CREATE DATABASE FACTURACION\_

GO

USE FACTURACION\_

GO

CREATE TABLE CLIENTES (

codCLi char(3) PRIMARY KEY,

nombreCli varchar(100),

apellidoCli varchar(100),

telefCli char(9)

)

CREATE TABLE FACTURAS (

nroFact char(3) PRIMARY KEY,

valorFact money default 0,

fechaFact date,

codCLi char(3),

FOREIGN KEY(codCLi) REFERENCES CLIENTES (codCLi)

)

CREATE TABLE PRODUCTOS (

codProd char(3) PRIMARY KEY,

nombreProd varchar(100),

unidades varchar(50),

cantidadStock dec(8,2),

precioUnitarioActual money

)

CREATE TABLE DETALLES (

nroFact char(3) not null,

codProd char(3) not null,

cantidadComprada dec(8,2),

precioUnitario money,

FOREIGN KEY(nroFact) REFERENCES FACTURAS (nroFact)

)

ALTER TABLE DETALLES ADD PRIMARY KEY(nroFact,codProd)

ALTER TABLE DETALLES ADD FOREIGN KEY(codProd) REFERENCES PRODUCTOS (codProd)

insert into CLIENTES values('C01','Jose','Ayala','092345678')

insert into CLIENTES values('C02','Maria','Paredes','062456789')

insert into CLIENTES values('C03','Marta','Ortega',null)

insert into FACTURAS values('001',default,'2011-03-23','C01')

insert into FACTURAS values('002',default,'2011-03-24','C01')

insert into FACTURAS values('003',default,'2011-03-23','C02')

insert into FACTURAS values('004',default,'2011-03-25','C02')

insert into PRODUCTOS values('P01','Arroz','libra',300,0.90)

insert into PRODUCTOS values('P02','Azucar','libra',100,1.20)

insert into PRODUCTOS values('P03','Aceite','litro',50,2.80)

insert into PRODUCTOS values('P04','Atun','lata de 500 gramos',300,1.20)

--El cliente Jose Ayala compra 5 Libras de arroz el 8 de Agosto del 2013 REALIZAR LA INSERSION

update FACTURAS set fechaFact='2013-08-08' where nroFact ='001'

--Selecciones

select \*from FACTURAS

select \*from DETALLES

select \*from PRODUCTOS

--Insersion de Datos Detalles

insert into DETALLES values ('001','P01',5,0.85)

--Actualizar Producto: Arroz

update FACTURAS set valorFact=5\*0.85 where nroFact='001'

update PRODUCTOS set cantidadStock=cantidadStock-5 where codProd='P01'

--El cliente Jose Ayala quiere ademas 4 litros de aceite en la MISMA FACTURA

insert into DETALLES values ('001','P03',4,2.5)

--Actualizar Producto: Aceite

update FACTURAS set valorFact=valorFact+(4\*2.50) where nroFact='001'

update PRODUCTOS set cantidadStock=cantidadStock-4 where codProd='P03'

--El cliente Jose Ayala compra 7 Libras de Arroz y 2 Libras de Azucar el 30 de diciembre del 2013

--con FACTURA 002 hacer el ingreso

update FACTURAS set fechaFact='2013-12-30' where nroFact ='002'

insert into DETALLES values ('002','P01',7,0.90)

insert into DETALLES values ('002','P02',2,1.20)

--Actualizar

update FACTURAS set valorFact=valorFact+(7\*0.90)+(2\*1.20) where nroFact='002'

update PRODUCTOS set cantidadStock=cantidadStock-7 where codProd='P01'

update PRODUCTOS set cantidadStock=cantidadStock-2 where codProd='P02'

--El cliente Maria Paredes compra 3 Libras de azucar y 2 latas de Atun el 30 de diciembre del 2013

--con FACTURA 003 hacer el ingreso

update FACTURAS set fechaFact='2013-12-30' where nroFact ='003'

insert into DETALLES values ('003','P02',3,1.20)

insert into DETALLES values ('003','P04',2,1.20)

--Actualizar

update FACTURAS set valorFact=valorFact+(3\*1.20)+(2\*1.20) where nroFact='003'

update PRODUCTOS set cantidadStock=cantidadStock-3 where codProd='P02'

update PRODUCTOS set cantidadStock=cantidadStock-2 where codProd='P04'

--El cliente Maria Paredes compra 5 Libras de arroz el 2 de enero del 2014

--con FACTURA 004 hacer el ingreso

update FACTURAS set fechaFact='2014-01-02' where nroFact ='004'

insert into DETALLES values ('004','P01',5,0.90)

--Actualizar

update FACTURAS set valorFact=valorFact+(5\*0.90)where nroFact='004'

update PRODUCTOS set cantidadStock=cantidadStock-5where codProd='P01'

--El cliente Marco Ortega compra 8 latas de Atun ,4 libras de Azucar y 1 litro de aceite el 6 de enero del 2014

--FACTURA 005

insert into CLIENTES values('C04','Marco','Ortega','035654545')

insert into FACTURAS values('005',default,'2014-01-06','C04')

insert into DETALLES values ('005','P02',4,1.20)

insert into DETALLES values ('005','P03',1,2.80)

insert into DETALLES values ('005','P04',8,1.20)

--Actualizar

update FACTURAS set valorFact=valorFact+(4\*1.20)+(1\*2.80)+(8\*1.20) where nroFact='005'

update PRODUCTOS set cantidadStock=cantidadStock-4where codProd='P02'

update PRODUCTOS set cantidadStock=cantidadStock-1 where codProd='P03'

update PRODUCTOS set cantidadStock=cantidadStock-8 where codProd='P04'

--Selecciones

select \*from FACTURAS

select \*from DETALLES

select \*from PRODUCTOS

select \*from CLIENTES

--Procediemiento Almacenado: Conjunto de intrucciones en SQL que se ejecutan con un nombre se utiliza

--cuando en muchas ocaciones tenemos que ejecutar un mismo conjunto de instrucciones. Ejemplo:

--CONSULTAS:

--Desplegar el Nombre y Apellido de los clientes que compraron azucar

select nombreCli,apellidoCli from CLIENTES c inner join FACTURAS f on c.codCLi= f.codCLi

inner join DETALLES d on f.nroFact=d.nroFact inner join PRODUCTOS p on d.codProd=p.codProd

where nombreProd='Azucar'

select nombreCli,apellidoCli from CLIENTES c ,FACTURAS f ,DETALLES d ,PRODUCTOS p

where c.codCLi= f.codCLi and f.nroFact=d.nroFact and d.codProd=p.codProd

and nombreProd='Azucar'

--PROCEDIMIENTO ALMACENADO

--Crear un procedimiento almacenado para la anterior consulta

create proc sp\_ClientePorProducto

as

select nombreCli,apellidoCli from CLIENTES c inner join FACTURAS f on c.codCLi= f.codCLi

inner join DETALLES d on f.nroFact=d.nroFact inner join PRODUCTOS p on d.codProd=p.codProd

where nombreProd='Azucar'

exec sp\_ClientePorProducto

--Procedimiento con parametros

alter proc sp\_ClientePorProducto1 @nomProd varchar(100)

as

select distinct (nombreCli),apellidoCli from CLIENTES c inner join FACTURAS f on c.codCLi= f.codCLi

inner join DETALLES d on f.nroFact=d.nroFact inner join PRODUCTOS p on d.codProd=p.codProd

where nombreProd=@nomProd

exec sp\_ClientePorProducto1 'Arroz'

exec sp\_ClientePorProducto1 'Atun'

exec sp\_ClientePorProducto1 'Aceite'

--Crear procedimiento almacenado que recibe como parametros el nombre de un producto, el mes y

--el año que se realizo una compra y despliega el detalle de esa compra junto con el nombre y apellido

--del cliente que hizo esa compra.

alter proc sp\_DetallesPorProductoMesyAño @nomProd varchar(100),@Mes int, @Año int

as

select nombreCli,apellidoCli,d.\* from CLIENTES c inner join FACTURAS f on c.codCLi= f.codCLi

inner join DETALLES d on f.nroFact=d.nroFact inner join PRODUCTOS p on d.codProd=p.codProd

where nombreProd=@nomProd and month(fechaFact)=@Mes and year(fechaFact)=@Año

exec sp\_DetallesPorProductoMesyAño 'Arroz',12,2013

exec sp\_DetallesPorProductoMesyAño 'Aceite',08,2013

exec sp\_DetallesPorProductoMesyAño 'Atun',01,2014

--Borrar un Procedimiento Almacenado(PA)

drop proc sp\_ClientePorProducto

--PA.- Se utlizan con frecuencia tanto para administracion como para ingreso de datos,consultas

--y tambien para ejecutar programas.

--TRIGGERS(DESENCADENADORES o DISPARADORES).-

--Son porciones de codigo en SQLo en cuelaquier lenguaje que se ejecutan en forma automatica

--cuando se realizan una insercion, una actualizacion o un borrado en una tabla.

--Son muy utiles para mantener la integridad de los datos.

--Crear un Trigger

create trigger tr\_ActualizaFacturaStock

on DETALLES after insert

as

declare @numfact char(3), @codProd char(3), @cantComprada decimal(8,2), @precUnit money

select @numfact=inserted.nroFact,@codProd=inserted.codProd,@cantComprada=inserted.cantidadComprada,

@precUnit=inserted.precioUnitario from inserted

update FACTURAS set valorFact=valorFact+@cantComprada\*@precUnit

where nroFact=@numfact

update PRODUCTOS set cantidadStock=cantidadStock-@cantComprada

where codProd=@codProd

insert into DETALLES values ('001','P04',2,1.20)

--Inserted.- Es una tabla temporal donde se almacena las ultimas filas que fueron insertadas en una tabla.

--Selecciones

select \*from FACTURAS

select \*from DETALLES

select \*from PRODUCTOS

select \*from CLIENTES

--Actualizar al borrar el detalle

create trigger tr\_BorraDetalle

on DETALLES after delete

as

declare @numfact char(3), @codProd char(3), @cantComprada decimal(8,2), @precUnit money

select @numfact=deleted.nroFact,@codProd=deleted.codProd,@cantComprada=deleted.cantidadComprada,

@precUnit=deleted.precioUnitario from deleted

update FACTURAS set valorFact=valorFact-@cantComprada\*@precUnit

where nroFact=@numfact

update PRODUCTOS set cantidadStock=cantidadStock+@cantComprada

where codProd=@codProd

delete from DETALLES where nroFact='001' and codProd='P04'

--Deleted.-Es una tabla temporal donde se almacena determinadas filas que han sido eliminadas en una tabla.

create trigger tr\_ActualizacionDetalles

on DETALLES after update

as

declare @numfact char(3), @codProd char(3), @cantComprada decimal(8,2), @precUnit money

select @numfact=deleted.nroFact,@codProd=deleted.codProd,@cantComprada=deleted.cantidadComprada,

@precUnit=deleted.precioUnitario from deleted

declare @cantCompradai decimal(8,2), @precUniti money

select @cantCompradai=inserted.cantidadComprada,

@precUniti=inserted.precioUnitario from inserted

update FACTURAS set valorFact=valorFact-@cantComprada\*@precUnit+@cantCompradai\*@precUniti

where nroFact=@numfact

update PRODUCTOS set cantidadStock=cantidadStock+@cantComprada-@cantCompradai

where codProd=@codProd

update DETALLES set cantidadComprada=7 where nroFact='001'and codProd='P01'