**ITSE - TECNICATURA SUPERIOR EN PROGRAMACIÓN**

**Base de Datos II**

**Trabajo Practico 3**

**Procedimientos Almacenados (SP)**

**Alumno: Exequiel Barco**

**Actividades**

**En base a la Base de Datos de “Gestión Ventas 1.0” creada en el TP 1 crear los siguientes SP y**

**ejecutarlos**

**Actividades y desarrollo**

1. Crear un SP ClienteAlta que reciba como parámetros de entrada los campos del registro cliente y lo agregue a la tabla.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ClienteAlta(

in vClienteNom varchar(50),

in vClienteCorreo varchar(50),

in vClienteDNI varchar(9))

BEGIN

INSERT INTO

Cliente (ClienteNom, ClienteCorreo,ClienteDNI)

VALUES (vClienteNom, vClienteCorreo, vClienteDNI);

END //

DELIMITER ;

call ClienteAlta('Prueba Alta', 'PruebaAlta@gmail.com', '411111111');

1. Crear un SP ClienteBaja que reciba como parámetros de entrada el documento del registro cliente y lo borre de la tabla.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ClienteBaja(

in vDNI varchar(9))

BEGIN

DELETE FROM

Cliente

WHERE ClienteDNI = vDNI;

END //

DELIMITER ;

call ClienteBaja('411111111');

1. Crear un SP ClienteModif que reciba como parámetros de entrada todos los campos del registro cliente y modifique solo los campos informados.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ClienteModif(

in vClienteNom varchar(50),

in vClienteCorreo varchar(50),

in vClienteDNI varchar(9))

BEGIN

UPDATE cliente

SET

ClienteNom = vClienteNom,

ClienteCorreo = vClienteCorreo,

ClienteDNI = vClienteDNI

WHERE ClienteDNI = vClienteDNI;

END //

DELIMITER ;

call ClienteModif('Barco Exequiel','BarcoExequiel@gmail.com','411111111');

1. Crear un SP ClienteConsulta que reciba como parámetros de entrada el documento del cliente y muestre todos sus datos.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ClienteConsulta(

in vClienteDNI varchar(9))

BEGIN

SELECT

ClienteNom AS Cliente,

ClienteCorreo as Correo,

ClienteDNI as DNI

FROM cliente

WHERE ClienteDNI = vClienteDNI;

END //

DELIMITER ;

call ClienteConsulta('411111111');

1. Crear un SP ClienteActualizacion que reciba como parámetros de entrada los campos del registro cliente y permita hacer Altas, Bajas, Modificaciones o Consulta.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ClienteActualizacion(

in parametro int,

in vClienteNom varchar(50),

in vClienteCorreo varchar(50),

in vClienteDNI varchar(9))

BEGIN

IF parametro = 1 THEN

call ClienteAlta(vClienteNom, vClienteCorreo, vClienteDNI);

END IF;

IF parametro = 2 THEN

call ClienteBaja(vClienteDNI);

END IF;

IF parametro = 3 THEN

call ClienteModif(vClienteNom, vClienteCorreo, vClienteDNI);

END IF;

IF parametro = 4 THEN

call ClienteConsulta(vClienteDNI);

END IF;

END //

DELIMITER ;

call ClienteActualizacion(1,'Prueba Alta','PruebaAlta@gmail','40000017');

call ClienteActualizacion(2, 'Baja Cliente', 'BajaaCliente@gmail.com','40000017');

call ClienteActualizacion(3,'ActualizacionCliente','ActualizacionCliente@gmail','40000017');

call ClienteActualizacion(4, 'Consulta Cliente', 'ConsultaCliente@gmail.com','40000017');

1. Crear un SP llamado “ClientesInsertar” que reciba como parámetros de entrada los campos Doc, Nom, Dom (o los campos que posea su tabla clientes) e inserte un registro en la tabla Clientes.

Ejecutar el SP devolviendo el último número de ClienteId insertado en un parámetro de salida IdCliente.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ClienteInsertar(

in vClienteNom varchar(50),

in vClienteCorreo varchar(50),

in vClienteDNI varchar(9))

BEGIN

INSERT INTO

cliente (ClienteNom,ClienteCorreo,ClienteDNI)

VALUES (vClienteNom,vClienteCorreo,vClienteDNI);

SELECT ClienteID AS ID FROM dbventas.cliente where ClienteDNI = vClienteDNI;

END //

DELIMITER ;

call ClienteInsertar('ClienteNuevo','ClienteNuevo@gmail.com','40000020');

1. Crear un SP llamado “ClientesCantFacturas” que devuelva la cantidad de facturas de un cliente determinado.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ClienteCantFacturas(

in vClienteDNI varchar(9))

BEGIN

SELECT

Cliente.ClienteDNI AS DNI,

Cliente.ClienteNom as Cliente,

Cliente.ClienteCorreo AS Correo,

count(VentaID) AS Facturas

FROM Venta

INNER JOIN Cliente ON Venta.ClienteID = cliente.ClienteId

WHERE Cliente.ClienteDNI = vClienteDNI;

END //

DELIMITER ;

call ClienteCantFacturas('42698355');

1. Crear un SP llamado “ProductoConsulta” que permita consultar todos los productos por nombre similar al valor ingresado y devuelva la cantidad de productos.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE ConsultaProducto(

in Filtro VARCHAR(30))

BEGIN

declare NombreSimilar varchar(30);

set NombreSimilar = Filtro;

SELECT ProdDesc AS Producto,

ProdStock as Stock

FROM Producto

WHERE ProdDesc LIKE NombreSimilar;

END //

DELIMITER ;

call ConsultaProducto('%hel%');

1. Crear un SP llamado “BajaFacturaDeCliente” que permita borrar la factura de un cliente.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE BajaFacturaDeCliente(

in vVentaID int)

BEGIN

DELETE

FROM DetalleVenta

Where VentaID = vVentaID;

DELETE

FROM Venta

WHERE

VentaID = vVentaID;

END //

DELIMITER ;

call BajaFacturaDeCliente('9');

1. Crear un SP llamado “BajaFacturaDeCliente2” que permita borrar la factura de un cliente y actualice en Productos.

**Respuesta:**

DELIMITER //

CREATE PROCEDURE BajaFacturaDeCliente2Cursor(in vVentaID INT)

BEGIN

DECLARE finCursor INT DEFAULT FALSE;

DECLARE IDVenta, IDProd, cantProd INT;

DECLARE Cursor1 CURSOR FOR SELECT VentaID, ProdID, ProdCantidad FROM

DetalleVenta WHERE VentaID = vVentaID;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finCursor = TRUE;

OPEN Cursor1;

FETCH Cursor1 INTO IDVenta, IDProd, cantProd;

WHILE finCursor = FALSE DO

UPDATE Producto SET ProdStock = ProdStock + cantProd WHERE ProdID = IDProd;

FETCH Cursor1 INTO IDVenta, IDProd, cantProd;

END WHILE;

DELETE FROM DetalleVenta WHERE VentaID = vVentaID;

DELETE FROM Venta WHERE VentaID = vVentaID;

CLOSE Cursor1;

END //

DELIMITER ;

call BajaFacturaDeCliente2Cursor(10);