

PANTONE 108 C

UNIVERSIDAD PRIVADA FRANZ TAMAYO

Defensa-Hito 3

Estudiante: FABIVA VEYRA RAMOS VERASTEGUI

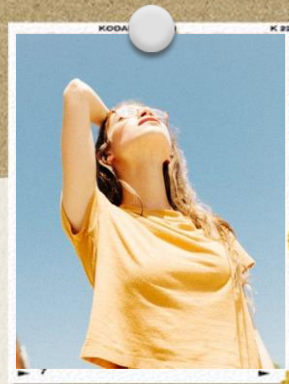
Asignatura: BASE DE DATOS II

Carrera: INGENIERÍA DE SISTEMAS

Sede: El Alto

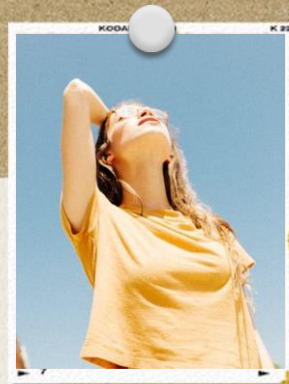
Paralelo: BDA (1)

Docente: Lic. William Barra Paredes



PANTONE 108 C

Manejo de conceptos.



1. Defina que es lenguaje procedural en MySQL

- ✗ Lenguajes procuderales o procedimentales: El usuario da órdenes para que se realicen las tareas pertinentes con el objetico de recuperar los datos requeridos. Es la base del lenguaje de consulta SQL.

2. Defina que es una función en MySQL.

- ✗ Las funciones es un fragmento de codigo en lenguaje procedural con piezas de código que reciben datos de entrada, realizan operaciones con ellos y luego devuelven un resultado.



3. *¿Qué cosas características debe de tener una función?*

MySQL funciona junto con varios programas de utilidad que soportan la administración de las bases de datos MySQL. Los comandos se envían a MySQLServer a través del cliente MySQL, que se instala en una computadora. MySQL fue desarrollado originalmente para manejar rápidamente grandes bases de datos.



4. ¿Cómo crear, modificar y cómo eliminar una función? Adjunte un ejemplo de su uso.

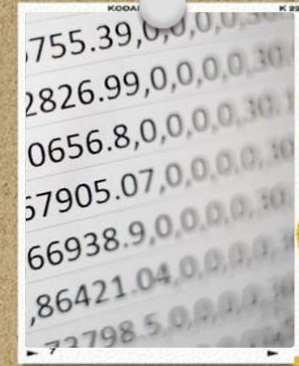
Crear una función CREATE FUNCTION

```
create function max_edad_est_masculino()  
  
returns integer  
  
begin  
  
return (  
  
select MAX(est.edad)  
  
from estudiantes AS est  
  
where est.sexo='masculino'  
  
);  
  
end;
```

Modificar una función CREATE OR REPLACE FUNCTION

```
Create or replace function max_edad_est_masculino()  
  
returns integer  
  
begin  
  
return (  
  
select MAX(est.edad)  
  
from estudiantes AS est  
  
where est.sexo='masculino'  
  
);  
  
end;
```

Eliminar una función DROP FUNCTION



5. Para qué sirve la función CONCAT y como funciona en MySQL

*Crear una función que muestre el uso de la función CONCAT?
La función debe concatenar 3 cadenas.*

CONCAT sirve para concatenar varias variables y más; en la que funciona insertando "Select CONCAT " seguido de la introducción de lo que queremos que muestre en paréntesis "('Hola', 'Mundo')".

6. Para qué sirve la función SUBSTRING y como funciona en MySQL

*¿Crear una función que muestre el uso de la función SUBSTRING?
La función recibe un nombre completo.
INPUT: Ximena Condori Mar
La función solo retorna el nombre.
OUTPUT: Ximena*

SUBSTRING sirve para contar una cadena de cierto lugar con un conteo determinado.



7. Para qué sirve la función STRCMP y como funciona en MYSQL

¿Crear una función que muestre el uso de la función STRCMP?

La función debe comparar 3 cadenas. Y deberá determinar si dos de ellas son iguales.

La función STRCMP compara carácter por carácter, si las cadenas son iguales retornara un valor 0, si la primera cadena es mayor retornara un valor positivo y si es menor retornara un valor negativo.

```
if (strcmp(str1,str2) == 0)
{
    printf("Son iguales!\n");
}
int strcmp(const char * s1,const char * s2);
```

8. Para qué sirve la función CHAR_LENGTH y LOCATE y como funciona en MYSQL

➤ **CHAR_LENGTH:** Para encontrar la longitud de una string dada (en caracteres). Cuenta el número de caracteres e ignora si los caracteres son de un solo byte o de varios bytes.

select char_length('Hola Mundo') as Mundo

➤ **LOCATE:** Es un comando del paquete findutils de GNU que provee una fácil y rápida manera de buscar archivos en el sistema completo basado en patrones de nombres.

Select locate ('La', 'Luna') as Locate



9. ¿Cual es la diferencia entre las funciones de agregación y funciones creados por el DBA?

Es decir funciones creadas por el usuario.

- X La diferencia entre las funciones de agregación son las que no se pueden utilizar en la clausula select y la funciones creados por el DBA son las que se pueden utilizar en la clausula select.

10. ¿Busque y defina a qué se referirá cuando se habla de parámetros de entrada y salida en MySQL?

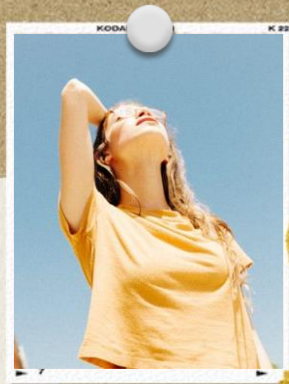
- X Se refiere a definir un parámetro en un procedimiento almacenado en MySQL es definirlo de entrada y salida en forma simultanea. Definimos un parámetro de entrada y salida mediante la palabra clave inout:

```
create procedure NOMBREPROCEDIMIENTO (inout NOMBREPARAMETRO TIPODEDATO)
begin
end
```



PANTONE 108 C

Parte practica



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



```
CREATE DATABASE HITO3;  
USE HITO3;
```

```
CREATE TABLE estudiantes (  
  nombres varchar (30),  
  apellidos varchar (50),  
  edad int,  
  fono int,  
  email varchar (100),  
  direccion varchar (100),  
  sexo varchar (10),  
  id_est int not null auto_increment primary key
```

```
); CREATE TABLE  
materias (  
  nombre_mat varchar (100),  
  cod_mat varchar (100),  
  id_mat int not null auto_increment primary key
```

```
);  
CREATE TABLE inscripcion (  
  semestre varchar (20),  
  gestion int,  
  id_est int not null,  
  id_mat int not null,  
  foreign key (id_est) references  
estudiantes(id_est),  
  foreign key (id_mat) references materias(id_mat)  
);
```

```
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)  
VALUES ('Miguel', 'Gonzales Veliz', 20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino'),  
('Sandra', 'Mavir Uria', 25, 2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino'),  
('Joel', 'Adubiri Mondar', 30, 2832117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino'),  
('Andrea', 'Arias Ballesteros', 21, 2832118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino'),  
('Santos', 'Montes Valenzuela', 24, 2832119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
```

```
INSERT INTO materias (nombre_mat, cod_mat)  
VALUES ('Introduccion a la Arquitectura', 'ARQ-101'),  
('Urbanismo y Diseño', 'ARQ-102'),  
('Dibujo y Pintura Arquitectonica', 'ARQ-103'),  
('Matematica discreta', 'ARQ-104'),  
('Fisica Basica', 'ARQ-105');
```

```
INSERT INTO inscripcion (semestre, gestion, id_est, id_mat)  
VALUES ('1er Semestre', 2018, 1, 1),  
('2do Semestre', 2018, 1, 2),  
('1er Semestre', 2019, 2, 4),  
('2do Semestre', 2019, 2, 3),  
('2do Semestre', 2020, 3, 3),  
('3er Semestre', 2020, 3, 1),  
('4to Semestre', 2021, 4, 4),  
('5to Semestre', 2021, 5, 5);
```

```
SELECT est.*  
FROM estudiantes AS est;  
SELECT mat.*  
FROM materias AS mat;  
SELECT ins.*  
FROM inscripcion AS ins;
```


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

estudiantes	
id_est	int(11)
nombres	varchar(30)
apellidos	varchar(50)
edad	int(11)
fono	int(11)
email	varchar(100)
direccion	varchar(100)
sexo	varchar(10)

materias	
id_mat	int(11)
nombre_mat	varchar(100)
cod_mat	varchar(100)

id_est

id_mat

inscripcion	
semestre	varchar(20)
gestion	int(11)
id_est	int(11)
id_mat	int(11)

```
64 SELECT est.*
65 FROM estudiantes AS est;
```

nombres	apellidos	edad	fono	email	direccion	sexo	id_est
1 Miguel	Gonzales Veliz	28	2832115	miguel@gmail.com	Av. 6 de Agosto	masculino	1
2 Sandra	Mavir Uria	25	2832116	sandra@gmail.com	Av. 6 de Agosto	femenino	2
3 Joel	Adubiri Mondar	38	2832117	joel@gmail.com	Av. 6 de Agosto	masculino	3
4 Andrea	Arias Ballesteros	21	2832118	andrea@gmail.com	Av. 6 de Agosto	femenino	4
5 Santos	Montes Valenzuela	24	2832119	santos@gmail.com	Av. 6 de Agosto	masculino	5

```
667 SELECT mat.*
668 FROM materias AS mat;
```

nombre_mat	cod_mat	id
1 Introduccion a la Arquitectura	ARQ-101	
2 Urbanismo y Diseño	ARQ-102	
3 Dibujo y Pintura Arquitectonico	ARQ-103	
4 Matematica discreta	ARQ-104	
5 Fisica Basica	ARQ-105	

```
670 SELECT ins.*
671 FROM inscripcion AS ins;
```

semestre	gestion	id_est	id_mat
1 1er Semestre	2018	1	1
2 2do Semestre	2018	1	2
3 1er Semestre	2019	2	4
4 2do Semestre	2019	2	3
5 2do Semestre	2020	3	3
6 3er Semestre	2020	3	1
7 4to Semestre	2021	4	4
8 5to Semestre	2021	5	5

12. Crear una función que genere la serie Fibonacci.

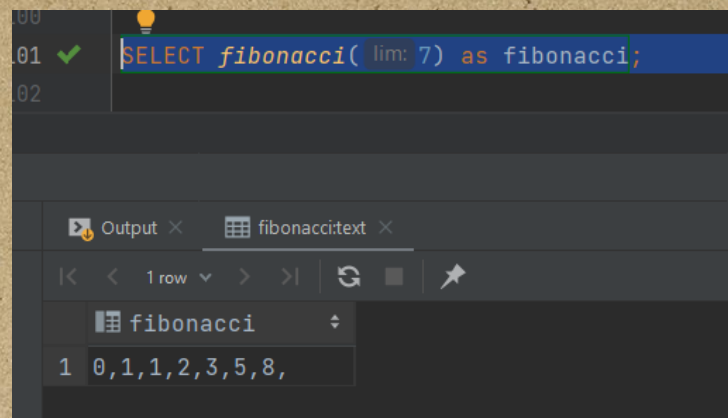
```
CREATE FUNCTION fibonacci(lim int)
RETURNS TEXT
BEGIN
  declare dato int default 0;
  declare dato1 int default 1;
  declare dato2 int default 0;
  declare x int default 1;
  declare respuesta text;
  IF lim >= 1
  THEN
    SET respuesta = concat(dato, ',');
  END IF;
  IF lim >= 2
  THEN
    SET respuesta = concat(respuesta, dato1, ',');
  END IF;
  if lim >= 3
  THEN
    WHILE x <= (lim - 2) do
      set dato2 = dato + dato1;
      set respuesta = concat(respuesta, dato2, ',');
      set dato = dato1;
      set dato1 = dato2;
      set x = x + 1;
    END WHILE;
  END IF;
  RETURN respuesta;
END;
SELECT fibonacci(7) as fibonacci;
```

La función recibe un límite(number)

La función debe de retornar una cadena.

Ejemplo para n=7. OUTPUT: 0, 1, 1, 2, 3, 5, 8.

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



```
00
01 ✓ SELECT fibonacci(lim: 7) as fibonacci;
02
```

Output x fibonacci:text x

1 row

fibonacci
1 0,1,1,2,3,5,8,

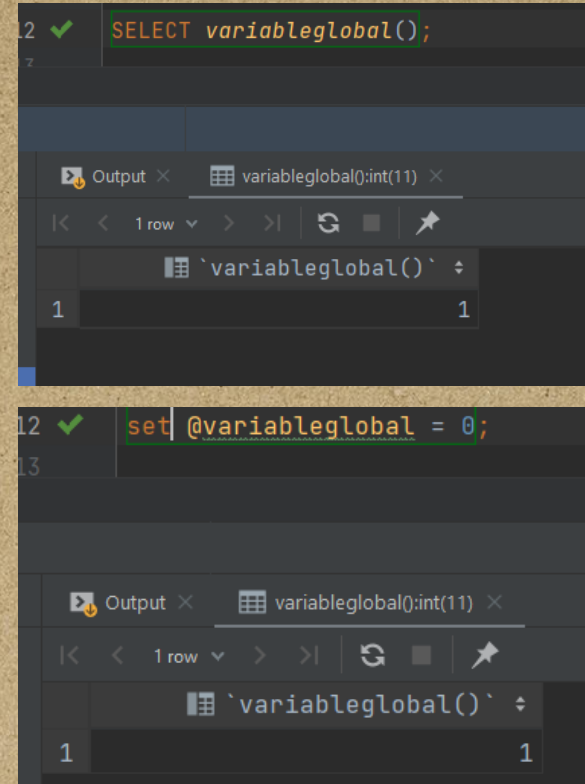
13. Crear una variable global a nivel BASE DE DATOS

Crear una función cualquiera.

La función debe retornar la variable global.

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

```
CREATE FUNCTION variableglobal ()  
RETURNS INT  
BEGIN  
    SET @variableglobal = 1;  
    RETURN @variableglobal;  
END;  
  
SELECT variableglobal();  
  
set @variableglobal = 0;
```



The first screenshot shows the SQL query `SELECT variableglobal();` being executed. The output window displays a single row with the value 1.

The second screenshot shows the SQL query `set @variableglobal = 0;` being executed. The output window displays a single row with the value 1.

14. Crear una función no recibe parámetros

(Utilizar WHILE, REPEAT & LOOP).

Previamente deberá de crear una función que obtenga la edad mínima de los estudiantes. La función no recibe ningún parámetro.

La función debe de retornar un número. (LA EDAD MINIMA). Si la edad mínima es PAR mostrar todos los pares empezando desde 0 a este ese valor de la edad mínima.

`paresImpares()`

1 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24,

Si la edad mínima es IMPAR mostrar descendentemente todos los impares hasta el valor 0.

`paresImpares()`

1 25, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1,

```
CREATE FUNCTION funtion1()
RETURNS TEXT
BEGIN
    declare edad int default 0;
    declare cont int default 0;
    declare res text default '';
    set edad = EdadMinima();
    IF (edad % 2 = 0)
    THEN
        WHILE (cont <= edad) do
            IF (cont % 2 = 0)
            THEN
                SET res = concat(res, cont, ', ');
            END IF;
            SET cont = cont + 1;
        END WHILE;
    ELSE
        SET cont = edad;
        WHILE (cont >= 0) do
            IF (cont % 2 = 1)
            THEN
                SET res = concat(res, cont, ', ');
            END IF;
            SET cont = cont - 1;
        END WHILE;
    END IF;
    RETURN res;
END;
```

SELECT funtion1();

1 funtion1()

1 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

15. Crear una función que determina cuantas veces se repite las vocales.

La función recibe una cadena y retorna un TEXT.

Retornar todas las vocales ordenadas e indicando la cantidad de veces que se repite en la cadena.

Resultado esperado

```
CREATE OR REPLACE FUNCTION CONTARVOCALES(CADENA TEXT)
RETURNS TEXT
BEGIN
    declare x int default 1;
    declare res text default '';
    declare lim int default char_length(cadena);
    declare letra char default '';
    declare a int default 0;
    declare e int default 0;
    declare i int default 0;
    declare o int default 0;
    declare u int default 0;
    WHILE x <= lim
    DO
        SET letra = substring(cadena,x, 1);
        IF letra = 'a'
        THEN
            SET
                a = a + 1;
        ELSE IF letra = 'e'
        THEN
            SET
                e = e + 1;
        ELSE IF letra = 'i'
```

```
        THEN
            SET
                i = i + 1;
        ELSE IF letra = 'o'
        THEN
            SET
                o = o + 1;
        ELSE IF letra = 'u'
        THEN
            SET
                u = u + 1;
        end if;
    end if;
    end if;
    end if;
    SET x = x + 1;
END WHILE;
```

```
SET res = concat('a:',a,',','e:',e,',','i:',i,',','o:',o,',','u:',u);
RETURN (res);
END;
```

```
SELECT ContarVocales('taller de base de datos') as Vocales;
```

Vocales

1 a:3 e:4 i:0 o:1 u: 0

16. Crear una función que recibe un parámetro INTEGER.

La función debe de retornar un texto(TEXT) como respuesta.

El parámetro es un valor numérico credit_number.

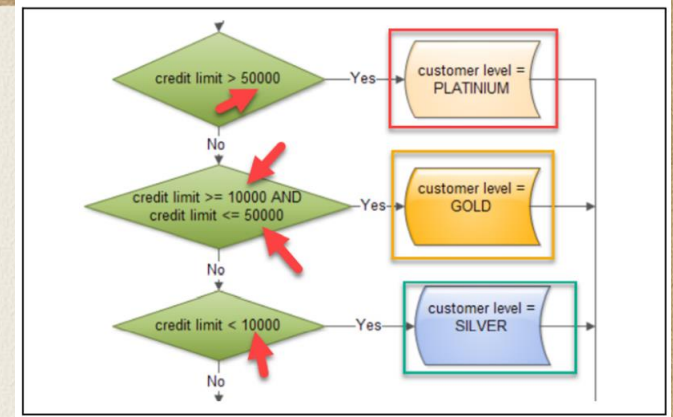
Si es mayor a 50000 es PLATINIUM.

Si es mayor igual a 10000 y menor igual a 50000 es GOLD.

Si es menor a 10000 es SILVER

La función debe retornar indicando si ese cliente es PLATINIUM, GOLD o SILVER en base al valor del credit_number.

Considere la imagen siguiente:



```
CREATE OR REPLACE FUNCTION usuario(credit_number INT)
RETURNS TEXT
BEGIN
    DECLARE respuesta text default "";
    SET respuesta = (
        SELECT CASE
            when credit_number > 50000 then 'PLATINIUM.'
            when credit_number >= 10000 && credit_number <= 50000 then 'GOLD.'
            when credit_number < 10000 then 'SILVER.'
        end);
    RETURN respuesta;
END;

select usuario(100000);
```

```
271
272 ✓ select usuario(100000);
273
```

Output x usuario(100000) x

1 row

```
'usuario(100000)'
1 PLATINIUM.
```


17. Crear una función que reciba un parámetro TEXT

En donde este parámetro deberá de recibir una cadena cualquiera y retorna un TEXT de respuesta.
Concatenar N veces la misma cadena reduciendo en uno en cada iteración hasta llegar a una sola letra.
Utilizar REPEAT y retornar la nueva cadena concatenada.
Considerar la siguiente imagen:

```
create or replace function descomponer(num text)
returns text
begin
  declare respuesta text default '';
  declare concatenar int default char_length(num);
  declare lim int default 1;
  declare x int default concatenar;
  repeat
    if concatenar >= lim
    then
      set respuesta = concat(substr(num, concatenar, x-1), ', ', respuesta);
      set concatenar = concatenar - 1;
    end if;
  until concatenar <= 0
  end repeat;
  return(respuesta);
end;
```

```
select descomponer('COMPUTADORA') as descomponer;
```

```
select descomponer('COMPUTADORA') as descomponer;
```

Output x descomponer x

1 row

descomponer

1 COMPUTADOR , OMPUTADORA , MPUTADORA , PUTADORA , UTADORA , TADORA , ADORA , DORA , ORA , RA , A ,



PANTONE 108 C

Gracias. ..

