BASE DE DATOS II

TAREA DE DEFENSA HITO 4

NOMBRE: APAZA LARA KEVIN MICHAEL
CÓDIGO: SIS13814908
DOCENTE: WILLIAM RODDY BARRA PAREDES
FECHA: Miércoles 15 de junio, 2022

MANEJO DE CONCEPTOS

Defina que es lenguaje procedural en MySQL.

Como en la programación clásica vendría siendo la delegación de tareas a funciones que pueden o no recibir parámetros.

Esta delegación de tareas cumplen tareas específicas para las cuales fueron creadas.

Defina que es una FUCNTION en MySQL.

Las funciones en MySQL pueden definirse como piezas de código que cumplen tareas específicas que pueden devolver valores únicos o tablas, siendo que existen las funciones de agregación y las funciones generadas por el DBA, las funciones de agregación sólo pueden devolver un valor a diferencia de las creadas por el DBA, la última puede o no recibir parámetros

Cuál es la diferencia entre funciones y procedimientos almacenados.

Entre las principales diferencias está que una función siempre debe de retornar un valor y un procedimiento puede no hacerlo.

Entre otra de sus diferencias están que el procedimiento puede ser invocado desde el entorno de desarrollo y una función no.

Cómo se ejecuta una función y un procedimiento almacenado

Funcion

Con la cláusula SELECT y enviando parámetro si es que es requerido,

Si es una función de agregación puede ir en la cláusula WHERE

Procedimiento almacenado

Con la cláusula CALL y enviando parametros si es requerido.

Defina que es una TRIGGER en MySQL.

Un trigger es un objeto de la base de datos que está asociado con una tabla y que se activa cuando ocurre un evento sobre la tabla.

En un trigger que papel juega las variables OLD y NEW

El alias OLD hace referencia a un registro ya existente antes de un update o delete. El alias NEW hace referencia a un registro nuevo después de un insert o update.

En un trigger que papel juega los conceptos(cláusulas) BEFORE o AFTER

BEFORE y AFTER especifican en qué momento se ejecutan (se ejecutan automáticamente).

BEFORE específica que se ejecute antes de una acción o un evento ya sea ALTER, INSERT, etc.

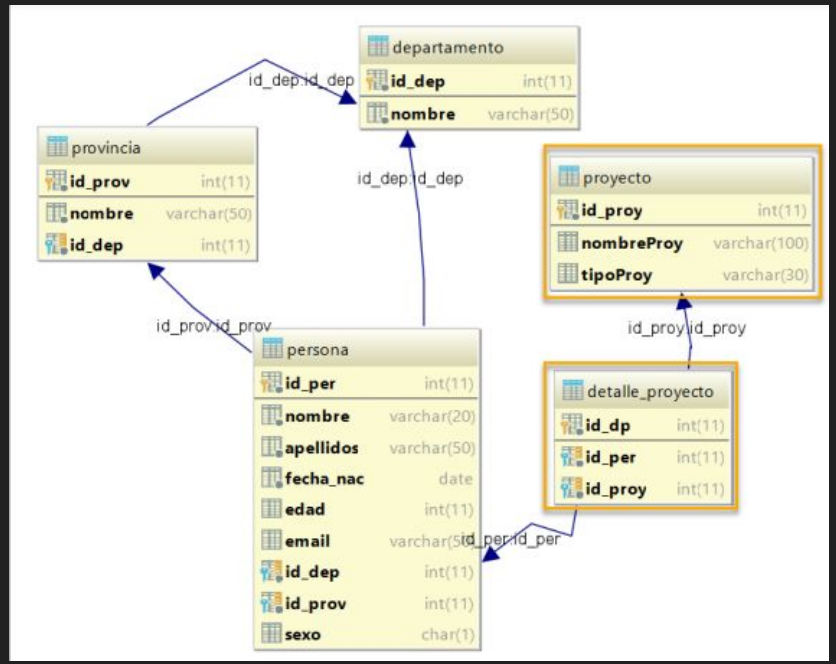
AFTER específica que se ejecute después de una acción o un evento.

A que se refiere cuando se habla de eventos en TRIGGERS

Los eventos que hacen que se ejecute un trigger son las operaciones de inserción (INSERT), borrado (DELETE) o actualización (UPDATE), ya que modifican los datos de una tabla.

Parte práctica

Crear la siguiente Base de datos y sus registros.



```sql
CREATE DATABASE defenza;

USE defenza;

CREATE TABLE departamento
(
id_dep INTEGER (11) AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombre VARCHAR(50)
);
CREATE TABLE proyecto
(
id_proy INTEGER (11) AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombreProy VARCHAR(100),
tipoProy VARCHAR (30),
estado VARCHAR(50)
);
CREATE TABLE provincia(
 id_prov int(11) AUTO_INCREMENT PRIMARY KEY,
 nombre VARCHAR(50),
 id_dep int(11),
 FOREIGN KEY (id_dep) REFERENCES departamento (id_dep)
);
```

```sql
CREATE TABLE persona
(
id_per INTEGER (11) AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombre VARCHAR(50),
apellidos VARCHAR(50),
fecha_nac date,
edad INTEGER(11),
email VARCHAR(50),
id_dep int(11),
id_prov int(11),
sexo CHAR(1),
FOREIGN KEY (id_prov) REFERENCES provincia (id_prov),
FOREIGN KEY (id_dep) REFERENCES departamento (id_dep)
);

CREATE TABLE detalle_proyecto
(
id_dp INTEGER (11) AUTO_INCREMENT PRIMARY KEY NOT NULL,
id_per int(11),
id_proy INTEGER(11),
FOREIGN KEY (id_proy) REFERENCES proyecto (id_proy),
FOREIGN KEY (id_per) REFERENCES persona (id_per)
);
```

```sql
INSERT INTO persona (id_per,nombre, apellidos, fecha_nac, edad, email,sexo)
VALUES (1,'kevin','apaza','2000-10-03',50,'kevin@gmail.com','M'),
(2, 'gracia','ticona','2005-01-04',30,'gracia@gmail.com','F');

INSERT INTO departamento (id_dep, nombre) VALUES (1,'cochabamba');
INSERT INTO departamento (id_dep, nombre) VALUES (2,'santa cruz');

INSERT INTO proyecto (id_proy,nombreProy, tipoProy) VALUES (1,'import','empresarial');
INSERT INTO proyecto (id_proy,nombreProy, tipoProy) VALUES (2,'import ii ','escolar');

INSERT INTO detalle_proyecto (id_dp) VALUES (1);
INSERT INTO detalle_proyecto (id_dp) VALUES (2);

INSERT INTO provincia (id_prov, nombre) VALUES (1,'rios seco');
INSERT INTO provincia (id_prov, nombre) VALUES (2,'mariscal');
```

10.Crear una función que sume los valores de la serie Fibonacci

```
create or replace function  fibo(limite integer) returns text
begin
   declare respuesta text default '';
   declare x integer default 0;
   declare y integer default 1;
   declare cont integer default 0;

   while cont!=limite do
       set respuesta= concat(respuesta,x,', ');
       set x=x+y;
       set y=x-y;
       set cont=cont+1;
   end while;

   return respuesta;
end;

select fibo(10);
```

	`fibo(10)`
1	0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

```
create or replace function  suma_fibo(Limita integer)
returns text
begin
        declare entrada text default '';
        declare espacio text default ' ';
        declare x int default 1;
        declare nVeces int default 0;
        declare letra char default '';
        declare limite int default 0;
        declare a int default 0;
        declare b int default 1;
        declare cont int default 0;
        declare aux int default 0;
        declare sumar int default 0;

        set entrada = fibo(limita);
        set limite = char_length(entrada);
        while x <= limite do
          set letra = substring(entrada, x, 1);
          if letra = espacio
            then
              set nVeces = nVeces + 1;
          end if;
          set x = x + 1;
        end while;
        while cont < nVeces
            do
                if cont = 0
                    then
                        set sumar = 0;
                    else
                        set sumar = sumar + b;
                        set aux = a;
                        set a = b;
                        set b = aux + a;
                end if;
                set cont = cont + 1;
        end while;
        return sumar;
end;
```

	`suma_fibo(10)`
1	88

Manejo de TRIGGERS I.

```
create trigger triger proy
   before insert on proyecto
   for each row
begin
   declare nombre text;
   set nombre=concat(new.nombreProy,',insertado');
   set new.nombreProy=nombre;
end;

insert into proyecto(id proy, nombreProy, tipoProy, estado)values
(3,'nuevo','general','reciente');

select * from proyecto;
```

	id_proy	nombreProy	tipoProy	estado
1	1	import	empresarial	<null>
2	2	import ii	escolar	<null>
3	3	nuevo,insertado	general	reciente

```
create trigger triger ejemplo2
   before update on proyecto
   for each row
begin
   declare nombre text;
   set nombre=concat(new.nombreProy,',actualizado');
   set new.nombreProy=nombre;
end;

update proyecto as pr
set pr.nombreProy ='pan'
where id_proy=3;

select * from proyecto;
```

	id_proy	nombreProy	tipoProy	estado
1	1	import	empresarial	<null>
2	2	import ii	escolar	<null>
3	3	pan,actualizado	general	reciente

Manejo de Triggers II.

```
create trigger calcula edad_para_persona
   before insert on persona
   for each row
begin
   declare edad_calc integer;
   set edad_calc=timestampdiff(year, new.fecha_nac,curdate());

   set new.edad=edad_calc;
end;
```

```
insert into persona( nombre,
fecha nac) values
('raul','1990-02-15');
```

```
select * from persona;
```

	id_per	nombre	apellidos	fecha_nac	edad	email
1	1	pan,actualizado	kevin	2000-10-03	50	kevin@gm
2	2	gracia	ticona	2005-01-04	30	gracia@g
3	3	raul	<null>	1990-02-15	32	<null>

Manejo de TRIGGERS III.

```
create table  copia persona(
      nombre  VARCHAR(50),
      apellidos  VARCHAR(50),
      fecha_nac  date,
      edad  INTEGER(11),
      email  VARCHAR(50),
      id_dep  int(11),
      id_prov  int(11),
      sexo  CHAR(1)
);
```

```
create trigger copia datos
   before insert on persona
   for each row
begin
   insert into copia_persona(nombre, apellidos, fecha_nac, edad, email, id_dep, id_prov,
sexo)values
       (new.nombre, new.apellidos, new.fecha_nac, new.edad, new.email, new.id_dep,
new.id_prov, new.sexo);
end;

insert into persona(nombre, apellidos, fecha nac,edad, email, id_dep, id_prov, sexo)values
('kevin','apaza','2000-01-01',15,'123@g.com',1,1,'m');

select * from copia_persona;
```

	nombre	apellidos	fecha_nac	edad	email	id_dep	id_prov	sexo
1	kevin	apaza	2000-01-01	22	123@g.com	1	1	m

Crear una consulta SQL que haga uso de todas las tablas.

```sql
create view vist as
select det.id_dp,proy.nombreProy,per.nombre,prov.nombre,dep.nombre
from detalle_proyecto as det
join proyecto as proy on det.id_proy = proy.id_proy
join persona as  per on det.id_per = per.id_per
join provincia as prov on per.id_prov = prov.id_prov
join departamento dep on per.id_dep = dep.id_dep
```

id_dp	nombreProy	per.nombre	prov.nombre	dep.nombre