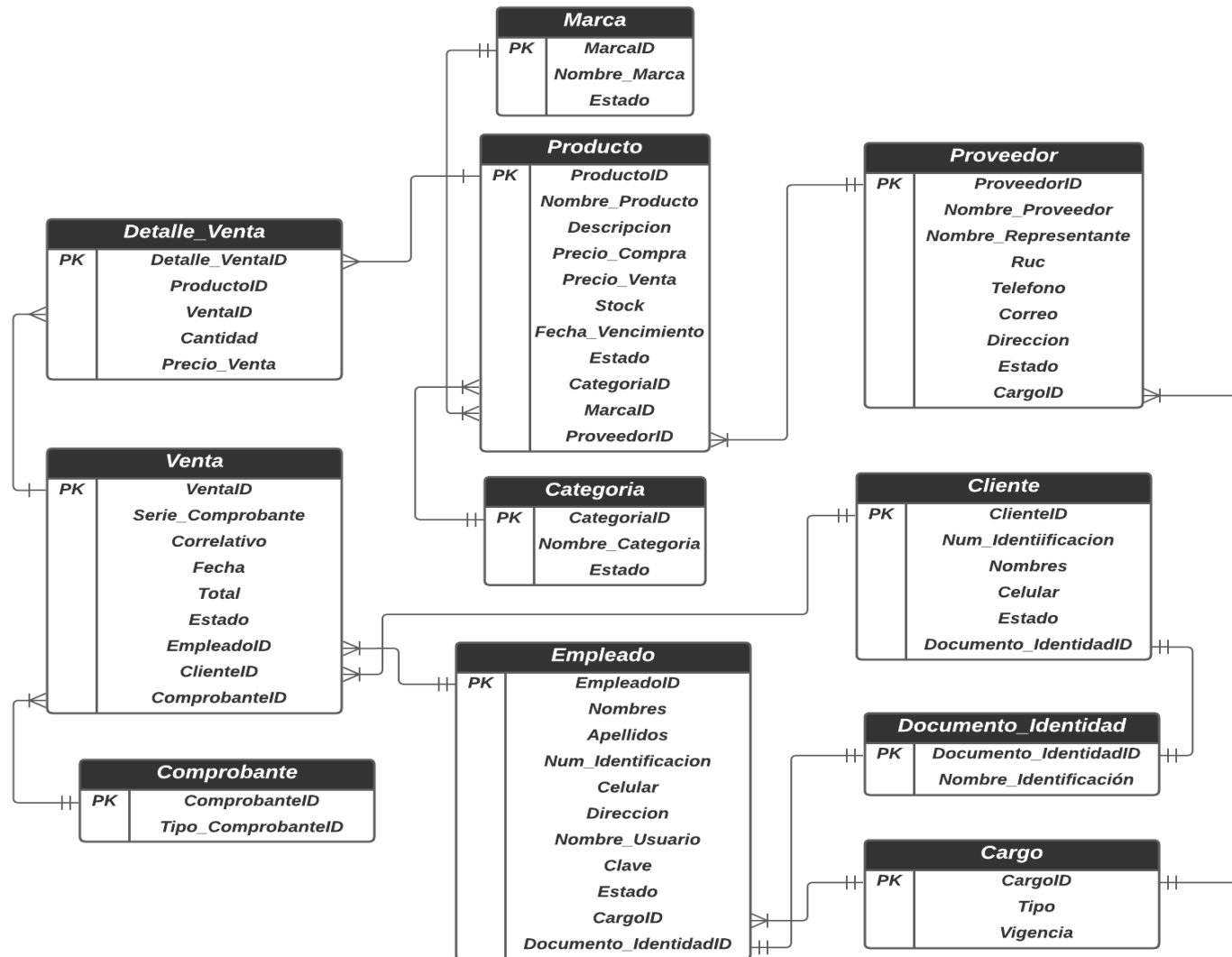
3.1.Caso Del Modelamiento

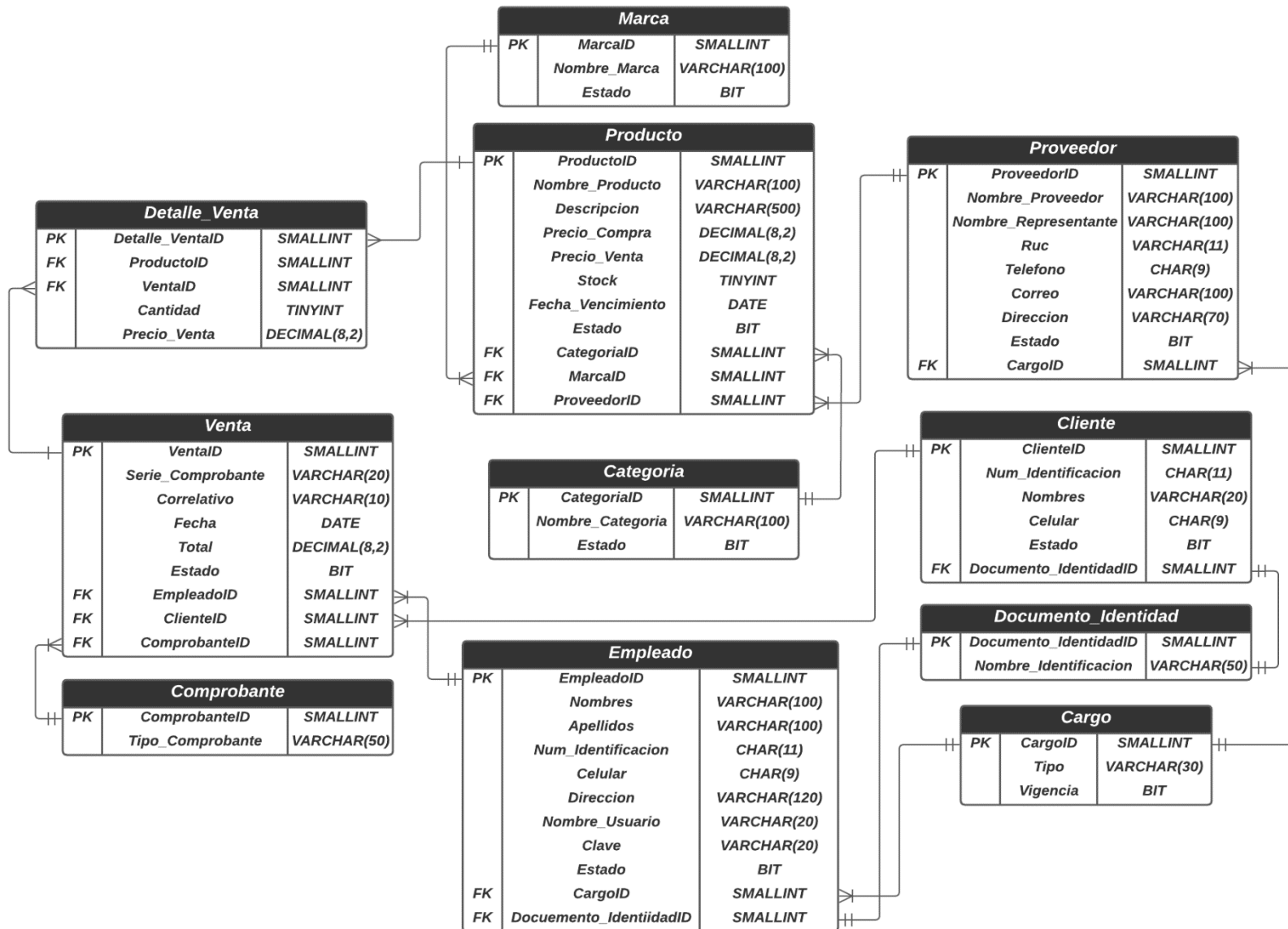
El siguiente proyecto por realizar, necesita la implementación de una base de datos para la tienda Señor del Gran Poder, la cual contará con las características requeridas para lograr su funcionamiento de manera correcta. El gerente de la empresa necesita tener un control de su personal de trabajo, los proveedores que ofrecerán los productos y registro de todas las ventas realizadas. Se sabe que cada producto contará con una marca determinada que permita registrar (Nombre de la Marca y Estado). A su vez está será ubicada en su categoría correspondiente, considerando (Nombre de la categoría y estado), por lo que 1 categoría tiene 1 o N productos y 1 o N productos tienen 1 categoría, teniendo en cuenta que del producto se debe considerar (Nombre Producto, Descripción, Precio de Compra, el precio de venta, Stock, Fecha de Vencimiento y Estado). 1 Proveedor tiene 1 o N productos y 1 o N productos tienen un proveedor. El proveedor contará con (Nombre Proveedor, Nombre Representante, Ruc, Telefono, Correo, Dirección, Estado). La compra establecida de un producto va a generar ventas tomando en cuenta (Serie Comprobante, Correlativo, Fecha, Total, Estado), por lo que 1 o N ventas detallará 1 o N productos y 1 o N productos estará detalladas en 1 o N ventas, con el fin de obtener un detalle de la cantidad de productos y el precio de venta total a pagar. Sobre su personal de trabajo se tomará como puntos principales (Nombres, Apellidos, Numero de Identificación, Celular, Nombre de Usuario, Clave, Estado) y se le asignará un cargo establecido (Tipo y Vigencia). Sobre su personal el gerente o Administrador tomará en cuenta que todos porten Documento de Identidad (DNI) y los registrará en la base de datos. Por último, 1 venta será adquirida 1 o N clientes registrando (Número Identificación, Nombres, Celular, Estado) y 1 cliente adquiere 1 o N ventas. La venta solo puede generar un único tipo de comprobante teniendo en cuenta (Tipo Comprobante).

3.2. Diagrama Entidad Relación - Conceptual



3.3. Diagrama Entidad Relación - Lógico





3.5.Diccionario de la base de datos

El diccionario de datos nos proporciona una descripción detallada de todas las tablas de la base de datos que se van a crear, de este modo el diccionario de datos contiene al menos todos los nombres y atributos y características para cada tabla.

Entidad	Atributos	Tipo de dato	Dimensión	Restricciones	Descripción
Categoria	CategoriaID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación de la Categoría
	Nombre_Categoria	VARCHAR	100		Nombre de la Categoría
	Estado	BIT			Categoría (1 - Activa o Inactiva - 0)
Marca	MarcaID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación de la Marca
	Nombre_Marca	VARCHAR	100		Nombre de la Marca
	Estado	BIT			Marca (1 - Activa o Inactiva - 0)
Cargo	CargoID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Cargo
	Tipo	VARCHAR	30		Nombre del Cargo
	Vigencia	BIT			Vigencia Cargo (1 - Activa o Inactiva - 0)
Proveedor	ProveedorID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Proveedor
	Nombre_Proveedor	VARCHAR	100		Nombre de la Empresa
	Nombre_Representante	VARCHAR	100		Nombre de la persona encargada de Reparto
	Ruc	VARCHAR	11		Registro Único de Contribuyentes de la Empresa
	Telefono	CHAR	9		Contacto de la Empresa
	Correo	VARCHAR	100		Contacto de la Empresa
	Direccion	VARCHAR	70		Dirección del lugar de la Empresa
	Estado	BIT			Cargo Existente (1 - Activa o Inactiva - 0)
	CargoID	SMALLINT		FOREIGN KEY	Referencia de la tabla Cargo
Producto	ProductoID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Producto
	Nombre_Producto	VARCHAR	100		Nombre del Producto
	Descripcion	VARCHAR	500		Descripción breve del Producto
	Precio_Compra	DECIMAL	(8,2)		Precio de compra externo
	Precio_Venta	DECIMAL	(8,2)		Precio de compra interno
	Stock	TINYINT			Stock de los Productos
	Fecha_Vencimiento	DATE			Fecha de caducidad de los Productos
	Estado	BIT			Producto en tienda (1 - Activo o Inactivo - 0)
	CategoriaID	SMALLINT		FOREIGN KEY	Referencia de la tabla Categoría
	MarcaID	SMALLINT		FOREIGN KEY	Referencia de la tabla Marca
Documento_Identidad	ProveedorID	SMALLINT		FOREIGN KEY	Referencia de la tabla Proveedor
	Documento_IdentidadID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Documento
	Nombre_Identificacion	VARCHAR	50		Tipo (DNI)
Empleado	EmpleadoID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Empleado
	Nombres	VARCHAR	100		Nombre del Trabajador
	Apellidos	VARCHAR	100		Apellidos del Trabajador
	Num_Identificacion	CHAR	11	UNIQUE	Número de documento de identificación
	Celular	CHAR	9		Número de contacto
	Direccion	VARCHAR	120		Dirección o lugar de pertenencia
	Nombre_Usuario	VARCHAR	20		Nombre de usuario de sistema
	Clave	VARCHAR	20		Dígito de 04 números
	Estado	BIT			Empleado (1 - Activo o Suspendido - 0)
	CargoID	SMALLINT		FOREIGN KEY	Referencia de la tabla Cargo
Comprobante	Documento_IdentidadID	SMALLINT		FOREIGN KEY	Referencia de tabla Documento de Identidad
	ComprobanteID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Comprobante
	Tipo_Comprobante	VARCHAR	50		Tipo - Boleta
Cliente	ClienteID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Cliente correspondiente
	Num_Identificacion	CHAR	11	UNIQUE	Número de documento de identificación
	Nombres	VARCHAR	20		Nombre del Cliente
	Celular	CHAR	9		Número de contacto
	Estado	BIT			Cliente frecuente (1 - Activo o Inactivo - 0)
	Documento_IdentidadID	SMALLINT		FOREIGN KEY	Referencia de la tabla Documento de Identidad

Venta	VentaID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación de la Venta
	Serie_Combprobante	VARCHAR	20		Serie del comprobante - B0001
	Correlativo	VARCHAR	10	UNIQUE	Correlativo Autoincrementable
	Fecha	DATE			Fecha de la venta
	Total	DECIMAL	(8.2)		Total de la compra establecida
	Estado	BIT			Estado (1 - No Amulado o Amulado - 0)
	EmpleadoID	SMALLINT		FOREIGN KEY	Referencia de la tabla Empleado
	ClienteID	SMALLINT		FOREIGN KEY	Referencia de la tabla Cliente
Detalle_Venta	ComprobanteID	SMALLINT		FOREIGN KEY	Referencia de la tabla Comprobante
	Detalle_VentaID	SMALLINT	Auto-Incrementable	PRIMARY KEY	Identificación del Detalle de Ventas
	VentaID	SMALLINT		FOREIGN KEY	Referencia de la tabla venta
	ProductoID	SMALLINT		FOREIGN KEY	Referencia de la tabla Producto
	Cantidad	TINYINT			Cantidad de Productos
	Precio_Venta	DECIMAL	(8.2)		Precio de la venta del Producto

CAPÍTULO 4: IMPORTACIÓN DEL DISEÑO EN SQL SERVER

4. Importación Del Diseño En SQL Server

4.1.Creación de la base de datos

Hacemos uso de la tabla maestra y verificamos si existe esa base datos. Si SQL detecta que existe, procede a eliminar y crear nuevamente la base de datos. Luego hace uso ya de la base de datos creada.

```
USE MASTER
GO

IF(DB_ID('Tienda_Version') IS NOT NULL)
DROP DATABASE Tienda_Version
CREATE DATABASE Tienda_Version
GO

USE Tienda_Version
GO
```

4.2.Creación de las tablas

- Creamos la tabla Categoría y le colocamos sus respectivos columnas o atributos.

```
CREATE TABLE Categoria
(
    CategoriaID          SMALLINT IDENTITY    NOT NULL,
    Nombre_Categoria    VARCHAR(100)          NOT NULL,
    Estado               BIT                   NOT NULL
)
GO
```

- Asignamos la llave primaria en la columna o atributo CategoriaID de la tabla Categoría.

```
ALTER TABLE Categoria
    ADD CONSTRAINT PK_Categoria PRIMARY KEY(CategoriaID)
GO
```

- Creamos la tabla Marca con sus respectivas columnas.

```
CREATE TABLE Marca
(
  MarcaID          SMALLINT IDENTITY          NOT NULL,
  Nombre_Marca     VARCHAR(100)              NOT NULL,
  Estado           BIT                        NOT NULL
)
GO
```

- Asignamos la llave primaria en la columna o atributo MarcaID de la tabla Marca.

```
ALTER TABLE MARCA
  ADD CONSTRAINT PK_MARCA PRIMARY KEY(MARCAID)
GO
```

- Creamos la tabla Cargo con sus respectivos atributos.

```
CREATE TABLE Cargo
(
  CargoID          SMALLINT IDENTITY          NOT NULL,
  Tipo             VARCHAR(30)                NOT NULL,
  Vigencia         BIT                        NULL
)
GO
```

- Asignamos la llave primaria en la columna o atributo CargoID de la tabla Cargo.

```
ALTER TABLE Cargo
  ADD CONSTRAINT PK_Cargo PRIMARY KEY(CargoID)
GO
```

- Creamos la tabla Proveedor con sus respectivos atributos.

```
CREATE TABLE Proveedor
(
ProveedorID          SMALLINT IDENTITY  NOT NULL,
Nombre_Proveedor     VARCHAR(100)       NOT NULL,
Nombre_Representante VARCHAR(100)       NULL,
Ruc                  VARCHAR(11)        NOT NULL,
Telefono             char(9)            null,
correo               VARCHAR(100)       NULL,
Direccion             VARCHAR(70)       NOT NULL,
Estado               BIT                NOT NULL,
CargoId              SMALLINT
)
GO
```

- Asignamos la llave primaria en la columna o atributo ProveedorID y Referenciamos la columna CargoID como llave Foránea en la tabla Proveedor.

```
ALTER TABLE Proveedor
ADD CONSTRAINT PK_Proveedor PRIMARY KEY(ProveedorID)
GO
```

```
ALTER TABLE PROVEEDOR
ADD CONSTRAINT FK_Cargo FOREIGN KEY (cargoId) REFERENCES cargo
GO
```

- Creamos la tabla Producto con sus respectivos atributos.

```
CREATE TABLE Producto
(
ProductoID          SMALLINT IDENTITY  NOT NULL,
Nombre_Producto     VARCHAR(100)       NOT NULL,
Descripcion          VARCHAR(500)      NOT NULL,
Precio_Compra        DECIMAL(8,2)      NOT NULL,
Precio_Venta         DECIMAL(8,2)      NOT NULL,
Stock               TINYINT            NOT NULL,
Fecha_Vencimiento    DATE              NULL,
Estado               BIT                NOT NULL,
CategoriaID          SMALLINT          ,
MarcaID              SMALLINT          ,
ProveedorID          SMALLINT
)
GO
```

- Asignamos la llave primaria en la columna o atributo ProductoID y referenciamos la columna CategoriaID, MarcaID y ProveedorID como llaves Foránea en la tabla Producto.

```
ALTER TABLE Producto
    ADD CONSTRAINT PK_Producto PRIMARY KEY(ProductoID)
GO

ALTER TABLE Producto
    ADD CONSTRAINT FK_Producto_Categoria FOREIGN KEY(CategoriaID) REFERENCES Categoria
GO

ALTER TABLE Producto
    ADD CONSTRAINT FK_Producto_Marca FOREIGN KEY(MarcaID) REFERENCES Marca
GO

ALTER TABLE Producto
    ADD CONSTRAINT FK_Producto_Proveedor FOREIGN KEY(ProveedorID) REFERENCES Proveedor
GO
```

- Creamos la tabla Documento de Identidad con su atributo.

```
CREATE TABLE Documento_Identidad
(
    Documento_IdentidadID    SMALLINT IDENTITY    NOT NULL,
    Nombre_Identificacion    VARCHAR(50)          NOT NULL,
)
GO
```

- Asignamos la llave primaria en la columna o atributo Documento_IdentidadID de la tabla Documento de Identidad.

```
ALTER TABLE Documento_Identidad
    ADD CONSTRAINT PK_Documento_Identidad PRIMARY KEY(Documento_IdentidadID)
GO
```

- Creamos la tabla Empleado con sus respectivos atributos.

```
CREATE TABLE Empleado
(
EmpleadoID          SMALLINT IDENTITY          NOT NULL,
Nombres              VARCHAR(100)              NOT NULL,
Apellidos            VARCHAR(100)              NOT NULL,
Num_Identificacion   CHAR(11)                  NOT NULL,
Celular              CHAR(9)                    NOT NULL,
Direccion            VARCHAR(120)              NULL,
Nombre_Usuario       VARCHAR(20)               NOT NULL,
Clave                VARCHAR(20)               NOT NULL,
Estado              BIT                        NULL,
CargoID              SMALLINT                   ,
Documento_IdentidadID SMALLINT
)
GO
```

- Asignamos la llave primaria en la columna o atributo EmpleadoID y referenciamos la columna CargoID y Documento_IdentidadID como llaves Foráneas en la tabla Empleado.

```
ALTER TABLE Empleado
ADD CONSTRAINT PK_Empleado PRIMARY KEY(EmpleadoID)
GO
```

```
ALTER TABLE Empleado
ADD CONSTRAINT FK_Empleado_Cargo FOREIGN KEY(CargoID) REFERENCES Cargo
GO
```

```
ALTER TABLE Empleado
ADD CONSTRAINT FK_Empleado_Documento_Identidad FOREIGN KEY(Documento_IdentidadID) REFERENCES Documento_Identidad
GO
```

- Asignamos la restricción Unique a la columna Num_Identificacion para evitar que los DNI se repitan

```
ALTER TABLE Empleado
ADD CONSTRAINT UC_Empleado_Num_Identificacion UNIQUE (Num_Identificacion)
GO
```


- Creamos la tabla Comprobante con sus respectivos atributos

```
CREATE TABLE Comprobante
(
ComprobanteID          SMALLINT IDENTITY          NOT NULL,
Tipo_Comprobante      VARCHAR(50)                NOT NULL
)
GO
```

- Asignamos la llave primaria en la columna ComprobanteID en la tabla Comprobante

```
ALTER TABLE Comprobante
ADD CONSTRAINT PK_Comprobante PRIMARY KEY(ComprobanteID)
GO
```

- Creamos la tabla Cliente con sus respectivos atributos

```
CREATE TABLE Cliente
(
ClienteID              SMALLINT IDENTITY          NOT NULL,
Num_Identificacion     CHAR(11)                  NOT NULL,
Nombres               VARCHAR(20)                NOT NULL,
Celular               CHAR(9)                   NULL,
Estado               BIT                        NULL,
Documento_IdentidadID SMALLINT                    ,
)
GO
```

- Asignamos la llave primaria en la columna o atributo ClienteID, referenciamos la columna Documento_IdentidadID como llave foránea y usamos la restricción Unique en la columna Num_Identificacion para evitar que los DNI se repitan, todo en la tabla Cliente

```
ALTER TABLE Cliente
ADD CONSTRAINT PK_Cliente PRIMARY KEY(ClienteID)
GO
```

```
ALTER TABLE Cliente
ADD CONSTRAINT FK_Cliente_Documento_IdentidadID FOREIGN KEY(Documento_IdentidadID) REFERENCES Documento_Identidad
GO
```

```
ALTER TABLE Cliente
ADD CONSTRAINT UC_Cliente_Num_Identificacion UNIQUE (Num_Identificacion)
GO
```

- Creamos la tabla Venta con sus respectivos atributos.

```
CREATE TABLE Venta
(
    VentaID          SMALLINT IDENTITY    NOT NULL,
    Serie_Comprobante VARCHAR(20)         NOT NULL,
    Correlativo      VARCHAR(10)          NOT NULL,
    Fecha            DATE                  NOT NULL,
    Total            Decimal(8,2)          NOT NULL,
    Estado           BIT                   NOT NULL,
    EmpleadoID       SMALLINT              ,
    ClienteID        SMALLINT              ,
    ComprobanteID    SMALLINT              ,
)
GO
```

- Asignamos la llave primaria a la columna VentaID, usamos la restricción Unique para la columna “Correlativo” y referenciamos la columna EmpleadoID, ClienteID, ComprobanteID como llaves foráneas, en la tabla Venta.

```
ALTER TABLE Venta
    ADD CONSTRAINT PK_Venta PRIMARY KEY(VentaID)
GO
```

```
ALTER TABLE Venta
    ADD CONSTRAINT UC_Venta_Correlativo UNIQUE (Correlativo)
GO
```

```
ALTER TABLE Venta
    ADD CONSTRAINT FK_Venta_Empleado FOREIGN KEY(EmpleadoID) REFERENCES Empleado
GO
```

```
ALTER TABLE Venta
    ADD CONSTRAINT FK_Venta_Cliente FOREIGN KEY(ClienteID) REFERENCES Cliente
GO
```

```
ALTER TABLE Venta
    ADD CONSTRAINT FK_Venta_Comprobante FOREIGN KEY(ComprobanteID) REFERENCES Comprobante
GO
```

- Creamos la tabla Detalle_Venta, Formada por las tablas Producto y Venta, con sus respectivos atributos.

```
CREATE TABLE Detalle_Venta
(
Detalle_VentaID    SMALLINT IDENTITY    NOT NULL,
VentaID            SMALLINT              ,
ProductoID         SMALLINT              ,
Cantidad           TINYINT              NOT NULL,
Precio_Venta       DECIMAL(8,2)         NOT NULL
)
GO
```

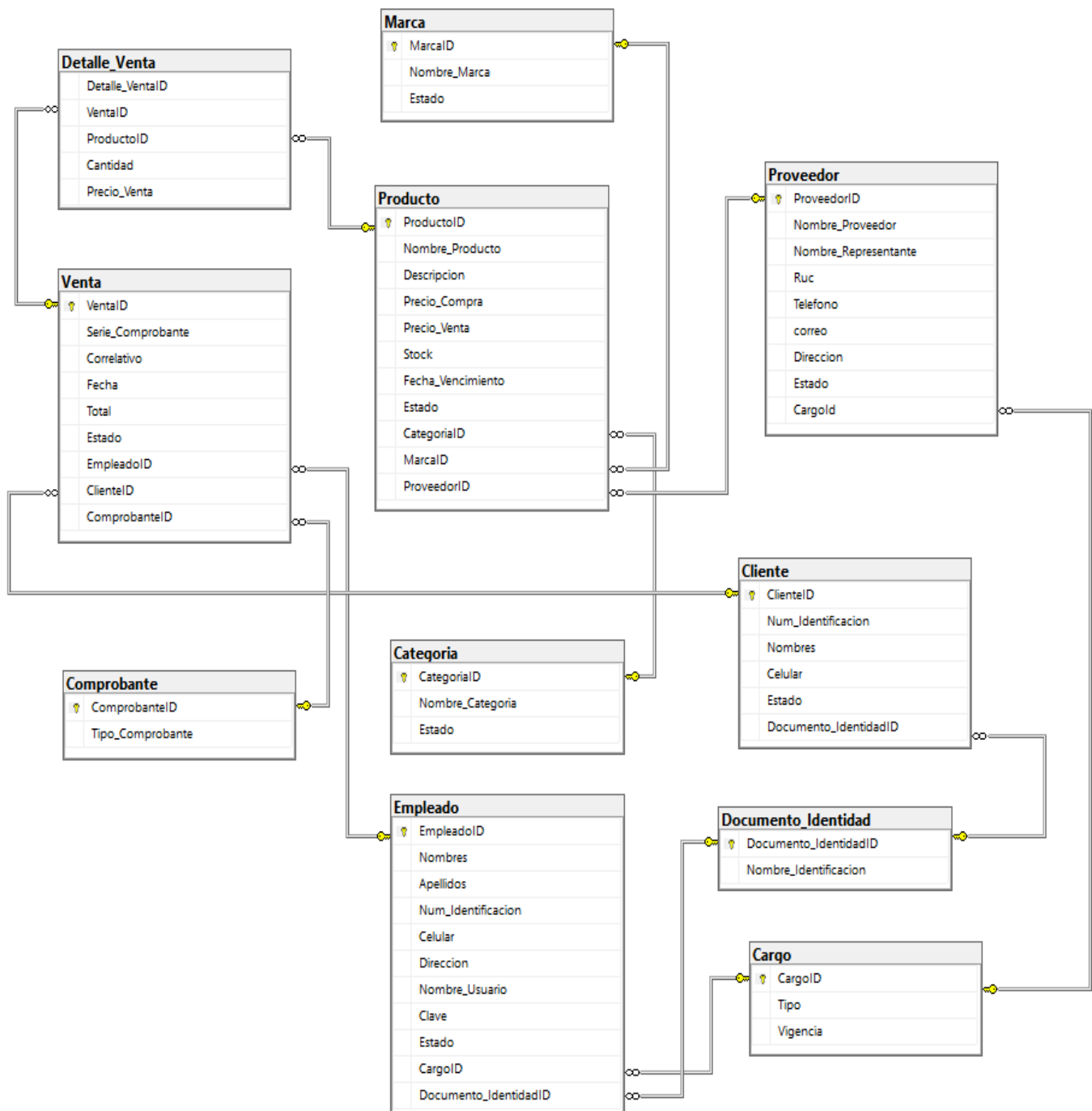
- Asignamos la llave primaria a la columna Detalle_VentaID y referenciamos las columnas

VentaID, ProductoID como llaves foráneas en la tabla Detalle Venta.

```
ALTER TABLE Detalle_Venta
    ADD CONSTRAINT FK_Detalle_Venta_Venta FOREIGN KEY(VentaID) REFERENCES Venta
GO
ALTER TABLE Detalle_Venta
    ADD CONSTRAINT FK_Detalle_Venta_Producto FOREIGN KEY(ProductoID) REFERENCES Producto
GO
```

4.3.Diagrama De Base De Datos

Procedemos a realizar el diagrama de la base de datos en SQL Server



CAPÍTULO 5:

REGISTRO DE DATOS EN LAS TABLAS

5. Registro De Datos En Las Tablas

Una vez que tenemos todas las tablas creadas con su respectivo tipo de datos, procedemos a hacer los registros respectivos por cada una de la tabla para almacenarla en la base de datos.

- Tabla Categoría

```

INSERT INTO Categoria VALUES
('ABARROTES', 1),
('DESAYUNOS', 1),
('AGUA Y BEBIDAS', 1),
('GALLETAS, SNACKS Y GOLOSINAS', 1),
('LACTEOS Y EMBUTIDOS', 1),
('HIGIENE Y SALUD', 1),
('LIMPIEZA', 1),
('COLCHONES Y MUEBLES', 1),
('MASCOTAS', 1)
GO

```

- Tabla Marca

```

INSERT INTO Marca VALUES
('DOÑA GUSTA', 1),
('SIBARITA', 1),
('MAGGI', 1),
('MARINA', 1),
('DON VICTORIO', 1),
('MOLITALIA', 1),
('POMAROLA', 1),
('ALACENA', 1),
('ALIANZA', 1),
('TARI', 1),
('CAPRI', 1),
('NICOLINI', 1),
('EL OLIVAR', 1),
('PRIMOR', 1),
('SAO', 1),
('FLORIDA', 1),
('CAMPOMAR', 1),
('ARICA', 1),

```

- Tabla Cargo

```

INSERT INTO CARGO VALUES
('ADMINISTRADOR', 1),
('SUPERVISOR', 1),
('GERENTE', 1),
('RECURSOS HUMANOS', 1),
('EMPLEADO', 1)
GO

```

- Proveedor

```

INSERT INTO Proveedor VALUES
('DISTRIBUIDOR PEPSICO SAC', 'JUAN MANUEL', '20147589632', '948515023', 'PEPSICO@GMAIL.COM', 'MIRAFLORES - CENTRO #450', 1,1),
('DISTRIBUIDOR P & G SAC', 'JOSE CARRANZA', '20147539632', '948515023', 'P&G@GMAIL.COM', 'HUANUCO - CENTRO #450', 1,1),
('DISTRIBUIDOR COCA COLA SAC', 'ALVARO JAIME', '20156342558', '946145236', 'COCACOLA@GMAIL.COM', 'SURCO - LIMA #963', 1,1),
('DISTRIBUIDOR NESTLE SAC', 'BRYAN STEVEN', '20197589632', '948515023', 'NESTLE@GMAIL.COM', 'LAMBAYEQUE - JUAN PABLO #834', 1,2),
('DISTRIBUIDOR GLORIA SAC', 'ALEXANDER FUENTES', '29156345415', '946758236', 'LACTEOS@GMAIL.COM', 'CHORILLOS', 1,2),
('DISTRIBUIDOR DEKASA SAC', 'MARK SANCHEZ', '22568953254', '946758236', 'DEKASA@GMAIL.COM', 'CALLAO - LIMA #256', 1,5),
('DISTRIBUIDOR SC JOHNSON SAC', 'JAMITO PEREZ', '22968953254', '946753236', 'SCJOHNSON@GMAIL.COM', 'CHICLAYO - LA VICTORIA #485', 1,5),
('DISTRIBUIDOR RICOCHET SAC', 'LORENZO DIAZ', '20196345415', '946758236', 'RICOCHET@GMAIL.COM', 'CHICLAYO - JLO #963', 1,1),
('DISTRIBUIDOR GRUPO MSK SAC', 'CARLOS ENRIQUE', '20156345415', '946758236', 'GRUPOMSK@GMAIL.COM', 'TRUJILLO - CENTRO #852', 1,2),
('DISTRIBUIDOR KFRAT SAC', 'JUAN CARLOS', '20157345415', '946758236', 'KRAFT@GMAIL.COM', 'LA MOLINA - CENTRO #858', 1,1)
GO

```

- Producto

```

INSERT INTO Producto VALUES
('CALDO DE GALLINA', 'CAJITA DE 6 UND', 0.99, 1.59, 20, '31/05/2024', 1, 1, 1, 1),
('PANQUITA', 'DOYPACK 250gr', 5.99, 7.70, 30, '15/02/2023', 1, 1, 2, 4),
('SAZONADOR DE ARROZ', 'PACK DE 6 UND', 1.00, 1.59, 20, '10/04/2022', 0, 1, 3, 4),
('PIMIENTA NEGRA MOLIDA', 'SOBRE DE 6 UND', 3.00, 3.89, 50, '10/10/2023', 1, 1, 2, 6),
('CANELA ENTERA', 'SOBRE 10gr', 3.00, 3.29, 40, '20/12/2023', 1, 1, 2, 4),
('AJINOMOTO', 'SOBRE 250gr', 5.00, 5.29, 70, '17/01/2024', 1, 1, 2, 4),
('ORÉGANO MOLIDA', 'PACK 6 UND x 3.5gr', 1.50, 1.99, 40, '20/12/2023', 1, 1, 2, 4),
('COMINO MOLIDO', 'SOBRE DE 6ND', 3.00, 3.89, 20, '18/09/2023', 1, 1, 2, 1),
('CALDO CONCENTRADA DE TOCINO', 'CAJITA 42gr', 1.00, 1.59, 60, '20/07/2023', 1, 1, 3, 4),
('HOJAS DE LAUREL', 'BOLSA 5gr', 1.80, 2.20, 10, '14/03/2022', 0, 1, 2, 4),
('HONGO Y LAUREL', 'SOBRE 5gr', 2.00, 2.49, 10, '23/03/2022', 0, 1, 2, 4),
('SAL DE MESA MARINA', 'BOLSA 1KG', 1.50, 2.20, 50, '14/05/2022', 0, 1, 4, 4),
('SAL ROSADA DE MARAS', 'DOYPACK 500g', 9.20, 10.39, 25, '18/07/2023', 1, 1, 4, 4),
('SALSA DE TOMATE', 'SOBRE 145gr', 1.40, 1.65, 10, '02/03/2022', 1, 1, 5, 4),
('SALSA DE TOMATE NICOLINI', 'SOBRE 145gr', 1.80, 2.20, 10, '14/03/2022', 1, 1, 12, 4),
('SALSA DE TOMATE SECRETA', 'SOBRE 175gr', 2.80, 3.20, 10, '14/03/2022', 1, 1, 7, 4),
('VINAGRETA POLLERA', 'SOBRE 85gr', 1.99, 3.00, 16, '18/05/2023', 1, 1, 8, 4),
('VINAGRETA MOSTAZA', 'SOBRE 85gr', 1.99, 2.50, 18, '16/10/2022', 1, 1, 8, 4),
('CREMA DE AJÍ', 'SOBRE 95gr', 3.30, 4.00, 50, '20/05/2024', 0, 1, 10, 4),
('FIDEOS TALLARINES', 'PAQUETE', 2.00, 3.70, 80, '04/10/2021', 1, 1, 9, 4),
('FIDEO CODITOS', 'PAQUETE', 2.00, 2.70, 80, '11/05/2021', 1, 1, 6, 4),
('SEMOLA', 'PAQUETE', 1.00, 1.70, 80, '10/02/2022', 1, 1, 6, 4),
('AVENA', 'PAQUETE', 2.50, 3.00, 80, '06/07/2023', 1, 2, 9, 1),
('MAYONESA A LA CENA', 'SOBRE', 4.50, 5.00, 40, '18/10/2023', 1, 1, 8, 4),
('FIDEOS SPAGHETTI', 'PAQUETE', 4.50, 5.00, 40, '18/10/2023', 1, 1, 6, 4),
('FIDEO TALLARIN', 'PAQUETE', 4.50, 5.00, 40, '18/10/2021', 1, 1, 6, 4),
('FIDEO CANUTO', 'PAQUETE', 0.80, 1.30, 20, '05/07/2023', 1, 1, 5, 4),
('FIDEO TALLARIN', 'PAQUETE', 1.50, 2.00, 30, '10/07/2024', 1, 1, 9, 4),

```

- Documento de Identidad


```

INSERT INTO Documento_Identidad VALUES
('DNI')
GO

```

- Empleado

```

INSERT INTO Empleado VALUES
('DANIEL AGAPITO', 'CURO BERNILLA', '14526369', '973408196', 'JOSE LEONARDO ORTIZ', 'DANIEL CURO', 1234, 1, 3, 1),
('FRANCO ALEXIS', 'SALAZAR SOPLAPUCO', '15857496', '934002550', 'LAS BRISAS', 'FRANCO ALEXIS', 1562, 1, 3, 1),
('KEVIN ENRIQUE', 'TEQUEN OBLITAS', '75968574', '939950378', 'LA VICTORIA', 'KEVIN ENRIQUE', 1478, 1, 3, 1),
('MARK STEPHANO', 'SANCHEZ MEDINA', '72271235', '946145406', 'FERREÑAFE', 'MARK STEPHANO', 2015, 1, 1, 1),
('JUAN JOSE', 'MEDINA SANCHEZ', '72871245', '946145496', 'FERREÑAFE', 'JUAN JOSE', 1488, 1, 3, 1)
GO

```

- Comprobante

```

INSERT INTO Comprobante VALUES
('BOLETA')
GO

```

- Cliente

```

INSERT INTO CLIENTE VALUES
('10421356273', 'SEBASTIAN', '946253124', 1, 2),
('72145689', 'CARLOS', '951145236', 1, 1),
('10401256233', 'PRISCILLA', '956325688', 0, 2),
('40129623', 'CLAUDIA', '956325488', 0, 1),
('10441256233', 'MELISSA', '959325688', 1, 2),
('10401246233', 'JUANITO', '956325688', 1, 2),
('46125623', 'TEO', '956225688', 0, 1)
GO

```


- Venta

```

INSERT INTO Venta VALUES ('B001', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '15/04/2022', 9.29, 1, 1, 5, 1)
INSERT INTO Venta VALUES ('B002', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '14/04/2022', 7.40, 1, 1, 2, 1)
GO
INSERT INTO Venta VALUES ('B003', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '14/04/2022', 220.00, 1, 4, 2, 1)
GO
INSERT INTO Venta VALUES ('B004', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '25/04/2022', 1.20, 1, 2, 3, 1)
GO
INSERT INTO Venta VALUES ('B005', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '25/04/2022', 6.00, 1, 4, 3, 1)
GO
INSERT INTO Venta VALUES ('B006', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '15/05/2022', 20.00, 1, 4, 4, 1)
GO
INSERT INTO Venta VALUES ('B007', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '16/05/2022', 1.20, 1, 3, 5, 1)
GO
INSERT INTO Venta VALUES ('B008', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '16/05/2022', 3.50, 1, 3, 6, 1)
GO
INSERT INTO Venta VALUES ('B009', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '17/05/2022', 42.00, 1, 5, 7, 1)
GO
INSERT INTO Venta VALUES ('B0010', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '18/07/2022', 8.50, 1, 3, 7, 1)
GO
INSERT INTO Venta VALUES ('B0011', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '15/07/2022', 8.00, 1, 3, 1, 1)
GO
INSERT INTO Venta VALUES ('B0012', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '16/07/2022', 9.59, 1, 2, 3, 1)
GO
INSERT INTO Venta VALUES ('B0013', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '20/07/2022', 4.70, 1, 2, 3, 1)
GO
INSERT INTO Venta VALUES ('B0014', (SELECT RIGHT ('00000000' + CONVERT(VARCHAR, ISNULL(MAX(VentaID), 0)+1), 4)
FROM Venta), '20/07/2022', 12.00, 1, 2, 4, 1)

```

- Detalle De Venta

```

INSERT INTO Detalle_Venta VALUES
(1,45,4,9.29)
GO
INSERT INTO Detalle_Venta VALUES
(2,50,8,7.4)
GO
INSERT INTO Detalle_Venta VALUES
(2,186,2,220.00)
GO
INSERT INTO Detalle_Venta VALUES
(3,85,2,1.20)
GO
INSERT INTO Detalle_Venta VALUES
(3,183,4,6.00)
GO
INSERT INTO Detalle_Venta VALUES
(3,174,3,20.00)
GO
INSERT INTO Detalle_Venta VALUES
(4,125,5,1.20)
GO
INSERT INTO Detalle_Venta VALUES
(5,190,5,3.50)
GO
INSERT INTO Detalle_Venta VALUES
(6,166,2,42.00)
GO
INSERT INTO Detalle_Venta VALUES
(7,78,2,8.50)
GO
INSERT INTO Detalle_Venta VALUES
(7,96,3,8.00)
GO
INSERT INTO Detalle_Venta VALUES
(7,145,3,9.59)
GO
INSERT INTO Detalle_Venta VALUES
(8,123,3,4.70)
GO
INSERT INTO Detalle_Venta VALUES
(9,172,1,12.00)
GO

```

CAPÍTULO 6:

CONSULTAS DE TABLAS

6. Consultas De Tablas

6.1.Consultas Básicas

```
--MOSTRAR LA CANTIDAD DE PRODUCTOS QUE ESTEN INACTIVOS
SELECT COUNT(*) AS 'CANTIDAD DE PRODUCTOS INACTIVOS'
FROM PRODUCTO
WHERE ESTADO = 0
GO
```

97 %

Resultados Mensajes

	CANTIDAD DE PRODUCTOS INACTIVOS
1	20

```
--MOSTRAR PRODUCTOS CUYO PRECIO DE VENTA SEA MAYORES A 5 SOLES Y ORDENAR DE MANERA ASCENDENTE
SELECT Nombre_Producto,Precio_Venta FROM Producto
WHERE Precio_Venta > 5
ORDER BY Precio_Venta ASC
GO
```

97 %

Resultados Mensajes

	Nombre_Producto	Precio_Venta
1	AJINOMOTO	5.29
2	TROZOS DE ATÚN	5.70
3	ATÚN DESMENUZADO	5.70
4	TROZOS DE ATÚN	5.89
5	LOMOS DE ATUN	5.99
6	TROZOS DE ATÚN	6.00
7	FILETE DE ATÚN	6.00
8	AGUA SAN MATEO	6.00
9	AGUA VIDA	6.00
10	PAPEL HIGIENICO	6.00
11	CROQUETAS	6.00
12	GASEOSA COCA COLA	6.20
13	AVENA PRE COCIDA	6.29
14	PALETAS	6.70
15	YOGURT DURAZNO	6.90
16	TROZOS DE ATÚN	7.00

```
--MOSTRAR LOS PRODUCTOS QUE EMPIECEN CON LA LETRA D USAR COMODIN (LIKE)
SELECT * FROM Producto
WHERE Nombre_Producto LIKE 'D%'
GO
```

97 %

Resultados Mensajes

	ProductoID	Nombre_Producto	Descripcion	Precio_Compra	Precio_Venta	Stock	Fecha_Vencimiento	Estado	CategorialID	MarcalID	ProveedorID
1	44	DURAZNOS ALMIBAR	LATA 822g	10.90	11.90	200	2023-04-15	1	1	18	4
2	45	DURAZNOS ALMIBAR	CUISINE & CO LATA 820 gr	7.99	9.29	140	2023-04-15	1	1	20	4
3	142	DESODORANTE CARBON	MACHO ALFA	10.00	11.00	10	2025-05-18	1	6	69	7
4	143	DESODORANTE LEÑADOR	LATA MÁXIMA	12.00	15.00	30	2021-05-26	1	6	69	6
5	144	DESODORANTE SPRAY	MAXIMA DURACION 48H	8.00	10.00	20	2023-05-17	1	6	69	6
6	145	DESODORANTE BARRA	BARRA FRESH	7.00	9.59	10	2023-05-05	1	6	69	6
7	160	DETERGENTE MARSELLA	AROS DE PODER	3.00	42.00	30	2023-07-06	1	7	64	2
8	161	DETERGENTE MARSELLA	SACA MANCHAS	5.00	42.00	14	2023-10-10	0	7	64	2
9	162	DETERGENTE ARIEL	TOQUE A BEBE	4.00	42.00	20	2023-07-02	1	7	66	2
10	163	DETERGENTE ACE	SACA MANCHAS DE 4KG	12.00	15.00	6	2023-10-07	1	7	63	2
11	164	DETERGENTE ACE	SACA MANCHAS DE 3KG	8.00	10.00	3	2025-07-04	0	7	63	2

```
-- MOSTRAR LA CANTIDAD DE PRODUCTOS REGISTRADOS DONDE EL STOCK SEA MAYOR O IGUAL A 20 Y LA FECHA
-- DE VENCIMIENTO SEA DE 15/04/2023
SELECT Nombre_Producto AS 'PRODUCTO', Stock AS 'STOCK', Fecha_Vencimiento AS 'FECHA' FROM Producto
WHERE Stock >=20 AND Fecha_Vencimiento = '15/04/2023'
GO
```

88 %

Resultados Mensajes

	PRODUCTO	STOCK	FECHA
1	DURAZNOS ALMIBAR	200	2023-04-15
2	DURAZNOS ALMIBAR	140	2023-04-15
3	PIÑAS ALMIBAR	20	2023-04-15
4	PIÑAS RODAJAS	156	2023-04-15

```
-- SELECCIONAR LOS PRODUCTOS CUYOS FECHAS DE VENCIMIENTO ESTEN ENTRE 15/04/2021 Y 07/06/2022
SELECT Nombre_Producto, Fecha_Vencimiento FROM Producto
WHERE Fecha_Vencimiento BETWEEN '15/04/2021' AND '07/06/2022'
GO
```

88 %

Resultados Mensajes

	Nombre_Producto	Fecha_Vencimiento
6	SALSA DE TOMATE NICOLINI	2022-03-14
7	SALSA DE TOMATE SECRETA	2022-03-14
8	FIDEOS TALLARINES	2021-10-04
9	FIDEO CODITOS	2021-05-11
10	SEMOLA	2022-02-10
11	FIDEO TALLARIN	2021-10-18
12	AGUA CIELO	2022-05-31
13	AGUA BENEDICTINO	2021-11-25
14	AGUA SAN MATEO	2022-03-26
15	AGUA SAN MATEO	2022-04-18
16	JAMONADA	2021-07-14
17	YOGURT FRUTOS SECOS	2021-06-30
18	DESODORANTE LEÑADOR	2021-05-26
19	ARENA DE GATOS	2021-05-06
20	CROQUETAS	2021-05-14
21	CROQUETAS	2021-06-14

```
--MOSTRAR PRODUCTOS CUYO PRECIO DE VENTA SEA MAYORES A 5 SOLES Y ORDENAR DE MANERA ASCENDENTE
SELECT Nombre_Producto,Precio_Venta FROM Producto
WHERE Precio_Venta > 5
ORDER BY Precio_Venta ASC
GO
```

88 %

Resultados Mensajes

	Nombre_Producto	Precio_Venta
1	AJINOMOTO	5.29
2	TROZOS DE ATÚN	5.70
3	ATÚN DESMENUZADO	5.70
4	TROZOS DE ATÚN	5.89
5	LOMOS DE ATUN	5.99
6	TROZOS DE ATÚN	6.00
7	FILETE DE ATÚN	6.00
8	AGUA SAN MATEO	6.00
9	AGUA VIDA	6.00
10	PAPEL HIGIENICO	6.00
11	CROQUETAS	6.00
12	GASEOSA COCA COLA	6.20
13	AVENA PRE COCIDA	6.29
14	PALETAS	6.70
15	YOGURT DURAZNO	6.90
16	TROZOS DE ATÚN	7.00
17	GASEOSA SANTA	7.20

Consulta ejecutada correcta... STEPHANO\STEPHANO (15.0 RTM) sa (68) Tienda_Version 00:00:00 94 filas

```
-- MOSTRAR LOS CLIENTES FRECUENTES QUE LLEGAN SIEMPRE A LA TIENDA, ORDENADOS DE
-- MANERA DESCENDENTE POR NOMBRES
SELECT * FROM Cliente
WHERE Estado = 1
ORDER BY Nombres DESC
GO
```

88 %

Resultados Mensajes

	ClienteID	Num_Identificacion	Nombres	Celular	Estado	Documento_IdentidadID
1	1	10421356273	SEBASTIAN	946253124	1	2
2	5	10441256233	MELISSA	959325688	1	2
3	6	10401246233	JUANITO	956325688	1	2
4	2	72145689	CARLOS	951145236	1	1

Consulta ejecut... STEPHANO\STEPHANO (15.0 RTM) sa (68) Tienda_Version 00:00:00 4 filas


```
-- MOSTRAR SOLO LOS PRODUCTOS CUYO NOMBRE EMPIECE CON UNA VOCAL ORDENADOS DE FORMA ASCENDENTE
----- 1º FORMA -----
SELECT Nombre_Producto
FROM Producto
GROUP BY Nombre_Producto
HAVING (LEFT(Nombre_Producto,1) = 'A') OR (LEFT(Nombre_Producto,1) = 'E') OR
(LEFT(Nombre_Producto,1) = 'I') OR (LEFT(Nombre_Producto,1) = 'O') OR (LEFT(Nombre_Producto,1) = 'U')
GO
```

88 %

Resultados Mensajes

	Nombre_Producto
1	ACEITE "EL OLIVAR"
2	ACEITE DE AJONJOLÍ
3	ACEITE DE OLIVA EXTRA
4	ACEITE DE SOYA BELLS
5	ACEITE DE SOYA SAO
6	ACEITE VEGETAL BELLS
7	ACEITE VEGETAL CAPRI
8	ACEITE VEGETAL CORAZON
9	ACEITE VEGETAL NICOLINI
10	ACEITE VEGETAL PREMIUN
11	ACEITE VEGETAL PRIMOR
12	ACONDICIONADOR
13	AGUA BENEDICTINO
14	AGUA CIELO
15	AGUA SAN MATEO
16	AGUA VIDA
17	AJINOMOTO

Consulta ejecutada correctamente. | STEPHANO\STEPHANO (15.0 RTM) | sa (68) | Tienda_Version | 00:00:00 | 35 filas

```
-- MOSTRAR SOLO LOS PRODUCTOS CUYO NOMBRE EMPIECE CON UNA VOCAL ORDENADOS DE FORMA ASCENDENTE
----- 2º FORMA -----
SELECT DISTINCT Nombre_Producto
FROM Producto
WHERE Nombre_Producto LIKE 'A%' OR Nombre_Producto LIKE 'E%' OR Nombre_Producto LIKE 'I%'
OR Nombre_Producto LIKE 'O%' OR Nombre_Producto LIKE 'U%'
ORDER BY Nombre_Producto
GO
```

88 %

Resultados Mensajes

	Nombre_Producto
1	ACEITE "EL OLIVAR"
2	ACEITE DE AJONJOLÍ
3	ACEITE DE OLIVA EXTRA
4	ACEITE DE SOYA BELLS
5	ACEITE DE SOYA SAO
6	ACEITE VEGETAL BELLS
7	ACEITE VEGETAL CAPRI
8	ACEITE VEGETAL CORAZON
9	ACEITE VEGETAL NICOLINI
10	ACEITE VEGETAL PREMIUN
11	ACEITE VEGETAL PRIMOR
12	ACONDICIONADOR
13	AGUA BENEDICTINO
14	AGUA CIELO
15	AGUA SAN MATEO
16	AGUA VIDA
17	AJINOMOTO

Consulta ejecutada correctamente... | STEPHANO\STEPHANO (15.0 RTM) | sa (68) | Tienda_Version | 00:00:00 | 35 filas

-- MOSTRAR SOLO LOS PRODUCTOS CUYO NOMBRE EMPIECE CON UNA VOCAL ORDENADOS DE FORMA ASCENDENTE
----- 3º FORMA -----

```

SELECT DISTINCT Nombre_Producto
FROM Producto
WHERE Nombre_Producto LIKE '[AEIOU]%'
ORDER BY Nombre_Producto
GO

```

88 %

Resultados Mensajes

	Nombre_Producto
1	ACEITE "EL OLIVAR"
2	ACEITE DE AJONJOLÍ
3	ACEITE DE OLIVA EXTRA
4	ACEITE DE SOYA BELLS
5	ACEITE DE SOYA SAO
6	ACEITE VEGETAL BELLS
7	ACEITE VEGETAL CAPRI
8	ACEITE VEGETAL CORAZON
9	ACEITE VEGETAL NICOLINI
10	ACEITE VEGETAL PREMIUN
11	ACEITE VEGETAL PRIMOR
12	ACONDICIONADOR
13	AGUA BENEDICTINO
14	AGUA CIELO
15	AGUA SAN MATEO
16	AGUA VIDA
17	AJONJOLÍ

✓ Consulta ejecutada correcta... STEPHANO\STEPHANO (15.0 RTM) sa (68) Tienda_Version 00:00:00 35 filas

6.2.Consultas De Múltiples Tablas

```
--MOSTRAR EL NOMBRE, PRECIO DE VENTA, CATEGORIA MENORES O IGUAL A 6 SOLES ORDENADOS POR NOMBRE DE PRODUCTO
SELECT P.Nombre_Producto AS 'PRODUCTO', P.Precio_Venta AS 'PRECIO DE VENTA', Nombre_Categoria AS 'CATEGORIA'
FROM Producto P JOIN Categoria C
ON P.CategoriaID = C.CategoriaID
WHERE Precio_Venta <= 6
ORDER BY Nombre_Producto ASC
GO
```

	PRODUCTO	PRECIO DE VENTA	CATEGORIA
1	220	2.50	AGUA Y BEBIDAS
2	ACONDICIONADOR	1.79	HIGIENE Y SALUD
3	AGUA BENEDICTINO	1.50	AGUA Y BEBIDAS
4	AGUA BENEDICTINO	2.50	AGUA Y BEBIDAS
5	AGUA BENEDICTINO	3.00	AGUA Y BEBIDAS
6	AGUA BENEDICTINO	5.00	AGUA Y BEBIDAS
7	AGUA CIELO	2.50	AGUA Y BEBIDAS
8	AGUA CIELO	3.50	AGUA Y BEBIDAS
9	AGUA CIELO	4.50	AGUA Y BEBIDAS
10	AGUA SAN MATEO	1.20	AGUA Y BEBIDAS
11	AGUA SAN MATEO	1.80	AGUA Y BEBIDAS
12	AGUA SAN MATEO	3.20	AGUA Y BEBIDAS
13	AGUA SAN MATEO	5.00	AGUA Y BEBIDAS
14	AGUA SAN MATEO	6.00	AGUA Y BEBIDAS
15	AGUA VIDA	2.00	AGUA Y BEBIDAS
16	AGUA VIDA	3.20	AGUA Y BEBIDAS
17	AGUA VIDA	6.00	AGUA Y BEBIDAS
18	AJINOMOTO	5.29	ABARROTES
19	AQUARIUSK	3.00	AGUA Y BEBIDAS
20	AQUARIUSK	3.00	AGUA Y BEBIDAS
21	AQUARIUSK	5.00	AGUA Y BEBIDAS
22	ARROZ AÑEJO	4.70	ABARROTES
23	ARROZ NO AÑEJO	3.80	ABARROTES
24	ATÚN DESMENUZADO	5.70	ABARROTES
25	ATÚN DESMENUZADO	4.99	ABARROTES
26	AVENA	3.00	DESAYUNOS
27	CAFE TRADICIONAL	1.20	DESAYUNOS
28	CALDO CONCENTRADA DE TOCINO	1.59	ABARROTES

Consulta ejecutada correctamente... STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 107 filas

```
/*MOSTRAR TODAS LAS VENTAS CON LAS SIGUIENTES COLUMNAS: NOMBRE PRODUCTO, */
/*CANTIDAD, PRECIO VENTA, TOTAL, CATEGORIA Y EL EMPLEADO QUE REALIZO LA VENTA*/

SELECT P.Nombre_Producto AS 'PRODUCTO', DV.Cantidad AS 'CANTIDAD',
DV.Precio_Venta AS 'PRECIO DE VENTA', DV.Precio_Venta * DV.Cantidad AS 'TOTAL',
C.Nombre_Categoria AS 'CATEGORIA', E.Nombres AS 'EMPLEADO' FROM Venta V JOIN Detalle_Venta DV
ON V.VentaID = DV.VentaID JOIN Producto P ON P.ProductoID = DV.ProductoID
JOIN Categoria C ON C.CategoriaID = P.CategoriaID JOIN Empleado E ON E.EmpleadoID = V.EmpleadoID
GO
```

	PRODUCTO	CANTIDAD	PRECIO DE VENTA	TOTAL	CATEGORIA	EMPLEADO
1	DURAZNOS ALMIBAR	4	9.29	37.16	ABARROTES	DANIEL AGAPITO
2	FILETE DE ATÚN	8	7.40	59.20	ABARROTES	DANIEL AGAPITO

```
--MOSTRAR TODOS LOS REGISTROS DE LA TABLA PRODUCTO CON SUS RESPECTIVAS CATEGORIAS
SELECT P.Nombre_Producto, C.Nombre_Categoria
FROM Producto P LEFT JOIN Categoria C
ON C.CategoriaID = P.CategoriaID
```

80 %

Resultados Mensajes

	Nombre_Producto	Nombre_Categoria
1	CALDO DE GALLINA	ABARROTES
2	PANQUITA	ABARROTES
3	SAZONADOR DE ARROZ	ABARROTES
4	PIMIENTA NEGRA MOLIDA	ABARROTES
5	CANELA ENTERA	ABARROTES
6	AJINOMOTO	ABARROTES
7	ORÉGANO MOLIDA	ABARROTES
8	COMINO MOLIDO	ABARROTES
9	CALDO CONCENTRADA DE TOCINO	ABARROTES
10	HOJAS DE LAUREL	ABARROTES
11	HONGO Y LAUREL	ABARROTES
12	SAL DE MESA MARINA	ABARROTES
13	SAL ROSADA DE MARAS	ABARROTES
14	SALSA DE TOMATE	ABARROTES
15	SALSA DE TOMATE NICOLINI	ABARROTES
16	SALSA DE TOMATE SECRETA	ABARROTES

Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 190 filas

```
--MOSTRAR LA FUSION DE TODOS LOS PROVEEDORES CONTRA TODAS LAS CATEGORIAS
SELECT PR.Nombre_Proveedor, C.Nombre_Categoria
FROM Proveedor PR CROSS JOIN Categoria C
```

80 %

Resultados Mensajes

	Nombre_Proveedor	Nombre_Categoria
1	DISTRIBUIDOR PEPSICO SAC	ABARROTES
2	DISTRIBUIDOR P & G SAC	ABARROTES
3	DISTRIBUIDOR COCA COLA SAC	ABARROTES
4	DISTRIBUIDOR NESTLE SAC	ABARROTES
5	DISTRIBUIDOR GLORIA SAC	ABARROTES
6	DISTRIBUIDOR DEKASA SAC	ABARROTES
7	DISTRIBUIDOR SC JOHNSON SAC	ABARROTES
8	DISTRIBUIDOR RICOCHET SAC	ABARROTES
9	DISTRIBUIDOR GRUPO MSK SAC	ABARROTES
10	DISTRIBUIDOR KFRAT SAC	ABARROTES
11	DISTRIBUIDOR PEPSICO SAC	DESAYUNOS
12	DISTRIBUIDOR P & G SAC	DESAYUNOS
13	DISTRIBUIDOR COCA COLA SAC	DESAYUNOS
14	DISTRIBUIDOR NESTLE SAC	DESAYUNOS
15	DISTRIBUIDOR GLORIA SAC	DESAYUNOS
16	DISTRIBUIDOR DEKASA SAC	DESAYUNOS

Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 90 filas

```
--MOSTRAR TODOS LOS REGISTROS DE LA TABLA CATEGORIA CON SUS RESPECTIVOS PRODUCTOS
SELECT P.Nombre_Producto, C.Nombre_Categoria
FROM Producto P RIGHT JOIN Categoria C
ON C.CategoriaID = P.CategoriaID
WHERE P.Fecha_Vencimiento IS NULL
```

80 %

Resultados Mensajes

	Nombre_Producto	Nombre_Categoria
1	PAPEL HIGIENICO	HIGIENE Y SALUD
2	ESPONJAS	LIMPIEZA
3	ESPONJAS	LIMPIEZA
4	ESPONJAS	LIMPIEZA
5	ESPONJAS	LIMPIEZA
6	ESCOBA	LIMPIEZA
7	PAPEL 3 HOJAS	LIMPIEZA
8	PAPEL TOALLA	LIMPIEZA
9	COLCHON	COLCHONES Y MUEBLES
10	COLCHON	COLCHONES Y MUEBLES
11	COLCHON	COLCHONES Y MUEBLES
12	ROPERO	COLCHONES Y MUEBLES
13	ROPERO	COLCHONES Y MUEBLES
14	ESCRITORIO	COLCHONES Y MUEBLES
15	PELOTA	MASCOTAS

✓ Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 15 filas

```
MUESTRE POR CADA PROVEEDOR TODAS LAS CATEGORÍAS E INDIQUE POR CADA COMBINACIÓN PROVEEDOR-CATEGORÍA CUANTOS PRODUCTOS SON DE CADA PROVEEDOR-CATEGORÍA. ORDENE POR PROVEEDOR Y CATEGORÍA
*/
SELECT PR.Nombre_Proveedor AS 'EMPRESA', C.Nombre_Categoria AS 'CATEGORIA', COUNT(ProductoID) AS 'CANTIDAD DE PRODUCTOS'
FROM Proveedor PR CROSS JOIN Categoria C LEFT JOIN Producto P
ON PR.ProveedorID = P.ProveedorID AND C.CategoriaID = P.CategoriaID
GROUP BY PR.Nombre_Proveedor, C.Nombre_Categoria
ORDER BY 1,2
GO
```

80 %

Resultados Mensajes

	EMPRESA	CATEGORIA	CANTIDAD DE PRODUCTOS
1	DISTRIBUIDOR COCA COLA SAC	ABARROTES	0
2	DISTRIBUIDOR COCA COLA SAC	AGUA Y BEBIDAS	35
3	DISTRIBUIDOR COCA COLA SAC	COLCHONES Y MUEBLES	0
4	DISTRIBUIDOR COCA COLA SAC	DESAYUNOS	0
5	DISTRIBUIDOR COCA COLA SAC	GALLETAS, SNACKS Y GOLOSINAS	0
6	DISTRIBUIDOR COCA COLA SAC	HIGIENE Y SALUD	0
7	DISTRIBUIDOR COCA COLA SAC	LACTEOS Y EMBUTIDOS	0
8	DISTRIBUIDOR COCA COLA SAC	LIMPIEZA	0
9	DISTRIBUIDOR COCA COLA SAC	MASCOTAS	0
10	DISTRIBUIDOR DEKASA SAC	ABARROTES	1
11	DISTRIBUIDOR DEKASA SAC	AGUA Y BEBIDAS	0
12	DISTRIBUIDOR DEKASA SAC	COLCHONES Y MUEBLES	3
13	DISTRIBUIDOR DEKASA SAC	DESAYUNOS	0
14	DISTRIBUIDOR DEKASA SAC	GALLETAS, SNACKS Y GOLOSINAS	0
15	DISTRIBUIDOR DEKASA SAC	HIGIENE Y SALUD	4
16	DISTRIBUIDOR DEKASA SAC	LACTEOS Y EMBUTIDOS	0

✓ Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 90 filas

6.3.Subconsultas

```
--MOSTRAR EL NOMBRE DEL PRODUCTO Y EL PRECIO DE VENTA QUE SEA MAYORES AL PRECIO DEL PRODUCTO MILO
SELECT Nombre_Producto AS 'PRODUCTO', Precio_Venta AS 'PRECIO' FROM Producto WHERE Precio_Venta >
(SELECT Precio_Venta FROM Producto where Nombre_Producto = 'MILO' )
```

80 %

Resultados Mensajes

	PRODUCTO	PRECIO
1	ACEITE VEGETAL PRIMOR	18.70
2	ACEITE VEGETAL BELLS	42.50
3	ACEITE DE OLIVA EXTRA	40.00
4	ACEITE VEGETAL CAPRI	17.70
5	ACEITE "EL OLIVAR"	25.40
6	ACEITE VEGETAL CAPRI	25.00
7	ACEITE VEGETAL BELLS	40.00
8	ACEITE VEGETAL CORAZON	40.00
9	PIÑAS ALMIBAR	21.50
10	ARROZ	155.00
11	ARROZ	185.00
12	AVENA CLÁSICA	40.00
13	LECHE GLORIA	25.00
14	CEREAL TRIGO CHOCOLATE	16.50
15	DESODORANTE LEÑADOR	15.00
16	SHAMPOO	19.00
17	DETERGENTE MARSELLA	42.00

Consulta ejecutada correctamente | STEPHANO\STEPHANO (15.0 RTM) | sa (68) | Tienda_Version | 00:00:00 | 37 filas

```
--MOSTRAR EL NOMBRE Y EL PRECIO DE VENTA DE UN DETERMINADO PRODUCTO Y LA DIFERENCIA CON
--EL PRECIO DE VENTA DEL PRODUCTO MÁS COSTOSO
SELECT NOMBRE_PRODUCTO,PRECIO_VENTA,
PRECIO_VENTA-(SELECT MAX(PRECIO_VENTA) AS 'PRECIO' FROM PRODUCTO) AS DIFERENCIA
FROM PRODUCTO
WHERE NOMBRE_PRODUCTO = 'AROMATIZANTE'

-- MOSTRAR EL NOMBRE, CATEGORIA Y PRECIO DEL PRODUCTO MÁS COSTOSO
SELECT P.Nombre_Producto,C.Nombre_Categoria, Precio_Venta
FROM Producto P JOIN Categoria C
ON P.CategoriaID = C.CategoriaID
```

80 %

Resultados Mensajes

	NOMBRE_PRODUCTO	PRECIO_VENTA	DIFERENCIA
1	AROMATIZANTE	15.00	-205.00

```
-- MOSTRAR EL NOMBRE, CATEGORIA Y PRECIO DEL PRODUCTO MÁS COSTOSO
SELECT P.Nombre_Producto,C.Nombre_Categoria, Precio_Venta
FROM Producto P JOIN Categoria C
ON P.CategoriaID = C.CategoriaID
WHERE Precio_Venta =
(SELECT MAX(Precio_Venta) FROM Producto)

-- ACTUALIZAR EL PRECIO DEL PRODUCTO MÁS COSTOSO
UPDATE Producto SET Precio_Venta = 230
WHERE Precio_Venta =
(SELECT MAX(Precio_Venta) FROM Producto);
```

80 %

Resultados Mensajes

	Nombre_Producto	Nombre_Categoria	Precio_Venta
1	CROQUETAS	MASCOTAS	220.00

```
--MOSTRAR EL NOMBRE DE LA CATEGORIA Y PRODUCTO, DONDE CUYA CATEGORÍA SEAN AQUELLAS QUE EMPIECEN CON LA LETRA 'D'
```

```
SELECT C.Nombre_Categoria, P.Nombre_Producto
FROM Producto P INNER JOIN Categoria C
ON P.CategoriaID = C.CategoriaID
WHERE C.Nombre_Categoria IN
(SELECT C.Nombre_Categoria FROM Categoria C WHERE Nombre_Categoria LIKE 'D%')
```

80 %

Resultados Mensajes

	Nombre_Categoria	Nombre_Producto
1	DESAYUNOS	AVENA
2	DESAYUNOS	MILO
3	DESAYUNOS	AVENA CLÁSICA
4	DESAYUNOS	AVENA TRADICIONAL
5	DESAYUNOS	AVENA PRE COCIDA
6	DESAYUNOS	COCOA
7	DESAYUNOS	LECHE GLORIA
8	DESAYUNOS	LECHE GLORIA
9	DESAYUNOS	CEREAL FRUTA AROS
10	DESAYUNOS	CEREAL TRIGO ATÓMICO
11	DESAYUNOS	CEREAL TRIGO CHOCOLATE
12	DESAYUNOS	CEREAL BAR
13	DESAYUNOS	CAFE TRADICIONAL

Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 13 filas


```
--MOSTRAR EL NOMBRE DE LA CATEGORIA Y PRODUCTO, DONDE CUYA CATEGORÍA SEAN AQUELLAS QUE EMPIECEN CON LA LETRA 'D'
SELECT C.Nombre_Categoria, P.Nombre_Producto
FROM Producto P INNER JOIN Categoria C
ON P.CategoriaID = C.CategoriaID
WHERE EXISTS
(SELECT * FROM Categoria C WHERE C.Nombre_Categoria LIKE 'D%' AND C.CategoriaID = P.CategoriaID)
GO
```

80 %

Resultados Mensajes

	Nombre_Categoria	Nombre_Producto
1	DESAYUNOS	AVENA
2	DESAYUNOS	MILO
3	DESAYUNOS	AVENA CLÁSICA
4	DESAYUNOS	AVENA TRADICIONAL
5	DESAYUNOS	AVENA PRE COCIDA
6	DESAYUNOS	COCOA
7	DESAYUNOS	LECHE GLORIA
8	DESAYUNOS	LECHE GLORIA
9	DESAYUNOS	CEREAL FRUTA AROS
10	DESAYUNOS	CEREAL TRIGO ATÓMICO
11	DESAYUNOS	CEREAL TRIGO CHOCOLATE
12	DESAYUNOS	CEREAL BAR
13	DESAYUNOS	CAFE TRADICIONAL

Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 13 filas

```
--MOSTRAR EL NOMBRE DE TODOS LOS PRODUCTOS Y LAS CATEGORIAS A LA QUE PERTENECE, SIN CONSIDERAR LA CATEGORIA DESAYUNOS
SELECT C.Nombre_Categoria, P.Nombre_Producto
FROM Producto P INNER JOIN Categoria C
ON P.CategoriaID = C.CategoriaID
WHERE NOT EXISTS
(SELECT * FROM Categoria C WHERE C.Nombre_Categoria LIKE 'D%' AND C.CategoriaID = P.CategoriaID)
```

80 %

Resultados Mensajes

	Nombre_Categoria	Nombre_Producto
1	ABARROTES	MAYONESA A LA CENA
2	ABARROTES	FIDEOS SPAGHETTI
3	ABARROTES	FIDEO TALLARIN
4	ABARROTES	FIDEO CANUTO
5	ABARROTES	FIDEO TALLARIN
6	ABARROTES	ACEITE VEGETAL PRIMOR
7	ABARROTES	ACEITE VEGETAL PRIMOR
8	ABARROTES	ACEITE VEGETAL PREMIUN
9	ABARROTES	ACEITE VEGETAL NICOLINI
10	ABARROTES	ACEITE DE SOYA BELLS
11	ABARROTES	ACEITE VEGETAL BELLS
12	ABARROTES	ACEITE DE SOYA SAO
13	ABARROTES	ACEITE VEGETAL BELLS
14	ABARROTES	ACEITE DE OLIVA EXTRA
15	ABARROTES	ACEITE VEGETAL CAPRI
16	ABARROTES	ACEITE "EL OLIVAR"

Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (65) Tienda_Version 00:00:00 177 filas

CAPÍTULO 7: FUNCIONES

7. Funciones

7.1. Funciones De Agregación

```
--MOSTRAR LA CANTIDAD DE PRODUCTOS Y PRECIO DE VENTA PROMEDIO DONDE EL NOMBRE DEL PROVEEDOR
--SEA DISTRIBUIDOR NESTLE SAC Y ADEMÁS QUE SEA DE LA CATEGORIAS ABARROTES, AGRUPADOS POR
--CODIGO DE PROVEEDOR Y RAZON SOCIAL
SELECT COUNT(ProductoID) AS 'CANTIDAD', AVG(P.Precio_Venta) AS 'PROMEDIO DE PRECIO VENTA', PR.Nombre_Proveedor AS 'PROVEEDOR'
FROM Producto P INNER JOIN Categoria C
ON P.CategoriaID = C.CategoriaID INNER JOIN Proveedor PR
ON P.ProveedorID = PR.ProveedorID
WHERE PR.Nombre_Proveedor = 'DISTRIBUIDOR NESTLE SAC'
GROUP BY PR.ProveedorID, PR.Nombre_Proveedor
```

	CANTIDAD	PROMEDIO DE PRECIO VENTA	PROVEEDOR
1	59	12.685932	DISTRIBUIDOR NESTLE SAC

- Comprobando Los Resultados

```
SELECT COUNT(P.ProductoID) AS 'CANTIDAD'
FROM Producto P INNER JOIN Proveedor PR
ON P.ProveedorID = PR.ProveedorID
WHERE PR.Nombre_Proveedor = 'DISTRIBUIDOR NESTLE SAC'
```

	CANTIDAD
1	59

```
SELECT AVG(P.Precio_Venta) AS 'PROMEDIO DE PRECIO DE VENTA'
FROM Producto P INNER JOIN Proveedor PR
ON P.ProveedorID = PR.ProveedorID
WHERE PR.Nombre_Proveedor = 'DISTRIBUIDOR NESTLE SAC'
```

	PROMEDIO DE PRECIO DE VENTA
1	12.685932


```
--PROMEDIO DEL PREIO DE COMPRA DE LOS PRODUCTOS
SELECT AVG(PRECIO_COMPRA) AS 'PROMEDIO DE COMPRA' FROM Producto

-- MOSTRAR EL PROMEDIO DE LOS PRODUCTOS QUE SON MAYORES A 2022 CON UN PRECIO DE VENTA MENOR A 8
SELECT AVG(Precio_Venta) AS 'PROMEDIO' FROM Producto
WHERE Fecha_Vencimiento > '01/01/2022'
```

PROMEDIO DE COMPRA	
1	12.934578

PROMEDIO	
1	13.047878

```
--LISTAR LOS CLIENTES CUYO EL PRECIO TOTAL DEL DETALLE FINAL SUPERE LOS 100 SOLES EN VENTAS
SELECT C.NOMBRES, SUM(DV.CANTIDAD * DV.PRECIO_VENTA) AS [TOTAL VENTAS]
FROM VENTA V INNER JOIN CLIENTE C
ON V.CLIENTEID = C.CLIENTEID INNER JOIN DETALLE_VENTA DV
ON DV.VENTAID = V.VENTAID
GROUP BY C.NOMBRES
HAVING SUM(DV.CANTIDAD * DV.PRECIO_VENTA) > 200
GO
```

	NOMBRES	TOTAL VENTAS
1	CARLOS	499.20

7.2.Funciones Escalares

```
--CREAMOS UNA FUNCION QUE DEVUELVA LA CANTIDAD DE PRODUCTOS PARA UNA DETERMINADA MARCA
CREATE FUNCTION TotalProductos
(
    --CREAMOS LA VARIABLE QUE VA ALMACENAR NUESTRA FUNCION
    @Codigo INTEGER
)
--RETORNAMOS EL TIPO DE DATO QUE HEMOS ASIGNADO A NUESTRA VARIABLE
RETURNS INTEGER
AS
BEGIN
    --DECLARAMOS LA VARIABLE DE RETORNO, EJMPLO CANTIDAD DE PRODUCTOS
    DECLARE @CANTIDAD INTEGER
    --AÑADIMOS AQUÍ LAS SENTENCIAS T-SQL PARA CALCULAR EL VALOR DE RETORNO, EJMPLO CANTIDAD
    SELECT @CANTIDAD = COUNT(Nombre_Producto) -- CUENTA LA CANTIDAD DE PRODUCTOS Y LOS ALMACENA EN @CANTIDAD
    FROM Producto
    WHERE MarcaID = @codigo
    --DEVOLVEMOS EL RESULTADO DE LA FUNCION VARIABLE DE RETORNO
    RETURN @CANTIDAD
END
GO
```

- Realizamos La Consulta La Función Creada

```
-- MOSTRAR POR CADA MARCA, EL NOMBRE Y CUANTOS PRODUCTOS TIENE CADA MARCA
SELECT Nombre_Marca, dbo.TotalProductos(MarcaID) AS 'TOTAL DE PRODUCTOS' FROM Marca
```

106 %

Resultados Mensajes

	Nombre_Marca	TOTAL DE PRODUCTOS
1	DOÑA GUSTA	1
2	SIBARITA	8
3	MAGGI	2
4	MARINA	2
5	DON VICTORIO	2
6	MOLITALIA	4
7	POMAROLA	1
8	ALACENA	3
9	ALIANZA	3
10	TARI	1
11	CAPRI	2
12	NICOLINI	2
13	EL OLIVAR	3
14	PRIMOR	6
15	SAO	1
16	FLORIDA	2
17	CAMPOMAR	2

Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (70) Tienda_Version 00:00:00 88 filas

```
--CREAR UNA FUNCION QUE DEVUELVA LA CANTIDAD DE PRODUCTOS PARA UNA DETERMINADA CATEGORIA
CREATE FUNCTION ConjuntoCategoria
(
    --CREAMOS LA VARIABLE QUE VA ALMACENAR NUESTRA FUNCION
    @Codigo INTEGER
)
--RETORNAMOS EL TIPO DE DATO QUE HEMOS ASIGNADO A NUESTRA VARIABLE
RETURNS INTEGER
AS
BEGIN
    --DECLARAMOS LA VARIABLE DE RETORNO, EJMPLO CANTIDAD DE PRODUCTOS
    DECLARE @Global INTEGER
    --AÑADIMOS AQUÍ LAS SENTENCIAS T-SQL PARA CALCULAR EL VALOR DE RETORNO, EJMPLO Global
    SELECT @Global = COUNT(Nombre_Producto) -- CUENTA LA CANTIDAD DE PRODUCTOS Y LOS ALMACENA EN @@Global
    FROM Producto
    WHERE CategoriaID = @codigo
    --DEVOLVEMOS EL RESULTADO DE LA FUNCION VARIABLE DE RETORNO
    RETURN @Global
END
GO
```

- Realizamos La Consulta De La Función Creada

```
-- MOSTRAR POR CADA CATEGORIA, EL NOMBRE Y CUANTOS PRODUCTOS TIENE CADA CATEGORIA
SELECT Nombre_Categoria, dbo.ConjuntoCategoria(CategoriaID) AS 'TOTAL DE PRODUCTOS' FROM Categoria
```

	Nombre_Categoria	TOTAL DE PRODUCTOS
1	ABARROTES	61
2	DESAYUNOS	13
3	AGUA Y BEBIDAS	40
4	GALLETAS, SNACKS Y GOLOSINAS	14
5	LACTEOS Y EMBUTIDOS	11
6	HIGIENE Y SALUD	14
7	LIMPIEZA	21
8	COLCHONES Y MUEBLES	6
9	MASCOTAS	10

Consulta ejecutada correctamente. STEPHANO\STEPHANO (15.0 RTM) sa (70) Tienda_Version 00:00:00 9 filas

CAPÍTULO 8: PROCEDIMIENTOS ALMACENADOS

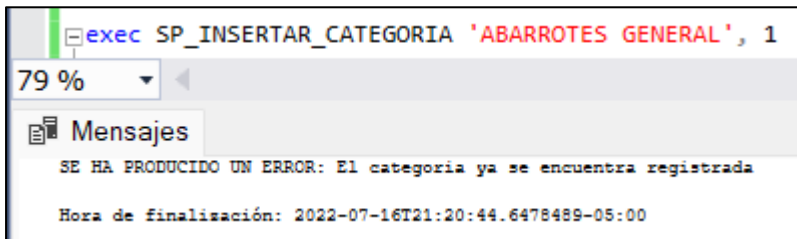
8. Procedimientos Almacenados

8.1.Procedimiento Almacenado Para Registrar

```
--CREAR UN PROCEDIMIENTO ALMACENADO PARA REGISTRAR UNA CATEGORIA
CREATE OR ALTER PROCEDURE SP_INSERTAR_CATEGORIA
(
    @Nombre_Categoria VARCHAR(100),
    @Estado BIT
)
AS BEGIN
    BEGIN TRY
        IF EXISTS (SELECT * FROM Categoria WHERE Nombre_Categoria = @Nombre_Categoria)
            RAISERROR('El categoria ya se encuentra registrada',16,1)
        ELSE
            IF (@Nombre_Categoria = '0')
                RAISERROR('Parámetro Invalido, la categoria no puede contener un valor 0', 16, 1)
            ELSE
                IF (@Nombre_Categoria IS NULL)
                    RAISERROR('El nombre de la categoria no puede ser nulo', 16, 1)
                ELSE
                    IF (@Nombre_Categoria = '')
                        RAISERROR('El nombre de la categoria no puede estar vacío', 16, 1)
                    ELSE
                        INSERT INTO Categoria VALUES (@Nombre_Categoria, @Estado)
                        PRINT 'Validación Correcta'
                    END TRY
                BEGIN CATCH
                    PRINT 'SE HA PRODUCIDO UN ERROR: ' + ERROR_MESSAGE()
                END CATCH
            END
        GO
```

- Comprobando cada RAISERROR

❖ Registrando Una Categoría Ya Existente



```
exec SP_INSERTAR_CATEGORIA 'ABARROTES GENERAL', 1
```

79 %

Mensajes

SE HA PRODUCIDO UN ERROR: El categoria ya se encuentra registrada

Hora de finalización: 2022-07-16T21:20:44.6478489-05:00

❖ Insertando una categoría indicándole con parámetro 0

```
exec SP_INSERTAR_CATEGORIA 0, 1
```

79 %

Mensajes

SE HA PRODUCIDO UN ERROR: Parámetro Invalido, la categoria no puede contener un valor 0

Hora de finalización: 2022-07-16T21:17:18.5241987-05:00

❖ Insertando una categoría con parámetros nulos

```
exec SP_INSERTAR_CATEGORIA null, 1
```

96 %

Mensajes

SE HA PRODUCIDO UN ERROR: El nombre de la categoria no puede ser nulo

Hora de finalización: 2022-07-16T21:25:01.9025730-05:00

❖ Insertando una categoría con parámetros vacío

```
exec SP_INSERTAR_CATEGORIA '', 1
```

79 %

Mensajes

SE HA PRODUCIDO UN ERROR: El nombre de la categoria no puede estar vacío

Hora de finalización: 2022-07-16T21:19:09.0847758-05:00

❖ Insertando una categoría nueva

```
exec SP_INSERTAR_CATEGORIA 'UTILES', 1
```

96 %

Mensajes

(1 fila afectada)
Validación Correcta

Hora de finalización: 2022-07-16T21:45:08.9994962-05:00

8.2.Procedimiento Almacenado Para Eliminar

```

CREATE OR ALTER PROCEDURE SP_ELIMINAR_CATEGORIA
(
    @CategoriaID SMALLINT
)
AS BEGIN
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM Categoria WHERE CategoriaID = @CategoriaID)
            RAISERROR('El categoria NO se encuentra registrada',16,1)
        ELSE
            IF EXISTS (SELECT * FROM Producto WHERE CategoriaID = @CategoriaID)
                RAISERROR('LA CATEGORIA SE ENCUENTRA ASIGNADA A 1 O MUCHOS PRODUCTOS',16,1)
            ELSE
                DELETE FROM Categoria WHERE CategoriaID = (@CategoriaID)
                PRINT 'Categoria Eliminada exitosamente'
            END TRY
        BEGIN CATCH
            PRINT 'SE HA PRODUCIDO UN ERROR: ' + ERROR_MESSAGE()
        END CATCH
    END
GO

```

- Comprobando cada RAISERROR

❖ Eliminando una categoría no existente

```

exec SP_ELIMINAR_CATEGORIA 12

```

96 %

Mensajes

SE HA PRODUCIDO UN ERROR: El categoria NO se encuentra registrada

Hora de finalización: 2022-07-16T21:51:12.6412978-05:00

❖ Eliminando una categoría relacionada con otra tabla

```

exec SP_ELIMINAR_CATEGORIA 2

```

96 %

Mensajes

SE HA PRODUCIDO UN ERROR: LA CATEGORIA SE ENCUENTRA ASIGNADA A 1 O MUCHOS PRODUCTOS

Hora de finalización: 2022-07-16T21:53:38.7360636-05:00

- ❖ Eliminando una categoría que no esté relacionada con otra tabla

```
exec SP_ELIMINAR_CATEGORIA 10
```

96 %

Mensajes

(1 fila afectada)
Categoría Eliminada exitosamente
Hora de finalización: 2022-07-16T21:55:27.5744639-05:00

8.3.Procedimiento Almacenado Para Buscar

```
CREATE OR ALTER PROCEDURE SP_BUSCAR_CATEGORIA
(
    @Nombre_Categoria VARCHAR(100)
)
AS BEGIN
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM Categoria WHERE Nombre_Categoria = @Nombre_Categoria)
            RAISERROR('El categoria no existe',16,1)
        ELSE
            IF (@Nombre_Categoria = '0')
                RAISERROR('Parámetro Invalido, no puede contener un 0', 16, 1)
            ELSE
                IF (@Nombre_Categoria IS NULL)
                    RAISERROR('El nombre de la categoria no puede ser nulo', 16, 1)
                ELSE
                    IF (@Nombre_Categoria = '')
                        RAISERROR('El nombre de la categoria no puede estar vacío', 16, 1)
                    ELSE
                        SELECT * FROM Categoria WHERE Nombre_Categoria = (@Nombre_Categoria)
                        PRINT 'Categoria Encontrada, Felicidades'
                    END TRY
                BEGIN CATCH
                    PRINT 'SE HA PRODUCIDO UN ERROR: ' + ERROR_MESSAGE()
                END CATCH
            END
        GO
```


8.4.Procedimiento Almacenado Para Actualizar

```
CREATE OR ALTER PROCEDURE SP_ACTUALIZAR_CATEGORIA
(
    @CategoriaID SMALLINT,
    @Nombre_Categoria VARCHAR(100),
    @Estado BIT
)
AS BEGIN
    BEGIN TRY
        IF NOT EXISTS (SELECT * FROM Categoria WHERE CategoriaID = @CategoriaID)
            RAISERROR('No puede actualizar una categoria que no existe',16,1)
        ELSE
            IF (@Nombre_Categoria IS NULL)
                RAISERROR('El nombre de la categoria no puede ser nulo', 16, 1)
            ELSE
                IF (@Nombre_Categoria = '')
                    RAISERROR('El nombre de la categoria no puede estar vacío', 16, 1)
                ELSE
                    UPDATE Categoria SET Nombre_Categoria = (@Nombre_Categoria), Estado = (@Estado) WHERE CategoriaID = (@CategoriaID)
                    PRINT 'CATEGORIA ACTUALIZADA'
            END TRY
        BEGIN CATCH
            PRINT 'SE HA PRODUCIDO UN ERROR: ' + ERROR_MESSAGE()
        END CATCH
    END
GO
```

```
EXEC SP_ACTUALIZAR_CATEGORIA 1, 'ABARROTES GENERAL', 1
EXEC SP_ACTUALIZAR_CATEGORIA 2, 'DESAYUNOS NUTRITIVOS', 1
EXEC SP_ACTUALIZAR_CATEGORIA 3, 'AGUAS Y BEBIDAS REFRESCANTES', 1
EXEC SP_ACTUALIZAR_CATEGORIA 4, 'GALLETAS, SNACKS Y GOLOSINAS GENIALES', 1
EXEC SP_ACTUALIZAR_CATEGORIA 5, 'LACTEOS Y EMBUTIDOS ALIMENTICIOS', 1
EXEC SP_ACTUALIZAR_CATEGORIA 6, 'HIGIENE Y SALUD PRIORIDAD', 1
EXEC SP_ACTUALIZAR_CATEGORIA 7, 'LIMPIEZA DEL HOGAR', 1
EXEC SP_ACTUALIZAR_CATEGORIA 8, 'COLCHONES Y MUEBLES DE CALIDAD', 1
EXEC SP_ACTUALIZAR_CATEGORIA 9, 'MASCOTAS PEQUEÑAS Y GRANDES', 1
```

95 %

Mensajes

```
(1 fila afectada)
CATEGORIA ACTUALIZADA

(1 fila afectada)
CATEGORIA ACTUALIZADA

(1 fila afectada)
CATEGORIA ACTUALIZADA

(1 fila afectada)
CATEGORIA ACTUALIZADA

(1 fila afectada)
CATEGORIA ACTUALIZADA

(1 fila afectada)
CATEGORIA ACTUALIZADA

(1 fila afectada)
CATEGORIA ACTUALIZADA
```

8.5.Procedimiento Almacenado para listar productos

```
-- CREANDO UN PROCEDIMIENTO ALMACENADO QUE ME DEVUELVA UNA LISTA
-- CON LA NOMBRE DE CATEGORIA, PRODUCTO Y LA FECHA DE VENCIMIENTO
-- CON UN PRECIO DE VENTA MENOR A LOS 3 SOLES
CREATE PROCEDURE SP_LISTA_PRODUCTOS
AS
BEGIN
    SELECT P.Nombre_Producto, C.Nombre_Categoria, P.Precio_Venta
    FROM Producto P INNER JOIN Categoria C
    ON C.CategoriaID = P.CategoriaID
    WHERE Precio_Venta <3
    ORDER BY P.Nombre_Producto
END
GO
```

- Ejecutando El Procedimiento Almacenado

----- EJECUTANDO EL PROCEDIMIENTO ALMACENADO

EXEC dbo.SP_LISTA_PRODUCTOS

95 %

Resultados Mensajes

	Nombre_Producto	Nombre_Categoria	Precio_Venta
1	220	AGUAS Y BEBIDAS REFRESCANTES	2.50
2	ACONDICIONADOR	HIGIENE Y SALUD PRIORIDAD	1.79
3	AGUA BENEDICTINO	AGUAS Y BEBIDAS REFRESCANTES	1.50
4	AGUA BENEDICTINO	AGUAS Y BEBIDAS REFRESCANTES	2.50
5	AGUA CIELO	AGUAS Y BEBIDAS REFRESCANTES	2.50
6	AGUA SAN MATEO	AGUAS Y BEBIDAS REFRESCANTES	1.20
7	AGUA SAN MATEO	AGUAS Y BEBIDAS REFRESCANTES	1.80
8	AGUA VIDA	AGUAS Y BEBIDAS REFRESCANTES	2.00
9	CAFE TRADICIONAL	DESAYUNOS NUTRITIVOS	1.20
10	CALDO CONCENTRADA DE TOCINO	ABARROTES GENERAL	1.59
11	CALDO DE GALLINA	ABARROTES GENERAL	1.59
12	CUATES	GALLETAS, SNACKS Y GOLOSINAS GENIALES	1.30
13	ESPONJAS	LIMPIEZA DEL HOGAR	2.00
14	ESPONJAS	LIMPIEZA DEL HOGAR	2.50
15	FIDEO CANUTO	ABARROTES GENERAL	1.30
16	FIDEO CODITOS	ABARROTES GENERAL	2.70
17	FIDEO TALLARIN	ABARROTES GENERAL	2.00
18	GALLETA PICARAS	GALLETAS, SNACKS Y GOLOSINAS GENIALES	2.80
19	GALLETA RELLENITA	GALLETAS, SNACKS Y GOLOSINAS GENIALES	2.50
20	GALLETA SODA	GALLETAS, SNACKS Y GOLOSINAS GENIALES	1.50
21	GALLETA TENTACION	GALLETAS, SNACKS Y GOLOSINAS GENIALES	2.50

Consulta ejecutada correctamente.

8.6.Procedimiento Almacenado Para Listar Cargos

```
-- CREANDO UN PROCEDIMIENTO ALMACENADO QUE ME DEVUELVA UNA LISTA
-- CON TODOS LOS EMPLEADOS REGISTRADOS CON SUS RESPECTIVOS CARGOS

CREATE PROCEDURE SP_LISTA_CARGOS
AS
BEGIN
    SELECT E.Nombres, E.Apellidos, CA.Tipo
    FROM Empleado E INNER JOIN Cargo CA
    ON CA.CargoID = E.CargoID
    ORDER BY Apellidos DESC
END
GO
```

- Ejecutando El Procedimiento Almacenado

```
-- EJECUTANDO EL PROCEDIMIENTO ALMACENADO
EXEC dbo.SP_LISTA_CARGOS
```

95 %

Resultados Mensajes

	Nombres	Apellidos	Tipo
1	KEVIN ENRIQUE	TEQUEN OBLITAS	EMPLEADO
2	MARK STEPHANO	SANCHEZ MEDINA	ADMINISTRADOR
3	FRANCO ALEXIS	SALAZAR SOPLAPUCO	EMPLEADO
4	JUAN JOSE	MEDINA SANCHEZ	EMPLEADO
5	DANIEL AGAPITO	CURO BERNILLA	EMPLEADO

8.7.Procedimiento Almacenado Para Listar Ventas

```
-- CREANDO UN PROCEDIMIENTO ALMACENADO QUE ME DEVUELVA UNA LISTA CON LAS
-- VENTAS REALIZADAS, LA CANTIDAD DE PRODUCTOS QUE LLEVA, EL TOTAL Y EL
-- EMPLEADO QUE REALIZA LA VENTA CON SU CARGO EN LA EMPRESA

CREATE PROCEDURE SP_LISTA_VENTAS
AS
BEGIN
    SELECT P.Nombre_Producto AS 'NOMBRE DE PRODUCTO', DV.Cantidad AS 'CANTIDAD', DV.Precio_Venta AS 'PRECIO DE VENTA',
    (DV.Precio_Venta * DV.Cantidad) AS 'TOTAL', E.Nombres AS 'EMPLEADO', CA.Tipo AS 'CARGO EN LA EMPRESA'
    FROM
    Detalle_Venta DV INNER JOIN Venta V
    ON V.VentaID = DV.VentaID INNER JOIN Producto P
    ON DV.ProductoID = P.ProductoID INNER JOIN Empleado E
    ON E.EmpleadoID = V.EmpleadoID INNER JOIN Cargo CA
    ON CA.CargoID = E.CargoID
END
GO
```

- Ejecutando el procedimiento almacenado

--EJECUTANDO EL PROCEDIMIENTO ALMACENADO

EXEC dbo.SP_LISTA_VENTAS

95 %

Resultados Mensajes

	NOMBRE DE PRODUCTO	CANTIDAD	PRECIO DE VENTA	TOTAL	EMPLEADO	CARGO EN LA EMPRESA
1	DURAZNOS ALMIBAR	4	9.29	37.16	DANIEL AGAPITO	EMPLEADO
2	FILETE DE ATÚN	8	7.40	59.20	DANIEL AGAPITO	EMPLEADO
3	CROQUETAS	2	220.00	440.00	DANIEL AGAPITO	EMPLEADO
4	AGUA SAN MATEO	2	1.20	2.40	MARK STEPHANO	ADMINISTRADOR
5	CROQUETAS	4	6.00	24.00	MARK STEPHANO	ADMINISTRADOR
6	LIMPIA BAÑOS	3	20.00	60.00	MARK STEPHANO	ADMINISTRADOR
7	PAPITAS FRITAS	5	1.20	6.00	FRANCO ALEXIS	EMPLEADO
8	PATÉ	5	3.50	17.50	MARK STEPHANO	ADMINISTRADOR
9	SUAVIZANTE DOWNY	2	42.00	84.00	MARK STEPHANO	ADMINISTRADOR
10	AGUA CIELO	2	8.50	17.00	KEVIN ENRIQUE	EMPLEADO
11	INKA COLA	3	8.00	24.00	KEVIN ENRIQUE	EMPLEADO
12	DESODORANTE BARRA	3	9.59	28.77	KEVIN ENRIQUE	EMPLEADO
13	CHUPETINES	3	4.70	14.10	KEVIN ENRIQUE	EMPLEADO
14	LIMPIA COCINA	1	12.00	12.00	JUAN JOSE	EMPLEADO
15	SUAVIZANTE DOWNY	1	42.00	42.00	JUAN JOSE	EMPLEADO
16	SALSA DE TOMATE NICOLINI	5	2.20	11.00	KEVIN ENRIQUE	EMPLEADO

8.8.Procedimiento Almacenado Cantidad De Productos Por Cliente

```
--CREAR UN PROCEDIMIENTO ALMACENADO QUE ME DEVUELVA LA CANTIDAD DE PRODUCTOS POR CADA CLIENTE
CREATE PROCEDURE SP_CANTIDAD_PRODUCTOS
AS
BEGIN
    SELECT C.Nombres, SUM(DV.Cantidad) AS 'CANTIDAD'
    FROM Cliente C INNER JOIN Venta V
    ON V.ClienteID = C.ClienteID INNER JOIN Detalle_Venta DV
    ON DV.VentaID = V.VentaID
    GROUP BY C.Nombres
END
GO
```

- Ejecutando el procedimiento almacenado

EXEC SP_CANTIDAD_PRODUCTOS

95 %

Resultados Mensajes

	Nombres	CANTIDAD
1	CARLOS	19
2	CLAUDIA	2
3	JUANITO	3
4	MELISSA	12
5	PRISCILLA	10
6	TEO	7

CAPÍTULO 9: VISTAS

9. Vistas

Una vista es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta.

9.1.Vistas Creadas

9.1.1. Vista – Mostrar Venta

```

/*CREAR UNA VISTA DONDE ME MUESTRE TODAS LAS VENTAS CON LAS SIGUIENTES COLUMNAS:
NOMBRE PRODUCTO,CANTIDAD, PRECIO VENTA, TOTAL, CATEGORIA Y EL EMPLEADO QUE REALIZO LA VENTA*/

CREATE OR ALTER VIEW VW_MOSTRAR_VENTAS
AS
SELECT P.Nombre_Producto AS 'PRODUCTO', DV.Cantidad AS 'CANTIDAD',
DV.Precio_Venta AS 'PRECIO DE VENTA', DV.Precio_Venta * DV.Cantidad AS 'TOTAL',
C.Nombre_Categoria AS 'CATEGORIA', E.Nombres AS 'EMPLEADO' FROM Venta V JOIN Detalle_Venta DV
ON V.VentaID = DV.VentaID JOIN Producto P ON P.ProductoID = DV.ProductoID
JOIN Categoria C ON C.CategoriaID = P.CategoriaID JOIN Empleado E ON E.EmpleadoID = V.EmpleadoID

```

SELECT * FROM VW_MOSTRAR_VENTAS

95 %

Resultados Mensajes

	PRODUCTO	CANTIDAD	PRECIO DE VENTA	TOTAL	CATEGORIA	EMPLEADO
1	DURAZNOS ALMIBAR	4	9.29	37.16	ABARROTES GENERAL	DANIEL AGAPITO
2	FILETE DE ATÚN	8	7.40	59.20	ABARROTES GENERAL	DANIEL AGAPITO
3	CROQUETAS	2	220.00	440.00	MASCOTAS PEQUEÑAS Y GRANDES	DANIEL AGAPITO
4	AGUA SAN MATEO	2	1.20	2.40	AGUAS Y BEBIDAS REFRESCANTES	MARK STEPHANO
5	CROQUETAS	4	6.00	24.00	MASCOTAS PEQUEÑAS Y GRANDES	MARK STEPHANO
6	LIMPIA BAÑOS	3	20.00	60.00	LIMPIEZA DEL HOGAR	MARK STEPHANO
7	PAPITAS FRITAS	5	1.20	6.00	GALLETAS, SNACKS Y GOLOSINAS GENIALES	FRANCO ALEXIS
8	PATÉ	5	3.50	17.50	MASCOTAS PEQUEÑAS Y GRANDES	MARK STEPHANO
9	SUAVIZANTE DOWNY	2	42.00	84.00	LIMPIEZA DEL HOGAR	MARK STEPHANO
10	AGUA CIELO	2	8.50	17.00	AGUAS Y BEBIDAS REFRESCANTES	KEVIN ENRIQUE
11	INKA COLA	3	8.00	24.00	AGUAS Y BEBIDAS REFRESCANTES	KEVIN ENRIQUE
12	DESODORANTE BARRA	3	9.59	28.77	HIGIENE Y SALUD PRIORIDAD	KEVIN ENRIQUE
13	CHUPETINES	3	4.70	14.10	GALLETAS, SNACKS Y GOLOSINAS GENIALES	KEVIN ENRIQUE
14	LIMPIA COCINA	1	12.00	12.00	LIMPIEZA DEL HOGAR	JUAN JOSE
15	SUAVIZANTE DOWNY	1	42.00	42.00	LIMPIEZA DEL HOGAR	JUAN JOSE
16	SALSA DE TOMATE NICOLINI	5	2.20	11.00	ABARROTES GENERAL	KEVIN ENRIQUE

9.1.2. Vista - Productos Mayores

```

/*CREAR UNA VISTA QUE PERMITA MOSTRAR EL NOMBRE, PRECIO DE VENTA, PRECIO MENORES O IGUAL A 6 SOLES
ORDENADOS POR NOMBRE DE PRODUCTO*/
CREATE OR ALTER VIEW VW_PRODUCTOS_MAYORES
AS
SELECT Nombre_Producto AS 'PRODUCTO', Precio_Venta AS 'PRECIO' FROM Producto WHERE Precio_Venta >
(SELECT Precio_Venta FROM Producto where Nombre_Producto = 'MILO' )
GO

```

SELECT * FROM VW_PRODUCTOS_MAYORES

95 %

Resultados Mensajes

	PRODUCTO	PRECIO
1	ACEITE VEGETAL PRIMOR	18.70
2	ACEITE VEGETAL BELLS	42.50
3	ACEITE DE OLIVA EXTRA	40.00
4	ACEITE VEGETAL CAPRI	17.70
5	ACEITE "EL OLIVAR"	25.40
6	ACEITE VEGETAL CAPRI	25.00
7	ACEITE VEGETAL BELLS	40.00
8	ACEITE VEGETAL CORAZON	40.00
9	PIÑAS ALMIBAR	21.50
10	ARROZ	155.00
11	ARROZ	185.00
12	AVENA CLÁSICA	40.00
13	LECHE GLORIA	25.00
14	CEREAL TRIGO CHOCOLATE	16.50
15	DESODORANTE LEÑADOR	15.00
16	SHAMPOO	19.00
17	DETERGENTE MARSELLA	42.00
18	DETERGENTE MARSELLA	42.00
19	DETERGENTE ARIEL	42.00
20	DETERGENTE ACE	15.00
21	SUAVIZANTE DOWNY	42.00
22	SUAVIZANTE DOWNY	42.00

✓ Consulta ejecutada correctamente.

9.1.3. Vista – Mostrar Empleado Cargo

```
--CREAR UNA VISTA QUE MUESTRE A LOS EMPLEADOS CON CARGOS
CREATE OR ALTER VIEW VW_MOSTRAR_EMPLEADO_CARGO
AS
SELECT (E.Nombres+' '+E.Apellidos) AS DATOS, C.Tipo AS CARGO
FROM Empleado E INNER JOIN Cargo C
ON C.CargoID = E.CargoID
GO
```

SELECT * FROM VW_MOSTRAR_EMPLEADO_CARGO

95 %

Resultados Mensajes

	DATOS	CARGO
1	DANIEL AGAPITO CURO BERNILLA	EMPLEADO
2	FRANCO ALEXIS SALAZAR SOPLAPUCO	EMPLEADO
3	KEVIN ENRIQUE TEQUEN OBLITAS	EMPLEADO
4	MARK STEPHANO SANCHEZ MEDINA	ADMINISTRADOR
5	JUAN JOSE MEDINA SANCHEZ	EMPLEADO

9.1.4. Vista - Producto Proveedor

```
--CREAR UNA VISTA QUE MUESTRE LOS PRODUCTOS CON SUS RESPECTIVOS PROVEEDORES
CREATE OR ALTER VIEW VW_PRODUCTO_PROVEEDOR
AS
SELECT P.Nombre_Producto AS 'PRODUCTO', PR.Nombre_Proveedor AS 'PROVEEDOR'
FROM Producto P JOIN Proveedor PR
ON PR.ProveedorID = P.ProveedorID
GO
```


SELECT * FROM VW_PRODUCTO_PROVEEDOR

95 %

Resultados Mensajes

	PRODUCTO	PROVEEDOR
1	CALDO DE GALLINA	DISTRIBUIDOR PEPSICO SAC
2	PANQUITA	DISTRIBUIDOR NESTLE SAC
3	SAZONADOR DE ARROZ	DISTRIBUIDOR NESTLE SAC
4	PIMIENTA NEGRA MOLIDA	DISTRIBUIDOR DEKASA SAC
5	CANELA ENTERA	DISTRIBUIDOR NESTLE SAC
6	AJINOMOTO	DISTRIBUIDOR NESTLE SAC
7	ORÉGANO MOLIDA	DISTRIBUIDOR NESTLE SAC
8	COMINO MOLIDO	DISTRIBUIDOR PEPSICO SAC
9	CALDO CONCENTRADA DE TOCINO	DISTRIBUIDOR NESTLE SAC
10	HOJAS DE LAUREL	DISTRIBUIDOR NESTLE SAC
11	HONGO Y LAUREL	DISTRIBUIDOR NESTLE SAC
12	SAL DE MESA MARINA	DISTRIBUIDOR NESTLE SAC
13	SAL ROSADA DE MARAS	DISTRIBUIDOR NESTLE SAC
14	SALSA DE TOMATE	DISTRIBUIDOR NESTLE SAC
15	SALSA DE TOMATE NICOLINI	DISTRIBUIDOR NESTLE SAC
16	SALSA DE TOMATE SECRETA	DISTRIBUIDOR NESTLE SAC
17	VINAGRETA POLLERA	DISTRIBUIDOR NESTLE SAC
18	VINAGRETA MOSTAZA	DISTRIBUIDOR NESTLE SAC
19	CREMA DE AJÍ	DISTRIBUIDOR NESTLE SAC
20	FIDEOS TALLARINES	DISTRIBUIDOR NESTLE SAC
21	FIDEO CODITOS	DISTRIBUIDOR NESTLE SAC
22	SEMOLA	DISTRIBUIDOR NESTLE SAC

Consulta ejecutada correctamente.

9.2.Vistas Encriptadas

```
CREATE OR ALTER VIEW VISTA_EMPLEADOS
WITH ENCRYPTION
AS
SELECT (E.Nombres+' '+E.Apellidos) AS 'DATOS GENERALES',
DI.Nombre_Identificacion AS 'DOCUMENTO', C.Tipo AS 'CARGO'
FROM Empleado E INNER JOIN Documento_Identidad DI
ON E.Documento_IdentidadID = DI.Documento_IdentidadID INNER JOIN Cargo C
ON C.CargoID = E.CargoID
GO
```

EXEC sp_helptext VISTA_EMPLEADOS

95 %

Mensajes

El texto para el objeto 'VISTA_EMPLEADOS' está cifrado.

Hora de finalización: 2022-07-17T20:11:25.1468842-05:00

CAPÍTULO 10: TRANSACCIONES

