Proyecto base de datos de una tienda

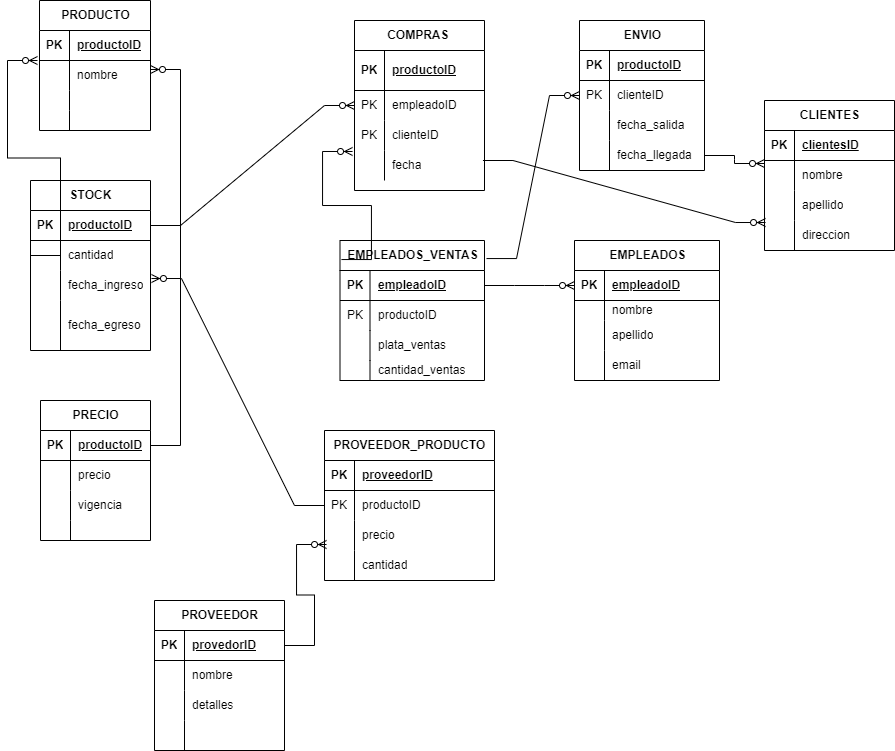
**Introducción al Proyecto de Base de Datos para la Tienda**

En el marco de nuestro proyecto de sistema de gestión para una tienda, hemos desarrollado una base de datos denominada "COMERCIO2". Este sistema busca brindar una solución integral para la administración de productos, clientes, empleados, ventas, compras, envíos, proveedores y sus respectivas interrelaciones.

**Objetivo del Proyecto:**

El principal objetivo de este proyecto es proporcionar una plataforma robusta y eficiente para la gestión de una tienda, facilitando la administración y seguimiento de productos, ventas, compras, clientes y otros aspectos esenciales para el funcionamiento de un comercio.

**Diagrama de entidad:**



**Tablas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***PRODUCTO*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | productoID | varchar(15) | not null | primary key para identificar el producto |
|  | nombre | varchar(50) | not null | nombre del producto |
|  |  |  |  |  |
| ***PRECIO*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | productoID | varchar(15) | not null | primary key para identificar el producto |
|  | precio | float | not null | valor monetario del producto |
|  | vigencia | date | null | vigencia del valor del producto |
|  |  |  |  |  |
| ***STOCK*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | productoID | varchar(15) | not null | primary key para identificar el producto |
|  | fecha\_ingreso | date | not null | fecha en la que ingreso el producto |
|  | fecha\_egreso | date | not null | fecha en la que egreso el producto |
|  | stock | tinyint | not null | cantidad de productos |
|  |  |  |  |  |
| ***EMPLEADOS*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | empleadoID | varchar(9) | not null | primary key para identificar a cada empleado |
|  | nombre | varchar(50) | not null | nombre del empleado |
|  | apellido | varchar(50) | not null | apellido del empleado |
|  | email | varchar(120) | not null | email del empleado |
|  |  |  |  |  |
| ***EMPLEADOS\_VENTAS*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | empleadoID | varchar(9) | not null | primary key para identificar al empleado que realizo la venta |
| FK | productoID | varchar(15) | not null | foreign key para identificar el producto vendido |
|  | plata\_ventas | float | null | dinero de la venta |
|  | cantidad\_ventas | int | null | cantidad de ventas realizadas |
|  |  |  |  |  |
| ***CLIENTES*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | clienteID | varchar(9) | not null | primary key para identificar a cada cliente |
|  | nombre | varchar(50) | not null | nombre del cliente |
|  | apellido | varchar(50) | not null | apellido del cliente |
|  | direccion | varchar(100) | not null | direccion del cliente |
|  |  |  |  |  |
| ***COMPRAS*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | productoID | varchar(15) | not null | primary key para identificar el producto que compraron |
| FK | empleadoID | varchar(9) | not null | foreign key para identificar quien realizo la venta |
| FK | clienteID | varchar(9) | not null | foreign key para identificar quien realizo la compa |
|  | fecha | datatime | null | fecha y hora para identificar cuando se realizo la operación |
|  |  |  |  |  |
| ***ENVIO*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | productoID | varchar(15) | not null | primary key para identificar el producto a enviar |
| FK | clienteID | varchar(9) | not null | foreign key para identificar al cliente que realizo la compra |
|  | fecha\_salida | datetime | null | fecha y hora que se realiza el envio |
|  | fecha\_llegada | datetime | null | fecha y hora que llega el envio |
|  |  |  |  |  |
| ***PROVEEDOR*** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DRESCRIPCION/ATRIBUTO** |
| PK | proveedorID | varchar(9) | not null | primary key para identificar al proveedor |
|  | nombre | varchar(50) | not null | nombre del proveedor |
|  | detalles | varchar(250) | null | campo para describir su servicio |
|  |  |  |  |  |
| **PROVEEDOR\_PRODUCTO** | | | | |
| **PK/FK** | **NOMBRE** | **TIPO DE DATO** | **N** | **DESCRIPCION/ATRIBUTO** |
| PK | proveedorID | varchar(9) | not null | primary key para identificar al proveedor |
| FK | productoId | varchar(15) | not null | foreign key para identificar al producto |
|  | precio | float | not null | valor del producto |
|  | cantidad | int | not null | cantidad de productos disponibles |

**Codigo:**

CREATE SCHEMA comercio;

USE comercio;

CREATE TABLE producto(

productoID VARCHAR(15) NOT NULL PRIMARY KEY,

nombre VARCHAR(50) NOT NULL

);

CREATE TABLE precio(

productoID VARCHAR(15) NOT NULL,

precio FLOAT NOT NULL,

vigencia DATE,

PRIMARY KEY (productoID),

FOREIGN KEY (productoID) REFERENCES producto (productoID)

);

CREATE TABLE stock(

productoID VARCHAR(15) NOT NULL,

fecha\_ingreso DATE NOT NULL,

fecha\_egreso DATE NOT NULL,

stock TINYINT NOT NULL,

PRIMARY KEY (productoID),

FOREIGN KEY (productoID) REFERENCES producto (productoID)

);

CREATE TABLE empleados(

empleadoID VARCHAR(9) NOT NULL PRIMARY KEY,

nombre VARCHAR(50) NOT NULL,

apellido VARCHAR(50) NOT NULL,

email VARCHAR(120) NOT NULL

);

CREATE TABLE empleados\_ventas(

empleadoID VARCHAR(9) NOT NULL,

productoID VARCHAR(15) NOT NULL,

plata\_ventas FLOAT NULL,

cantidad\_ventas INT NULL,

PRIMARY KEY (empleadoID, productoID),

FOREIGN KEY (empleadoID) REFERENCES empleados (empleadoID),

FOREIGN KEY (productoID) REFERENCES producto (productoID)

);

CREATE TABLE clientes(

clienteID VARCHAR(9) NOT NULL PRIMARY KEY,

nombre VARCHAR (50) NOT NULL,

apellido VARCHAR (50) NOT NULL,

direccion VARCHAR (100) NOT NULL

);

CREATE TABLE compras(

productoID VARCHAR(15) NOT NULL,

empleadoID VARCHAR(9) NOT NULL,

clienteID VARCHAR(9) NOT NULL,

fecha DATETIME,

PRIMARY KEY(productoID, empleadoID, clienteID),

FOREIGN KEY (productoID) REFERENCES producto (productoID),

FOREIGN KEY (empleadoID) REFERENCES empleados (empleadoID),

FOREIGN KEY (clienteID) REFERENCES clientes (clienteID)

);

CREATE TABLE envio(

productoID VARCHAR(15) NOT NULL,

clienteID VARCHAR(9) NOT NULL,

fecha\_salida DATETIME,

fecha\_llegada DATETIME,

PRIMARY KEY(productoID, clienteID),

FOREIGN KEY (productoID) REFERENCES producto (productoID),

FOREIGN KEY (clienteID) REFERENCES clientes (clienteID)

);

CREATE TABLE proveedor(

proveedorID VARCHAR(9) NOT NULL PRIMARY KEY,

nombre VARCHAR(50) NOT NULL,

detalles VARCHAR(250) NULL

);

CREATE TABLE proveedor\_producto(

proveedorID VARCHAR (9) NOT NULL,

productoID VARCHAR(15) NOT NULL,

precio FLOAT NOT NULL,

cantidad INT NOT NULL,

PRIMARY KEY (proveedorID, productoID),

FOREIGN KEY (proveedorID) REFERENCES proveedor (proveedorID),

FOREIGN KEY (productoID) REFERENCES producto (productoID)

);

**Insercion de datos:**

USE COMERCIO2;

INSERT INTO producto (productoID, nombre)

VALUES

('1', 'Puerta de madera'),

('2', 'Ventana de aluminio'),

('3', 'Ventiluz de vidrio'),

('4', 'Mosquitero de plástico'),

('5', 'Tornillo de acero'),

('6', 'Grifería cromada'),

('7', 'Portón de hierro'),

('8', 'Chapa de acero inoxidable'),

('9', 'Espejo de cristal'),

('10', 'Mesada de granito');

INSERT INTO precio (productoID, precio, vigencia)

VALUES

('1', 150.00, '2023-05-01'),

('2', 75.00, '2023-10-25'),

('3', 50.00, '2023-02-12'),

('4', 10.00, '2023-04-01'),

('5', 0.50, '2023-03-21'),

('6', 90.00, '2023-06-18'),

('7', 200.00, '2023-07-21'),

('8', 80.00, '2023-04-04'),

('9', 120.00, '2023-09-11'),

('10', 250.00, '2023-01-25');

INSERT INTO stock (productoID, fecha\_ingreso, fecha\_egreso, stock)

VALUES

('1', '2023-01-01', '2023-01-10', 100),

('2', '2023-01-01', '2023-01-15', 200),

('3', '2023-01-01', '2023-01-05', 150),

('4', '2023-01-01', '2023-01-20', 300),

('5', '2023-01-01', '2023-01-25', 5000),

('6', '2023-01-01', '2023-01-30', 100),

('7', '2023-01-01', '2023-02-05', 50),

('8', '2023-01-01', '2023-02-10', 200),

('9', '2023-01-01', '2023-02-15', 300),

('10', '2023-01-01', '2023-02-20', 80);

INSERT INTO empleados (empleadoID, nombre, apellido, email)

VALUES

('1001', 'Juan', 'Perez', 'juan@gmail.com'),

('1002', 'María', 'Gomez', 'maria@gmail.com'),

('1003', 'Carlos', 'Lopez', 'carlos@gmail.com'),

('1004', 'Laura', 'Martinez', 'laura@gmail.com'),

('1005', 'Pedro', 'Rodriguez', 'pedro@gmail.com'),

('1006', 'Ana', 'Sanchez', 'ana@gmail.com'),

('1007', 'Luis', 'Fernandez', 'luis@gmail.com'),

('1008', 'Eva', 'Garcia', 'eva@gmail.com'),

('1009', 'Miguel', 'Diaz', 'miguel@gmail.com'),

('1010', 'Carmen', 'Torres', 'carmen@gmail.com');

INSERT INTO empleados\_ventas (empleadoID, productoID, plata\_ventas, cantidad\_ventas)

VALUES

('1001', '1', 1500.00, 10),

('1002', '2', 750.00, 5),

('1003', '3', 500.00, 7),

('1004', '4', 100.00, 20),

('1005', '5', 5.00, 100),

('1006', '6', 900.00, 8),

('1007', '7', 2000.00, 2),

('1008', '8', 800.00, 10),

('1009', '9', 1200.00, 6),

('1010', '10', 2500.00, 4);

INSERT INTO clientes (clienteID, nombre, apellido, direccion)

VALUES

('2001', 'Luis', 'Gonzalez', 'Bermejo 1232'),

('2002', 'Maria', 'Sanchez', 'Casacuberta 435'),

('2003', 'Pedro', 'Martinez', 'Pehuajo 4574'),

('2004', 'Laura', 'Fernandez', 'Panama 798'),

('2005', 'Carlos', 'Lopez', 'Quindio 980'),

('2006', 'Ana', 'Torres', 'Lamadrid 23525'),

('2007', 'Juan', 'Diaz', 'Linares 14445'),

('2008', 'Eva', 'Garcia', 'Fragio 223'),

('2009', 'Miguel', 'Soto', 'Uruguay 580'),

('2010', 'Carmen', 'Rivera', 'Timoteo 43');

INSERT INTO compras (productoID, empleadoID, clienteID, fecha)

VALUES

('1', '1001', '2001', '2023-01-10 08:15:30'),

('2', '1002', '2002', '2023-01-15 14:30:45'),

('3', '1003', '2003', '2023-01-20 12:00:00'),

('4', '1004', '2004', '2023-01-25 10:45:30'),

('5', '1005', '2005', '2023-02-01 16:20:15'),

('6', '1006', '2006', '2023-02-05 09:30:00'),

('7', '1007', '2007', '2023-02-10 11:55:30'),

('8', '1008', '2008', '2023-02-15 14:45:15'),

('9', '1009', '2009', '2023-02-20 17:10:45'),

('10', '1010', '2010', '2023-02-25 08:30:30');

INSERT INTO envio (productoID, clienteID, fecha\_salida, fecha\_llegada)

VALUES

('1', '2001', '2023-01-12 10:00:00', '2023-01-14 15:30:00'),

('2', '2002', '2023-01-16 09:45:00', '2023-01-19 12:15:00'),

('3', '2003', '2023-01-21 11:30:00', '2023-01-24 16:00:00'),

('4', '2004', '2023-01-26 14:45:00', '2023-01-29 09:30:00'),

('5', '2005', '2023-02-02 12:30:00', '2023-02-04 17:45:00'),

('6', '2006', '2023-02-06 10:15:00', '2023-02-09 15:00:00'),

('7', '2007', '2023-02-11 13:30:00', '2023-02-13 10:15:00'),

('8', '2008', '2023-02-16 15:45:00', '2023-02-19 09:00:00'),

('9', '2009', '2023-02-21 09:30:00', '2023-02-23 14:45:00'),

('10', '2010', '2023-02-26 11:15:00', '2023-02-28 16:30:00');

INSERT INTO proveedor (proveedorID, nombre, detalles)

VALUES

('3001', 'Libermet SRL', 'Proveedor de materiales de construcción'),

('3002', 'Don Pepe SA', 'Proveedor de equipos y herramientas'),

('3003', 'Nino SRL', 'Proveedor de muebles y decoración'),

('3004', 'Alvaro SRL', 'Proveedor de iluminación'),

('3005', 'Quijote SA', 'Proveedor de metales'),

('3006', 'Valo SRL', 'Proveedor de electrónica'),

('3007', 'Fernandez HNOS', 'Proveedor de materiales eléctricos'),

('3008', 'Mujica SA', 'Proveedor de mecanica'),

('3009', 'Cacho INC', 'Proveedor de madera'),

('3010', 'Juanito SRL', 'Proveedor de productos químicos');

INSERT INTO proveedor\_producto (proveedorID, productoID, precio, cantidad)

VALUES

('3001', '1', 1450.00, 100),

('3002', '2', 7000.00, 150),

('3003', '3', 4800.00, 200),

('3004', '4', 800.00, 300),

('3005', '5', 1200.00, 5000),

('3006', '6', 850.00, 250),

('3007', '7', 1900.00, 80),

('3008', '8', 750.00, 120),

('3009', '9', 1100.00, 180),

('3010', '10', 2400.00, 50);

**Creacion de vistas:**

Use COMERCIO2;

/\*vista para los productos y su precio vigente\*/

CREATE VIEW vista\_productos\_precios AS

SELECT p.productoID, p.nombre, precio.precio, precio.vigencia

FROM producto p

JOIN precio ON p.productoID = precio.productoID;

/\* vista ventas empleados\*/

CREATE VIEW vista\_ventas\_empleados AS

SELECT ev.empleadoID, e.nombre, e.apellido, SUM(ev.plata\_ventas) AS total\_ventas

FROM empleados\_ventas ev

JOIN empleados e ON ev.empleadoID = e.empleadoID

GROUP BY ev.empleadoID, e.nombre, e.apellido;

/\*vista compras clientes\*/

CREATE VIEW vista\_compras\_clientes AS

SELECT c.clienteID, cl.nombre, cl.apellido, p.nombre AS producto, c.fecha

FROM compras c

JOIN producto p ON c.productoID = p.productoID

JOIN clientes cl ON c.clienteID = cl.clienteID;

/\*vista para proveedoresy productos\*/

CREATE VIEW vista\_productos\_proveedores AS

SELECT pp.productoID, p.nombre AS producto, pr.nombre AS proveedor, pp.precio, pp.cantidad

FROM proveedor\_producto pp

JOIN producto p ON pp.productoID = p.productoID

JOIN proveedor pr ON pp.proveedorID = pr.proveedorID;

/\*visto stock total\*/

CREATE VIEW vista\_stock\_total AS

SELECT p.productoID, p.nombre, COALESCE(s.stock, 0) AS stock\_actual

FROM producto p

LEFT JOIN (

SELECT productoID, SUM(stock) AS stock

FROM stock

GROUP BY productoID

) s ON p.productoID = s.productoID;

**Funciones:**

use comercio2;

/\*funcion para calcular el precio promedio\*/

DELIMITER //

CREATE FUNCTION calcularPrecioPromedio(productoID VARCHAR(15)) RETURNS FLOAT DETERMINISTIC

BEGIN

DECLARE avg\_price FLOAT;

SELECT AVG(precio) INTO avg\_price

FROM precio

WHERE productoID = productoID;

RETURN avg\_price;

END;

//

DELIMITER ;

/\* funcion para obtener stock\*/

DELIMITER //

CREATE FUNCTION obtenerStock(productoID VARCHAR(15)) RETURNS TINYINT DETERMINISTIC

BEGIN

DECLARE stock\_actual TINYINT;

SELECT COALESCE(stock, 0) INTO stock\_actual

FROM stock

WHERE productoID = productoID;

RETURN stock\_actual;

END;

//

DELIMITER ;

**Stored procedures:**

use comercio2;

/\* Este procedimiento te permite consultar una tabla especificada de forma ordenada.

tablaNombre: El nombre de la tabla que deseas consultar.

campoOrden: El nombre del campo por el cual deseas ordenar los resultados.

ordenDireccion: La direccion del ordenamiento ('ASC' para ascendente, 'DESC' para descendente)\*/

DELIMITER //

CREATE PROCEDURE ConsultaOrdenada(IN tablaNombre VARCHAR(255), IN campoOrden VARCHAR(255), IN ordenDireccion VARCHAR(4))

BEGIN

SET @query = CONCAT('SELECT \* FROM ', tablaNombre, ' ORDER BY ', campoOrden, ' ', ordenDireccion);

PREPARE stmt FROM @query;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

END;

//

DELIMITER ;

/\*Este procedimiento te permite insertar o eliminar un registro especifico en una tabla.

tablaNombre: El nombre de la tabla en la que deseas realizar la accion.

accion: La accion a realizar ('INSERT' para insertar, 'DELETE' para eliminar).

registroID: El ID del registro que deseas insertar o eliminar\*/

DELIMITER //

CREATE PROCEDURE InsertarEliminarRegistro(IN tablaNombre VARCHAR(255), IN accion VARCHAR(7), IN registroID VARCHAR(15))

BEGIN

IF accion = 'INSERT' THEN

/\*Insertar un nuevo registro\*/

INSERT INTO tablaNombre (productoID) VALUES (registroID);

SELECT 'Registro insertado en la tabla ', tablaNombre;

ELSEIF accion = 'DELETE' THEN

-- Eliminar un registro existente

DELETE FROM tablaNombre WHERE productoID = registroID;

SELECT 'Registro eliminado de la tabla ', tablaNombre;

ELSE

SELECT 'Acción no válida. Use "INSERT" o "DELETE".';

END IF;

END;

//

DELIMITER ;

/\*Ejemplo de consulta ordenada ascendente en la tabla producto\*/

CALL ConsultaOrdenada('producto', 'productoID', 'ASC');

/\*Ejemplo de consulta ordenada (descendente) en la tabla empleados\*/

CALL ConsultaOrdenada('empleados', 'nombre', 'DESC');

/\*Ejemplo de insercion de un registro en la tabla stock\*/

CALL InsertarEliminarRegistro('stock', 'INSERT', '11');

/\*Ejemplo de eliminación de un registro en la tabla clientes\*/

CALL InsertarEliminarRegistro('clientes', 'DELETE', '2003');

**Triggers:**

use comercio2;

/\*tabla de registro de operaciones\*/

CREATE TABLE producto\_insert\_log (

    logID INT AUTO\_INCREMENT PRIMARY KEY,

    accion VARCHAR(10) NOT NULL,

    usuario VARCHAR(50) NOT NULL,

    fecha DATE NOT NULL,

    hora TIME NOT NULL,

    productoID VARCHAR(15)

);

/\*trigger before en la tabla producto\*/

DELIMITER //

CREATE TRIGGER producto\_before\_insert

BEFORE INSERT ON producto

FOR EACH ROW

BEGIN

    /\*Registro de accion previa a la insercion,

registrara el usuario, la fecha y hora antes de la insercion\*/

    INSERT INTO producto\_insert\_log (accion, usuario, fecha, hora)

    VALUES ('Insert', USER(), CURDATE(), CURTIME());

END;

//

DELIMITER ;

/\*triggers de after en la tabla producto\*/

DELIMITER //

CREATE TRIGGER producto\_after\_insert

AFTER INSERT ON producto

FOR EACH ROW

BEGIN

    /\*registro de accion posterior a la insercion,

    registrara el usuario, la fecha, la hora y el ID del producto insertado\*/

    INSERT INTO producto\_insert\_log (accion, usuario, fecha, hora, productoID)

    VALUES ('Insert', USER(), CURDATE(), CURTIME(), NEW.productoID);

END;

//

DELIMITER ;

**Sentencias:**

Use comercio2;

 /\*transaccion para la tabla producto\*/

START TRANSACTION;

/\*eliminar algunos registros en la tabla producto\*/

DELETE FROM producto WHERE productoID IN ('1', '2', '3');

/\*ROLLBACK;\*/

/\*COMMIT;\*/

/\*transaccion para la tabla producto\*/

START TRANSACTION;

/\*insertar registros en la tabla producto\*/

INSERT INTO producto (productoID, nombre) VALUES

    ('11', 'pala'),

    ('12', 'martillo'),

    ('13', 'destornillador'),

    ('14', 'llave francesa'),

    ('15', 'pico'),

    ('16', 'soldadora'),

    ('17', 'maza'),

    ('18', 'pegamento');

/\*agregar SAVEPOINT\*/

SAVEPOINT savepoint\_4;

/\*insertar mas registros en la tabla producto\*/

INSERT INTO producto (productoID, nombre) VALUES

    ('19', 'tornillo'),

    ('20', 'clavo'),

    ('21', 'espatula'),

    ('22', 'lija'),

    ('23', 'pincel'),

    ('24', 'brocha'),

    ('25', 'cortina'),

    ('26', 'tiner');

/\*agregar SAVEPOINT despues de la insercion\*/

SAVEPOINT savepoint\_8;

/\*sentencia para eliminar el SAVEPOINT de los primeros 4 registros insertados\*/

/\*RELEASE SAVEPOINT savepoint\_4;\*/

/\*sentencia para eliminar el SAVEPOINT de los últimos 4 registros\*/

/\*RELEASE SAVEPOINT savepoint\_8;\*/

/\*ROLLBACK;\*/

/\*COMMIT;\*/