TAREA HABILITADORA HITO 4

NOMBRE: Juan Elian Alvarez Vallejos

CARRERA: Ing. De Sistemas

MATERIA: Base de Datos II

DOCENTE: William Roddy Barra

GESTION: II – 2019

|  |
| --- |
| CREATE DATABASE ONG;  USE ONG;  CREATE TABLE proyecto  (  id\_proy integer auto\_increment primary key not null,  nombreProy varchar(100),  tipoproy varchar(30)  );  CREATE TABLE detalle\_proyecto  (  id\_dp INTEGER auto\_increment primary key  not null,  id\_per INTEGER,  id\_proy integer,  foreign key(id\_per) REFERENCES persona (id\_per),  foreign key (id\_proy) REFERENCES proyecto (id\_proy)  );  ALTER TABLE provincia ADD column id\_dep integer;  ALTER TABLE provincia ADD FOREIGN KEY (id\_dep) REFERENCES departamento(id\_dep);  CREATE TABLE persona  (    id\_per integer auto\_increment primary key not null,    nombre varchar(20),    apellidos varchar(50),    fecha\_nac date,    edad int,    email varchar(50),    id\_dep INTEGER,    id\_prov integer,    foreign key (id\_dep) references DEPARTAMENTO (id\_dep),    foreign key (id\_prov) references provincia (id\_prov)  );  CREATE TABLE DEPARTAMENTO  (    id\_dep integer auto\_increment primary key not null,    nombre varchar(50)  );  CREATE TABLE provincia  (    id\_prov integer auto\_increment primary key not null,    nombre varchar(50),    id\_dep integer,    foreign key (id\_dep) references DEPARTAMENTO (id\_dep)  ); |

EJERCICIO 1

Crear una Vista.

● La Vista debe de llamarse personasMujeresDepartamento.

● La consulta de la vista debe reflejar como campos nombres y apellidos concatenados, la edad y la fecha de

nacimiento.

● Obtener todas las personas del sexo femenino que hayan nacido en el departamento de Cochabamba en

donde la fecha sea: ○ fecha\_nac = '1993-10-10'.

|  |
| --- |
| DROP FUNCTION ObtenerPersona;  CREATE FUNCTION ObtenerPersona(fecha\_nac date,departamento TEXT,sexo TEXT)  RETURNS BOOLEAN  BEGIN  DECLARE result BOOLEAN DEFAULT FALSE;  IF (fecha\_nac = '1993-10-10' and departamento ='Cochabamba' AND sexo = 'femenino') THEN Set result= true;  end if;  RETURN result;    end;    CREATE VIEW personasMujeresDepartamento AS  SELECT CONCAT(p.nombres,'-',p.apellidos,'-',p.f\_nacimiento,'',d.nom\_departamento) As Detalle\_persona  FROM persona p INNER JOIN departamento d on p.id\_depto = d.id\_departamento WHERE ObtenerPersona(p.f\_nacimiento,d.nom\_departamento,p.sexo);    SELECT v1.Detalle\_persona FROM personasMujeresDepartamento AS v1; |

EJERCICIO 2

Crear una TRIGGER.

● El trigger debe de llamarse backupPersonasUpdate.

● El evento debe de ejecutarse en un BEFORE UPDATE.

● Crear un tabla llamada backupPersonasUpdate.

● Esta nueva tabla tiene que tener 2 campos oldNombre y newNombre.

● Cada vez que se modifique el registro de una persona, se debe de insertar un nuevo registro en la tabla

backupPersonasUpdate, en donde el primer campo es el nombre que se está modificando y el segundo campo es el nuevo nombre.

|  |
| --- |
| DROP TABLE backupPersonasUpdate;  CREATE TABLE backupPersonasUpdate ( operation CHAR(1) NOT NULL, -- ('D', 'U', 'I') stamp TIMESTAMP NOT NULL, userid TEXT NOT NULL, hostname TEXT NOT NULL, oldNombre varchar(50), newNombre VARCHAR(50), id\_Persona integer not null );  DROP TRIGGER backupPersonasUpdate;  CREATE TRIGGER backupPersonasUpdate BEFORE UPDATE ON persona FOR EACH ROW  BEGIN  INSERT INTO backupPersonasUpdate(operation, stamp, userid, hostname,oldNombre, newNombre, id\_Persona)  SELECT 'U' ,now(),user(),@@hostname,old.nombres,NEW.nombres,OLD.id\_persona; end;    UPDATE persona set nombres= 'Javier' WHERE id\_persona =1; |

EJERCICIO 3

Crear una TRIGGER.

● El trigger debe de llamarse calculaEdad.

● El evento debe de ejecutarse en un BEFORE INSERT.

● Cada vez que se inserte un registro en la tabla PERSONA, el trigger debe de calcular la edad en función a la

edad en función a la fecha de nacimiento.

|  |
| --- |
| ALTER TABLE persona ADD COLUMN edad integer;  CREATE TRIGGER calculaedad  BEFORE INSERT ON persona #NEW FOR EACH ROW BEGIN  DECLARE edad INTEGER DEFAULT 0;  SELECT TIMESTAMPDIFF(YEAR ,NEW.f\_nacimiento, CURDATE())  INTO edad;  SET NEW.edad =edad; end;    INSERT INTO persona ( nombres, apellidos, direccion, f\_nacimiento, telefono, email, sexo, id\_depto, id\_prov, edad)  VALUES ('Roberto','Pena','Almafuerte 123','1991-1028',49118657,'robertito@gmail.com','masculino',3,1,0); |

EJERCICIO 4

Crear un TRIGGER BEFORE o AFTER INSERT para la tabla PROYECTO.

● El nombre del TRIGGER deberá ser triggerInsert\_Proyecto

● Deberá de crear una tabla de AUDITORIA en donde esta tabla deberá de tener 2 columnas. ○ El 1er campo debe de guardar el nuevo idProy insertado. ○ El 2do campo debe de guardar el nombre de proyecto y tipo de proyecto concatenados separados por

un espacio.

■ Ejemplo: nombreProy: ‘’Educacion para Ancianos”, tipoProy: “Educacion”.

■ Resultado: ‘’Educacion para Ancianos - Educacion”.

|  |
| --- |
| CREATE TABLE auditoria  (  operation CHAR(1) NOT NULL, stamp TIMESTAMP NOT NULL, userid TEXT NOT NULL, hostname TEXT NOT NULL, nombreProy varchar(100), id\_Proy integer not null    );    CREATE TRIGGER triggerInsert\_Proyecto  BEFORE INSERT on proyecto  FOR EACH ROW BEGIN  INSERT INTO auditoria(operation, stamp, userid, hostname, nombreProy, id\_Proy) SELECT 'I',NOW(),USER(),@@hostname,CONCAT(New.nombreProy,' - ',NEW.tipoproy),NEW.id\_proy; end;  DROP TRIGGER triggerInsert\_Proyecto;    INSERT INTO proyecto (id\_proy, nombreProy, tipoproy)  VALUES (1,'Logitech','Tecnologia'); |

EJERCICIO 5

Crear un TRIGGER BEFORE o AFTER para INSERT, UPDATE y DELETE para la tabla DETALLE\_PROYECTO.

● El nombre del TRIGGER deberá ser triggerForDetProy

● Deberá de crear una tabla de AUDITORIA similar al siguiente ejemplo.

|  |
| --- |
| CREATE TABLE auditoria\_detalle\_proyecto (  operation CHAR(1) NOT NULL, -- ('D', 'U', 'I') stamp TIMESTAMP NOT NULL, userid TEXT NOT NULL,  ……………………………………  todos los campos de la tabla detalle\_proyecto  ); |

● Debe de crear un Procedimiento Almacenado o Stored Procedure (SP). ○ Este SP debe recibir parámetros de entrada con los valores a insertar en la tabla de AUDITORIA.

● Los TRIGGERS deben de utilizar este SP, cada trigger debe de enviar los parámetros de inserción de la tabla de

AUDITORIA

|  |
| --- |
| CREATE TABLE auditoria\_detalle\_proyecto (  operation CHAR(1) NOT NULL, -- ('D', 'U', 'I')  stamp TIMESTAMP NOT NULL,  userid TEXT NOT NULL,  id\_proy integer not null,  nombreProy varchar(50),  tipoproy TEXT  );  CREATE PROCEDURE insertar\_detalleProy\_en\_tablas(in operation char(1),in id\_proy integer, in nombreProy varchar(50)  ,in tipoproy text)  BEGIN  INSERT INTO auditoria\_detalle\_proyecto(operation, stamp, userid, id\_proy, nombreProy,tipoproy) SELECT  operation,now(),user(),id\_proy,nombreProy,tipoproy;  end;  CREATE trigger insertarDetProyecto  AFTER insert on proyecto  FOR EACH ROW  BEGIN  CALL insertar\_detalleProy\_en\_tablas('I',NEW.id\_proy,new.nombreProy,NEW.tipoproy);  end;  CREATE TRIGGER ActualizarDetProyecto  AFTER update on proyecto  for each row  BEGIN  call insertar\_detalleProy\_en\_tablas('U',OLD.id\_proy,NEW.nombreProy,NEW.tipoproy);  end;  CREATE TRIGGER EliminarDetProyecto  BEFORE DELETE on proyecto  for each row  begin  call insertar\_detalleProy\_en\_tablas('D',old.id\_proy,old.nombreProy,OLD.tipoproy);  end; |

EJERCICIO 6

|  |
| --- |
| alter table auditoria\_detalle\_proyecto add column diaDelaSemana varchar(12) default '';  create procedure auditoria\_detalle\_tabla(IN operacion char,IN id\_dp integer,IN id\_per integer,IN id\_proy integer, IN diaDelaSemana text )  begin  insert into auditoria\_detalle\_proyecto  (operation, stamp, userid, id\_dp, id\_per, id\_proy,diaDelaSemana)  select operacion, now(), user(),id\_dp, id\_per, id\_proy,diaDelaSemana;  end;  create function getdayname(nowlocale varchar(10))  returns text  begin  declare day text default '';  set @@lc\_time\_names=nowlocale;  select dayname(now()) into day;  return day;  end;  create trigger dias\_para\_insertar  before insert on detalle\_proyecto  for each row  begin  declare day text default '';  select dayname(now()) into day;  if getdayname('es\_BO')=('sabado')||('domingo')  then  signal sqlstate '45000'  set message\_text ='no se admite enserciones dias sabados y domingos!!';  else  call auditoria\_detalle\_tabla('I',NEW.id\_dp, NEW.id\_per, NEW.id\_proy,day);  end if;  end; |