

HITO 4, EVALUACIÓN PROCESUAL BASE DE DATOS 2

UNIVERSIDAD PRIVADA FRANZ TAMAYO
DEFENSA HITO 4 – EVALUACION PROCESUAL

Estudiante: Univ. Jhomar Huaycho Quispe

Asignatura: BASE DE DATOS II

Carrera: INGENIERÍA DE SISTEMAS

Paralelo: BDA (1)

Docente: Lic. William Barra Paredes

fecha: 26/11/2022

GITHUB: <https://github.com/JHOMARHUAYCHO/BASE-DE-DATOS-2/tree/main/HITO%204/PRACTICA%20H4>

VIDEO EXPOSICION :

https://drive.google.com/drive/folders/1VmteNzWSQ7sy_Kj7LjQk_k1frn_wJCUf8?usp=share_link

PARTE TEORICA

- **1 Defina que es lenguaje procedural en Mysql**
- R Es el manejo de estructuras (bucles, sentencias, variables , etc.) dentro del gestor de base de datos

- **2. Defina que es una FUCNTION en Mysql.**
- R. es una rutina creada que puede recibir parámetros, para realizar un proceso y devolver un resultado en concreto.

- **3. Cuál es la diferencia entre funciones y procedimientos almacenados.**
- R. aunque ambas realizan operaciones, la diferencia mas notable es que una función no devuelve un valor único; mientras que un procedimiento almacenado no nos devuelve ningún valor.

- **4. Cómo se ejecuta una función y un procedimiento almacenado.**
- R. un procedimiento almacenado se ejecuta con el comando CALL mientras que una función se ejecuta con la sentencia SELECT.

- **5. Defina que es un TRIGGER en Mysql.**
- R un TRIGGER es una instrucción almacenada dentro de la base de datos que se ejecuta de manera automática ante un evento INSERT, UPDATE y DELETE

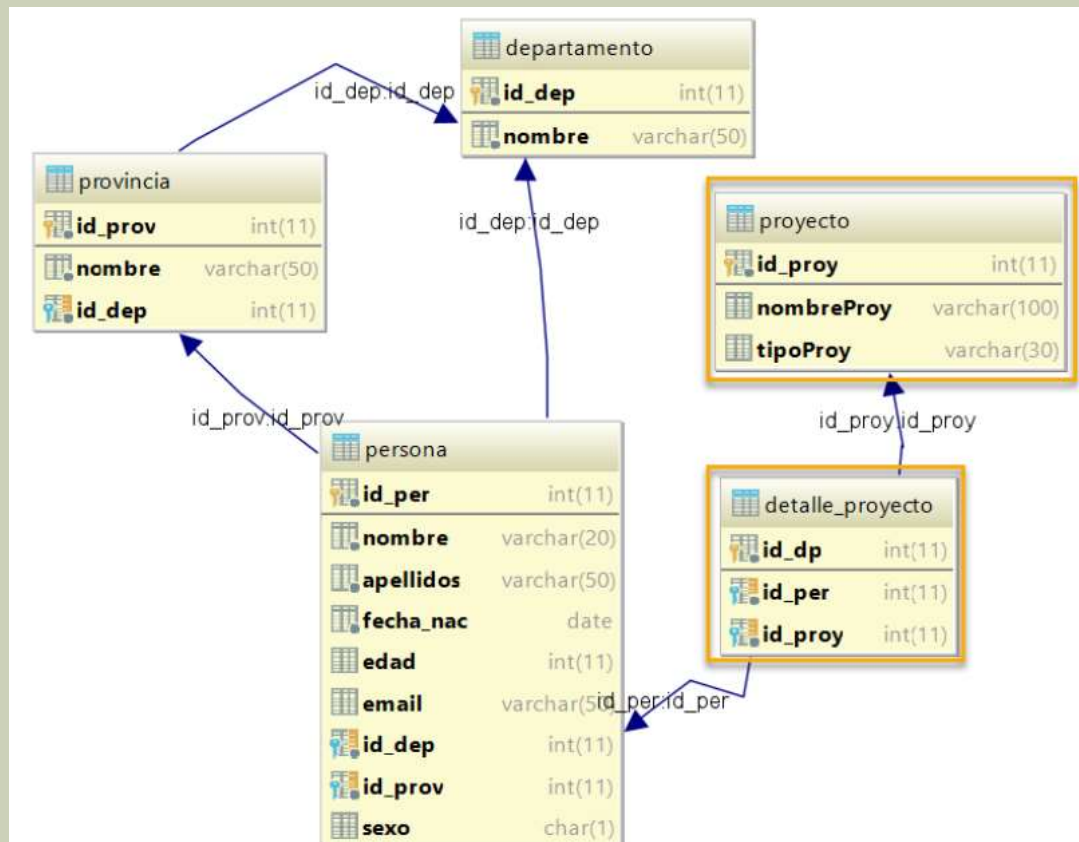
- **6. En un trigger que papel juega las variables OLD y NEW**
- R. la variable OLD nos permite acceder a la vieja variable y la sentencia NEW nos permite acceder a la nueva variables; antes o después de un evento UPDATE.

- **7. En un trigger que papel juega los conceptos(cláusulas) BEFORE o AFTER**
- R. la clausula BEFOR 'antes' hace que el trigger se ejecute antes del evento, mientras que AFTER 'después' hace que el trigger se ejecute después del evento

- **8. A que se refiere cuando se habla de eventos en TRIGGERS**
- R. Los eventos son acciones que se realizan , en el caso de los trigger se refiere específicamente al 'UPDATE= MODIFICAR', 'DELETE= BORRAR' Y 'INSERT= INSERTAR'

PARTE PRACTICA

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



```
CREATE DATABASE PRACTICA_H4;
USE PRACTICA_H4;

CREATE TABLE DEPARTAMENTO
(
    ID_DEP INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    NOMBRE VARCHAR(50)
);

CREATE TABLE PROVINCIA
(
    ID_PROV INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    NOMBRE VARCHAR(50),
    ID_DEP INT NOT NULL,
    FOREIGN KEY (ID_DEP) REFERENCES DEPARTAMENTO(ID_DEP)
);

CREATE TABLE PROYECTO
(
    ID_PROY INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    NOMBRE_PROY VARCHAR(100),
    TIPO_PROY VARCHAR(30)
);

CREATE TABLE PERSONA
(
    ID_PER INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
    NOMBRE VARCHAR(20),
    APELLIDOS VARCHAR(50),
    FECHA_NAC DATE,
    EDAD INT,
    EMAIL VARCHAR(50),
    ID_DEP INT NOT NULL ,
    ID_PROV INT NOT NULL,
    GENERO VARCHAR(1),
    FOREIGN KEY (ID_PROV) REFERENCES PROVINCIA(ID_PROV),
    FOREIGN KEY (ID_DEP) REFERENCES DEPARTAMENTO(ID_DEP)
);

CREATE TABLE DETALLE_PROYECTO
(
    ID_DP INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
    ID_PER INT NOT NULL ,
    ID_PROY INT NOT NULL ,

    FOREIGN KEY (ID_PROY) REFERENCES PROYECTO(ID_PROY),
    FOREIGN KEY (ID_PER) REFERENCES PERSONA(ID_PER)
);
```

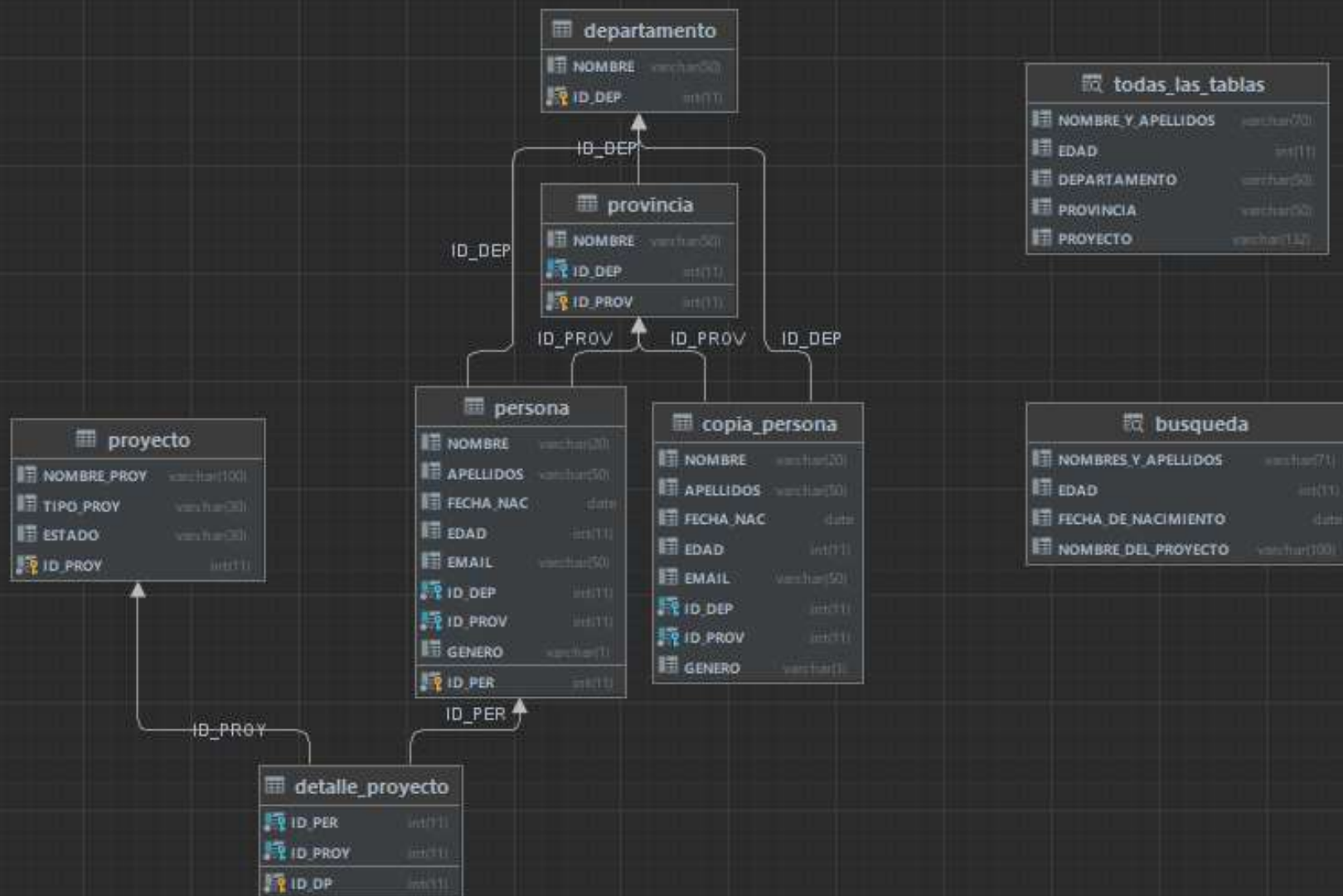
```
INSERT INTO DEPARTAMENTO(NOMBRE)
VALUES ('LA PAZ'),
       ('SANTA CRUZ'),
       ('BENI'),
       ('ORURO'),
       ('CHUQUISACA'),
       ('TARIJA'),
       ('PANDO'),
       ('POTOSI'),
       ('COCHABAMBA');
```

```
INSERT INTO PROVINCIA(NOMBRE, ID_DEP)
VALUES ('VIACHA',1),
       ('ROBORE',2),
       ('MAGDALENA',3),
       ('CHALLAPATA',4),
       ('TARABUCO',5),
       ('VILLAMONTES',6),
       ('GUAYARAMERIN',7),
       ('UYUNI',8),
       ('AIQUILE',9);
```

```
INSERT INTO PERSONA (NOMBRE, APELLIDOS, FECHA_NAC, EDAD, EMAIL, ID_DEP, ID_PROV, GENERO)
VALUES ('RODRIGO','MENODZA UGARTE','1999-11-21',23,'RODRIGO@GMAIL.COM',1,1,'M'),
       ('MARIA','LAURA TORREZ','1999-12-16',25,'MARIA@GAMIL.COM',2,2,'F'),
       ('AUGUSTO','MEDRANO LOZA','1998-09-12',24,'AUGUSTO@GMAIL.COM',3,3,'M'),
       ('MARIANO','FERNANDEZ GUTIERREZ','1995-09-12',27,'MARIANO@GMAIL.COM',4,4,'M'),
       ('LORENA','ZAMUDIO LLANOS','2000-09-12',22,'LORENA@GMAIL.COM',5,5,'F');
```

```
INSERT INTO PROYECTO(NOMBRE_PROY, TIPO_PROY)
VALUES ('ANIMALES','BIOLOGIA'),
       ('PREHISTORIA','ANTROPOLOGIA'),
       ('MICROBIOS','BIOLOGIA'),
       ('REDES Y SISTEMAS','TECNOLOGIA'),
       ('FLUJO MAGNETICO','FISICA');
```

```
INSERT INTO DETALLE_PROYECTO(ID_PER, ID_PROY)
VALUES (1,2),
       (2,1),
       (3,4),
       (4,3);
```



PREGUNTA 10

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


```
CREATE OR REPLACE FUNCTION SERIE_FIBONANCI(NUMBER INTEGER)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
    DECLARE A INTEGER DEFAULT 0;
```

```
    DECLARE B INTEGER DEFAULT 1;
```

```
    DECLARE AUX INTEGER DEFAULT 0;
```

```
    DECLARE CONTADOR INTEGER DEFAULT 0;
```

```
    DECLARE CADENA TEXT DEFAULT '';
```

```
    SET CADENA = CONCAT(A, ',', B);
```

```
    IF NUMBER=1 THEN SET CADENA='0';
```

```
    ELSEIF NUMBER=2 THEN SET CADENA='0,1';
```

```
    ELSEIF NUMBER<=0 THEN SET CADENA='EL NUMERO DEBE SER MAYOR A CERO';
```

```
    ELSE
```

```
        REPEAT
```

```
            SET AUX=A+B;
```

```
            SET CADENA = CONCAT(CADENA, ',', AUX);
```

```
            SET A=B;
```

```
            SET B=AUX;
```

```
            SET CONTADOR=CONTADOR+1;
```

```
            UNTIL CONTADOR = NUMBER-2 END REPEAT;
```

```
    END IF;
```

```
    RETURN CADENA;
```

```
END;
```

```

CREATE OR REPLACE FUNCTION CONTAR_FIBONANCI(SERIE TEXT)
RETURNS INTEGER
BEGIN
  DECLARE SUMA INTEGER DEFAULT 0;
  DECLARE CONT INTEGER DEFAULT 1;
  DECLARE FINAL INTEGER DEFAULT CHAR_LENGTH(SERIE);

  REPEAT
    SET SUMA =SUMA + SUBSTRING(SERIE,CONT,1);
    SET CONT=CONT+2;
  until CONT> FINAL END REPEAT;

  RETURN SUMA;
end;

SELECT CONTAR_FIBONANCI(SERIE_FIBONANCI(7));

```

	<i>'SERIE_FIBONANCI(6)'</i>
1	0,1,1,2,3,5

Output <i>CONTAR_FIBONANCI</i> (SER... <i>FIBONANCI</i> (6)):int(11) ×	
	<i>'CONTAR_FIBONANCI</i> (<i>SERIE_FIBONANCI</i> (6))'
1	12

PREGUNTA 11

- ○ 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**

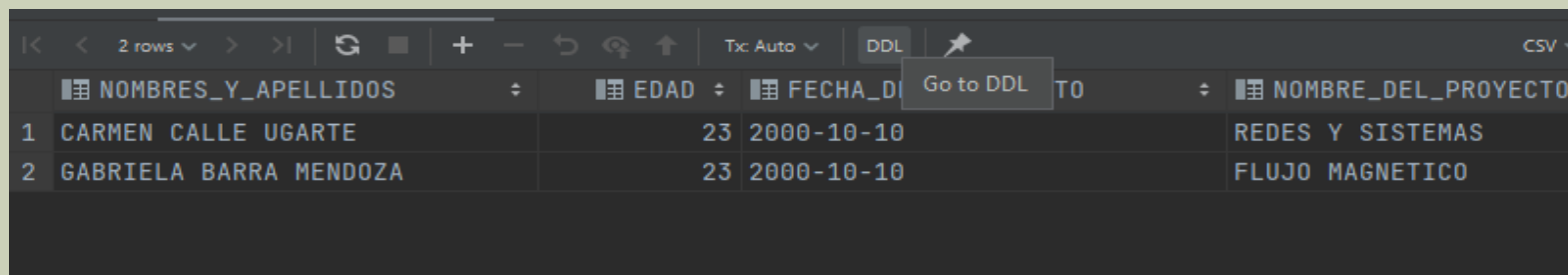
```
INSERT INTO DEPARTAMENTO(NOMBRE)
VALUES('EL ALTO');
```

```
INSERT INTO PERSONA (NOMBRE, APELLIDOS, FECHA_NAC, EDAD, EMAIL, ID_DEP, ID_PROV, GENERO)
VALUES ('CARMEN','CALLE UGARTE','2000-10-10',23,'CARMEN@GMAIL.COM',10,1,'F'),
      ('GABRIELA','BARRA MENDOZA','2000-10-10',23,'GABRIELA@GMAIL.COM',10,1,'F');
INSERT INTO DETALLE_PROYECTO(ID_PER, ID_PROY)
VALUES (6,4),
      (7,5);
```

```
CREATE OR REPLACE VIEW BUSQUEDA AS
SELECT CONCAT(PERSONA.NOMBRE,',',PERSONA.APELLIDOS) AS NOMBRES_Y_APELLIDOS,
       PERSONA.EDAD AS EDAD,
       PERSONA.FECHA_NAC AS FECHA_DE_NACIMIENTO,
       PROYECTO.NOMBRE_PROY AS NOMBRE_DEL_PROYECTO
FROM PERSONA
INNER JOIN DEPARTAMENTO ON PERSONA.ID_DEP = DEPARTAMENTO.ID_DEP
INNER JOIN DETALLE_PROYECTO ON PERSONA.ID_PER = DETALLE_PROYECTO.ID_PER
INNER JOIN PROYECTO ON DETALLE_PROYECTO.ID_PROY = PROYECTO.ID_PROY

WHERE PERSONA.GENERO='F' AND DEPARTAMENTO.NOMBRE='EL ALTO' AND PERSONA.FECHA_NAC='2000-10-10';

SELECT * FROM BUSQUEDA;
```



	NOMBRES_Y_APELLIDOS	EDAD	FECHA_DE_NACIMIENTO	NOMBRE_DEL_PROYECTO
1	CARMEN CALLE UGARTE	23	2000-10-10	REDES Y SISTEMAS
2	GABRIELA BARRA MENDOZA	23	2000-10-10	FLUJO MAGNETICO

PREGUNTA 12

- 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
- ○ 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.

```
ALTER TABLE PROYECTO ADD (ESTADO VARCHAR(30));
```

```
INSERT INTO PROYECTO(NOMBRE_PROY, TIPO_PROY)  
VALUES ('EDUCACION PERSONAS ESPECIALES','EDUCACION'),  
      ('PLANTACION DE ARBOLES','FORESTACION'),  
      ('LOS AZTECAS','CULTURA');
```

```
CREATE OR REPLACE TRIGGER UPDATE_TIP_PROYCT  
BEFORE UPDATE ON PROYECTO  
FOR EACH ROW  
BEGIN  
    IF NEW.TIPO_PROY='EDUCACION'OR NEW.TIPO_PROY='FORESTACION'OR NEW.TIPO_PROY='CULTURA'  
    THEN SET NEW.ESTADO='ACTIVO';  
    ELSE  
        SET NEW.ESTADO='INACTIVO';  
    END IF;  
END;
```

```
CREATE OR REPLACE TRIGGER ADD_TIP_PROYCT  
BEFORE INSERT ON PROYECTO  
FOR EACH ROW  
BEGIN  
    IF NEW.TIPO_PROY='EDUCACION'OR NEW.TIPO_PROY='FORESTACION'OR NEW.TIPO_PROY='CULTURA'  
    THEN SET NEW.ESTADO='ACTIVO';  
    ELSE  
        SET NEW.ESTADO='INACTIVO';  
    END IF;  
end;
```

ID_PROY	NOMBRE_PROY	TIPO_PROY	ESTADO
1	1 ANIMALES	BIOLOGIA	<null>
2	2 PREHISTORIA	ANTROPOLOGIA	<null>
3	3 MICROBIOS	BIOLOGIA	<null>
4	4 REDES Y SISTEMAS	TECNOLOGIA	<null>
5	5 FLUJO MAGNETICO	FISICA	<null>
6	6 EDUCACION PERSONAS ESPECIALES	EDUCACION	<null>
7	7 PLANTACION DE ARBOLES	FORESTACION	<null>
8	8 LOS AZTECAS	CULTURA	<null>
9	9 COMUNIDADES EN BOLIVIA	ANTROPOLOGIA	INACTIVO
10	10 REGADO DE PLANTAS	FORESTACION	ACTIVO

PREGUNTA 13

- 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 OR REPLACE TRIGGER AGREGAR_EDAD
BEFORE INSERT ON PERSONA
FOR EACH ROW
BEGIN
    SET NEW.EDAD= TIMESTAMPDIFF(YEAR,NEW.FECHA_NAC,CURDATE());
end;
```

ID_PER	NOMBRE	APELLIDOS	FECHA_NAC	EDAD	EMAIL	ID_DEP
6	CARMEN	CALLE UGARTE	2000-10-10	23	CARMEN@GMAIL.COM	10
7	GABRIELA	BARRA MENDOZA	2000-10-10	23	GABRIELA@GMAIL.COM	10
8	ALEJANDRA	RAMOS ORTEGA	2000-12-15	22	ALEJANDRA@GMAIL.COM	5
9	PEDRO	SALASAR ORTEGA	1999-12-15	22	PEDRO@GMAIL.COM	3
10	PEDRO	SALASAR ORTEGA	1996-12-15	25	PEDRO@GMAIL.COM	3

PREGUNTA 14

- 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 COPIA_PERSONA
(
  NOMBRE VARCHAR(20),
  APELLIDOS VARCHAR(50),
  FECHA_NAC DATE,
  EDAD INT,
  EMAIL VARCHAR(50),
  ID_DEP INT NOT NULL ,
  ID_PROV INT NOT NULL,
  GENERO VARCHAR(1),
  FOREIGN KEY (ID_PROV) REFERENCES PROVINCIA(ID_PROV),
  FOREIGN KEY (ID_DEP) REFERENCES DEPARTAMENTO(ID_DEP)
);

CREATE OR REPLACE TRIGGER COPIAR_PERSONA
BEFORE INSERT ON PERSONA
FOR EACH ROW
BEGIN
  INSERT INTO COPIA_PERSONA(NOMBRE, APELLIDOS, FECHA_NAC, EDAD, EMAIL, ID_DEP, ID_PROV, GENERO)
  VALUES(NEW.NOMBRE,NEW.APELLIDOS,NEW.FECHA_NAC,NEW.EDAD,NEW.EMAIL,NEW.ID_DEP,NEW.ID_PROV,NEW.GENERO);
end;

INSERT INTO PERSONA(NOMBRE, APELLIDOS, FECHA_NAC, EDAD, EMAIL, ID_DEP, ID_PROV, GENERO)
VALUES ('ALEJANDRA','RAMOS ORTEGA','2000-12-15',22,'ALEJANDRA@GMAIL.COM',5,5,'F');

```

	NOMBRE	APELLIDOS	FECHA_NAC	EDAD	EMAIL	ID_DEP	ID_PROV	GENERO
1	ALEJANDRA	RAMOS ORTEGA	2000-12-15	22	ALEJANDRA@GMAIL.COM	5	5	F

PREGUNTA 15

- 15. Crear una consulta SQL que haga uso de todas las tablas.
- ○ La consulta generada convertirlo a VISTA

```
CREATE OR REPLACE VIEW TODAS_LAS_TABLAS AS
SELECT
  CONCAT(PERSONA.NOMBRE, PERSONA.APELLIDOS) AS NOMBRE_Y_APELLIDOS,
  PERSONA.EDAD AS EDAD,
  DEPARTAMENTO.NOMBRE AS DEPARTAMENTO,
  PROVINCIA.NOMBRE AS PROVINCIA,
  CONCAT(PROYECTO.NOMBRE_PROY, ': ', TIPO_PROY) AS PROYECTO
FROM PERSONA
INNER JOIN DEPARTAMENTO ON PERSONA.ID_DEP = DEPARTAMENTO.ID_DEP
INNER JOIN PROVINCIA ON PERSONA.ID_PROV = PROVINCIA.ID_PROV
INNER JOIN DETALLE_PROYECTO ON PERSONA.ID_PER = DETALLE_PROYECTO.ID_PER
INNER JOIN PROYECTO ON DETALLE_PROYECTO.ID_PROY = PROYECTO.ID_PROY;

SELECT * FROM (TODAS_LAS_TABLAS);
```

	NOMBRE_Y_APELLIDOS	EDAD	DEPARTAMENTO	PROVINCIA	PROYECTO
1	RODRIGOMENODZA UGARTE	23	LA PAZ	VIACHA	PREHISTORIA: ANTROPOLOGIA
2	MARIALAURA TORREZ	25	SANTA CRUZ	ROBORE	ANIMALES: BIOLOGIA
3	AUGUSTOMEDRANO LOZA	24	BENI	MAGDALENA	REDES Y SISTEMAS: TECNOLOGIA
4	MARIANO FERNANDEZ GUTIERREZ	27	ORURO	CHALLAPATA	MICROBIOS: BIOLOGIA
5	CARMENCALLE UGARTE	23	EL ALTO	VIACHA	REDES Y SISTEMAS: TECNOLOGIA
6	GABRIELABARRA MENDOZA	23	EL ALTO	VIACHA	FLUJO MAGNETICO: FISICA