**UTN-FRGP/TUP FINAL LABORATORIO DE COMPUTACIÓN III**

|  |  |  |
| --- | --- | --- |
| Nombre y Apellido: Herrera Emanuel | | |
| DNI: 37866505 | | Legajo: 23402 |
| Fecha: | | EXAMEN FINAL Tomo: Folio: |
| CALIFICACIÓN |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COMPETENCIAS ESPECIFICAS  EVALUADAS | **0**  **No lo**  **hace** | **2**  **Lo hace con errores sustanciales** | **4**  **Lo hace con errores que deben mejorarse** | **6**  **Lo hace correctamente**  **(Aceptable)** | **8**  **Lo hace muy bien** | **10**  **Lo hace excelente mente** |
| Aplicar vocabulario y terminología propia de la materia. |  |  |  |  |  |  |
| Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas. |  |  |  |  |  |  |
| Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores. |  |  |  |  |  |  |
| Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma |  |  |  |  |  |  |
| Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje. |  |  |  |  |  |  |

**UTN-FRGP/TUP FINAL LABORATORIO DE COMPUTACIÓN III**

|  |  |  |
| --- | --- | --- |
| Nombre y Apellido: | | |
| DNI: | | Legajo: |
| Fecha: | | EXAMEN FINAL Tomo: Folio: |
| CALIFICACIÓN |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COMPETENCIAS ESPECIFICAS  EVALUADAS | **0**  **No lo**  **hace** | **2**  **Lo hace con errores sustanciales** | **4**  **Lo hace con errores que deben mejorarse** | **6**  **Lo hace correctamente**  **(Aceptable)** | **8**  **Lo hace muy bien** | **10**  **Lo hace excelente mente** |
| Aplicar vocabulario y terminología propia de la materia. |  |  |  |  |  |  |
| Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas. |  |  |  |  |  |  |
| Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores. |  |  |  |  |  |  |
| Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma |  |  |  |  |  |  |
| Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje. |  |  |  |  |  |  |

**UTN-FRGP/TUP FINAL LABORATORIO DE COMPUTACIÓN III**

|  |  |  |
| --- | --- | --- |
| Nombre y Apellido: | | |
| DNI: | | Legajo: |
| Fecha: | | EXAMEN FINAL Tomo: Folio: |
| CALIFICACIÓN |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COMPETENCIAS ESPECIFICAS  EVALUADAS | **0**  **No lo**  **hace** | **2**  **Lo hace con errores sustanciales** | **4**  **Lo hace con errores que deben mejorarse** | **6**  **Lo hace correctamente**  **(Aceptable)** | **8**  **Lo hace muy bien** | **10**  **Lo hace excelente mente** |
| Aplicar vocabulario y terminología propia de la materia. |  |  |  |  |  |  |
| Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas. |  |  |  |  |  |  |
| Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores. |  |  |  |  |  |  |
| Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma |  |  |  |  |  |  |
| Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje. |  |  |  |  |  |  |

**UTN-FRGP/TUP FINAL LABORATORIO DE COMPUTACIÓN III**

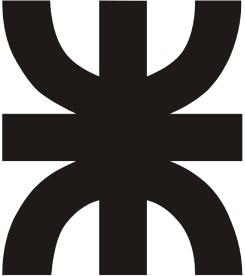
|  |  |  |
| --- | --- | --- |
| Nombre y Apellido: | | |
| DNI: | | Legajo: |
| Fecha: | | EXAMEN FINAL Tomo: Folio: |
| CALIFICACIÓN |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| COMPETENCIAS ESPECIFICAS  EVALUADAS | **0**  **No lo**  **hace** | **2**  **Lo hace con errores sustanciales** | **4**  **Lo hace con errores que deben mejorarse** | **6**  **Lo hace correctamente**  **(Aceptable)** | **8**  **Lo hace muy bien** | **10**  **Lo hace excelente mente** |
| Aplicar vocabulario y terminología propia de la materia. |  |  |  |  |  |  |
| Aplicar los resultados del autoaprendizaje, actualización e investigación en la resolución de problemas. |  |  |  |  |  |  |
| Resolver situaciones problemáticas planteadas como proyectos interdisciplinares integradores. |  |  |  |  |  |  |
| Desempeñar las actividades dentro del equipo con responsabilidad individual, grupal, ética y social, en tiempo y forma |  |  |  |  |  |  |
| Participar activamente en el aula (virtual o presencial), realizando aportes a la comunidad de aprendizaje. |  |  |  |  |  |  |

UNIVERSIDAD TECNOLÓGICA NACIONAL - FRGP

# Laboratorio de Computación III

# Base de Datos para un E-commerce de electrodomésticos



**Autores:**

Ariel Sebastián Tapia

Cesar Ignacio Aquino Tello

Germán Vázquez

Emanuel Herrera

**Profesores:**

Ariel Hernán Herrera

Diego Ariel Español

**2021 - Primer Cuatrimestre**

ÍNDICE

[INTRODUCCIÓN 1](#_Toc77279793)

[D.E.R 2](#_Toc77279794)

[CÓDIGO 3](#_Toc77279795)

[STORES PROCEDURES 12](#_Toc77279796)

[TRIGGERS 33](#_Toc77279797)

[INSERTS 37](#_Toc77279798)

[CONSULTAS 45](#_Toc77279799)

# INTRODUCCIÓN

En el presente trabajo se desarrolló una base de datos SQL con 16 tablas para una cadena de negocios que se dedica a la comercialización de artículos electrónicos. Esta no solo cuenta con locales físicos, también posee con un sitio web (ecommerce).

De esta manera, el proyecto cuenta con tres tipos de usuarios:

Usuarios: Esta es una tabla general que posee datos en común entre los distintos tipos de usuarios.

* Personal: encargados de la recepción de mercadería.
* Clientes: realizan pedidos.
* Proveedores: no poseen acceso a una cuenta, son registrados por el administrador y se utiliza esta tabla para saber de dónde provienen los artículos.

El manejo de los artículos y la modificación de stock se hace a través de 6 tablas (cabe mencionar que un artículo solo puede provenir de un único proveedor):

* Artículos: Cuenta con toda la información relevante de un artículo específico.
* Artículos\_por\_Proveedor: En esta tabla se especifica el Cuit del proveedor encargado del artículo, además de su stock, código y precio unitario.
* Recepcion\_Artículos: Se reciben pedidos de un proveedor.
* Detalles\_Recepción\_Artículos: Es indicado el código del artículo recibido, la cantidad y el precio pagado por la mercadería. (Se valida que el artículo ingresado sea del mismo proveedor indicado en la recepción general).
* Ventas: Se establece toda la información necesaria como la fecha, método de pago e información sobre el cliente.
* Detalles\_Ventas: Indica la cantidad de stock vendida de cada artículo (Se valida que la cantidad ingresada sea menor o igual al stock disponible)

El resto de tablas son del tipo complementarias:

1. Perfiles
2. Provincias
3. Medios de pago
4. Marcas
5. Estados
6. Categorías

# D.E.R

Diagrama

Descripción generada automáticamente

# CÓDIGO

/\* CREA DB \*/

CREATE DATABASE Lab3TIF

ON

(

NAME = Lab3TIF\_dat,

FILENAME = 'C:\Trabajo Integrador Final\Lab3TIF.mdf'

)

GO

USE Lab3TIF

GO

**/\* TABLA ESTADOS \*/**

CREATE TABLE Estados(

EST\_codigo INT IDENTITY(1,1) NOT NULL,

EST\_nombre VARCHAR(25) NOT NULL,

CONSTRAINT PK\_Estados PRIMARY KEY (EST\_codigo)

)

GO

**/\* TABLA PERFILES \*/**

CREATE TABLE Perfiles(

PER\_codigo INT IDENTITY(1,1) NOT NULL,

PER\_nombre VARCHAR(25) NOT NULL,

CONSTRAINT PK\_Perfiles PRIMARY KEY (PER\_codigo)

)

GO

**/\* TABLA PROVINCIAS \*/**

CREATE TABLE Provincias(

PROV\_codigo INT IDENTITY(1,1) NOT NULL,

PROV\_nombre VARCHAR(50) NOT NULL,

CONSTRAINT PK\_Provincias PRIMARY KEY (PROV\_codigo)

)

GO

**/\* MEDIOS DE PAGO \*/**

CREATE TABLE Medios\_de\_Pago(

MDP\_codigo INT IDENTITY(1,1) NOT NULL,

MDP\_nombre VARCHAR(50) NOT NULL,

MDP\_otros\_detalles VARCHAR(100) NULL,

MDP\_estado\_cod INT NOT NULL,

CONSTRAINT PK\_Medios\_de\_pago PRIMARY KEY (MDP\_codigo),

CONSTRAINT FK\_Medios\_de\_pago\_Estados FOREIGN KEY (MDP\_estado\_cod) REFERENCES Estados(EST\_codigo)

)

GO

**/\* TABLA USUARIOS \*/**

CREATE TABLE Usuarios(

USU\_id INT IDENTITY(1,1) NOT NULL,

USU\_perfil\_cod INT NOT NULL,

USU\_nombre VARCHAR(50) NOT NULL,

USU\_apellido VARCHAR(50) NOT NULL,

USU\_telefono VARCHAR(50) NOT NULL,

USU\_email VARCHAR(50) NOT NULL,

USU\_direccion VARCHAR(50) NOT NULL,

USU\_ciudad VARCHAR(50) NOT NULL,

USU\_provincia\_cod INT NOT NULL,

USU\_ruta\_img VARCHAR(100) NOT NULL,

USU\_estado\_cod INT NULL,

CONSTRAINT PK\_Usuarios PRIMARY KEY (USU\_id),

CONSTRAINT FK\_Usuarios\_Perfiles FOREIGN KEY (USU\_perfil\_cod) REFERENCES Perfiles(PER\_codigo),

CONSTRAINT FK\_Usuarios\_Estados FOREIGN KEY (USU\_estado\_cod) REFERENCES Estados(EST\_codigo),

CONSTRAINT FK\_Usuarios\_Provincias FOREIGN KEY (USU\_provincia\_cod) REFERENCES Provincias(PROV\_codigo)

)

GO

**/\* TABLA PROVEEDORES \*/**

CREATE TABLE Proveedores(

PROV\_cuit INT NOT NULL,

PROV\_usu\_id INT NOT NULL,

PROV\_razon\_social VARCHAR(50) NOT NULL,

PROV\_nombre\_contacto VARCHAR(50) NOT NULL,

CONSTRAINT PK\_Proveedores PRIMARY KEY (PROV\_cuit),

CONSTRAINT FK\_Proveedores\_Usuarios FOREIGN KEY (PROV\_usu\_id) REFERENCES Usuarios(USU\_id),

)

GO

**/\* TABLA PERSONAL \*/**

CREATE TABLE Personal(

PER\_codigo INT IDENTITY(1,1) NOT NULL,

PER\_usu\_id INT NOT NULL,

PER\_dni VARCHAR(25) NOT NULL,

PER\_username VARCHAR(50) NOT NULL,

PER\_contrasenia VARCHAR(50) NOT NULL,

CONSTRAINT PK\_Personal PRIMARY KEY (PER\_codigo),

CONSTRAINT FK\_Personal\_Usuarios FOREIGN KEY (PER\_usu\_id) REFERENCES Usuarios(USU\_id),

)

GO

**/\* TABLA CLIENTES \*/**

CREATE TABLE Clientes(

CLI\_cuit INT NOT NULL,

CLI\_usu\_id INT NOT NULL,

CLI\_dni VARCHAR(25) NOT NULL,

CLI\_username VARCHAR(50) NOT NULL,

CLI\_contrasenia VARCHAR(50) NOT NULL,

CONSTRAINT PK\_Clientes PRIMARY KEY (CLI\_cuit),

CONSTRAINT FK\_Clientes\_Usuarios FOREIGN KEY (CLI\_usu\_id) REFERENCES Usuarios(USU\_id),

)

GO

**/\* TABLA MARCAS \*/**

CREATE TABLE Marcas(

MAR\_codigo INT IDENTITY(1,1) NOT NULL,

MAR\_nombre VARCHAR(255) NOT NULL,

MAR\_descripcion VARCHAR(100) NULL,

MAR\_ruta\_imagen VARCHAR(100) NULL,

MAR\_estado\_cod INT NULL,

CONSTRAINT PK\_Marcas PRIMARY KEY (MAR\_codigo),

CONSTRAINT FK\_Marcas\_Estado FOREIGN KEY (MAR\_estado\_cod) REFERENCES Estados(EST\_codigo)

)

GO

**/\* TABLA CATEGORIAS \*/**

CREATE TABLE Categorias(

CAT\_codigo INT IDENTITY(1,1) NOT NULL,

CAT\_nombre VARCHAR(50) NOT NULL,

CAT\_descripcion VARCHAR(100) NULL,

CAT\_ruta\_imagen VARCHAR(100) NULL,

CAT\_estado\_cod INT NULL,

CONSTRAINT PK\_Categorias PRIMARY KEY (CAT\_codigo),

CONSTRAINT FK\_Categorias\_Estado FOREIGN KEY (CAT\_estado\_cod) REFERENCES Estados(EST\_codigo)

)

GO

**/\* TABLA ARTICULOS \*/**

CREATE TABLE Articulos(

ART\_codigo INT IDENTITY(1,1) NOT NULL,

ART\_marca\_cod INT NOT NULL,

ART\_categoria\_cod INT NOT NULL,

ART\_nombre VARCHAR(50) NOT NULL,

ART\_descripcion VARCHAR(150) NOT NULL,

ART\_punto\_pedido INT NOT NULL,

ART\_precio\_lista DECIMAL(18,2) DEFAULT 0.00,

ART\_ruta\_imagen VARCHAR(100) NULL,

ART\_estado\_cod INT NULL,

CONSTRAINT PK\_Articulos PRIMARY KEY (ART\_codigo),

CONSTRAINT FK\_Articulos\_Marcas FOREIGN KEY (ART\_marca\_cod) REFERENCES Marcas(MAR\_codigo),

CONSTRAINT FK\_Articulos\_Categorias FOREIGN KEY (ART\_categoria\_cod) REFERENCES Categorias(CAT\_codigo),

CONSTRAINT FK\_Articulos\_Estado FOREIGN KEY (ART\_estado\_cod) REFERENCES Estados(EST\_codigo)

)

GO

**/\* TABLA ARTICULOS POR PROVEEDORES \*/**

CREATE TABLE Articulos\_por\_Proveedor(

AXP\_codigo INT IDENTITY(1,1) NOT NULL,

AXP\_prov\_cuit INT NOT NULL,

AXP\_articulo\_cod INT NOT NULL,

AXP\_stock\_actual INT NOT NULL DEFAULT 0,

AXP\_stock\_anterior INT NOT NULL DEFAULT 0,

AXP\_entrada INT NOT NULL DEFAULT 0,

AXP\_salida INT NOT NULL DEFAULT 0,

AXP\_precio\_unitario DECIMAL(18,2) DEFAULT (0.00),

CONSTRAINT PK\_Articulos\_por\_Proveedor PRIMARY KEY (AXP\_codigo),

CONSTRAINT FK\_Articulos\_por\_Proveedor\_Articulos FOREIGN KEY (AXP\_articulo\_cod) REFERENCES Articulos(ART\_codigo),

CONSTRAINT FK\_Articulos\_por\_Proveedor\_Proveedores FOREIGN KEY (AXP\_prov\_cuit) REFERENCES Proveedores(PROV\_cuit),

CONSTRAINT UQ\_Articulos\_por\_Proveedor\_AXP\_articulo\_cod UNIQUE (AXP\_articulo\_cod)

)

GO

**/\* TABLA VENTAS\*/**

CREATE TABLE Ventas(

VEN\_codigo INT IDENTITY(1,1) NOT NULL,

VEN\_cli\_cuit INT NOT NULL,

VEN\_medio\_pago\_cod INT NOT NULL,

VEN\_fecha SMALLDATETIME NOT NULL,

VEN\_fecha\_requerida SMALLDATETIME NULL,

VEN\_fecha\_envio SMALLDATETIME NULL,

VEN\_total\_facturado DECIMAL(18,2) DEFAULT (0.00),

VEN\_codigo\_estado INT NULL,

CONSTRAINT PK\_Ventas PRIMARY KEY (VEN\_codigo),

CONSTRAINT FK\_Ventas\_Clientes FOREIGN KEY (VEN\_cli\_cuit) REFERENCES Clientes(CLI\_cuit),

CONSTRAINT FK\_Ventas\_Medios\_de\_Pago FOREIGN KEY (VEN\_medio\_pago\_cod) REFERENCES Medios\_de\_Pago(MDP\_codigo),

CONSTRAINT FK\_Ventas\_codigo\_estado FOREIGN KEY (VEN\_codigo\_estado) REFERENCES Estados(EST\_codigo)

)

GO

**/\* TABLA DETALLES DE VENTA \*/**

CREATE TABLE DetalleVenta(

DTV\_venta\_codigo INT NOT NULL,

DTV\_prov\_cuit INT NOT NULL,

DTV\_articulo\_cod INT NOT NULL,

DTV\_cantidad\_unidades INT NULL,

DTV\_precio\_unitario DECIMAL(18,2) DEFAULT (0.00),

CONSTRAINT PK\_DetalleVenta PRIMARY KEY (DTV\_venta\_codigo, DTV\_articulo\_cod),

CONSTRAINT FK\_DetalleVenta\_Ventas FOREIGN KEY (DTV\_venta\_codigo) REFERENCES Ventas(VEN\_codigo),

CONSTRAINT FK\_DetalleVenta\_Proveedores FOREIGN KEY (DTV\_prov\_cuit) REFERENCES Proveedores(PROV\_cuit),

CONSTRAINT FK\_DetalleVenta\_Articulos FOREIGN KEY (DTV\_articulo\_cod) REFERENCES Articulos(ART\_codigo)

)

GO

**/\* TABLA RECEPCION ARTICULOS\*/**

CREATE TABLE Recepcion\_Articulos(

RECART\_codigo INT IDENTITY(1,1) NOT NULL,

RECART\_per\_cod INT NOT NULL,

RECART\_prov\_cuit INT NOT NULL,

RECART\_fecha SMALLDATETIME NOT NULL,

RECART\_total\_facturado DECIMAL(18,2) DEFAULT (0.00),

RECART\_estado\_cod INT NOT NULL,

CONSTRAINT PK\_Recepcion\_Articulos PRIMARY KEY (RECART\_codigo),

CONSTRAINT FK\_Recepcion\_Articulos\_Personal FOREIGN KEY (RECART\_per\_cod) REFERENCES Personal(PER\_codigo),

CONSTRAINT FK\_Recepcion\_Articulos\_Proveedores FOREIGN KEY (RECART\_prov\_cuit) REFERENCES Proveedores(PROV\_cuit),

CONSTRAINT FK\_Recepcion\_Articulos\_Estados FOREIGN KEY (RECART\_estado\_cod) REFERENCES Estados(EST\_codigo)

)

GO

**/\* TABLA DETALLES RECEPCION ARTICULOS \*/**

CREATE TABLE DetalleRecepcion\_Articulos(

DRECART\_codigo INT NOT NULL,

DRECART\_prov\_cuit INT NOT NULL,

DRECART\_articulo\_cod INT NOT NULL,

DRECART\_cantidad\_unidades INT NULL,

DRECART\_precio\_unitario DECIMAL(18,2) DEFAULT (0.00),

DRECART\_numero\_de\_linea INT IDENTITY (1,1),

CONSTRAINT PK\_DetalleRecepcion\_Articulos PRIMARY KEY (DRECART\_codigo, DRECART\_numero\_de\_linea),

CONSTRAINT FK\_DetalleRecepcion\_Articulos\_Recepcion\_Articulos FOREIGN KEY (DRECART\_codigo)

REFERENCES Recepcion\_Articulos(RECART\_codigo),

CONSTRAINT FK\_DetalleRecepcion\_Articulos\_Articulos FOREIGN KEY (DRECART\_articulo\_cod) REFERENCES Articulos(ART\_codigo)

)

GO

## STORES PROCEDURES

USE Lab3TIF

GO

--AGREGAR CLIENTES

CREATE PROCEDURE spAgregarCliente

@CLI\_cuit INT,

@CLI\_usu\_id INT,

@CLI\_dni VARCHAR(25),

@CLI\_username VARCHAR(50),

@CLI\_contrasenia VARCHAR(50)

AS

DECLARE @perfil int

SELECT @perfil=USU\_perfil\_cod FROM Usuarios WHERE USU\_id=@CLI\_usu\_id

IF @perfil=2

BEGIN

INSERT INTO Clientes(CLI\_cuit,CLI\_usu\_id,CLI\_dni,CLI\_username,CLI\_contrasenia)

SELECT @CLI\_cuit,@CLI\_usu\_id,@CLI\_dni,@CLI\_username,@CLI\_contrasenia

END

ELSE PRINT 'Tipo de usuario incorrecto'

GO

-- AGREGAR PERSONAL

CREATE PROCEDURE spAgregarPersonal

@PER\_usu\_id INT,

@PER\_dni VARCHAR(25),

@PER\_username VARCHAR(50),

@PER\_contrasenia VARCHAR(50)

AS

DECLARE @perfil int

SELECT @perfil=USU\_perfil\_cod FROM Usuarios WHERE USU\_id=@PER\_usu\_id

IF @perfil=1

BEGIN

INSERT INTO Personal(PER\_usu\_id,PER\_dni,PER\_username,PER\_contrasenia)

SELECT @PER\_usu\_id,@PER\_dni,@PER\_username,@PER\_contrasenia

END

ELSE PRINT 'Tipo de usuario incorrecto'

GO

--AGREGAR DETALLEVENTA

CREATE PROCEDURE spAgregarDetalleVenta

@DTV\_venta\_codigo INT,

@DTV\_articulo\_cod INT,

@DTV\_cantidad\_unidades INT

AS

-- Se obtiene el cuit proveniente del proveedor que despacha el producto ingresado, 1 producto solo puede ser dado por un único proveedor.

DECLARE @cuitdelproveedor int

SELECT @cuitdelproveedor=AXP\_prov\_cuit from Articulos\_por\_Proveedor WHERE AXP\_articulo\_cod=@DTV\_articulo\_cod

INSERT INTO DetalleVenta(DTV\_venta\_codigo,DTV\_prov\_cuit,DTV\_articulo\_cod,DTV\_cantidad\_unidades)

SELECT @DTV\_venta\_codigo,@cuitdelproveedor,@DTV\_articulo\_cod,@DTV\_cantidad\_unidades

GO

-- AGREGAR PROVINCIAS

CREATE PROCEDURE spAgregarProvincias(

@PROV\_nombre VARCHAR(50)

)

AS

INSERT INTO Provincias(PROV\_nombre)

VALUES(@PROV\_nombre)

GO

-- EDITAR PROVINCIAS

CREATE PROCEDURE spEditarProvincias(

@PROV\_nombre VARCHAR(50),

@PROV\_codigo INT

)

AS

UPDATE Provincias

SET

PROV\_nombre=@PROV\_nombre

WHERE

PROV\_codigo= @PROV\_codigo

GO

-- AGREGAR ESTADOS

CREATE PROCEDURE spAgregarEstados(

@EST\_nombre VARCHAR(25)

)

AS

INSERT INTO Estados(EST\_nombre)

VALUES(@EST\_nombre)

GO

-- EDITAR ESTADOS

CREATE PROCEDURE spEditarEstados(

@EST\_nombre VARCHAR(50),

@EST\_codigo INT

)

AS

UPDATE Estados

SET

EST\_nombre=@EST\_nombre

WHERE

EST\_codigo= @EST\_codigo

GO

-- AGREGAR PERFILES

CREATE PROCEDURE spAgregarPerfiles(

@PER\_nombre VARCHAR(25)

)

AS

INSERT INTO Perfiles(PER\_nombre)

VALUES(@PER\_nombre)

GO

-- EDITAR PERFILES

CREATE PROCEDURE spEditarPerfiles(

@PER\_nombre VARCHAR(50),

@PER\_codigo INT

)

AS

UPDATE Perfiles

SET

PER\_nombre=@PER\_nombre

WHERE

PER\_codigo= @PER\_codigo

GO

-- AGREGAR MEDIOS DE PAGO

CREATE PROCEDURE spAgregarMedios\_de\_Pago(

@MDP\_nombre VARCHAR(50),

@MDP\_otros\_detalles VARCHAR(100)

)

AS

INSERT INTO Medios\_de\_Pago(MDP\_nombre,MDP\_otros\_detalles,MDP\_estado\_cod)

VALUES(@MDP\_nombre,@MDP\_otros\_detalles,2)

GO

-- EDITAR MEDIOS DE PAGO

CREATE PROCEDURE spEditarMedios\_de\_Pago(

@MDP\_codigo INT,

@MDP\_nombre VARCHAR(50),

@MDP\_otros\_detalles VARCHAR(100),

@MDP\_estado\_cod INT

)

AS

UPDATE Medios\_de\_Pago

SET

MDP\_nombre=@MDP\_nombre,

MDP\_otros\_detalles=@MDP\_otros\_detalles,

MDP\_estado\_cod=@MDP\_estado\_cod

WHERE

MDP\_codigo= @MDP\_codigo

GO

-- AGREGAR RECEPCION DE ARTICULO

CREATE PROCEDURE spAgregarRecepcion\_Articulos(

@RECART\_per\_cod INT,

@RECART\_prov\_cuit INT

)

AS

INSERT INTO Recepcion\_Articulos(RECART\_per\_cod,RECART\_prov\_cuit,RECART\_fecha,RECART\_estado\_cod)

VALUES(@RECART\_per\_cod,@RECART\_prov\_cuit,getdate(),2)

GO

-- EDITAR RECEPCION DE ARTICULO

CREATE PROCEDURE spEditarRecepcion\_Articulos(

@RECART\_codigo INT,

@RECART\_per\_cod INT,

@RECART\_prov\_cuit INT,

@RECART\_fecha SMALLDATETIME,

@RECART\_estado\_cod INT

)

AS

UPDATE Recepcion\_Articulos

SET

RECART\_per\_cod=@RECART\_per\_cod,

RECART\_prov\_cuit=@RECART\_prov\_cuit,

RECART\_fecha=@RECART\_fecha,

RECART\_estado\_cod=@RECART\_estado\_cod

WHERE

RECART\_codigo=@RECART\_codigo

GO

-- AGREGAR USUARIO

CREATE PROCEDURE spAgregarUsuarios(

@USU\_perfil\_cod INT,

@USU\_nombre VARCHAR(50),

@USU\_apellido VARCHAR(50),

@USU\_telefono VARCHAR(50),

@USU\_email VARCHAR(50),

@USU\_direccion VARCHAR(50),

@USU\_ciudad VARCHAR(50),

@USU\_provincia\_cod INT,

@USU\_ruta\_img VARCHAR(100)

)

AS

INSERT INTO Usuarios(USU\_perfil\_cod,USU\_nombre,USU\_apellido,USU\_telefono,USU\_email,USU\_direccion,USU\_ciudad,USU\_provincia\_cod,USU\_ruta\_img,USU\_estado\_cod)

VALUES(@USU\_perfil\_cod,@USU\_nombre,@USU\_apellido,@USU\_telefono,@USU\_email,@USU\_direccion,@USU\_ciudad,@USU\_provincia\_cod,@USU\_ruta\_img,2)

GO

-- EDITAR USUARIO

CREATE PROCEDURE spEditarUsuarios(

@USU\_id INT,

@USU\_perfil\_cod INT,

@USU\_nombre VARCHAR(50),

@USU\_apellido VARCHAR(50),

@USU\_telefono VARCHAR(50),

@USU\_email VARCHAR(50),

@USU\_direccion VARCHAR(50),

@USU\_ciudad VARCHAR(50),

@USU\_provincia\_cod INT,

@USU\_ruta\_img VARCHAR(100),

@USU\_estado\_cod INT

)

AS

UPDATE USUARIOS

SET USU\_perfil\_cod=@USU\_perfil\_cod,

USU\_nombre=@USU\_nombre,

USU\_apellido=@USU\_apellido,

USU\_telefono=@USU\_telefono,

USU\_email=@USU\_email,

USU\_direccion=@USU\_direccion,

USU\_ciudad=@USU\_ciudad,

USU\_provincia\_cod=@USU\_provincia\_cod,

USU\_ruta\_img=@USU\_ruta\_img,

USU\_estado\_cod=@USU\_estado\_cod

WHERE USU\_id=@USU\_id

GO

-- BAJA USUARIO

CREATE PROCEDURE spBajaUsuario

(

@USU\_id INT,

@USU\_estado\_cod INT

)

AS

UPDATE USUARIOS

SET

USU\_estado\_cod=@USU\_estado\_cod

WHERE

USU\_id=@USU\_id

GO

-- AGREGAR MARCA

CREATE PROCEDURE spAgregarMarca

(

@MAR\_nombre VARCHAR(255),

@MAR\_descripcion VARCHAR,

@MAR\_ruta\_imagen VARCHAR,

@MAR\_estado\_cod INT

)

AS

INSERT INTO Marcas

(

MAR\_nombre,

MAR\_descripcion,

MAR\_ruta\_imagen,

MAR\_estado\_cod

)

VALUES

(

@MAR\_nombre,

@MAR\_descripcion,

@MAR\_ruta\_imagen,

@MAR\_estado\_cod

)

GO

-- MODIFICAR MARCA

CREATE PROCEDURE spModificarMarca

(

@MAR\_codigo INT,

@MAR\_nombre VARCHAR(255),

@MAR\_descripcion VARCHAR,

@MAR\_ruta\_imagen VARCHAR,

@MAR\_estado\_cod INT

)

AS

UPDATE Marcas

SET

MAR\_nombre=@MAR\_nombre,

MAR\_descripcion=@MAR\_descripcion,

MAR\_ruta\_imagen=@MAR\_ruta\_imagen,

MAR\_estado\_cod=@MAR\_estado\_cod

WHERE

MAR\_codigo= @MAR\_codigo

GO

-- DAR DE BAJA MARCA

CREATE PROCEDURE spBajaMarca

(

@MAR\_codigo INT,

@MAR\_estado\_cod INT

)

AS

UPDATE Marcas

SET

MAR\_estado\_cod=@MAR\_estado\_cod

WHERE

MAR\_codigo= @MAR\_codigo

GO

-- AGREGAR CATEGORIA

CREATE PROCEDURE spAgregarCategoria

(

@CAT\_nombre VARCHAR(255),

@CAT\_descripcion VARCHAR(255),

@CAT\_ruta\_imagen VARCHAR(255),

@CAT\_estado\_cod INT

)

AS

INSERT INTO Categorias

(

CAT\_nombre,

CAT\_descripcion,

CAT\_ruta\_imagen,

CAT\_estado\_cod

)

VALUES

(

@CAT\_nombre,

@CAT\_descripcion,

@CAT\_ruta\_imagen,

@CAT\_estado\_cod

)

GO

-- MODIFICAR CATEGORIA

CREATE PROCEDURE spModificarCategoria

(

@CAT\_codigo INT,

@CAT\_nombre VARCHAR(255),

@CAT\_descripcion VARCHAR(255),

@CAT\_ruta\_imagen VARCHAR(255),

@CAT\_estado\_cod INT

)

AS

UPDATE Categorias

SET

CAT\_nombre=@CAT\_nombre,

CAT\_descripcion=@CAT\_descripcion,

CAT\_ruta\_imagen=@CAT\_ruta\_imagen,

CAT\_estado\_cod=@CAT\_estado\_cod

WHERE

CAT\_codigo=@CAT\_codigo

GO

-- DAR DE BAJA CATEGORIA

CREATE PROCEDURE spBajaCategoria

(

@CAT\_codigo INT,

@CAT\_estado\_cod INT

)

AS

UPDATE Categorias

SET

CAT\_estado\_cod=@CAT\_estado\_cod

WHERE

CAT\_codigo=@CAT\_codigo

GO

-- AGREGAR ARTICULO

CREATE PROCEDURE spAgregarArticulo

(

@ART\_marca\_cod INT,

@ART\_categoria\_cod INT,

@ART\_nombre VARCHAR(255),

@ART\_descripcion VARCHAR(255),

@ART\_punto\_pedido INT,

@ART\_precio\_lista DECIMAL(18,2),

@ART\_ruta\_imagen VARCHAR(255),

@ART\_estado\_cod INT

)

AS

INSERT INTO Articulos

(

ART\_marca\_cod,

ART\_categoria\_cod,

ART\_nombre,

ART\_descripcion,

ART\_punto\_pedido,

ART\_precio\_lista,

ART\_ruta\_imagen,

ART\_estado\_cod

)

VALUES

(

@ART\_marca\_cod,

@ART\_categoria\_cod,

@ART\_nombre,

@ART\_descripcion,

@ART\_punto\_pedido,

@ART\_precio\_lista,

@ART\_ruta\_imagen,

@ART\_estado\_cod

)

GO

-- MODIFICAR ARTICULO

CREATE PROCEDURE spModificarArticulo

(

@ART\_codigo INT,

@ART\_marca\_cod INT,

@ART\_categoria\_cod INT,

@ART\_nombre VARCHAR(255),

@ART\_descripcion VARCHAR(255),

@ART\_punto\_pedido INT,

@ART\_precio\_lista DECIMAL(18,2),

@ART\_ruta\_imagen VARCHAR(255),

@ART\_estado\_cod INT

)

AS

UPDATE Articulos

SET

ART\_marca\_cod=@ART\_marca\_cod,

ART\_categoria\_cod=@ART\_categoria\_cod,

ART\_nombre=@ART\_nombre,

ART\_descripcion=@ART\_descripcion,

ART\_punto\_pedido=@ART\_punto\_pedido,

ART\_precio\_lista=@ART\_precio\_lista,

ART\_ruta\_imagen=@ART\_ruta\_imagen,

ART\_estado\_cod=@ART\_estado\_cod

WHERE ART\_codigo=@ART\_codigo

GO

-- DAR DE BAJA ARTICULO

CREATE PROCEDURE spBajaArticulo (

@ART\_codigo INT,

@ART\_estado\_cod INT

)

AS

UPDATE Articulos

SET

ART\_estado\_cod = @ART\_estado\_cod

WHERE

ART\_codigo = @ART\_codigo

GO

-- ACTUALIZA ARTICULOS\_POR\_PROVEEDOR EN CASO DE REALIZARSE UN

-- CONTROL DE INVENTARIOS --NO ES LO MISMO QUE UN TRIGGER--

CREATE PROCEDURE spActualizarStock

(

@AXP\_prov\_cuit INT,

@AXP\_articulo\_cod INT,

@AXP\_entrada INT,

@AXP\_salida INT,

@AXP\_precio\_unitario DECIMAL(18,2)

)

AS

IF EXISTS (SELECT \* FROM Articulos\_por\_Proveedor WHERE AXP\_prov\_cuit=@AXP\_prov\_cuit AND AXP\_articulo\_cod=@AXP\_articulo\_cod)

BEGIN

IF EXISTS (SELECT \* FROM Articulos\_por\_Proveedor WHERE AXP\_stock\_actual >= @AXP\_salida)

BEGIN

UPDATE Articulos\_por\_Proveedor

SET

AXP\_entrada=@AXP\_entrada,

AXP\_salida=@AXP\_salida,

AXP\_precio\_unitario =@AXP\_precio\_unitario ,

AXP\_stock\_anterior=AXP\_stock\_actual,

AXP\_stock\_actual=AXP\_stock\_actual+@AXP\_entrada-@AXP\_salida

WHERE AXP\_prov\_cuit=@AXP\_prov\_cuit

AND AXP\_articulo\_cod=@AXP\_articulo\_cod

END

END

ELSE

BEGIN

INSERT INTO Articulos\_por\_Proveedor(AXP\_prov\_cuit,AXP\_articulo\_cod,AXP\_precio\_unitario ,AXP\_stock\_actual,AXP\_entrada)

VALUES(@AXP\_prov\_cuit,@AXP\_articulo\_cod,@AXP\_precio\_unitario ,@AXP\_entrada,@AXP\_entrada)

END

GO

CREATE PROCEDURE spAgregarEstado

(

@codigoestado int ,

@nombreestado varchar (25)

)

AS

INSERT INTO Estados(EST\_codigo,EST\_nombre)

VALUES(@codigoestado,@nombreestado)

GO

CREATE PROCEDURE spAgregarProveedor

(

@cuit int,

@usuarioid int,

@razonsocial varchar (50),

@nombredecontacto varchar (50)

)

AS

DECLARE @perfil int

SELECT @perfil=USU\_perfil\_cod FROM Usuarios WHERE USU\_id=@usuarioid

IF @perfil=3

BEGIN

INSERT INTO Proveedores(PROV\_cuit,PROV\_usu\_id,PROV\_razon\_social,PROV\_nombre\_contacto)

VALUES(@cuit,@usuarioid,@razonsocial,@nombredecontacto)

END

ELSE PRINT 'Tipo de usuario incorrecto'

GO

CREATE PROCEDURE spAgregarVenta

(

@cuitcliente int,

@mediodepago int,

@codigodeestado int

)

AS

DECLARE @date smalldatetime

SET @date=getdate()

INSERT INTO Ventas(VEN\_cli\_cuit,VEN\_medio\_pago\_cod,VEN\_fecha,VEN\_fecha\_requerida,VEN\_fecha\_envio,VEN\_codigo\_estado)

VALUES(@cuitcliente ,@mediodepago,@date,@date,null,@codigodeestado )

GO

CREATE PROCEDURE spAgregarDetalleRecepcionDeArticulos

(

@codigo int,

@codigodearticulo int,

@cantidaddeunidades int,

@preciounitario decimal (18,2)

)

AS

-- Se obtiene el cuit proveniente del producto ingresado, 1 producto solo puede ser dado por un único proveedor.

DECLARE @cuitdeproveedor int,@cuitrecepcion int

SELECT @cuitdeproveedor=AXP\_prov\_cuit from Articulos\_por\_Proveedor WHERE AXP\_articulo\_cod=@codigodearticulo

-- Se compara el cuit del proveedor para que este corresponda con el de la recepcion general.

SELECT @cuitrecepcion=RECART\_prov\_cuit from Recepcion\_Articulos WHERE RECART\_codigo=@codigo

-- Validacion

IF(@cuitrecepcion=@cuitdeproveedor)

BEGIN

INSERT INTO DetalleRecepcion\_Articulos(DRECART\_codigo,DRECART\_prov\_cuit,DRECART\_articulo\_cod,DRECART\_cantidad\_unidades,DRECART\_precio\_unitario)

VALUES(@codigo ,@cuitdeproveedor ,@codigodearticulo ,@cantidaddeunidades ,@preciounitario)

END

ELSE PRINT 'El artículo ingresado no pertenece al proveedor de cuit '+ CAST(@cuitrecepcion AS VARCHAR)+'.'

GO

CREATE PROCEDURE spAgregarArt\_Prov

(

@AXP\_prov\_cuit int,

@AXP\_articulo\_cod int

)

AS

INSERT INTO Articulos\_por\_Proveedor(AXP\_prov\_cuit,AXP\_articulo\_cod,AXP\_stock\_actual,AXP\_stock\_anterior,AXP\_entrada,AXP\_salida,AXP\_precio\_unitario)

VALUES(@AXP\_prov\_cuit,@AXP\_articulo\_cod,0,0,0,0,0)

GO

## TRIGGERS

USE Lab3TIF

GO

**--Trigger que pone el precio del artículo “automáticamente”.**

CREATE TRIGGER TR\_InsertPrecio\_DV

on DetalleVenta

after INSERT

as

BEGIN

--SE INSERTA PRECIO DEL ARTÍCULO

DECLARE @precio decimal(18,2),@cart int,@cven int, @cantComprar int

SELECT @cart=DTV\_articulo\_cod, @cven=DTV\_venta\_codigo, @cantComprar=DTV\_cantidad\_unidades FROM inserted

SELECT @precio= ART\_precio\_lista FROM Articulos WHERE ART\_codigo=@cart

UPDATE DetalleVenta SET DTV\_precio\_unitario=@precio WHERE DTV\_venta\_codigo=@cven and DTV\_articulo\_cod=@cart

--SE ACTUALIZA EL TOTAL DE LA FACTURA DE LA VENTA

UPDATE Ventas SET VEN\_total\_facturado=VEN\_total\_facturado+@precio\*@cantComprar WHERE VEN\_codigo=@cven

END

GO

**--TRIGGER PARA ACTUALIZAR EL STOCK CUANDO SE VENDE UN PRODUCTO**

CREATE TRIGGER TR\_ActualizarStock

ON DetalleVenta

AFTER INSERT

AS

BEGIN

DECLARE @Cart int, @CantCompr int, @stock int

SELECT @CantCompr=DTV\_cantidad\_unidades,@Cart=DTV\_articulo\_cod FROM inserted

SELECT @stock=AXP\_stock\_actual FROM Articulos\_por\_Proveedor WHERE AXP\_articulo\_cod=@Cart

IF(@CantCompr<=@stock)

UPDATE Articulos\_por\_Proveedor SET AXP\_stock\_actual=@stock-@CantCompr,AXP\_stock\_anterior=@stock,AXP\_salida+=@CantCompr WHERE AXP\_articulo\_cod=@Cart

ELSE

ROLLBACK

END

GO

CREATE TRIGGER REESTABLECER\_IDENTITY

ON Recepcion\_Articulos

AFTER INSERT

AS

BEGIN

DBCC CHECKIDENT ('DetalleRecepcion\_Articulos', reseed,0);

END

GO

CREATE TRIGGER TR\_ObtenerPrecioAxP

ON Articulos\_por\_Proveedor

AFTER INSERT

AS

BEGIN

DECLARE @precio decimal(18,2),@Cart int,@CAxP int

SELECT @Cart=AXP\_articulo\_cod, @CAxP=AXP\_codigo FROM inserted

SELECT @precio=ART\_precio\_lista FROM Articulos WHERE ART\_codigo=@Cart

-- Settea el precio obtenido de la tabla artículos

UPDATE Articulos\_por\_Proveedor SET AXP\_precio\_unitario=@precio WHERE AXP\_codigo=@CAxP

END

GO

CREATE TRIGGER TR\_ActualizarRecepcion

ON DetalleRecepcion\_Articulos

AFTER INSERT

AS

BEGIN

DECLARE @CodRecepcion int,@CodArt int,@CantRecibida int,@Stock\_ant int,@Precio decimal(18,2)

SELECT @CodRecepcion=DRECART\_codigo,@CodArt=DRECART\_articulo\_cod,@CantRecibida=DRECART\_cantidad\_unidades, @Precio=DRECART\_precio\_unitario FROM inserted

SELECT @Stock\_ant=AXP\_stock\_anterior FROM Articulos\_por\_Proveedor WHERE AXP\_articulo\_cod=@CodArt

-- Actualiza el valor del stock al ingresar un artículo

UPDATE Articulos\_por\_Proveedor SET AXP\_stock\_actual+=@CantRecibida,AXP\_stock\_anterior=@Stock\_ant,AXP\_entrada+=@CantRecibida WHERE AXP\_articulo\_cod=@CodArt

-- Actualiza el valor total del registro Recepcion\_Artículo

UPDATE Recepcion\_Articulos SET RECART\_total\_facturado+=@CantRecibida\*@Precio WHERE RECART\_codigo=@CodRecepcion

END

GO

## INSERTS

USE Lab3TIF

GO

-- PROVINCIAS

BEGIN

EXEC spAgregarProvincias 'Buenos Aires'

EXEC spAgregarProvincias 'Capital Federal'

EXEC spAgregarProvincias 'Catamarca'

EXEC spAgregarProvincias 'Chaco'

EXEC spAgregarProvincias 'Chubut'

EXEC spAgregarProvincias 'Córdoba'

EXEC spAgregarProvincias 'Corrientes'

EXEC spAgregarProvincias 'Entre Ríos'

EXEC spAgregarProvincias 'Formosa'

EXEC spAgregarProvincias 'Jujuy'

EXEC spAgregarProvincias 'La Pampa'

EXEC spAgregarProvincias 'La Rioja'

EXEC spAgregarProvincias 'Mendoza'

EXEC spAgregarProvincias 'Misiones'

EXEC spAgregarProvincias 'Neuquén'

EXEC spAgregarProvincias 'Río Negro'

EXEC spAgregarProvincias 'Salta'

EXEC spAgregarProvincias 'San Juan'

EXEC spAgregarProvincias 'San Luis'

EXEC spAgregarProvincias 'Santa Cruz'

EXEC spAgregarProvincias 'Santa Fe'

EXEC spAgregarProvincias 'Santiago del Estero'

EXEC spAgregarProvincias 'Tierra del Fuego'

EXEC spAgregarProvincias 'Tucumán'

END

-- ESTADOS

BEGIN

EXEC spAgregarEstados 'Baja'

EXEC spAgregarEstados 'Alta'

END

-- PERFILES

BEGIN

EXEC spAgregarPerfiles 'Personal'

EXEC spAgregarPerfiles 'Clientes'

EXEC spAgregarPerfiles 'Proveedores'

END

-- MEDIOS DE PAGO

BEGIN

EXEC spAgregarMedios\_de\_Pago 'Efectivo','Método de pago en efectivo'

EXEC spAgregarMedios\_de\_Pago 'Débito','Método de pago a través de tarjeta de débito'

EXEC spAgregarMedios\_de\_Pago 'Crédito','Método de pago a través de tarjeta de crédito'

END

-- USUARIOS

EXEC spAgregarUsuarios 2,'German','Vazquez','12345','germanv99@mail.com','Direccion32','Ciudad12',2,'img61.png'

EXEC spAgregarUsuarios 2,'Lautaro','Diaz','12345','lautaroDiaz@mail.com','Direccion','Ciudad',3,'img6.png'

EXEC spAgregarUsuarios 2,'Javier','Gonzalez','011 5123-5234','javigonzalez92@mail.com','Direccion1','Ciudad2',8,'img5.png'

EXEC spAgregarUsuarios 1,'Antonio Jose','Salazar','011 3817-7518','neippuharonnoi-2953@yopmail.com','Moreno 3653','Lomas del Mirador',1,'img13.png'

EXEC spAgregarUsuarios 1,'Tania','Pellicer','011 6356-9715','fodettaummuja-9696@yopmail.com','Larrea 1325','Villa Del Parque',15,'img32.png'

EXEC spAgregarUsuarios 3,'Otilia','Prada','011 4641-1359','neippuharonnoi-2953@yopmail.com','Sanicolas 526','La Rioja',12,'img1.png'

EXEC spAgregarUsuarios 3,'Laura','Maria Cuellar','011 4922-4801','bruxoyaralau-4667@yopmail.com','General Pico','General Pico',11,'img4.png'

EXEC spAgregarUsuarios 3,'Garcia','Herrera','011 2107-5861','priboikosibro-4325@yopmail.com','Alsina 520','Morón',9,'img2.png'

--CLIENTES

EXEC spAgregarCliente 1234567,1,'41282723',Germanv99,'123456'

EXEC spAgregarCliente 1234576,2,'39203123',Laudiaz92,'123123'

EXEC spAgregarCliente 1234555,3,'39912123',JavierGz,'asdasd'

EXEC spAgregarCliente 1234555,4,'42123331',AntonyJ,'asd123'

EXEC spAgregarCliente 3121234,5,'35123331',TaniaPellicer95,'asd123'

EXEC spAgregarCliente 3121234,6,'41123331',OtaliaPrada,'asd123'

EXEC spAgregarCliente 3121139,7,'37123331',LauraCuellar,'asd123'

EXEC spAgregarCliente 1234444,8,'32123331',GraciaHerrera,'asd123'

--PERSONAL

EXEC spAgregarPersonal 1,'41282723',Germanv99,'123456'

EXEC spAgregarPersonal 2,'39203123',Laudiaz92,'123123'

EXEC spAgregarPersonal 3,'39912123',JavierGz,'asdasd'

EXEC spAgregarPersonal 4,'42123331',AntonyJ,'asd123'

EXEC spAgregarPersonal 5,'35123331',TaniaPellicer95,'asd123'

EXEC spAgregarPersonal 6,'41123331',OtaliaPrada,'asd123'

EXEC spAgregarPersonal 7,'37123331',LauraCuellar,'asd123'

EXEC spAgregarPersonal 8,'32123331',Garcia,'asd123'

--PROVEDORES

EXEC spAgregarProveedor 1234567,1,'Samsonite','German Vazquez'

EXEC spAgregarProveedor 1234576,2,'Amec','Lautaro Diaz'

EXEC spAgregarProveedor 1234555,3,'Lenovo','Javier Gonzalez'

EXEC spAgregarProveedor 1234555,4,'Samsonite','Antonio Jose Salazar'

EXEC spAgregarProveedor 3121234,5,'Amec','Tania Pellicer'

EXEC spAgregarProveedor 3121234,6,'Lenovo','Otalia Prada'

EXEC spAgregarProveedor 3121139,7,'Samsonite','Laura Maria Cuellar'

EXEC spAgregarProveedor 1234444,8,'Amec','Garcia Herrera'

-- CATEGORIAS

INSERT INTO Categorias (CAT\_nombre,CAT\_descripcion,CAT\_ruta\_imagen,CAT\_estado\_cod)

SELECT 'Cámaras y Accesorios','Descubrí miles de productos','Imagenes/categorias/camaras-y-accesorios.png',2 UNION

SELECT 'Celulares y Teléfonos','Encontrá lo mejor en smartphones','Imagenes/categorias/celulares-y-telefonos.png',2 UNION

SELECT 'Computación','Todo para armar tu Home Office','Imagenes/categorias/computacion.png',2 UNION

SELECT 'Consolas y Videojuegos','Lo mejor del mundo gamer','Imagenes/categorias/consolas-y-videojuegos.png',2 UNION

SELECT 'Electrónica Audio y Video','Encontrá lo mejor en electrónica','Imagenes/categorias/electronica-audio-y-video.png',2 UNION

SELECT 'Herramientas','Hacé realidad tus proyectos','Imagenes/categorias/herramientas.png',2 UNION

SELECT 'Televisores','Viví tu entretenimiento con los mejores TV','Imagenes/categorias/televisores.png',2

GO

-- MARCAS

INSERT INTO Marcas (MAR\_nombre,MAR\_descripcion,MAR\_ruta\_imagen,MAR\_estado\_cod)

SELECT 'ActiVision','','Imagenes/marcas/ActiVision.png',2 UNION

SELECT 'Admiral','One Touch','Imagenes/marcas/Admiral.png',2 UNION

SELECT 'Alcatel','','Imagenes/marcas/Alcatel.png',2 UNION

SELECT 'AMD','','Imagenes/marcas/AMD.png',2 UNION

SELECT 'Apple','Piensa diferente','Imagenes/marcas/Apple.png',2 UNION

SELECT 'Atma','','Imagenes/marcas/Atma.png',2 UNION

SELECT 'Benro','','Imagenes/marcas/Benro.png',2 UNION

SELECT 'BGH','','Imagenes/marcas/BGH.png',2 UNION

SELECT 'Black & Decker','','Imagenes/marcas/Black & Decker.png',2 UNION

SELECT 'BlackBerry','','Imagenes/marcas/BlackBerry.png',2 UNION

SELECT 'Bosch','','Imagenes/marcas/Bosch.png',2 UNION

SELECT 'Canon','','Imagenes/marcas/Canon.png',2 UNION

SELECT 'Celestron','','Imagenes/marcas/Celestron.png',2 UNION

SELECT 'Corsair','','Imagenes/marcas/Corsair.png',2 UNION

SELECT 'Dell','','Imagenes/marcas/Dell.png',2

GO

-- ARTICULOS

EXEC spAgregarArticulo 1,7,'Smart TV BGH','Dimensiones: 96.8cm de ancho,56.8cm de alto,7.8cm de profundidad',4,'41999.00','Imagenes/articulos/5.png',2

EXEC spAgregarArticulo 2,2,'Moto G9 Power','Con la pantalla Max Vision HD+ de 6.8"',12,'35999.00','Imagenes/articulos/10.png',2

EXEC spAgregarArticulo 3,7,'Smart TV Noblex','Su resolución es 4K',5,'81499.00','Imagenes/articulos/1.png',2

EXEC spAgregarArticulo 4,2,'Nokia 23 M','Memoria interna de 32GB. Apto para tarjeta SD de 512GB',12,'15999.00','Imagenes/articulos/6.png',2

EXEC spAgregarArticulo 5,2,'Philco P241','Memoria interna de 512MB. Apto para tarjeta SD de 64GB',8,'9599.00','Imagenes/articulos/11.png',2

EXEC spAgregarArticulo 6,6,'Taladro eléctrico percutor Philco','Velocidad mínima de rotación: 3000rpm.',20,'9212.00','Imagenes/articulos/12.png',2

EXEC spAgregarArticulo 7,7,'Smart TV Philips','Con YouTube,Netflix,Web browser.',3,'43999.00','Imagenes/articulos/3.png',2

EXEC spAgregarArticulo 8,2,'Samsung Galaxy A52','Tiene 4 cámaras traseras de 64Mpx/12Mpx/5Mpx/5Mpx.',8,'55999.00','Imagenes/articulos/7.png',2

EXEC spAgregarArticulo 9,2,'Samsung Galaxy Note20','Pantalla Dynamic AMOLED 2X de 6.9',2,'155999.00','Imagenes/articulos/9.png',2

EXEC spAgregarArticulo 10,7,'Smart TV Samsung','Tecnología HDR para una calidad de imagen mejorada.',3,'75999.00','Imagenes/articulos/2.png',2

EXEC spAgregarArticulo 11,7,'Smart TV TCL','Con función Screen Share',6,'32999.00','Imagenes/articulos/4.png',2

EXEC spAgregarArticulo 12,2,'Xiaomi Poco X3','Dispositivo liberado para que elijas la compañía telefónica que prefieras',3,'61200.00','Imagenes/articulos/8.png',2

-- ARTICULOS\_POR\_PROVEEDOR

EXEC spAgregarArt\_Prov 1234444,1

EXEC spAgregarArt\_Prov 3121139,2

EXEC spAgregarArt\_Prov 3121234,3

EXEC spAgregarArt\_Prov 1234444,4

EXEC spAgregarArt\_Prov 3121139,5

EXEC spAgregarArt\_Prov 1234444,6

EXEC spAgregarArt\_Prov 3121234,7

EXEC spAgregarArt\_Prov 1234444,8

EXEC spAgregarArt\_Prov 3121139,9

EXEC spAgregarArt\_Prov 1234444,10

EXEC spAgregarArt\_Prov 3121139,11

EXEC spAgregarArt\_Prov 3121234,12

-- RECEPCION ARTICULO

EXEC spAgregarRecepcion\_Articulos 1,1234444

EXEC spAgregarRecepcion\_Articulos 1,3121139

EXEC spAgregarRecepcion\_Articulos 2,3121234

-- DETALLE RECEPCION ARTICULO

-- CodRec,CodArt,Cantidad,PrecioComprado

EXEC spAgregarDetalleRecepcionDeArticulos 1,1,50,2000

EXEC spAgregarDetalleRecepcionDeArticulos 2,2,20,550

EXEC spAgregarDetalleRecepcionDeArticulos 3,3,5520,150

EXEC spAgregarDetalleRecepcionDeArticulos 1,4,1029,250

EXEC spAgregarDetalleRecepcionDeArticulos 2,5,421,2050

EXEC spAgregarDetalleRecepcionDeArticulos 1,6,2878,850

EXEC spAgregarDetalleRecepcionDeArticulos 3,7,1270,1356

EXEC spAgregarDetalleRecepcionDeArticulos 1,8,240,1020

EXEC spAgregarDetalleRecepcionDeArticulos 2,9,190,3030

EXEC spAgregarDetalleRecepcionDeArticulos 1,10,100,1523

EXEC spAgregarDetalleRecepcionDeArticulos 2,11,200,40

EXEC spAgregarDetalleRecepcionDeArticulos 3,12,400,127

-- Ingresos erroneos (validados)

EXEC spAgregarDetalleRecepcionDeArticulos 3,1,50,2000

EXEC spAgregarDetalleRecepcionDeArticulos 1,2,20,550

EXEC spAgregarDetalleRecepcionDeArticulos 2,3,5520,150

-- VENTAS

EXEC spAgregarVenta 1234555,2,2

EXEC spAgregarVenta 1234567,2,2

EXEC spAgregarVenta 1234576,2,2

-- DETALLE VENTA

EXEC spAgregarDetalleVenta 1,4,10

EXEC spAgregarDetalleVenta 1,2,5

EXEC spAgregarDetalleVenta 1,1,8

EXEC spAgregarDetalleVenta 1,3,2

EXEC spAgregarDetalleVenta 1,9,3

EXEC spAgregarDetalleVenta 2,6,9

EXEC spAgregarDetalleVenta 2,9,2

EXEC spAgregarDetalleVenta 2,1,4

EXEC spAgregarDetalleVenta 3,5,10

EXEC spAgregarDetalleVenta 3,7,4

EXEC spAgregarDetalleVenta 3,12,5

## CONSULTAS

--Consultas

--los artículos mayores de una determinada cantidad en stock.

SELECT ART\_nombre AS 'Artículo',

Articulos\_por\_Proveedor.AXP\_stock\_actual AS 'Stock'

FROM Articulos\_por\_Proveedor

INNER JOIN Articulos ON Articulos.ART\_codigo = Articulos\_por\_Proveedor.AXP\_codigo

WHERE Articulos\_por\_Proveedor.AXP\_stock\_actual > 1000

GO

-- o con un procedimiento almacenado

CREATE PROCEDURE ArticulosMayorStock

@stock INT

AS

SELECT ART\_nombre AS 'Artículo',

Articulos\_por\_Proveedor.AXP\_stock\_actual AS 'Stock'

FROM Articulos\_por\_Proveedor

INNER JOIN Articulos ON Articulos.ART\_codigo = Articulos\_por\_Proveedor.AXP\_codigo

WHERE Articulos\_por\_Proveedor.AXP\_stock\_actual > @stock

GO

EXEC ArticulosMayorStock 1000

--CANTIDAD DE VENTAS DETALLADAS POR VENTA, CANTIDAD DE ARTÍCULOS COMPRADOS Y MONTO.

SELECT DTV\_venta\_codigo AS [VENTA] ,

COUNT(\*) AS [Cantidad de Compras],

SUM(DTV\_cantidad\_unidades) AS [Total de Unidades],

SUM(DTV\_precio\_unitario\*DTV\_cantidad\_unidades) AS [MONTO]

FROM DetalleVenta GROUP BY DTV\_venta\_codigo

GO

--CANTIDAD DE UNIDADES VENDIDAS DE CADA ARTÍCULO, ORDENADAS DE MAYOR A MENOR ( MÁS DEMANDADA A LA MENOS DEMANDADA)

SELECT AXP\_articulo\_cod AS [CÓDIGO],A.ART\_nombre AS [NOMBRE], AXP\_salida AS [CANTIDAD VENDIDA] FROM Articulos\_por\_Proveedor AS AP INNER JOIN Articulos AS A

ON A.ART\_codigo=AP.AXP\_articulo\_cod

ORDER BY AXP\_salida DESC

GO

-- DETALLE DE VENTA DE UNA FACTURA DADA

SELECT SUM(DTV\_precio\_unitario \* DTV\_cantidad\_unidades) AS 'Total Parcial',

art\_nombre 'Nombre Artículo',

DTV\_cantidad\_unidades 'Unidades',

DTV\_precio\_unitario 'Precio'

FROM DetalleVenta

INNER JOIN articulos on DTV\_articulo\_cod = ART\_codigo

WHERE DTV\_venta\_codigo = 2

GROUP BY art\_nombre,DTV\_articulo\_cod,DTV\_cantidad\_unidades,DTV\_precio\_unitario

-- O CON UN PROCEDIMIENTO ALMACENADO

CREATE PROCEDURE DetalleVentaPorFactura

@CodFactura INT

AS

SELECT SUM(DTV\_precio\_unitario \* DTV\_cantidad\_unidades) AS 'Total Parcial',

art\_nombre 'Nombre Artículo',

DTV\_cantidad\_unidades 'Unidades',

DTV\_precio\_unitario 'Precio'

FROM DetalleVenta

INNER JOIN articulos on DTV\_articulo\_cod = ART\_codigo

WHERE DTV\_venta\_codigo = @CodFactura

GROUP BY art\_nombre,DTV\_articulo\_cod,DTV\_cantidad\_unidades,DTV\_precio\_unitario

GO

EXEC DetalleVentaPorFactura 2

-- CONSULTA QUE MUESTRA DATOS DEL PERSONAL

SELECT

USU\_nombre AS 'Nombre',

USU\_apellido AS 'Apellido',

USU\_telefono AS 'Teléfono',

USU\_email AS 'eMail',

USU\_direccion AS 'Dirección',

USU\_ciudad AS 'Ciudad',

PROV\_nombre AS 'Provincia',

EST\_nombre AS 'Estado'

FROM Usuarios

INNER JOIN Provincias

ON Provincias.PROV\_codigo = Usuarios.USU\_provincia\_cod

INNER JOIN Estados

ON Estados.EST\_codigo = Usuarios.USU\_estado\_cod

INNER JOIN

Perfiles

ON Perfiles.PER\_codigo = Usuarios.USU\_perfil\_cod

WHERE Perfiles.PER\_codigo = 1

-- CONSULTA QUE MUESTRA DATOS DE LOS CLIENTES

SELECT

USU\_nombre AS 'Nombre',

USU\_apellido AS 'Apellido',

USU\_telefono AS 'Teléfono',

USU\_email AS 'eMail',

USU\_direccion AS 'Dirección',

USU\_ciudad AS 'Ciudad',

PROV\_nombre AS 'Provincia',

EST\_nombre AS 'Estado'

FROM Usuarios

INNER JOIN Provincias

ON Provincias.PROV\_codigo = Usuarios.USU\_provincia\_cod

INNER JOIN Estados

ON Estados.EST\_codigo = Usuarios.USU\_estado\_cod

INNER JOIN

Perfiles

ON Perfiles.PER\_codigo = Usuarios.USU\_perfil\_cod

WHERE Perfiles.PER\_codigo = 2

-- CONSULTA QUE MUESTRA DATOS DE LOS PROVEEDORES

SELECT

USU\_nombre AS 'Nombre',

USU\_apellido AS 'Apellido',

USU\_telefono AS 'Teléfono',

USU\_email AS 'eMail',

USU\_direccion AS 'Dirección',

USU\_ciudad AS 'Ciudad',

PROV\_nombre AS 'Provincia',

EST\_nombre AS 'Estado'

FROM Usuarios

INNER JOIN Provincias

ON Provincias.PROV\_codigo = Usuarios.USU\_provincia\_cod

INNER JOIN Estados

ON Estados.EST\_codigo = Usuarios.USU\_estado\_cod

INNER JOIN

Perfiles

ON Perfiles.PER\_codigo = Usuarios.USU\_perfil\_cod

WHERE Perfiles.PER\_codigo = 3