



# PRACTICA BASE DE DATOS HITO 2

---

Estudiante: Henry Javier Huarachi Quispe

Docente: William Barra

# MANEJO DE CONCEPTOS

---

1. ¿A que se refiere cuando se habla de bases de datos relacionales?

- Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas.

2. ¿A que se refiere cuando se habla de bases de datos no relacionales?

- son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje SQL para las consultas

# ¿Qué es MySQL y MariaDB?. Explique si existen diferencias o son iguales

---

- Mariadb es un código completamente abierto que ofrece mayor rendimiento y es mas ligero a mySql que posee módulos de código cerrado

MariaDB

MySQL

MIN  
STDEV  
SUM AVG  
MAX COUNT

# ¿Qué son las funciones de agregación?

son aquellas funciones que vienen por defecto en el MySQL



## 5. ¿Qué llegaría a ser XAMPP, WAMP SERVER o LAMP?

- XAMPP y WAMP son servidores locales. Al probar tu software localmente en estos servidores, puedes encontrar errores y fallos graves antes de comenzar a funcionar.

MIN  
STDDEV  
SUM  
AVG  
MAX  
COUNT

The screenshot shows a MySQL IDE interface. On the left, the 'Database Explorer' pane displays the database structure for '@localhost [2]', including schemas like 'base', 'information\_schema', 'mysql', 'performance\_schema', 'sys', and 'tareahito3', as well as various tables and routines. The main editor window shows a SQL script defining a function named 'ejempp3'. The function takes three varchar parameters and returns a varchar. It uses conditional logic to compare the parameters and return a message. Below the editor, the 'Services' pane shows the execution of the function with the input 'CAMISA', 'FUTBOL', and 'CAMISA', resulting in the output 'HAY DOS IGUALES'.

```

select ejempp2( palabra: 'maria antonieta velasco', parte: 17, tamaño: 7)
#ejemplo 3
create or replace function ejempp3 (palabra1 varchar(30),palabra2 varchar(30),palabra3 varchar(30))
returns varchar(90)
begin
    declare resultado varchar(60)default '';
    if STRCMP(palabra1,palabra2)=0 THEN SET resultado= 'HAY DOS IGUALES';
    ELSEIF STRCMP(palabra1,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';
    ELSEIF STRCMP(palabra2,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';
    ELSE
        SET resultado='NINGUNO ES IGUAL A OTRO';
    END IF;
    RETURN resultado;
end;
SELECT ejempp3( palabra1: 'CAMISA', palabra2: 'FUTBOL', palabra3: 'CAMISA')
  
```

Output: 1 HAY DOS IGUALES

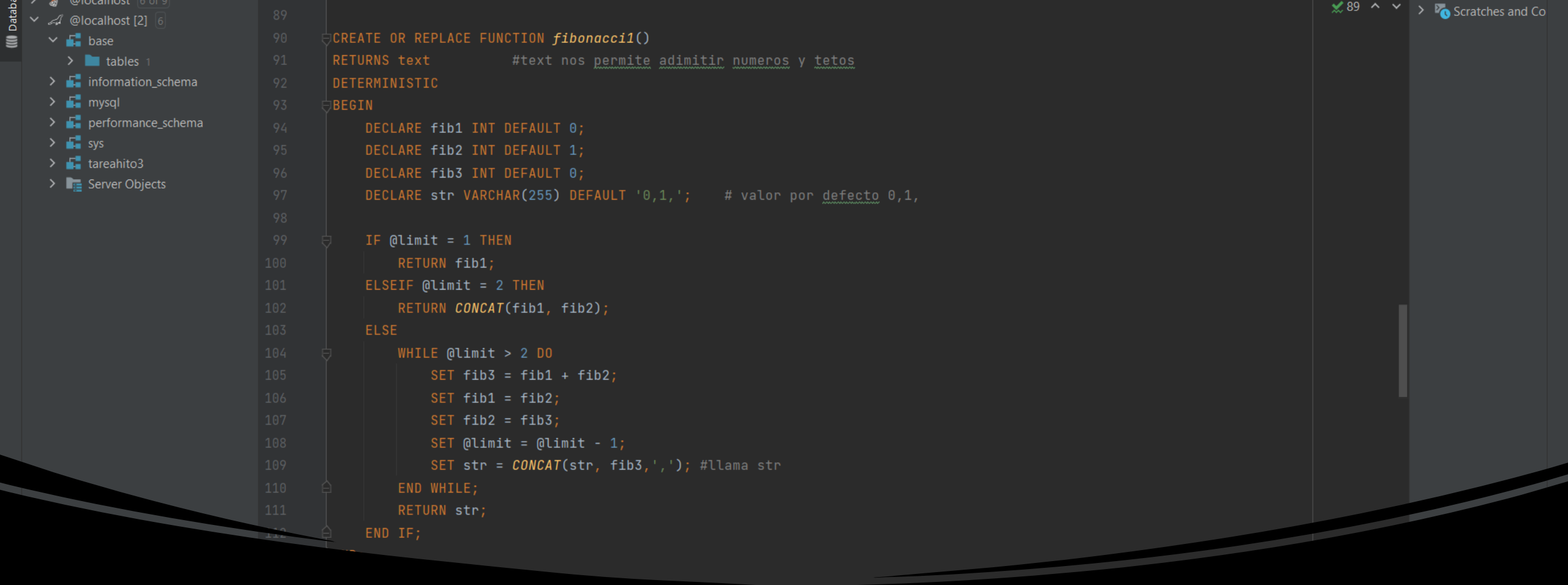
6. ¿Cuál es la diferencia entre las funciones de agregación y funciones creadas por el DBA? Es decir funciones creadas por el usuario.

- Las funciones de agregación son aquellas funciones que vienen por defecto en el MySQL. Las funciones creadas por el DBA son funciones que el usuario necesita crear en base a su necesidad usando también funciones de agregación.



# 8. Que es DML y DDL?

- DML:son aquellas utilizadas para insertar, borrar, modificar y consultar los datos de una base de datos.
- DDL:son aquellas utilizadas para la creación de una base de datos y todos sus componentes: tablas, índices, relaciones, disparadores (triggers)



```
89
90 CREATE OR REPLACE FUNCTION fibonacci1()
91 RETURNS text          #text nos permite admitir numeros y tetos
92 DETERMINISTIC
93 BEGIN
94     DECLARE fib1 INT DEFAULT 0;
95     DECLARE fib2 INT DEFAULT 1;
96     DECLARE fib3 INT DEFAULT 0;
97     DECLARE str VARCHAR(255) DEFAULT '0,1,';    # valor por defecto 0,1,
98
99     IF @limit = 1 THEN
100         RETURN fib1;
101     ELSEIF @limit = 2 THEN
102         RETURN CONCAT(fib1, fib2);
103     ELSE
104         WHILE @limit > 2 DO
105             SET fib3 = fib1 + fib2;
106             SET fib1 = fib2;
107             SET fib2 = fib3;
108             SET @limit = @limit - 1;
109             SET str = CONCAT(str, fib3, ','); #llama str
110         END WHILE;
111         RETURN str;
112     END IF;
```

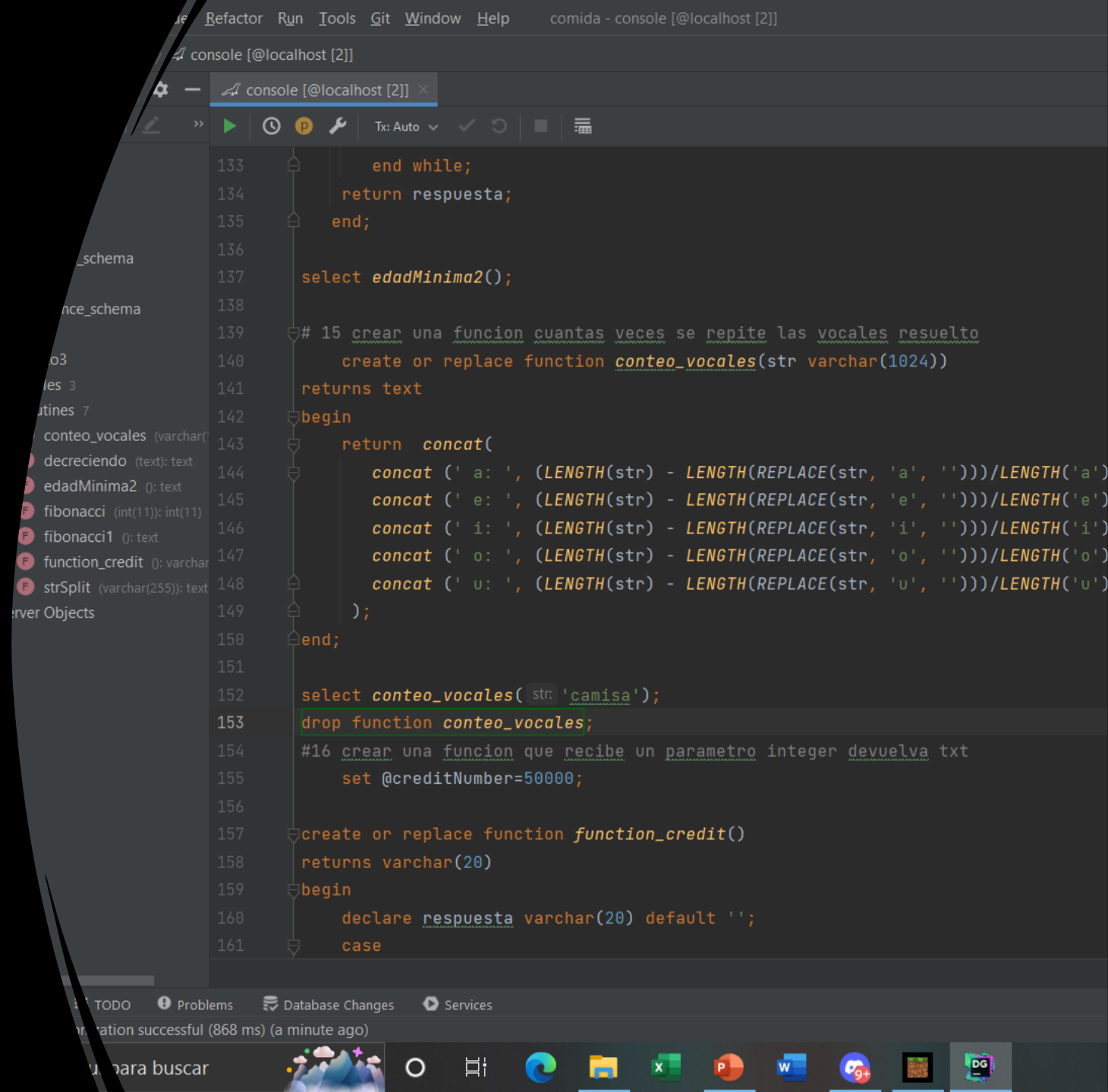
9. ¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parametros, etc

- Una function debe Tener : nombre , operaciones , parametros , returns , return



# 10. ¿Cómo crear, modificar y cómo eliminar una función?

- Una function se crea con : create function (nombre)
- Se elimina con : drop function(nombre de la funcion)
- Create or replace function (nombre) para modificar la funcion..



The screenshot shows a SQL script in a development environment. The script includes the following SQL statements:

```
133     end while;
134     return respuesta;
135 end;
136
137 select edadMinima2();
138
139 # 15 crear una funcion cuantas veces se repite las vocales resuelto
140 create or replace function conteo_vocales(str varchar(1024))
141 returns text
142 begin
143     return concat(
144         concat (' a: ', (LENGTH(str) - LENGTH(REPLACE(str, 'a', '')))/LENGTH('a')),
145         concat (' e: ', (LENGTH(str) - LENGTH(REPLACE(str, 'e', '')))/LENGTH('e')),
146         concat (' i: ', (LENGTH(str) - LENGTH(REPLACE(str, 'i', '')))/LENGTH('i')),
147         concat (' o: ', (LENGTH(str) - LENGTH(REPLACE(str, 'o', '')))/LENGTH('o')),
148         concat (' u: ', (LENGTH(str) - LENGTH(REPLACE(str, 'u', '')))/LENGTH('u'))
149     );
150 end;
151
152 select conteo_vocales(str: 'camisa');
153 drop function conteo_vocales;
154
155 #16 crear una funcion que recibe un parametro integer devuelva txt
156 set @creditNumber=50000;
157
158 create or replace function function_credit()
159 returns varchar(20)
160 begin
161     declare respuesta varchar(20) default '';
162     case
```

The interface also shows a sidebar with a database schema tree on the left and a status bar at the bottom indicating a successful operation.

# PARTE PRACTICA

11. Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER.



```
INSERT INTO CLIENTE(FULLNAME, LASTNAME, EDAD, DOMICILIO)
VALUES('PEDRO','SALAZAR GOMEZ',19,'RIO SECO'),
      ('MARIA','DELGADO MENDOZA',20,'BALLIVIAN');

INSERT INTO PEDIDO(ARTICULO, COSTO, FECHA)
VALUES('LECHE',21.2,'2022-08-02'),
      ('CARNE',7.4,'2022-09-12');

INSERT INTO DETALLE_PEDIDO(ID_PEDIDO, ID_CLIENTE)
VALUES(1,2),
      (2,1);
```

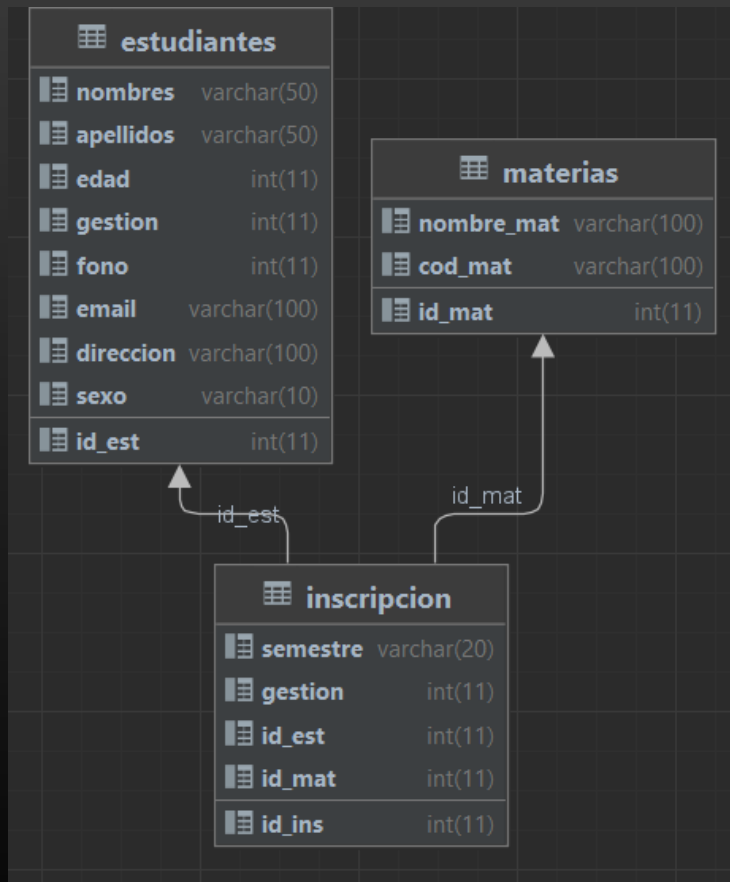
## 12. Crear una consulta SQL en base al ejercicio anterior.

---

#12

```
SELECT DETALLE_PEDIDO.ID_DETALLE_PEDIDO AS CODIGO_COMPRA,  
       CONCAT(CLIENTE.FULLNAME,CLIENTE.LASTNAME) AS NOMBRE_COMPLETO,  
       PEDIDO.ARTICULO,PEDIDO.COSTO  
  
FROM DETALLE_PEDIDO  
INNER JOIN CLIENTE ON DETALLE_PEDIDO.ID_CLIENTE = CLIENTE.ID_CLIENTE  
INNER JOIN PEDIDO ON DETALLE_PEDIDO.ID_PEDIDO = PEDIDO.ID_PEDIDO;
```

# 13. Crear un función que compare dos códigos de materia.



```
RETURNS BOOLEAN
BEGIN
    DECLARE RESPUESTA BOOLEAN;

    IF CODI_MAT=NOMBRE_MAT
    THEN
        SET RESPUESTA=1;
    END IF;

    RETURN RESPUESTA;
END;

SELECT ESTUDIANTES.id_est, ESTUDIANTES.NOMBRES, ESTUDIANTES.APELLIDOS,
       MATERIAS.NOMBRE_MAT, MATERIAS.COD_MAT
FROM inscripcion
INNER JOIN estudiantes ON inscripcion.id_est = estudiantes.id_est
INNER JOIN materias ON inscripcion.id_mat = materias.id_mat
WHERE COMPARA_MATERIAS( CODI_MAT: materias.cod_mat, NOMBRE_MAT: 'ARQ-105');
```

COMPARA\_MATERIAS()

Result 29

1 row

id_est	NOMBRES	APELLIDOS	NOMBRE_MAT	COD_MAT
5	Santos	Montes Valenzuela	Fisica Basica	ARQ-105

14. Crear una función que permita obtener el promedio de las edades del género masculino o femenino de los estudiantes inscritos en la asignatura ARQ-104.

```
CREATE OR REPLACE FUNCTION PROMEDIO_GENERO_MATERIA( GENERO VARCHAR(11), MATERIA_COD VARCHAR(100))
RETURNS INTEGER
BEGIN
    DECLARE PROMEDIO INTEGER DEFAULT 0;

    SELECT AVG(estudiantes.edad) INTO PROMEDIO
    FROM inscripcion
    INNER JOIN estudiantes ON inscripcion.id_est = estudiantes.id_est
    INNER JOIN materias ON inscripcion.id_mat = materias.id_mat
    WHERE estudiantes.sexo= GENERO AND materias.cod_mat=MATERIA_COD;
    RETURN PROMEDIO;
END;
```

```
SELECT PROMEDIO_GENERO_MATERIA( GENERO: 'femenino', MATERIA_COD: 'ARQ-104');
```

#15

```
CREATE OR REPLACE FUNCTION CONCATENAR_PALABRAS (A1 VARCHAR(30), A2 VARCHAR(30), A3 VARCHAR(30))
RETURNS VARCHAR(90)
BEGIN
```

it PROMEDIO\_GENERO\_MATE...o','ARQ-104'):int(11) X

1 row > > | ↺ ■ ★

PROMEDIO\_GENERO\_MATERIA('femenino','ARQ-104')` ↕

23

# 15. Crear una función que permita concatenar 3 cadenas.

---

```
151
152 CREATE OR REPLACE FUNCTION CONCATENAR_PALABRAS (A1 VARCHAR(30),A2 VARCHAR(30),A3 VARCHAR(30))
153 RETURNS VARCHAR(90)
154 BEGIN
155     DECLARE CADENA_TOTAL VARCHAR(90) DEFAULT '';
156     SET CADENA_TOTAL= CONCAT('(',A1,',') ('(',A2,',') ('(',A3,',')');
157
158     RETURN CADENA_TOTAL ;
159 END;
160
161 ✓ SELECT CONCATENAR_PALABRAS( A1: 'PEPITO', A2: 'PEP', A3: '50') AS CONCATENADO_TOTAL;
162
163 #16
164
165
166 CREATE OR REPLACE FUNCTION EST_SEGUN_EDAD_GENERO(SEXO VARCHAR(11),EDAD INTEGER)
167 RETURNS BOOLEAN
168 BEGIN
169     DECLARE SUMA INTEGER DEFAULT 0;
```

Output    CONCATENADO\_TOTAL:varchar(90) ×

1 row

CONCATENADO_TOTAL
1 (PEPITO) (PEP) (50)

# 16. Crear una función de acuerdo a lo siguiente:

```
CREATE OR REPLACE FUNCTION EST_SEGUN_EDAD_GENERO(SEX0 VARCHAR(11),EDAD INTEGER)
RETURNS BOOLEAN
BEGIN
    DECLARE SUMA INTEGER DEFAULT 0;
    DECLARE SIoNO BOOLEAN;

    SELECT SUM(ESTUDIANTES.EDAD) INTO SUMA
    FROM estudiantes
    WHERE estudiantes.SEX0=SEX0;

    IF SUMA%2=0 AND SUMA>EDAD
    THEN
        SET SIoNO=1;
    END IF;
    RETURN SIoNO;
END;

SELECT ESTUDIANTES.NOMBRES, ESTUDIANTES.APELLIDOS,ESTUDIANTES.EDAD,INSCRIPCION.SEMESTRE
FROM inscripcion
INNER JOIN estudiantes ON inscripcion.id_est = estudiantes.id_est;
```

Output SUM(EDAD):int(11) X

2 rows > >| ↺ ■ ★

☰ `SUM(EDAD)` ▾

46

74



# 17. Crear una función de acuerdo a lo siguiente:

```
CREATE OR REPLACE FUNCTION BUSCA_NOMBRES_APELLIDOS(nombre varchar(100),apellidos varchar(100),nombre_c varchar (100),apellido_c var

RETURNS BOOLEAN
BEGIN
    DECLARE DECISION BOOLEAN;
    if nombre=nombre_c and apellidos=apellido_c
    then
        set DECISION=1;
    end if;
    RETURN DECISION;
END;

SELECT EST.*
from estudiantes as est
where BUSCA_NOMBRES_APELLIDOS( nombre: est.nombres, apellidos: est.apellidos, nombre_c: 'sandra', apellido_c: 'mavir uria')
```

put tareahito2.estudiantes

id_est	nombres	apellidos	edad	gestion	fono	email	direccion	sexo
2	Sandra	Mavir Uria	25	<null>	2832116	sandra@gmail.com	Av. 6 de Agosto	femenino