**PROCEDIMIENTOS ALMACENADOS**

**Subprogramas**

**Son importantes para no repetir instrucciones o por cuestión de seguridad**

**Trabajan con:**

**Select Update Insert Delete**

**Procedimiento para insertar una ciudad**

delimiter //

create procedure insciudad(in \_idciudad varchar(5), in \_descripcion varchar(30))

Begin

insert into ciudad values(\_idciudad,\_descripcion);

end

//

Delimiter ;

**Ejecutar un procedimiento almacenado**

Call insciudad(‘106’,’Marinilla’)

**Procedimiento para consultar las ciudades**

delimiter //

create procedure consciudad()

Begin

Select \* from ciudad;

end

//

Delimiter ;

**Procedimiento para eliminar una ciudad**

delimiter //

create procedure eliciudad (in \_idciudad varchar(5))

begin

delete from ciudad c where c.Idciudad = \_idciudad

end

//

delimiter ;

**Procedimiento para modificar una ciudad**

delimiter //

create procedure modiciudad(in \_descripcion varchar(30), in \_idciudad varchar(5))

Begin

update ciudad c set descripcion = \_descripcion where c.Idciudad = \_idciudad;

end

//

delimiter ;

**Ver los procedimientos almacenados creados**

Show procedure status; (visualiza los procedimientos creados)

**Para borrar un procedimiento**

Drop procedure y el nombre del procedimiento.

**Ejercicios**

* Crear un procedimiento almacenado que permita visualizar la información de las facturas por una nro de factura específico
* Crear un procedimiento almacenada que permite eliminar una factura específica
* Crear un procedimiento almacenado que permita el apellido Mesa por Marulanda del cliente con idcliente 2002
* Crear un procedimiento almacenado que permita insertar un registro de cliente

**Introducción a trigger**

Los triggers o disparadores, son procedimientos automáticos que se activan después de una acción, esta acción puede ser un insert, delete o update

Se trabaja con dos identificadores que son after (despues de) before (antes de)

El after es el caso más común de los triggers, se activa después de una sentencia y afecta solo tablas distintas a la que lo esta invocando, situación que no sucede con el before que solo afecta la tabla que lo invoca

La sintaxis general es

delimiter //

create trigger nombre after (insert, delete, update) on tabla

for each row //recorre fila por fila (arreglos o matrices)

begin

sentencia

end

//

delimiter ;

Ejemplo crear un trigger que permita actualizar la comisión de los vendedores cada vez que se hagan ventas

ejemplo

delimiter //

create trigger valorcomision after insert on factura

for each row

begin

update vendedor v set comision = (select (sum(f.Valor\_total)-sum(f.iva)) \* 0.1 from factura f group by f.Idvendedor having f.Idvendedor = v.Idvendedor);

end

//

delimiter ;

Ejercicio 2

Crear un trigger que permita actualizar la existencia cada vez que se realicen ventas de productos.