

# **TAREA FINAL HITO-4**

## **BASE DE DATOS II**

# DEFINA QUE ES LENGUAJE PROCEDURAL EN MYSQL

El lenguaje procedural en MySQL se refiere a la capacidad de MySQL para ejecutar procedimientos almacenados y funciones definidos por el usuario. Un procedimiento almacenado es un conjunto de instrucciones SQL que se almacenan en la base de datos y se pueden invocar y ejecutar posteriormente.



# DEFINA QUE ES UNA FUNCION EN MYSQL

En MySQL, una función es un objeto de base de datos que se utiliza para realizar cálculos o manipulaciones de datos específicas y devuelve un valor como resultado. Las funciones en MySQL se utilizan principalmente para simplificar y reutilizar la lógica de procesamiento de datos en consultas SQL y procedimientos almacenados.

```
create function serieFibonacci(limt int)
  returns text
begin
  declare resp text default '';
  declare num1 text default 0;
  declare num2 text default 1;
  declare num3 text default 0;
  declare cont int default 0;
  while cont < limt do
    set resp = CONCAT(resp,num3,',');
    set num1 = num2;
    set num2 = num3;
    set num3 = num1 + num2;
    set cont = cont + 1;
  end while;
  return resp;
end;
```

# CUÁL ES LA DIFERENCIA ENTRE FUNCIONES Y PROCEDIMIENTOS ALMACENADOS

Estos son algunas diferencias

**Modificación de datos:** Funciones: No pueden realizar operaciones de inserción, actualización o eliminación de registros. Están principalmente diseñadas para cálculos y transformaciones de datos.

**Procedimientos almacenados:** Pueden realizar cualquier tipo de operación en la base de datos, incluida la modificación de datos. Pueden contener instrucciones SQL para realizar inserciones, actualizaciones, eliminaciones y otras operaciones de manipulación de datos.

**Contexto de ejecución:**

**Funciones:** Se evalúan en el contexto de una expresión o consulta SQL. Pueden ser utilizadas en cualquier lugar donde se permita una expresión en una consulta.

**Procedimientos almacenados:** Se ejecutan de forma independiente y deben ser invocados explícitamente. Contienen un conjunto de instrucciones SQL y lógica de programación más compleja.



# CÓMO SE EJECUTA UNA FUNCIÓN Y UN PROCEDIMIENTO ALMACENADO.

Ejecución de una función:

Para ejecutar una función en MySQL, se utiliza la función como parte de una expresión o consulta SQL. Puedes llamar a una función en una consulta SELECT, WHERE o cualquier otra cláusula que acepte expresiones.

Ejemplo:

```
SELECT nombre(parámetro)
```

Ejecución de un procedimiento almacenado:

Para ejecutar un procedimiento almacenado en MySQL, se utiliza la sentencia CALL, seguida del nombre del procedimiento almacenado y los parámetros, si es necesario.

Ejemplo:

```
CALL nombres(parametros)
```

# DEFINA QUE ES UNA TRIGGER EN MYSQL.

La función principal de un trigger es permitir la ejecución automática de acciones o lógica de programación definida por el usuario cuando se cumple una condición o evento específico en una tabla. Los triggers se utilizan para garantizar la integridad de los datos, aplicar reglas de negocio, auditar cambios o realizar acciones adicionales cuando se modifican los datos en una tabla.

```
create or replace trigger TRIGGER_NUEVOS_DATOS
  before insert
  on PERSONA
  for each row
  begin
    insert into NEW(NOMBRE, APELLIDO, fecha_de_nac,
    values (NEW.nombre,NEW.apellido,NEW.fecha_de
  end;
```



# EN UN TRIGGER QUE PAPEL JUEGA LAS VARIABLES OLD Y NEW

Las variables OLD y NEW son variables especiales que se refieren a la fila afectada por el evento. Aquí está su significado:

**OLD:** La variable OLD almacena los valores antiguos (previos a la operación) de las columnas afectadas. Es útil en los triggers AFTER UPDATE y AFTER DELETE, ya que permite acceder a los valores originales antes de la actualización o eliminación.

**NEW:** La variable NEW almacena los nuevos valores (posteriores a la operación) de las columnas afectadas. Es útil en los triggers AFTER INSERT y AFTER UPDATE, ya que proporciona acceso a los valores actualizados o recién insertados en la tabla.

# EN UN TRIGGER QUE PAPEL JUEGA LOS CONCEPTOS(CLÁUSULAS) BEFORE O AFTER

## BEFORE (Antes):

- Si se utiliza la cláusula "BEFORE" en el trigger, el trigger se activa antes de que se realice la operación que desencadenó el evento.
- En un trigger "BEFORE INSERT", se puede acceder y modificar los valores de las columnas de la fila antes de que se realice la inserción en la tabla.
- En un trigger "BEFORE UPDATE", se puede acceder y modificar los valores de las columnas de la fila antes de que se realice la actualización en la tabla.

## AFTER (Después):

- Si se utiliza la cláusula "AFTER" en el trigger, el trigger se activa después de que se haya realizado la operación que desencadenó el evento.
- En un trigger "AFTER INSERT", se puede acceder a los valores de las columnas de la fila después de que se haya realizado la inserción en la tabla.
- En un trigger "AFTER UPDATE", se puede acceder a los valores de las columnas de la fila después de que se haya realizado la actualización en la tabla.
- En un trigger "AFTER DELETE", se puede acceder a los valores de las columnas de la fila después de que se haya realizado la eliminación de la tabla.



# A QUE SE REFIERE CUANDO SE HABLA DE EVENTOS EN TRIGGERS

Cuando se habla de "eventos" en el contexto de los triggers, se refiere a las acciones que ocurren en una tabla de la base de datos y que activan el trigger. Los eventos pueden ser operaciones de inserción (INSERT), actualización (UPDATE) o eliminación (DELETE) de registros en una tabla específica.

Cada vez que se realiza una operación de inserción, actualización o eliminación en la tabla, se genera un evento correspondiente. Estos eventos son capturados por los triggers asociados a la tabla y desencadenan la ejecución de las acciones definidas en el trigger.

En resumen, los eventos en los triggers son las operaciones de inserción, actualización o eliminación de registros en una tabla que desencadenan la ejecución del código del trigger asociado.