UNIFRANZ

Internacionalízate

BASE DE DATOS PRACTICA H4

Nombre: Henry Javier Huarachi Quispe

Docente: William Barra

MANEJO DE CONCEPTOS

Defina que es lenguaje procedural en MySQL.

- Los procedimientos almacenados MySQL son conjuntos de comandos SQL almacenados en el servidor. Esto nos permite trabajar con grandes cantidades de datos.

DEFINA QUE ES UNA FUNCIÓN EN MYSQL.

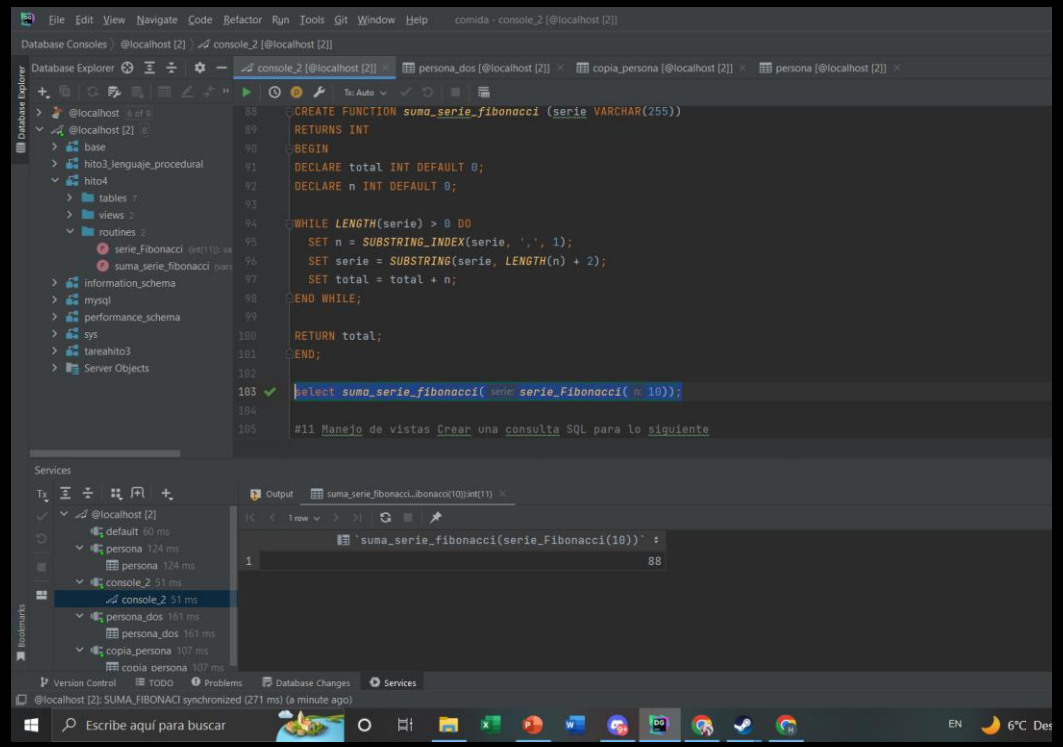
- Es una rutina creada para tomar unos parámetros, procesarlos y retornar en un salida

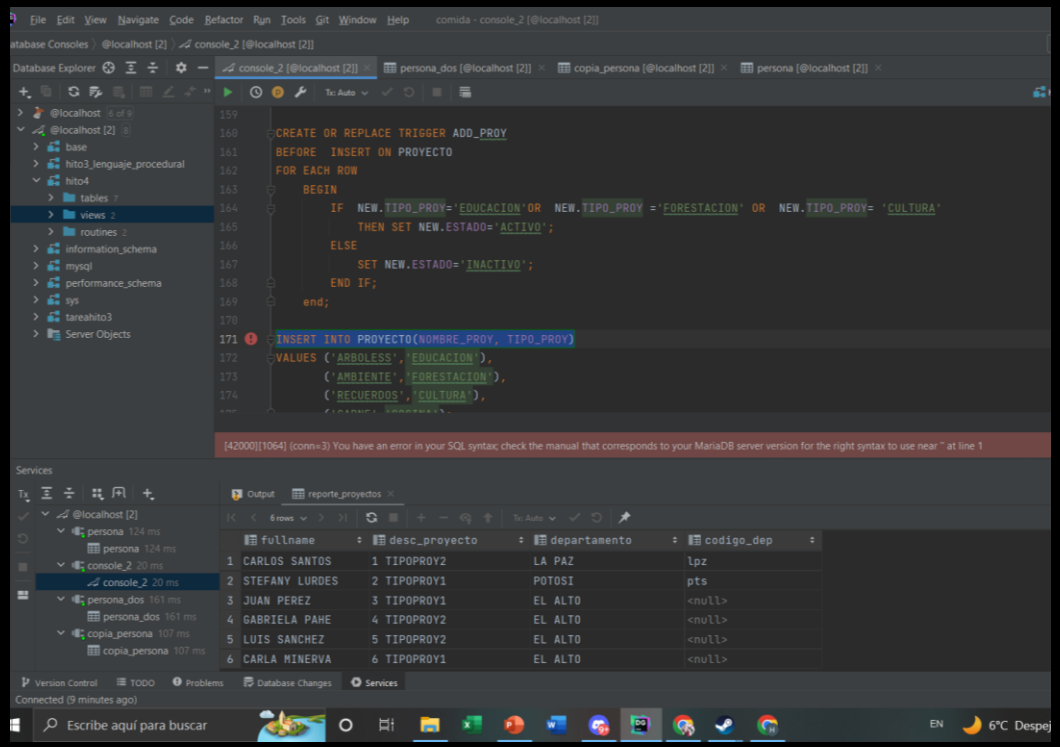
```
CREATE OR REPLACE TRIGGER ADD_PROY
BEFORE  INSERT ON PROYECTO
FOR EACH ROW
   BEGIN
      IF  NEW.TIPO_PROY='EDUCACION'OR  NEW.TIPO_PROY ='FORESTACION' OR  NEW.TIPO_PROY= 'CULTURA'
         THEN SET NEW.ESTADO='ACTIVO';
      ELSE
         SET NEW.ESTADO='INACTIVO';
      END IF;
   end;
```

```
CREATE OR REPLACE FUNCTION SUMA_FIBONACI(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;
```

3¿Cuál es la diferencia entre funciones y procedimientos almacenados?

- La diferencia entre una función y un procedimiento almacenado es que la función en si puede usarse de forma manual respecto a un procedimiento almacenado que se ejecuta automáticamente





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

- La función se ejecuta de manera manual en si en una consulta. Muy distinto del procedimiento que se ejecuta por si solo

5.Defina que es una TRIGGER en MySQL.

- Un trigger en si es una función almacenada mas usada para verificar si hay cambios en una tabla en si con las condiciones de after y before además de tener dos variables el old y el new.

```
CREATE OR REPLACE TRIGGER insert_persona_dos
BEFORE INSERT ON persona
FOR EACH ROW
BEGIN
    INSERT INTO persona_dos(id_dep,id_prov,nombre,apellidos,fecha_nac,edad,genero,email)
    VALUES (NEW.id_dep,NEW.id_prov,NEW.nombre,NEW.APELLIDOS,NEW.fecha_nac,NEW.edad,NEW.genero,NEW.email);
END;
```

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

Estas dos variables identifican el antes y el después

Old: Esta variable nos habla sobre como estuvo antes.

New: Esta variable nos muestra el respectivo cambio que tuvo.

```
CREATE OR REPLACE TRIGGER insert_persona_dos
BEFORE INSERT ON persona
FOR EACH ROW
BEGIN
    INSERT INTO persona_dos(id_dep,id_prov,nombre,apellidos,fecha_nac,edad,genero,email)
    VALUES (NEW.id_dep,NEW.id_prov,NEW.nombre,NEW.APELLIDOS,NEW.fecha_nac,NEW.edad,NEW.genero,NEW.email);
END;

INSERT INTO persona(id_dep,id_prov,nombre, apellidos,fecha_nac,edad,genero,email)
VALUES (2,2,'carlos', 'paucar', '1998-11-12',24,'m','carlos@gmail.com');

SELECT * FROM copia_persona;
```

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

Lo que indican estas dos clausulas es si el trigger se ejecutara antes o después.

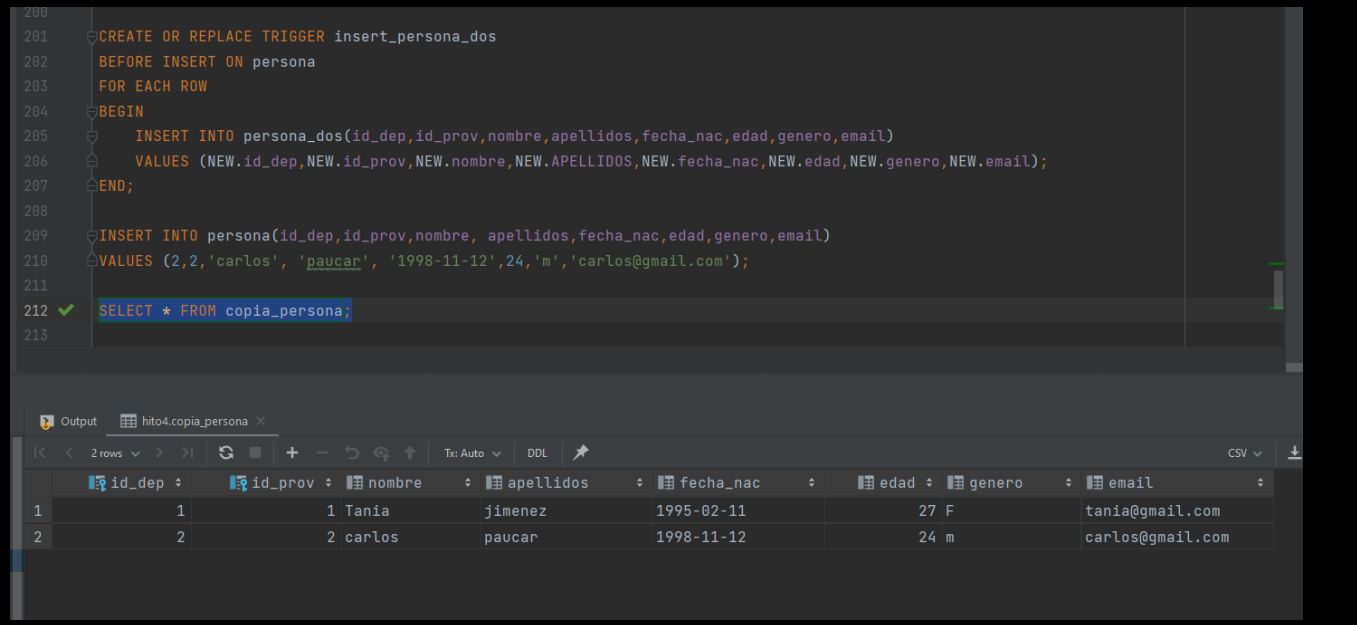
Before: antes

After. después

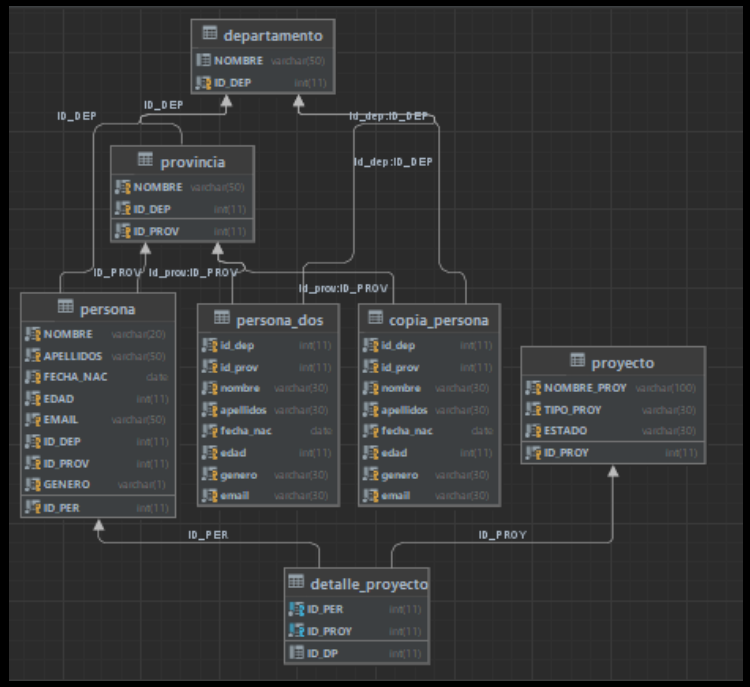
```
CREATE OR REPLACE TRIGGER ADD_PROY
BEFORE  INSERT ON PROYECTO
FOR EACH ROW
    BEGIN
        IF  NEW.TIPO_PROY='EDUCACION'OR  NEW.TIPO_PROY ='FORESTACION' OR  NEW.TIPO_PROY= 'CULTURA'
            THEN SET NEW.ESTADO='ACTIVO';
        ELSE
            SET NEW.ESTADO='INACTIVO';
        END IF;
  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

INSERT INTO PROYECTO(NOMBRE_PROY, TIPO_PROY)
VALUES ('ARBOLESS','EDUCACION'),
       ('AMBIENTE','FORESTACION'),
       ('RECUERDOS','CULTURA'),
       ('CARNE','COCINA');
SELECT PROY.*
```

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

Los eventos son aquello que hace que el trigger funcione como cuando se modifica algún registro de alguna tabla

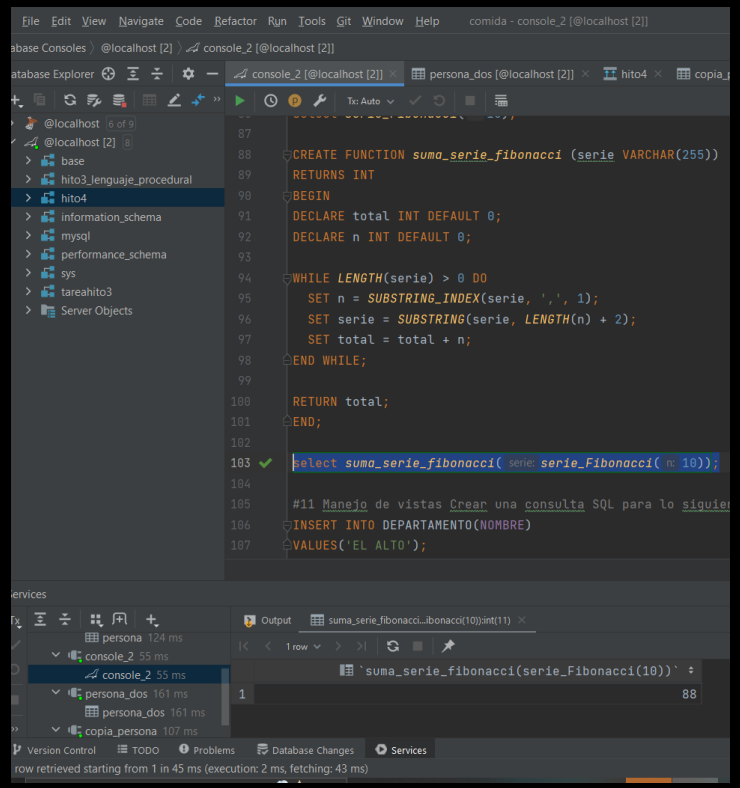


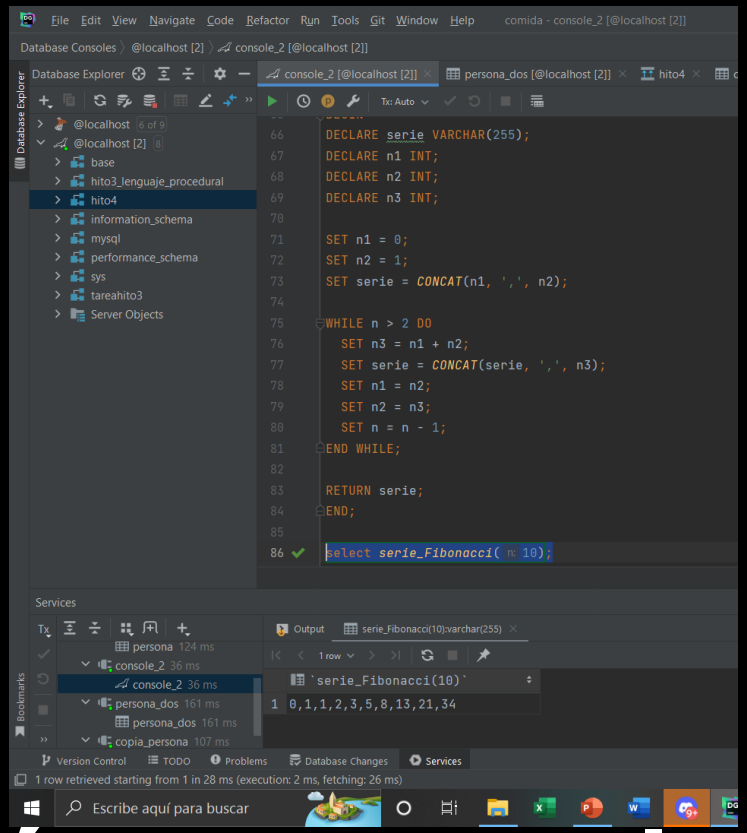
```sql
INSERT INTO  DEPARTAMENTO(NOMBRE)
VALUES ('LA PAZ'),
       ('POTOSI');
INSERT INTO  PROVINCIA(NOMBRE, ID_DEP)
VALUES ('VIACHA',1),
       ('UYUNI',2);
INSERT INTO PERSONA (NOMBRE, APELLIDOS, FECHA_NAC, EDAD, EMAIL, ID_DEP, ID_PROV, GENERO)
VALUES ('CARLOS','SANTOS','2000-12-16',21,'CARLOS1123@GMAIL.COM',1,1,'M'),
       ('STEFANY','LURDES','1998-04-10',24,'TEFFY45@GAMIL.COM',2,2,'F');
INSERT INTO  PROYECTO(NOMBRE_PROY, TIPO_PROY)
VALUES('PROYECTO1','TIPOPROY1'),
      ('PROYECTO2','TIPOPROY2');
INSERT INTO DETALLE_PROYECTO(ID_PER, ID_PROY)
VALUES (1,2),
       (2,1);
```



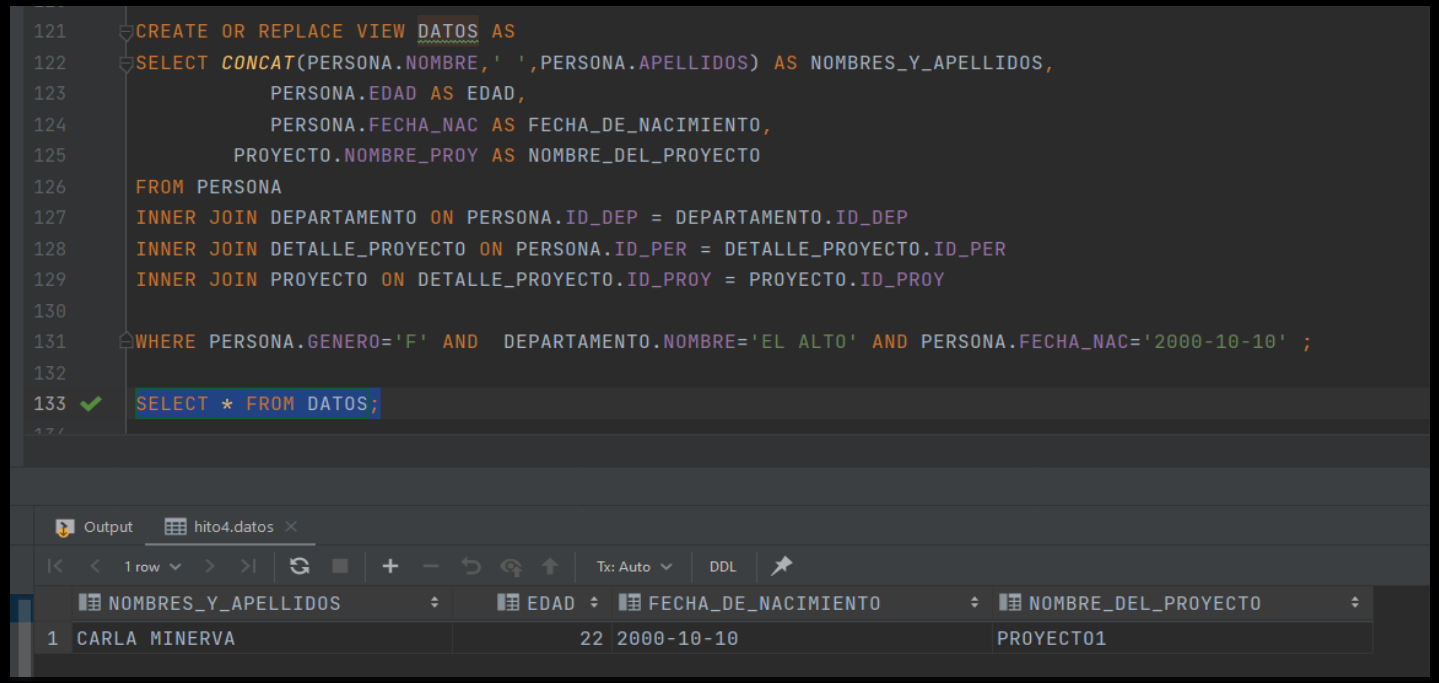
PARTE PRACTICA

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





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



11.Manejo de vistas

```
CREATE OR REPLACE TRIGGER ADD_PROY
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;
```

Output | Result 25

10 rows

ID_PROY	NOMBRE_PROY	TIPO_PROY	ESTADO
1	PROYECTO1	TIPOPROY1	<null>
2	PROYECTO2	TIPOPROY2	<null>
3	EDUCACION PERSONAS ESPECIALES	EDUCACION	<null>
4	PLANTACION DE ARBOLES	FORESTACION	<null>
5	LOS AZTECAS	CULTURA	<null>
6	ARBOLESS	EDUCACION	ACTIVO
7	ARBOLESS	EDUCACION	ACTIVO
8	AMBIENTE	FORESTACION	ACTIVO
9	RECUERDOS	CULTURA	ACTIVO
10	CARNE	COCINA	INACTIVO

12.Manejo de TRIGGERS I.

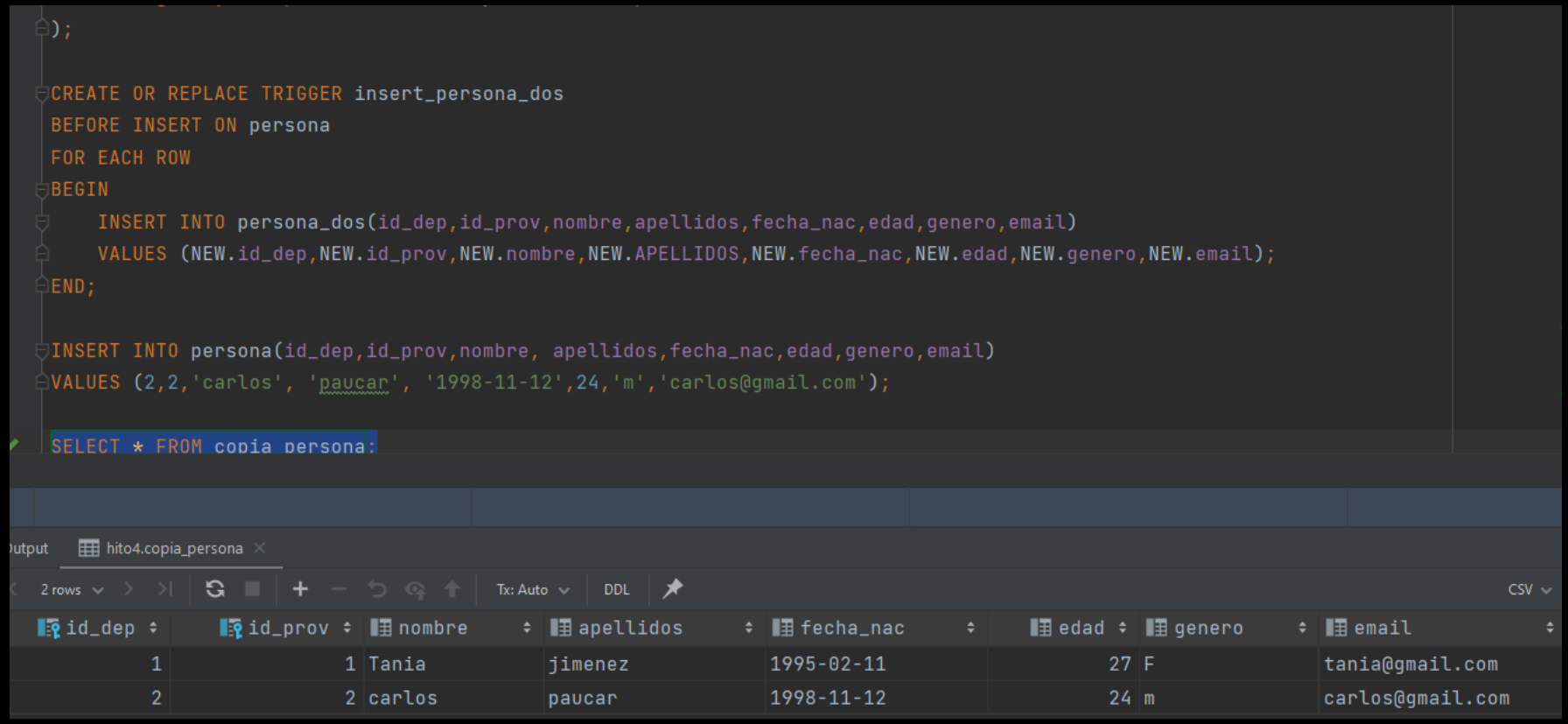
```
FROM PROYECTO AS PROY;
#13 Manejo de Triggers II El trigger debe de llamarse calculaEdad
CREATE OR REPLACE TRIGGER CALCULAEDAD
BEFORE INSERT ON PERSONA
FOR EACH ROW
    BEGIN
        SET NEW.EDAD= TIMESTAMPDIFF(YEAR,NEW.FECHA_NAC,CURDATE());
    end;

INSERT INTO  PERSONA(NOMBRE, APELLIDOS, FECHA_NAC, EMAIL, ID_DEP, ID_PROV, GENERO)
VALUES  ('JOSE','CARRASCO','1994-12-17','PEPE12@GMAIL.COM',3,1,'M');

SELECT *FROM PERSONA;
#14 Manejo de TRIGGERS III.
```

itput | Result 27

9 rows

ID_PER	NOMBRE	APELLIDOS	FECHA_NAC	EDAD	EMAIL	ID_DEP
6	CARLA	MINERVA	2000-10-10	22	CARLY123@GMAIL.COM	3
10	JOSE	CARRASCO	1994-12-17	27	PEPE12@GMAIL.COM	3
11	Tania	jimenez	1995-02-11	27	tania@gmail.com	1

13.Manejo de Triggers II.



```
);

CREATE OR REPLACE TRIGGER insert_persona_dos
BEFORE INSERT ON persona
FOR EACH ROW
BEGIN
    INSERT INTO persona_dos(id_dep,id_prov,nombre,apellidos,fecha_nac,edad,genero,email)
    VALUES (NEW.id_dep,NEW.id_prov,NEW.nombre,NEW.APELLIDOS,NEW.fecha_nac,NEW.edad,NEW.genero,NEW.email);
END;

INSERT INTO persona(id_dep,id_prov,nombre, apellidos,fecha_nac,edad,genero,email)
VALUES (2,2,'carlos', 'paucar', '1998-11-12',24,'m','carlos@gmail.com');

SELECT * FROM copia_persona;
```

utput | hito4.copia_persona

2 rows | Tx: Auto | DDL | CSV

id_dep	id_prov	nombre	apellidos	fecha_nac	edad	genero	email
1	1	Tania	jimenez	1995-02-11	27	F	tania@gmail.com
2	2	carlos	paucar	1998-11-12	24	m	carlos@gmail.com

14.Manejo de TRIGGERS III.

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

	fullname	desc_proyecto	departamento	codigo_dep
1	CARLOS SANTOS	1 TIPOPROY2	LA PAZ	lpz
2	STEFANY LURDES	2 TIPOPROY1	POTOSI	pts
3	JUAN PEREZ	3 TIPOPROY1	EL ALTO	<null>
4	GABRIELA PAHE	4 TIPOPROY2	EL ALTO	<null>
5	LUIS SANCHEZ	5 TIPOPROY2	EL ALTO	<null>
6	CARLA MINERVA	6 TIPOPROY1	EL ALTO	<null>

```sql
create or replace view reporte_proyectos as
    select concat(per.nombre,' ',per.apellidos) as fullname,
           concat(dp.id_per,' ',proy.TIPO_PROY) as desc_proyecto,
           dep.nombre as departamento,
    case
    when dep.nombre= 'la paz' then 'lpz'
    when dep.nombre ='potosi' then 'pts'
    end as codigo_dep
    from persona as per
    inner join  departamento as dep on per.id_dep = dep.id_dep
    inner join detalle_proyecto as dp on per.id_per = dp.id_per
    inner join proyecto as proy on dp.id_proy = proy.id_proy;
select *
from reporte_proyectos;
```