



BASE DE DATOS II

LUIS ALVAREZ MEDINA

MANEJO DE CONCEPTOS



01

Defina que es lenguaje procedural en MySQL

Es la capacidad de escribir y ejecutar un programa o bloques de código que contienen lógica y acciones más complejas dentro de la base de datos. Haciendo uso de procedimientos almacenados, funciones y triggers.

02

Defina que es una FUNCTION en MySQL

Una función "FUNCTION" es un encapsulador de procedimientos lógicos los cuales devuelven un resultado.

03

Cual es la diferencia entre funciones y procedimientos almacenados

Las funciones devuelven un valor calculado el cual se puede utilizar en consultas SQL, por otro lado un procedimiento almacenado son conjuntos de instrucciones SQL, el cual se lo llama con la instrucción CALL.

04

Como se ejecuta una función y un procedimiento almacenado

Para ejecutar una función se utiliza la instrucción: select función();

Para ejecutar un procedimiento almacenado se utiliza la instrucción: Call procedimiento();

05

Defina que es un TRIGGER en MySQL

En MySQL un trigger es un objeto en base de datos el cual lo utilizamos para ejecutar automáticamente una serie de acciones cuando ocurren ciertos eventos dentro de una tabla ya sea before o after de ejecutar las siguientes instrucciones insert, update y delete.

06

En un trigger que papel juega las variables OLD y NEW

OLD: se lo utiliza para acceder a los valores antiguos después de un update o un delete de algún registro

NEW: se lo utiliza para acceder a los valores nuevos después de un insert o un update de algún registro

07

En un trigger que papel juega las clausulas BEFORE or AFTER

BEFORE: esta la utilizamos cuando queremos utilizar lógica adicional o modificar valores antes de que se realicen las operaciones principales.

AFTER: esta la utilizamos cuando se desea realizar acciones con lo valores nuevos y antiguos después de que se realicen las operaciones principales.

08

A que se refiere cuando se habla de eventos en TRIGGERS

*Se refiere a las acciones especificas que ocurren dentro de una tabla de la base de datos
Estos eventos serian los siguientes:*

INSERT: Ingresar registros de una tabla.

UPDATE: Actualizar registros de una tabla.

DELETE: Eliminar registros de una tabla

PARTE

PRACTICA



CREAR LA SIGUIENTE BASE DE DATOS Y SUS REGISTROS

Agregar mínimamente 2 registros para cada tabla

```
create table departamento(
  id_dep int auto_increment primary key,
  nombre varchar(50)
);

create table provincia(
  id_prov int auto_increment primary key,
  nombre varchar(50),
  id_dep int,
  foreign key (id_dep) references departamento(id_dep)
);

create table persona(
  id_per int auto_increment primary key,
  nombre varchar(20),
  apellidos varchar(50),
  fecah_nac date,
  edad int,
  email varchar(50),
  id_dep int,
  id_prov int,
  sexo char(1),
  foreign key (id_dep) references departamento(id_dep),
  foreign key (id_prov) references provincia(id_prov)
);
```

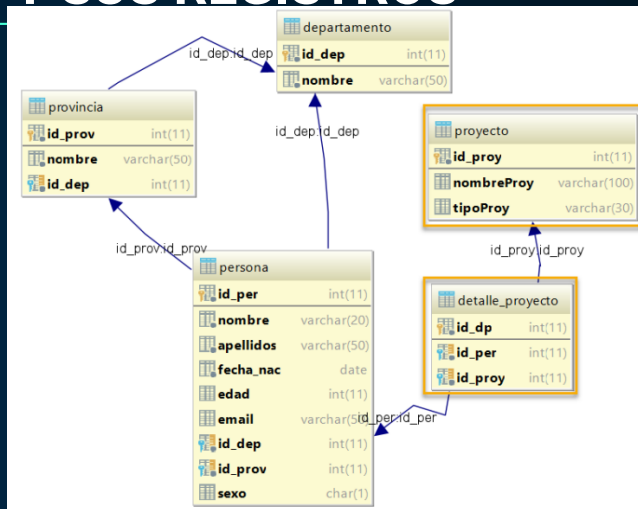
```
create table proyecto (
  id_proy int auto_increment primary key,
  nombreProy varchar(100),
  tipoProy varchar(30)
);

create table detalle_proyecto(
  id_dp int auto_increment primary key,
  id_per int,
  id_proy int,
  foreign key (id_per) references persona(id_per),
  foreign key (id_proy) references proyecto(id_proy)
);
```

```
insert into departamento(nombre) values ('La Paz');
insert into provincia(nombre, id_dep) values ('Pedro Domingo Murillo',1),
('Quillacollo',2);

insert into persona(nombre, apellidos, fecah_nac, edad, email, id_dep, id_prov, sexo)
values ('Luis','Alvarez Medina','2004-01-26',19,'LuisAlvarez@gmail.com',2,2,'M'),
('Yamiley','Lipe Chura','2000-07-23',23,'YamileyLipe@gmail.com',1,1,'F'),
('Carolina','Alanoca Paucara','2000-10-10',23,'CarolinaPucara@gmail.com',1,1,'F');
insert into proyecto(nombreProy, tipoProy)
values ('Conce y Aprende','Educativo'),
('Conexion Cultural','Cultura');

insert into detalle_proyecto(id_per, id_proy)
values (1,1),
(2,2),
(3,1);
```



Crear Una función que sume los valores de la serie Fibonacci.

- El objetivo es sumar todos los números de la serie fibonacci desde una cadena.
- Es decir usted tendrá solo la cadena generada con los primeros N números de la serie fibonacci y a partir de ellos deberá sumar los números de esa serie.

◦ Ejemplo:
suma_serie_fibonacci(mi_metodo_que_retorna_la_serie(10))

■ Note que previamente deberá crear una función que retorne una cadena con la serie fibonacci hasta un cierto valor. 1. Ejemplo: 0,1,1,2,3,5,8,.....

■ Luego esta función se deberá pasar como parámetro a la función que suma todos los valores de esa serie generada.

```
create or replace function serieFibonacci(limt int)
returns text
begin
    declare resp text default '';
    declare n1 int default 0;
    declare n2 int default 1;
    declare nr int default 0;

    while (limt > 0) do
        set resp = concat(resp,n1,',');
        set nr = n1+n2;
        set n1 = n2;
        set n2 = nr;
        set limt = limt-1;
    end while;
    return resp;
end;
```

```
create or replace function sumarFibonacci(limt int)
returns int
begin
```

```
    declare cont int default limt;
    declare ubic int default 0;
    declare cant int default 0;
    declare numf int default 0;
    declare ubic2 text default 0;
    declare sum int default 0;
    while (cont>0) do
        set ubic2 = ubic;
        set ubic = locate(',',serieFibonacci(limt),ubic2+1);
        set cant = ubic-ubic2;
        set numf = substr(serieFibonacci(limt),ubic2+1,cant-1);
        set sum = numf+sum;
        set cont = cont-1;
    end while;
    return sum;
end;

select sumarFibonacci(10);
```

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

```
`sumarFibonacci(10)`
1 88
```


Manejo de vistas.

- Crear una consulta SQL para lo siguiente.
- La consulta de la vista debe reflejar como campos:
 1. nombres y apellidos concatenados
 2. la edad
 3. fecha de nacimiento.
 4. Nombre del proyecto
- Obtener todas las personas del sexo femenino que hayan nacido en el departamento de El Alto en donde la fecha de nacimiento sea:
 1. fecha_nac = '2000-10-10'

```
create or replace view identificadorPersona as
select concat(per.nombre,' ', per.apellidos), per.edad, per.fecah_nac,p.nombreProy from persona per
join departamento d on per.id_dep = d.id_dep
join detalle_proyecto dp on per.id_per = dp.id_per
join proyecto p on dp.id_proy = p.id_proy
where d.nombre = 'La Paz' and per.fecah_nac = '2000-10-10' and per.sexo = 'F';

select * from identificadorPersona;
```

	concat(per.nombre,' ', per.apellidos)	edad	fecah_nac	nombreProy
1	Carolina Alanoca Paucara	23	2000-10-10	Conce y Aprende

Manejo de TRIGGERS I

○ Crear TRIGGERS Before or After para INSERT y UPDATE aplicado a la tabla PROYECTO

■ Debera de crear 2 triggers minimamente.

○ Agregar un nuevo campo a la tabla PROYECTO.

■ El campo debe llamarse ESTADO

○ Actualmente solo se tiene habilitados ciertos tipos de proyectos.

■ EDUCACION, FORESTACION y CULTURA

○ Si al hacer insert o update en el campo tipoProy llega los valores EDUCACION, FORESTACIÓN o CULTURA, en el campo ESTADO colocar el valor ACTIVO. Sin embargo se llegat un tipo de proyecto distinto colocar INACTIVO

```
alter table proyecto add column Estado varchar(15);
```

```
create or replace trigger Activoinactivo
before insert
on proyecto
for each row
begin
    if(new.tipoProy in ('Educacion','Forestacion','cultura')) then
        set new.Estado = 'Activo';
    else
        set new.Estado='Inactivo';
    end if;
end;
```

```
create or replace trigger Activoinactivo_update
before update
on proyecto
for each row
begin
    if(new.tipoProy in ('Educacion','Forestacion','cultura')) then
        set new.Estado = 'Activo';
    else
        set new.Estado='Inactivo';
    end if;
end;
```

```
insert into proyecto (nombreProy, tipoProy) VALUES ('Cortar Arboles','Forestacion');
insert into proyecto (nombreProy, tipoProy) VALUES ('Cortar Arboles','venta');
select * from proyecto;
update proyecto set tipoProy = 'Educacion' where id_proy =2;
update proyecto set tipoProy = 'Claro' where id_proy =1;
select * from proyecto;
```

	id_proy	nombreProy	tipoProy	Estado
1	1	Conce y Aprende	venta	Inactivo
2	2	Conexion Cultural	Educacion	Activo
3	3	Cortar Arboles	Forestacion	Activo
4	4	Cortar Arboles	venta	Inactivo
5	5	Cortar Arboles	Forestacion	Activo
6	6	Cortar Arboles	venta	Inactivo

	id_proy	nombreProy	tipoProy	Estado
1	1	Conce y Aprende	Claro	Inactivo
2	2	Conexion Cultural	Educacion	Activo
3	3	Cortar Arboles	Forestacion	Activo
4	4	Cortar Arboles	venta	Inactivo
5	5	Cortar Arboles	Forestacion	Activo
6	6	Cortar Arboles	venta	Inactivo

Manejo de TRIGGERS II

- El trigger debe de llamarse calculaEdad.
- El evento debe de ejecutarse en un BEFORE INSERT.
- Cada vez que se inserta un registro en la tabla PERSONA, el trigger debe de calcular la edad en función a la fecha de nacimiento.

```
create or replace trigger CalcularEdad
before insert
on persona
for each row
begin
    set new.edad= timestampdiff(year, new.fecah_nac,curdate());

end;

insert into persona(nombre, apellidos, fecah_nac, email, id_dep, id_prov, sexo)
values ('jhon','Apaza','2001-06-26','LuisAlvarez@gmail.com',2,2,'M');

select * from persona;
```

8	8	jhon	Apaza	2001-06-26	21	LuisAlvarez@gmail.com	2	2	M
---	---	------	-------	------------	----	-----------------------	---	---	---

Manejo de TRIGGERS III

- Crear otra tabla con los mismos campos de la tabla persona (Excepto el primary key id_per).
- No es necesario que tenga PRIMARY KEY.
- Cada vez que se haga un INSERT a la tabla persona estos mismos valores deben insertarse a la tabla copia.
- Para resolver esto deberá de crear un trigger before insert para la tabla PERSONA

```
create table persona_copia(
  nombre varchar(20),
  apellidos varchar(50),
  fecah_nac date,
  edad int,
  email varchar(50),
  id_dep int,
  id_prov int,
  sexo char(1)
);
```

```
create or replace trigger hacerCopia
  before insert
  on persona
  for each row
  begin
    insert into persona_copia
    (nombre, apellidos, fecah_nac, edad, email, id_dep, id_prov, sexo)
    SELECT
    new.nombre, new.apellidos, new.fecah_nac, new.edad, new.email, new.id_dep, new.id_prov, new.sexo;
  end;

insert into persona(nombre, apellidos, fecah_nac, email, id_dep, id_prov, sexo)
values ('kevin', 'apaza', '2000-06-26', 'LuisAlvarez@gmail.com', 1, 1, 'M');

select * from persona;
select * from persona_copia;
```

7	7	kevin	apaza	2000-06-26	22	LuisAlvarez@gmail.com	1	1	M
8	8	jhon	Apaza	2001-06-26	21	LuisAlvarez@gmail.com	2	2	M

	nombre	apellidos	fecah_nac	edad	email	id_dep	id_prov	sexo
1	kevin	apaza	2000-06-26	22	LuisAlvarez@gmail.com	1		1 M
2	jhon	Apaza	2001-06-26	21	LuisAlvarez@gmail.com	2		2 M

Crear una consulta SQL que haga uso de todas las tablas

- La consulta generada convertirlo a VISTA

```
create or replace view Usandotablas as
select concat(per.nombre,' ', per.apellidos), per.edad, per.fecah_nac,
p.nombreProy as Proyecto,p.tipoProy,p.Estado,
prov.nombre as provincia ,d.nombre as departamento,dp.id_proy
from persona per
join departamento d on per.id_dep = d.id_dep
join detalle_proyecto dp on per.id_per = dp.id_per
join proyecto p on dp.id_proy = p.id_proy
join provincia prov on per.id_prov = prov.id_prov;

select * from usandotablas
```

	'concat(per.nombre,' ', per.apellidos)'	edad	fecah_nac	Proyecto	tipoProy	Estado	provincia	departamento	id_proy
1	Yamiley Lipe Chura	23	2000-07-23	Conexion Cultural	Educacion	Activo	Pedro Domingo Murillo	La Paz	2
2	Carolina Alanoca Paucara	23	2000-10-10	Conce y Aprende	Claro	Inactivo	Pedro Domingo Murillo	La Paz	1
3	Luis Alvarez Medina	19	2004-01-26	Conce y Aprende	Claro	Inactivo	Quillacollo	Cochabamba	1

**GRACIAS POR SU
ATENCIÓN!!**
