**AP06-AA7-EV04 Construcción de elementos de programación para el proyecto de formación.**

**Mauricio Andrés Pulgarín Araque**

**Husney Bayona Rincón**

**Cristian Leonardo Ospina Hamon**

**SENIGRAFT**

**2019**

**INTRODUCCIÓN**

En la presente actividad se realizan los correspondiente puntos 10 procedimientos almacenados, 5 funciones y 5 triggers correspondientes al proyecto.

**Tabla de Contenido**

Procedimientos Almacenados……………………………………………………………...4

Funciones…………………………………………………………………………………..8

Triggers……………………………………………………………………………………13

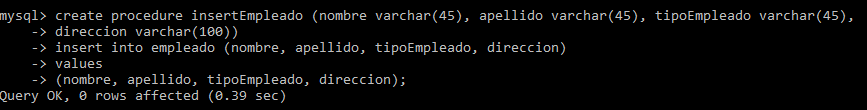
create procedure insertEmpleado (nombre varchar(45), apellido varchar(45), tipoEmpleado varchar(45),

-> direccion varchar(100))

-> insert into empleado (nombre, apellido, tipoEmpleado, direccion)

-> values

-> (nombre, apellido, tipoEmpleado, direccion);



---------------------------------------------------------------------------------------------------------------------------

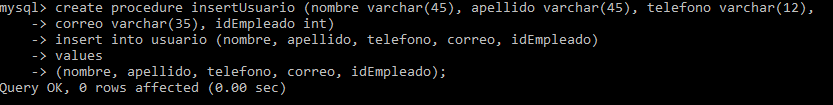
create procedure insertUsuario (nombre varchar(45), apellido varchar(45), telefono varchar(12),

-> correo varchar(35), idEmpleado int)

-> insert into usuario (nombre, apellido, telefono, correo, idEmpleado)

-> values

-> (nombre, apellido, telefono, correo, idEmpleado);



---------------------------------------------------------------------------------------------------------------------------

create procedure insertPaciente (cedula varchar(15), nombre varchar(45), apellido varchar(45),

-> informe varchar(200), direccion varchar(45), horaRecogida time, horaEntrega time, lugarInicio varchar(45),

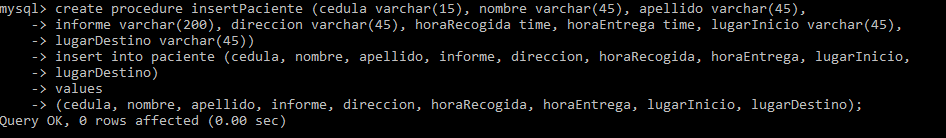
-> lugarDestino varchar(45))

-> insert into paciente (cedula, nombre, apellido, informe, direccion, horaRecogida, horaEntrega, lugarInicio,

-> lugarDestino)

-> values

-> (cedula, nombre, apellido, informe, direccion, horaRecogida, horaEntrega, lugarInicio, lugarDestino);



---------------------------------------------------------------------------------------------------------------------------

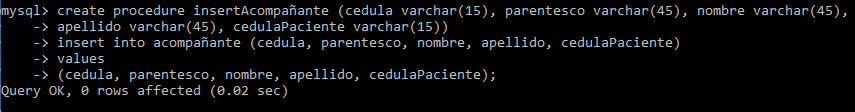
create procedure insertAcompañante (cedula varchar(15), parentesco varchar(45), nombre varchar(45),

-> apellido varchar(45), cedulaPaciente varchar(15))

-> insert into acompañante (cedula, parentesco, nombre, apellido, cedulaPaciente)

-> values

-> (cedula, parentesco, nombre, apellido, cedulaPaciente);



---------------------------------------------------------------------------------------------------------------------------

create procedure insertAmbulancia (placa varchar(10), seguros varchar(80), idEstado int, camillero int,

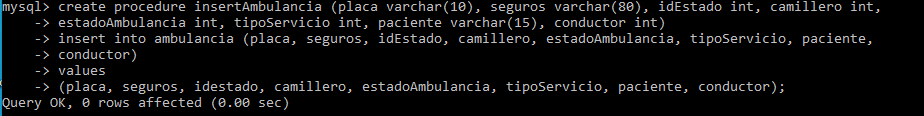
-> estadoAmbulancia int, tipoServicio int, paciente varchar(15), conductor int)

-> insert into ambulancia (placa, seguros, idEstado, camillero, estadoAmbulancia, tipoServicio, paciente,

-> conductor)

-> values

-> (placa, seguros, idestado, camillero, estadoAmbulancia, tipoServicio, paciente, conductor);

---------------------------------------------------------------------------------------------------------------------------

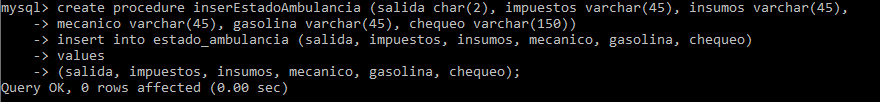
create procedure inserEstadoAmbulancia (salida char(2), impuestos varchar(45), insumos varchar(45),

-> mecanico varchar(45), gasolina varchar(45), chequeo varchar(150))

-> insert into estado\_ambulancia (salida, impuestos, insumos, mecanico, gasolina, chequeo)

-> values

-> (salida, impuestos, insumos, mecanico, gasolina, chequeo);

---------------------------------------------------------------------------------------------------------------------------

create procedure insertInformes (estadoPaciente varchar(250), estadoAmbulancia varchar(150),

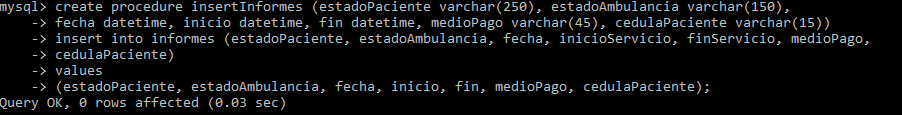
-> fecha datetime, inicio datetime, fin datetime, medioPago varchar(45), cedulaPaciente varchar(15))

-> insert into informes (estadoPaciente, estadoAmbulancia, fecha, inicioServicio, finServicio, medioPago,

-> cedulaPaciente)

-> values

-> (estadoPaciente, estadoAmbulancia, fecha, inicio, fin, medioPago, cedulaPaciente);

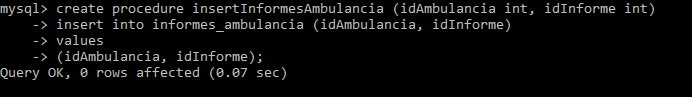
---------------------------------------------------------------------------------------------------------------------------

create procedure insertInformesAmbulancia (idAmbulancia int, idInforme int)

-> insert into informes\_ambulancia (idAmbulancia, idInforme)

-> values

-> (idAmbulancia, idInforme);

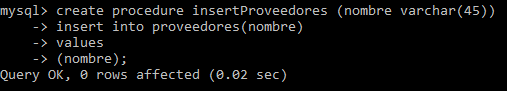
---------------------------------------------------------------------------------------------------------------------------

create procedure insertProveedores (nombre varchar(45))

-> insert into proveedores(nombre)

-> values

-> (nombre);



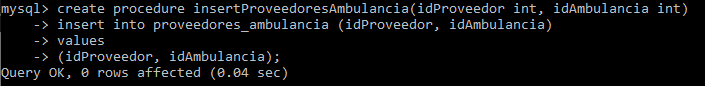
---------------------------------------------------------------------------------------------------------------------------

create procedure insertProveedoresAmbulancia(idProveedor int, idAmbulancia int)

-> insert into proveedores\_ambulancia (idProveedor, idAmbulancia)

-> values

-> (idProveedor, idAmbulancia);

---------------------------------------------------------------------------------------------------------------------------

**Funciones:**

DELIMITER //

mysql> CREATE FUNCTION totalEmpleados()

-> RETURNS INT

-> BEGIN

-> DECLARE total INT;

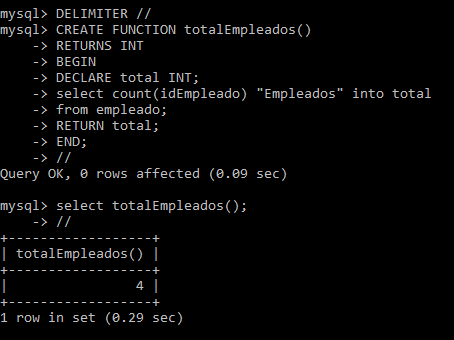
-> select count(idEmpleado) "Empleados" into total

-> from empleado;

-> RETURN total;

-> END;

-> //



---------------------------------------------------------------------------------------------------------------------------

DELIMITER //

mysql> CREATE FUNCTION totalAmbulancias()

-> RETURNS INT

-> BEGIN

-> DECLARE total INT;

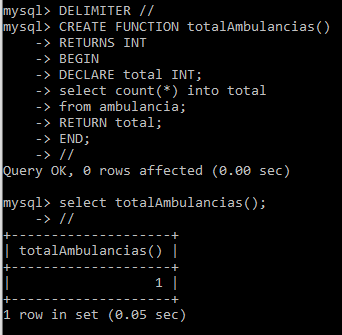
-> select count(\*) into total

-> from ambulancia;

-> RETURN total;

-> END;

-> //



---------------------------------------------------------------------------------------------------------------------------

DELIMITER //

mysql> CREATE FUNCTION totalUsuarios()

-> RETURNS INT

-> BEGIN

-> DECLARE total INT;

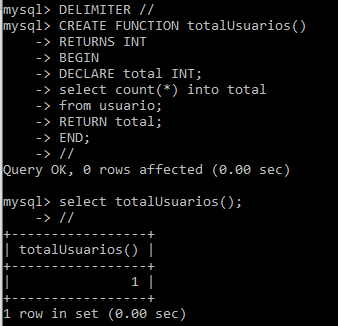
-> select count(\*) into total

-> from usuario;

-> RETURN total;

-> END;

-> //



---------------------------------------------------------------------------------------------------------------------------

DELIMITER //

mysql> CREATE FUNCTION pacienteIdxCedula(cedula varchar(15))

-> RETURNS INT

-> BEGIN

-> DECLARE id INT;

-> select idInformes into id

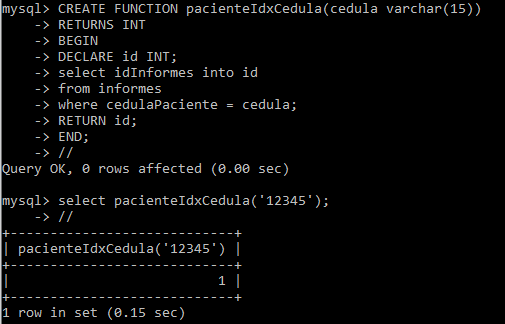
-> from informes

-> where cedulaPaciente = cedula;

-> RETURN id;

-> END;

-> //



---------------------------------------------------------------------------------------------------------------------------

DELIMITER //

mysql> CREATE FUNCTION totalPacientes()

-> RETURNS INT

->

-> BEGIN

-> DECLARE total INT;

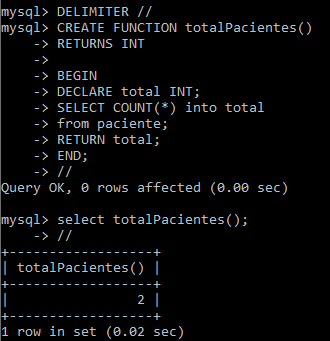
-> SELECT COUNT(\*) into total

-> from paciente;

-> RETURN total;

-> END;

-> //



---------------------------------------------------------------------------------------------------------------------------

Triggers:

Actualización Usuarios:

create table registroUsuarios(

-> idRegistro int primary key auto\_increment,

-> idUsuario\_antiguo int,

-> nombre\_antiguo varchar(45),

-> apelido\_antiguo varchar(45),

-> telefono\_antiguo varchar(12),

-> correo\_antiguo varchar(35),

-> idEmpleado\_antiguo int,

-> idUsuario\_nuevo int,

-> nombre\_nuevo varchar(45),

-> apellido\_nuevo varchar(45),

-> telefono\_nuevo varchar(45),

-> correo\_nuevo varchar(35),

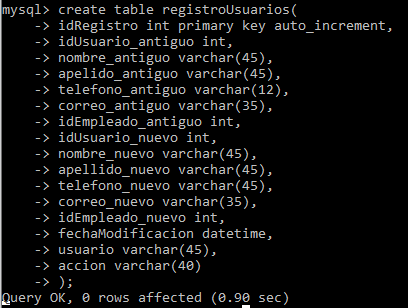
-> idEmpleado\_nuevo int,

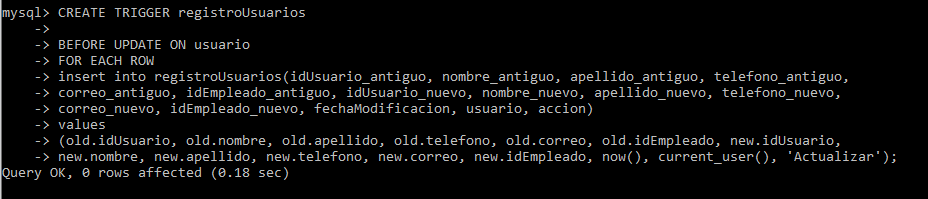
-> fechaModificacion datetime,

-> usuario varchar(45),

-> accion varchar(40)

-> );





---------------------------------------------------------------------------------------------------------------------------

Usuarios Eliminados

create table usuariosEliminados(

-> idRegistro int primary key auto\_increment,

-> idUsuario\_eliminado int,

-> nombre\_eliminado varchar(45),

-> apellido\_eliminado varchar(45),

-> telefono\_eliminado varchar(12),

-> correo\_eliminado varchar(35),

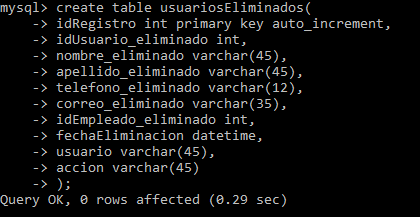
-> idEmpleado\_eliminado int,

-> fechaEliminacion datetime,

-> usuario varchar(45),

-> accion varchar(45)

-> );



CREATE TRIGGER registroUsuariosEliminados

-> BEFORE DELETE ON usuario

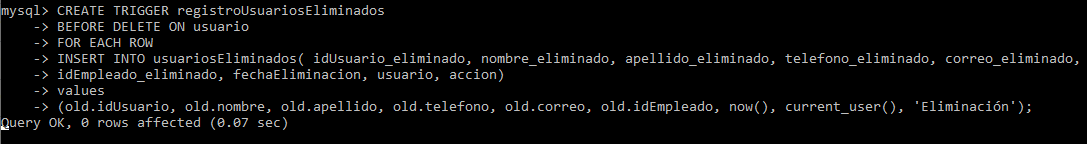
-> FOR EACH ROW

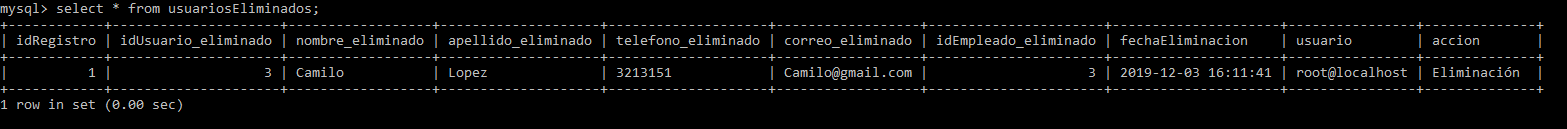
-> INSERT INTO usuariosEliminados( idUsuario\_eliminado, nombre\_eliminado, apellido\_eliminado, telefono\_eliminado, correo\_eliminado,

-> idEmpleado\_eliminado, fechaEliminacion, usuario, accion)

-> values

-> (old.idUsuario, old.nombre, old.apellido, old.telefono, old.correo, old.idEmpleado, now(), current\_user(), 'Eliminación');





---------------------------------------------------------------------------------------------------------------------------

Empleados Actualizados:

create table registroEmpleadosActualizados(

-> idRegistro int primary key auto\_increment,

-> idEmpleado\_antiguo int,

-> nombre\_antiguo varchar(45),

-> apellido\_antiguo varchar(45),

-> tipoEmpleado\_antiguo varchar(45),

-> direccion\_angitua varchar(100),

-> idEmpleado\_nuevo int,

-> nombre\_nuev varchar(45),

-> apellido\_nuevo varchar(45),

-> tipoEmpleado\_nuevo varchar(45),

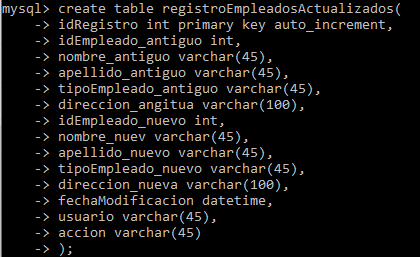
-> direccion\_nueva varchar(100),

-> fechaModificacion datetime,

-> usuario varchar(45),

-> accion varchar(45)

-> );



CREATE TRIGGER registroEmpleados

-> BEFORE UPDATE ON empleado

-> FOR EACH ROW

-> INSERT INTO registroEmpleadosActualizados (idEmpleado\_antiguo, nombre\_antiguo, apellido\_antiguo,

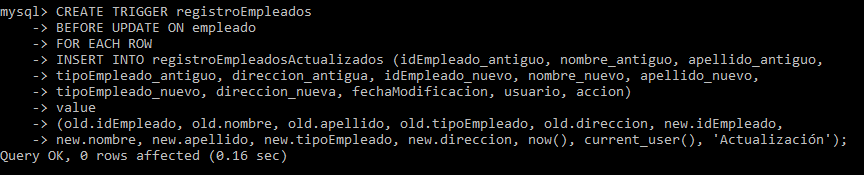
-> tipoEmpleado\_antiguo, direccion\_antigua, idEmpleado\_nuevo, nombre\_nuevo, apellido\_nuevo,

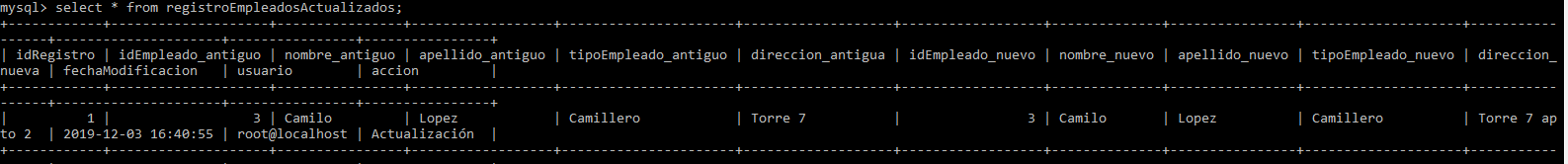
-> tipoEmpleado\_nuevo, direccion\_nueva, fechaModificacion, usuario, accion)

-> value

-> (old.idEmpleado, old.nombre, old.apellido, old.tipoEmpleado, old.direccion, new.idEmpleado,

-> new.nombre, new.apellido, new.tipoEmpleado, new.direccion, now(), current\_user(), 'Actualización');



---------------------------------------------------------------------------------------------------------------------------

Empleados Eliminados

CREATE TABLE empleadosEliminados(

-> idRegistro int primary key auto\_increment,

-> idEmpleado\_eliminado int,

-> nombre\_eliminado varchar(45),

-> apellido\_eliminado varchar(45),

-> tipoEmpleado\_eliminado varchar(45),

-> direccion\_eliminada varchar(100)

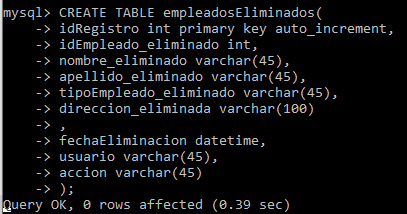
-> ,

-> fechaEliminacion datetime,

-> usuario varchar(45),

-> accion varchar(45)

-> );



CREATE TRIGGER registroEmpleadosEliminados

-> BEFORE DELETE ON empleado

-> FOR EACH ROW

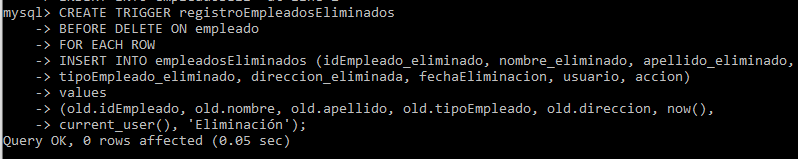
-> INSERT INTO empleadosEliminados (idEmpleado\_eliminado, nombre\_eliminado, apellido\_eliminado,

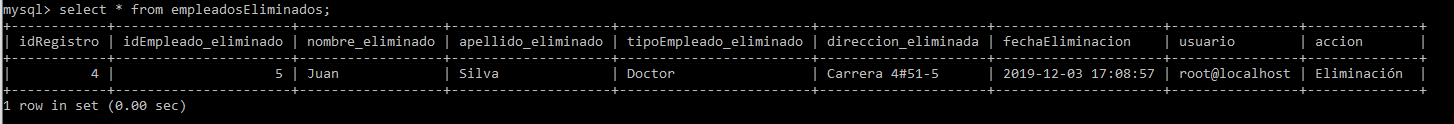
-> tipoEmpleado\_eliminado, direccion\_eliminada, fechaEliminacion, usuario, accion)

-> values

-> (old.idEmpleado, old.nombre, old.apellido, old.tipoEmpleado, old.direccion, now(),

-> current\_user(), 'Eliminación');



---------------------------------------------------------------------------------------------------------------------------

Ingreso de Nuevos Empleados:

CREATE TABLE nuevosEmpleados(

-> idRegistro int primary key auto\_increment,

-> idEmpleado\_nuevo int,

-> nombre\_nuevo varchar(45),

-> apellido\_nuevo varchar(45),

-> tipoEmpleado\_nuevo varchar(45),

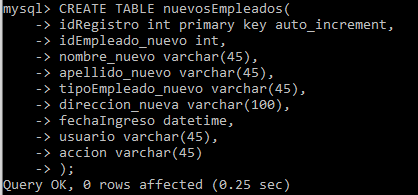
-> direccion\_nueva varchar(100),

-> fechaIngreso datetime,

-> usuario varchar(45),

-> accion varchar(45)

-> );



CREATE TRIGGER registroNuevosEmpelados

-> AFTER INSERT ON empleado

-> FOR EACH ROW

-> INSERT INTO nuevosEmpleados (idEmpleado\_nuevo, nombre\_nuevo, apellido\_nuevo,

-> tipoEmpleado\_nuevo, direccion\_nueva, fechaIngreso, usuario, accion)

-> values

-> (new.idEmpleado, new.nombre, new.apellido, new.tipoEmpleado, new.direccion,

-> now(), current\_user(), 'Ingreso');

