***Proyecto Facturador.sql***

****

* **Alumno: Ayala Kevin Elias**
* **Tutor: Cabana Fernando**
* **Profesor: Redondo Camilo**
* **Comisión : 40750**

***El proyecto detallado a continuación tiene la función de facturar con el nombre de la persona, recibo, producto, ciudad, cantidad y la garantía del producto comprado***

***Creación de las tablas***

***y***

***base de datos***

* ***Aca creamos la base de datos en la que crearemos las tablas***

*Créate database facturador;*

* ***Creamos la tabla cliente que lo guardara son los datos de la persona que se registre para la venta***

*create table Cliente(*

*ID\_Cliente int primary key not null auto\_increment,*

*nombre varchar (1000) not null,*

*apellido varchar(1000) not null,*

*email varchar(1000) not null,*

*dni integer unsigned not null,*

*provincia varchar(1000) not null,*

*localidad varchar(1000) not null*

*);*

* ***Creamos la tabla stock que lo hace es llevar un control de los productos que hay guardados***

*DROP TABLE IF EXISTS `Stock`;*

*create table Stock(*

*ID\_Stock int not null auto\_increment primary key,*

*ID\_Producto int , -- FK*

*cantidad int unsigned not null);*

* ***Creamos la tabla productos que lo hace es guardar los datos de los productos ingresados***

***create table Producto(***

***ID\_Producto integer primary key not null auto\_increment,***

***marca varchar (1000) not null,***

***modelo varchar(1000) not null,***

***serial\_number varchar(1000) not null,***

***precio integer unsigned not null,***

***producto\_disponible boolean default false***

***);***

* ***Creamos la tabla garantía y aca se guardan los datos de las tablas,Stock,Producto y Cliente, para poder usarlos de referencia con la garantía del producto***

***DROP TABLE IF EXISTS `Garantia`;***

***create table Garantia(***

***ID\_Garantia int primary key not null auto\_increment,***

***ID\_Producto int , -- FK***

***ID\_Stock int , -- FK***

***ID\_Cliente int -- FK***

* ***Creamos la tabla recibo, consultamos los datos de todas las otras tablas para poder generar un recibo con los datos del cliente y el producto comprado***

***DROP TABLE IF EXISTS `Recibo`;***

***create table Recibo(***

***ID\_Recibo int primary key not null auto\_increment,***

***ID\_Garantia int,-- fk***

***ID\_Producto int , -- FK***

***ID\_Stock int , -- FK***

***ID\_Cliente int, -- FK***

***ID\_Provincia int, -- FK***

***ID\_Localidades int -- FK***

***);***

* ***Creamos la tabla Provincia para poder consultar a cual pertenece nuestro cliente***

***DROP TABLE IF EXISTS `Provincias`;***

***CREATE TABLE Provincias (***

***ID\_Provincia int not null auto\_increment primary key,***

***provincia varchar(1000) NOT NULL***

***);***

* ***Creamos la tabla localidades para poder consultar a qué provincia y que localidad pertenece nuestro cliente***

***CREATE TABLE Localidades (***

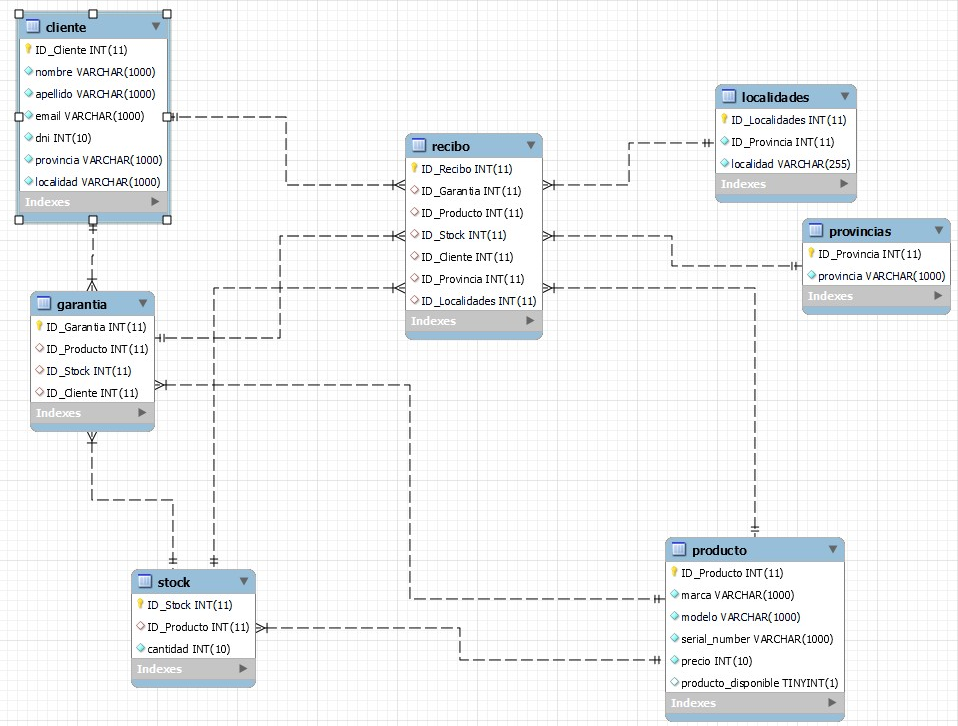
***ID\_Localidades int not null auto\_increment primary key,***

***ID\_Provincia int NOT NULL,***

***localidad varchar(255) NOT NULL***

***);***

***Diagrama entidad relación***

******

***Se crea las tablas de relación***

***de uno a muchos***

***alter table Stock***

***ADD foreign key (ID\_Producto)***

***REFERENCES Producto(ID\_Producto);***

***ALTER TABLE Garantia -- en la tabla Garantia***

***ADD FOREIGN KEY (ID\_Producto) -- agregame como clave foranea (ID\_PRODUCTO)***

***REFERENCES Producto (ID\_Producto); -- de la tabla Producto con clave primaria (ID\_PRODUCTO)***

***ALTER TABLE Garantia -- en la tabla Garantia***

***ADD FOREIGN KEY (ID\_Stock) -- agregame como clave foranea (ID\_Stock)***

***REFERENCES Stock (ID\_Stock); -- de la tabla Stock con clave primaria (ID\_Stock)***

***ALTER TABLE Garantia -- en la tabla Garantia***

***ADD FOREIGN KEY (ID\_Cliente) -- agregame como clave foranea (ID\_Cliente)***

***REFERENCES Cliente (ID\_Cliente); -- de la tabla Cliente con clave primaria (ID\_Cliente)***

***ALTER TABLE Recibo -- en la tabla Recibo***

***ADD FOREIGN KEY (ID\_Cliente) -- agregame como clave foranea (ID\_Cliente)***

***REFERENCES Cliente (ID\_Cliente); -- de la tabla Cliente con clave primaria (ID\_Cliente)***

***ALTER TABLE Recibo -- en la tabla Recibo***

***ADD FOREIGN KEY (ID\_Stock) -- agregame como clave foranea (ID\_Stock)***

***REFERENCES Stock (ID\_Stock); -- de la tabla Stock con clave primaria (ID\_Stock)***

***ALTER TABLE Recibo -- en la tabla Recibo***

***ADD FOREIGN KEY (ID\_Producto) -- agregame como clave foranea (ID\_PRODUCTO)***

***REFERENCES Producto (ID\_Producto); -- de la tabla Producto con clave primaria (ID\_PRODUCTO)***

***ALTER TABLE Recibo -- en la tabla Recibo***

***ADD FOREIGN KEY (ID\_Garantia) -- agregame como clave foranea (ID\_Garantia)***

***REFERENCES Garantia(ID\_Garantia); -- de la tabla Garantia con clave primaria (ID\_Garantia)***

***ALTER TABLE Recibo -- en la tabla Recibo***

***ADD FOREIGN KEY (ID\_Provincia) -- agregame como clave foranea (ID\_Provincia) producto***

***REFERENCES Provincias (ID\_Provincia); -- de la tabla provincias con clave primaria (ID\_Provincia)***

***ALTER TABLE Recibo -- en la tabla Recibo***

***ADD FOREIGN KEY (ID\_Localidades) -- agregame como clave foranea (ID\_Localidades)***

***REFERENCES Localidades (ID\_Localidades); -- de la tabla localidades con clave primaria (ID\_Localidades)***

***CREACION Y USO DE FUNCIONES***

******

*Función I : Lo que hacemos en la función 1 es buscar cuantas personas/clientes viven en la localidad ingresada por parámetro, esto lo busca en la tabla CLIENTEBK y nos retorna la unidad de personas viviendo en la localidad ingresada.*

*Función II : Lo que hacemos en la función 2 es buscar cuantas personas/clientes viven en la provincia ingresada por parámetro, esto lo busca en la tabla CLIENTEBK y nos retorna la unidad de personas viviendo en la provincia ingresada.*

******

*Función III : Lo que hacemos en la función 3 es buscar cuantos productos son del mismo modelo ingresado por parámetro esto lo busca en la tabla PRODUCTOBK y nos retorna la unidad de productos que cumplan con el modelo ingresado*

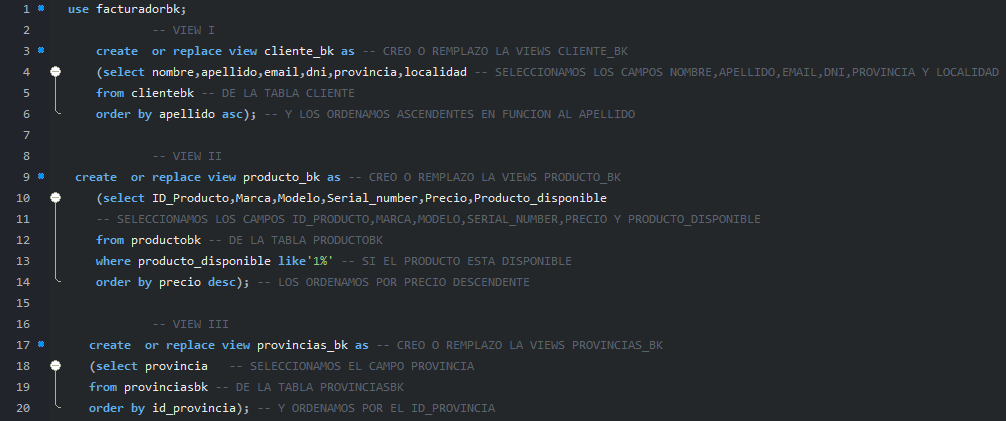
*Función IV : Lo que hacemos en la función 4 es buscar cuantos productos tienen la misma marca ingresada por parámetro esto lo busca en la tabla PRODUCTOBK y nos retorna la unidad de productos que cumplan con la marca ingresado*

******

*Función V : Lo que hacemos en la función 5 es buscar cuantos productos están disponibles que cumplan con los parámetros ingresados, para que esto funcione tenemos que unir la tabla STOCKBK y la tabla PRODUCTOBK por medio del ID\_PRODUCTO que es parte de la tabla STOCKBK después ingresamos la marca y el modelo que son los parámetros a tener en cuenta y me consulta cuantos productos que cumplan esos parámetros están disponibles en el stock.*

*Función IV : Lo que hacemos en la función 6 es buscar cuantos productos iguales se vendieron para esto consultamos en la tabla PRODUCTOBK el parámetro ingresado que va a ser igual al campo SERIAL\_NUMBER y así nos va a decir la cantidad de veces que se vendió ese producto.*

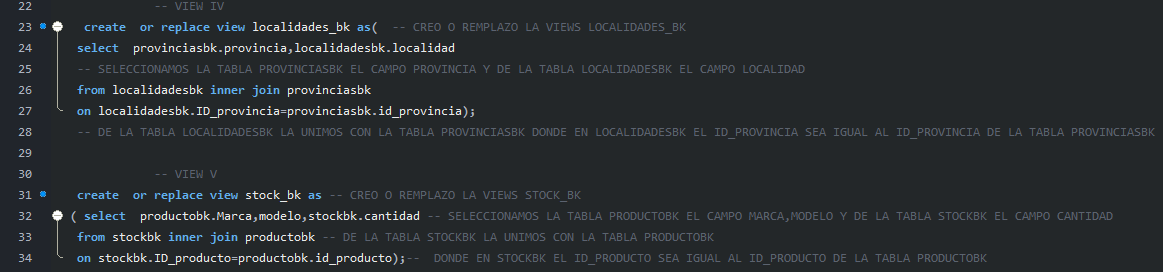
***CREACION Y USO DE VIEWS***



*Views I: Lo que hacemos en la Views 1 lo que hacemos es seleccionar los campos nombres, apellido, email, DNI, provincia y localidad de la tabla CLIENTEBK nos muestran todos los datos y los ordenamos ascendentes en función al apellido.*

*Views II: Lo que hacemos en la Views 2 lo que hacemos es seleccionar los Campos id\_producto, marca, modelo, serial\_number, precio y producto\_disponible de la tabla PRODUCTBK, filtramos los productos que estén disponibles y los ordenan en precio descendente.*

*Views III: Lo que hacemos en la Views 3 lo que hacemos es seleccionar el campo provincia de la tabla PROVINCIABK nos muestra las provincias y los ordenamos en base a su ID se ordenan descendente.*

******

*Views IV: Lo que hacemos en la Views 4 lo que hacemos es seleccionar el campo provincia de la tabla PROVINCIASBK y localidad de la tabla LOCALIDADESBK.*

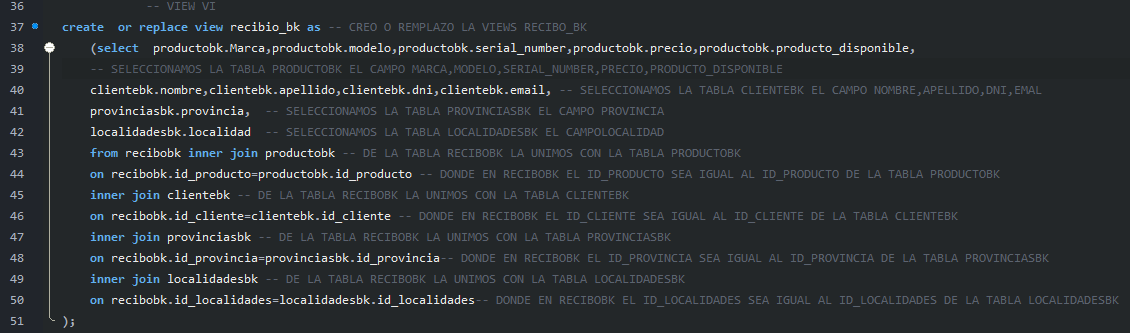
*INNER JOIN*

*Unimos las tablas LOCALIDADESBK con la tabla PROVINCIASBK donde en la tabla principal el campo ID\_PROVINCIA sea igual al ID\_PROVINCIA de la tabla PROVINCIASBK.*

*Views V: Lo que hacemos en la Views 5 lo que hacemos es seleccionar el campo marca, modelo de la tabla PRODUCTOBK y cantidad de la tabla STOCKBK.*

*INNER JOIN*

*Unimos las tablas SOTCKBK con la tabla PRODUCTOBK donde en la tabla principal el campo ID\_PRODUCTO sea igual al ID\_ PRODUCTO de la tabla PRODUCTOBK nos muestra la marca, el modelo y la cantidad de los productos.*



*Views VI: Lo que hacemos en la Views 6 lo que hacemos es seleccionar el campo marca, modelo, Serial\_number, precio, producto\_disponible de la tabla PRODUCTOBK, nombre, apellido, dni, email de la tabla CLIENTEBK, provincia de la tabla PROVINCASBK, localidad de la tabla LOCALIDADESBK.*

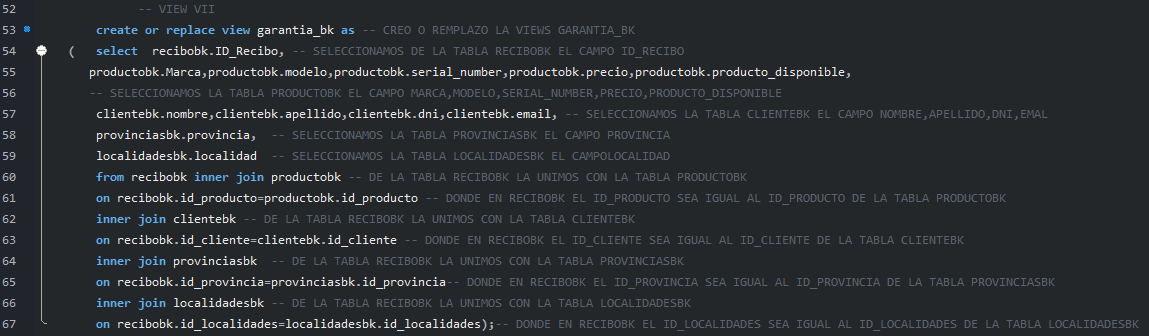
*INNER JOIN*

*Unimos las tablas RECIBOBK con la tabla PRODUCTOBK donde en la tabla principal el campo ID\_PRODUCTO sea igual al ID\_ PRODUCTO de la tabla PRODUCTOBK nos muestra la marca, el modelo, el precio, el serial\_number y la cantidad de los productos.*

*Unimos las tablas RECIBOBK con la tabla CLIENTEBK donde en la tabla principal el campo ID\_CLIENTE sea igual al ID\_ CLIENTE de la tabla CLIENTEBK nos muestra nombre, apellido, dni, y email.*

*Unimos las tablas RECIBOBK con la tabla PROVINCIASBK donde en la tabla principal el campo ID\_PROVINCIA sea igual al ID\_PROVINCIA de la tabla PROVINCIASBK nos muestra la provincia del cliente.*

*Unimos las tablas RECIBOBK con la tabla LOCALIDADESBK donde en la tabla principal el campo ID\_LOCALIDAD sea igual al ID\_LOCALIDAD de la tabla LOCALIDADESBK nos muestra la localidad del cliente.*

**

*Views VII: Lo que hacemos en la Views 7 lo que hacemos es seleccionar el campo id\_recibo de la tabla RECIBOBK nos va a mostrar el id del recibo para la garantía, marca, modelo, Serial\_number, precio, producto\_disponible de la tabla PRODUCTOBK, nombre, apellido, dni, email de la tabla CLIENTEBK, provincia de la tabla PROVINCASBK, localidad de la tabla LOCALIDADESBK.*

*INNER JOIN*

*Unimos las tablas RECIBOBK con la tabla PRODUCTOBK donde en la tabla principal el campo ID\_PRODUCTO sea igual al ID\_ PRODUCTO de la tabla PRODUCTOBK nos muestra la marca, el modelo, el precio, el serial\_number y la cantidad de los productos.*

*Unimos las tablas RECIBOBK con la tabla CLIENTEBK donde en la tabla principal el campo ID\_CLIENTE sea igual al ID\_ CLIENTE de la tabla CLIENTEBK nos muestra nombre, apellido, dni, y email.*

*Unimos las tablas RECIBOBK con la tabla PROVINCIASBK donde en la tabla principal el campo ID\_PROVINCIA sea igual al ID\_PROVINCIA de la tabla PROVINCIASBK nos muestra la provincia del cliente.*

*Unimos las tablas RECIBOBK con la tabla LOCALIDADESBK donde en la tabla principal el campo ID\_LOCALIDAD sea igual al ID\_LOCALIDAD de la tabla LOCALIDADESBK nos muestra la localidad del cliente.*

***CREACION Y USO DE LOS***

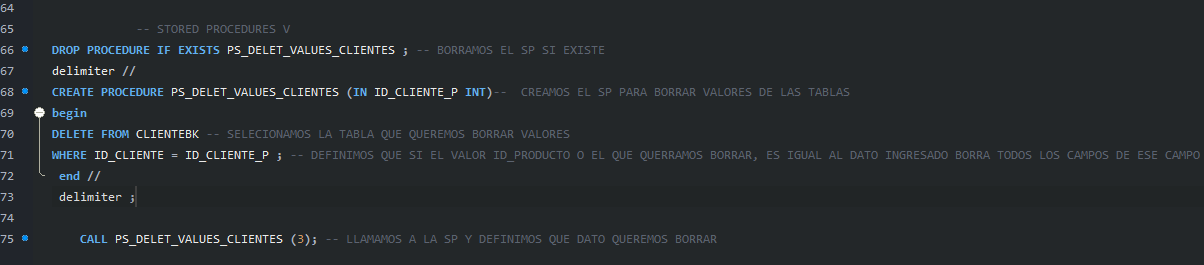
***STORE PROCEDURES***

******

*Stored Produceres I: Lo que el SP 1 hace es que podamos ingresar datos en la tabla seleccionada, siempre y cuando concuerden con los campos seleccionados por ejemplo seleccionamos la tabla PRODUCTOBK y llenamos los datos en el mismo orden de los campos.*

*Stored Produceres II: Lo que el SP 2 hace es que podamos borrar datos en la tabla seleccionada, siempre y cuando el ID del producto ingresado concuerde con el ID del producto en la tabla PRODUCTOBK.*

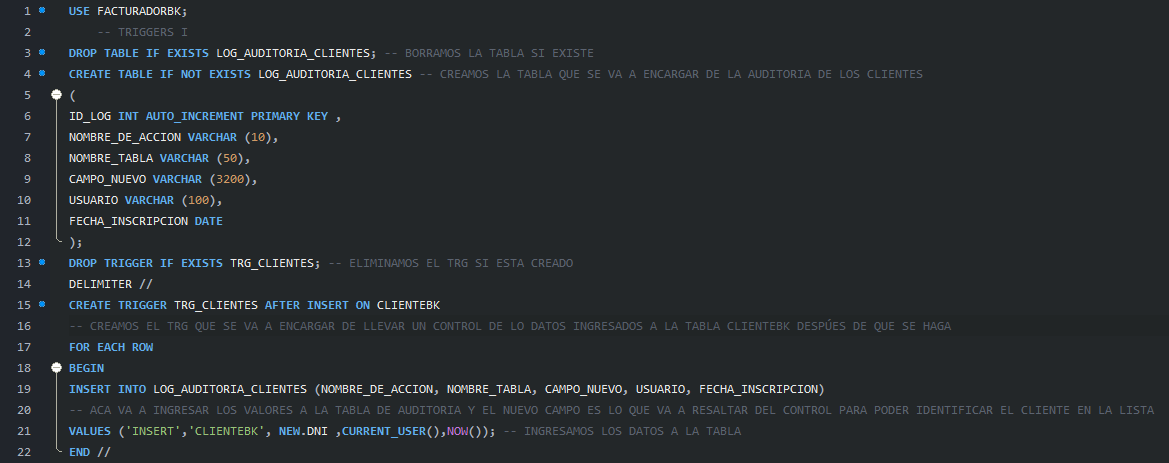
******

******

*Stored Produceres III: Lo que el SP 3 hace es ordenar los productos en forma ascendente o descendente de la tabla PRODUCTOBK así lo decidamos al ingresar el producto y el orden en el que lo queremos ver.*

*Stored Produceres IV: Lo que el SP 4 hace es que podamos ingresar datos en la tabla seleccionada, siempre y cuando concuerden con los campos seleccionados por ejemplo seleccionamos la tabla CLIENTEBK y llenamos los datos en el mismo orden de los campos.*

*Stored Produceres V: Lo que el SP 5 hace es que podamos borrar datos en la tabla seleccionada, siempre y cuando el ID del cliente ingresado concuerde con el ID del cliente en la tabla CLIENTEBK.*

******

*TRIGGER I: Lo que el SP 1 es llevar un registro de los datos ingresados a la tabla CLIENTEBK , lo es campo nuevo que en este caso es DNI es lo que nos va a permitir al nuevo cliente, también nos va a mostrar el usuario que lo ingreso y la fecha del ingreso.*