



Tarea HIT04

14.06.2022

—

Base de Datos II

Unifranz

Sede el Alto

Hito4

Objetivos de la tarea.

1. Mostrar el manejo de **programación a nivel de Base de Datos**.
2. Mostrar el manejo del **lenguaje procedural**.
3. Mostrar el manejo de **funciones, vistas y triggers en MySQL**.

Consigna.

Esta tarea se divide en dos partes:

- La primera parte corresponde a la **parte TEÓRICA necesaria**, en donde se encuentra un conglomerado de preguntas relacionadas a BASES DE DATOS RELACIONALES.
- La segunda parte corresponde a la **parte PRÁCTICA necesaria** en donde deberá realizar y crear funciones manejando el motor de base de datos MySQL manejando el concepto del lenguaje procedural.

Consideraciones sobre la entrega.

La tarea es abierta, usted decide cómo entregar su tarea. Puede agregar imágenes, enlaces, etc. Sin embargo, considerar los siguientes enunciados.

- Deberá generar una presentación hecha en **POWER POINT**. Después este mismo archivo deberá ser convertido a un **archivo PDF**.
- Después de tener disponible la presentación, debe de **GENERAR UN VIDEO** explicando todo su contenido o lo que considere necesario e importante.
 - a. El video debe de tener como mínimo 15 minutos.
 - b. En el video debe estar presente su cámara web.
 - Si no tiene cámara web puede apoyarse en [IRIUM](#) que básicamente convierte su celular en una cámara web.
- Después de tener disponible el video, el video lo puede subir a **youtube, vimeo** o a drive u otra cualquier plataforma.
- Finalmente, todo lo generado, es decir:
 - a. El archivo **ppt** (powerpoint).
 - b. El archivo **pdf** (el powerpoint convertido a PDF).

- c. El **video** subido a alguna plataforma.
- d. IMPORTANTE:
 - Tiene que ser subido a la plataforma **GITHUB**.
 - Para hacer este proceso tiene que tener sus archivos ya disponibles.
 - Crear la carpeta **HITO4** en nuestro repositorio de github.
- En la plataforma **MOODLE** solo deberá de subir una carátula referenciando a su tarea que se encuentra en github.

Ejemplo:

<p>UNIVERSIDAD PRIVADA FRANZ TAMAYO</p> <p>DEFENSA HITO 4 - TAREA FINAL</p> <p>Estudiante: Univ. Nombres Apellidos</p> <p>Asignatura: BASE DE DATOS II</p> <p>Carrera: INGENIERÍA DE SISTEMAS</p> <p>Sede: El Alto</p> <p>Paralelo: BDA (1)</p> <p>Docente: Lic. William Barra Paredes</p> <p>fecha: xx/xx/2022</p> <p>GITHUB: https://github.com/dheevi/base-de-datos-ii (aquí va el enlace a su cuenta de github)</p>
--

A partir de este punto están las preguntas **TEÓRICAS** y **PRÁCTICAS** a resolverse.

Manejo de conceptos.

1. Defina que es lenguaje procedural en MySQL.

El objeto compartido que contienen el manejador del **lenguaje** ha de ser compilado e instalado

2. Defina que es una **FUCNTION** en MySQL.

Tipos de funciones

Funciones de cadenas – operan en tipos de datos de cadena

Funciones numéricas : opere en tipos de datos numéricos.

- 3.Cuál es la diferencia entre funciones y procedimientos almacenados.

Una **rutina** almacenada es un **procedimiento** o una **función**.

Un **procedimiento** se invoca mediante una sentencia CALL y sólo puede devolver valores mediante variables de salida. Una **función** puede ser llamada desde dentro de una sentencia **como** cualquier otra **función** y puede devolver un valor escalar.

4. Cómo se ejecuta una función y un procedimiento almacenado.

5. Defina que es una **TRIGGER** en MySQL.

El **trigger MySQL** es un objeto de la base de datos que está asociado con una tabla. Se activará cuando una acción definida se ejecute en la tabla.

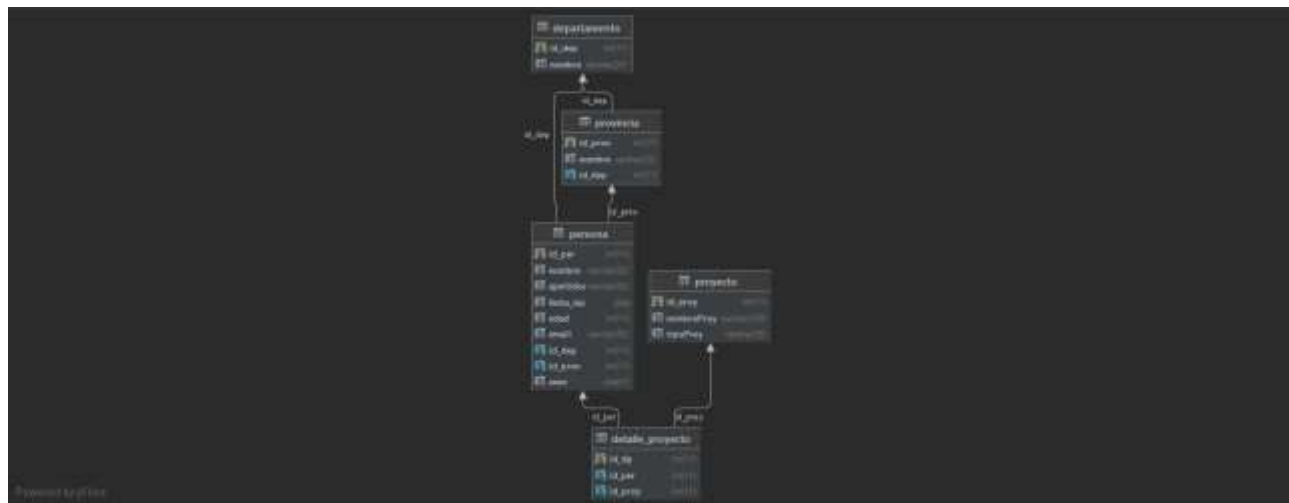
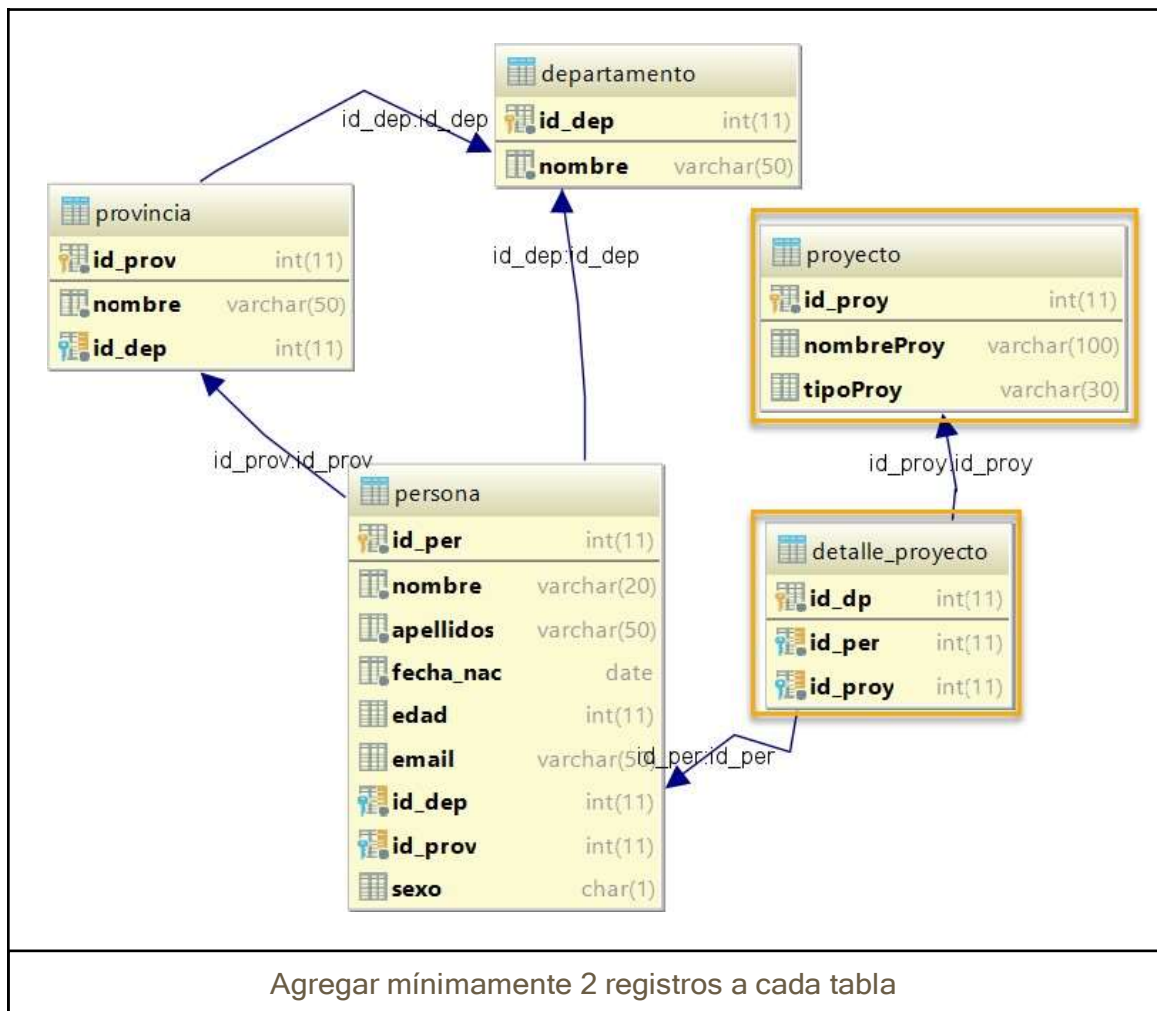
6. En un trigger que papel juega las variables **OLD** y **NEW**

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

8. A que se refiere cuando se habla de eventos en TRIGGERS

Parte practica

9. Crear la siguiente Base de datos y sus registros.



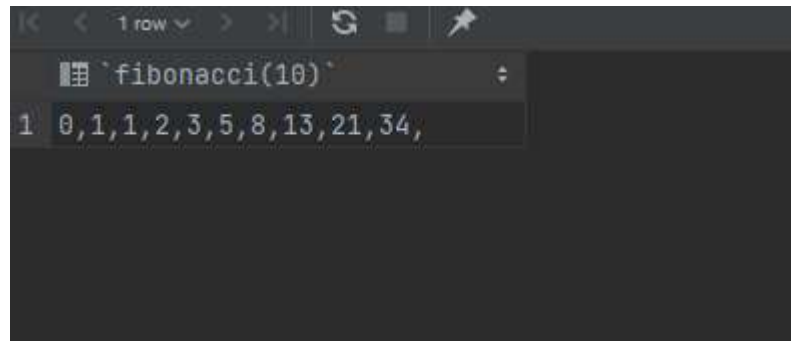
10. Crear una función que sume los valores de la serie Fibonacci.

- 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,.....



```
create function fibonacci(limite int)
returns text
begin
  declare a int default 0;
  declare b int default 1;
  declare c int default 0;
  declare x int default 1;
  declare response text;

  if limite >= 1
  then
    set response = concat(a, ',');
  end if;

  if limite >= 2
  then
    set response = concat(response, b, ',');
  end if;

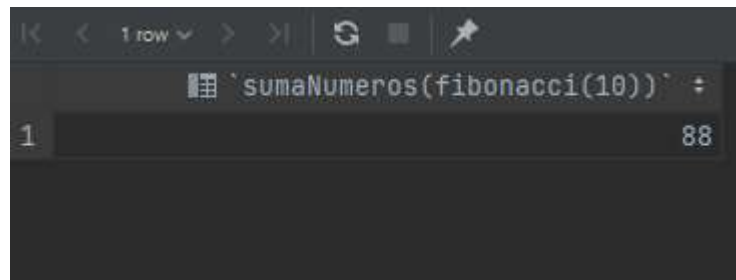
  if limite >= 3
  then
    while x <= (limite - 2) do
      set c=a+b;
      set response = concat(response, c, ',');
      set a = b;
      set b = c;
      set x = x+1;
    end while;
  end if;
  return response;
end;
```

```
SELECT fibonacci(10);
```

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



- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.



```
CREATE FUNCTION sumaNumeros(cadena VARCHAR(100))
RETURNS INT
BEGIN
    DECLARE numero VARCHAR(50) DEFAULT '';
    DECLARE total INTEGER DEFAULT 0;
    DECLARE ObtenCaracter VARCHAR(1);
    DECLARE cuanto INTEGER DEFAULT 1;

    IF LENGTH(cadena) > 0 THEN
        WHILE (cuanto <= LENGTH(cadena)) DO
            SET ObtenCaracter = SUBSTRING(cadena, cuanto, 1);
            /* Suma de Valores*/
            IF ObtenCaracter = ',' THEN
                SET total = total + numero;
                SET numero = '';
            ELSE
                /* Concatenar numero */
                SET numero = CONCAT(numero, ObtenCaracter);
            END IF;
            SET cuanto = cuanto + 1;
        END WHILE;
        RETURN total;
    ELSE
        RETURN 0;
    END IF;
END;
```

11. 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'

LA CONSULTA GENERADA PREVIAMENTE CONVERTIR EN UNA VISTA

```
CREATE VIEW
manejo_vista AS
SELECT CONCAT(p.nombre, p.apellidos) as nombre_comp, p.edad, p.fecha_nac,
pr.nombreProy, dep.*
FROM persona p
INNER JOIN detalle_proyecto dp on dp.id_per = p.id_per
INNER JOIN departamento dep on p.id_dep = dep.id_dep
INNER JOIN proyecto pr on dp.id_proy = pr.id_proy
WHERE p.sexo = 'F'
AND p.fecha_nac = '2000-10-10'

select *from persona;
SELECT * from manejo_vista
```

	nombre_comp	edad	fecha_nac	nombreProy	id_dep	nombre
1	MariaMamani Condori	23	2000-10-10	Ambiente	1	La Paz

12. 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**

```
# EJERCICIO 12
# TRIGGER 1
CREATE TRIGGER tipo_proyecto BEFORE INSERT
ON proyecto
FOR EACH ROW
BEGIN
  IF NEW.tipoProy = 1 THEN
    SET NEW.tipoProy = 'EDUCACION';
  ELSE IF NEW.tipoProy = 2 THEN
    SET NEW.tipoProy = 'FORESTACION';
  ELSE
    SET NEW.tipoProy = 'CULTURA';
  END IF;
END;

INSERT INTO proyecto(nombreProy, tipoProy) values('Proyecto Clinica', 1);

# TRIGGER 2
CREATE TRIGGER nombre_proyecto BEFORE INSERT
```

```
    ON proyecto
  FOR EACH ROW
  BEGIN
    IF NEW.nombreProy = '' THEN
      SET NEW.nombreProy = 'No se asigno nombre.';
    END IF;
  END;

INSERT INTO proyecto(nombreProy, tipoProy) values('', 1);

ALTER TABLE proyecto ADD estado varchar(10);
# TRIGGER 3
CREATE TRIGGER estado BEFORE INSERT
  ON proyecto
  FOR EACH ROW
  BEGIN
    IF (NEW.tipoProy = 'EDUCACION' OR NEW.tipoProy = 'FORESTACION' OR
NEW.tipoProy = 'CULTURA') THEN
      SET NEW.estado = 'Activo';
    ELSE
      SET NEW.estado = 'Inactivo';
    END IF;
  END;
```

- 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**
- Adjuntar el **código SQL generado y una imagen de su correcto funcionamiento.**

13. 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.
- Adjuntar el **código SQL generado y una imagen de su correcto funcionamiento.**

```
CREATE TRIGGER calcularEdad BEFORE INSERT
  ON persona
  FOR EACH ROW
  BEGIN
    SET NEW.edad = TIMESTAMPDIFF(YEAR,NEW.fecha_nac,CURDATE());
  END;

INSERT INTO persona(nombre, apellidos, fecha_nac, email, id_dep, id_prov, sexo)
VALUES('marcos', 'condori', '2000-06-18', 'marco@gmail.com', 1,1,'M');
```

```
mysql>
mysql>
2022-06-14 20:23:44] completed in 194 ms
mysql> CREATE TRIGGER calcularEdad BEFORE INSERT
mysql>
  ON persona
  FOR EACH ROW
  BEGIN
    SET NEW.edad = TIMESTAMPDIFF(YEAR,NEW.fecha_nac,CURDATE());
  END;
2022-06-14 20:27:02] completed in 124 ms
mysql> INSERT INTO persona(nombre, apellidos, fecha_nac, email, id_dep, id_prov, sexo) VALUES('marcos', 'condori', '2000-06-18', 'marco@gmail.com', 1,1,'M');
2022-06-14 20:27:09] 1 row affected in 0.02 ms
```

14. 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**.

- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.

```

○ create table auditoria(
    nombre varchar(25),
    apellidos varchar(50),
    fecha_nac date,
    edad int(11),
    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 TRIGGER audit BEFORE INSERT
ON persona
FOR EACH ROW
BEGIN
    INSERT INTO auditoria(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, email, id_dep,
id_prov, sexo) VALUES ('Cristian', 'Suxo', '2000-01-01',
'cris@gmail.com', 1,1,'M');

```

```

mysql> CREATE TRIGGER audit BEFORE INSERT
ON persona
FOR EACH ROW
BEGIN
    INSERT INTO auditoria(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;
[2022-06-16 20:29:45] completed in 265 ms
mysql> INSERT INTO persona(nombre, apellidos, fecha_nac, email, id_dep, id_prov, sexo) VALUES('Cristian', 'Suxo', '2000-01-01', 'cris@gmail.com', 1,1,'M')
[2022-06-16 20:29:54] 1 row affected in 72 ms

```

nombre	apellidos	fecha_nac	edad	email	id_dep	id_prov	sexo
1 Cristian	Suxo	2000-01-01	22	cris@gmail.com	1	1	M

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

- La consulta generada convertirlo a VISTA
- ```

CREATE VIEW
detalle_general as
SELECT CONCAT(p.nombre, ' ', p.apellidos) as nombre_comp, p.edad,
p.fecha_nac, pro.nombreProy, d.nombre as departamento, pr.nombre as
provincia
FROM persona p
INNER JOIN departamento d on d.id_dep = p.id_dep
INNER JOIN provincia pr on pr.id_prov = p.id_prov
INNER JOIN detalle_proyecto dp on dp.id_per = p.id_per
INNER JOIN proyecto pro on pro.id_proy = dp.id_proy
WHERE p.id_per = 2

```

| nombre_comp            | edad | fecha_nac  | nombreProy | departamento | provincia |
|------------------------|------|------------|------------|--------------|-----------|
| 1 Maria Mamani Condori | 23   | 2000-10-10 | Ambiente   | La Paz       | Murillo   |