

BASE DE DATOS

Tarea hito 3

Estudiante: Ilia Araceli Sarzo Laura

Codigo: SIS14125434

Docente: William Barra

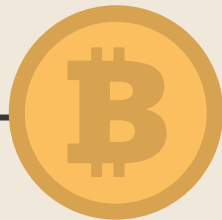


UNIFRANZ

UNIVERSIDAD PRIVADA

FRANZ TAMAYO

Manejo de Conceptos



Defina que es lenguaje procedural en My SQL

Es en síntesis la programación a nivel de DBA, en donde generamos distintas estructuras de control dentro de funciones almacenadas

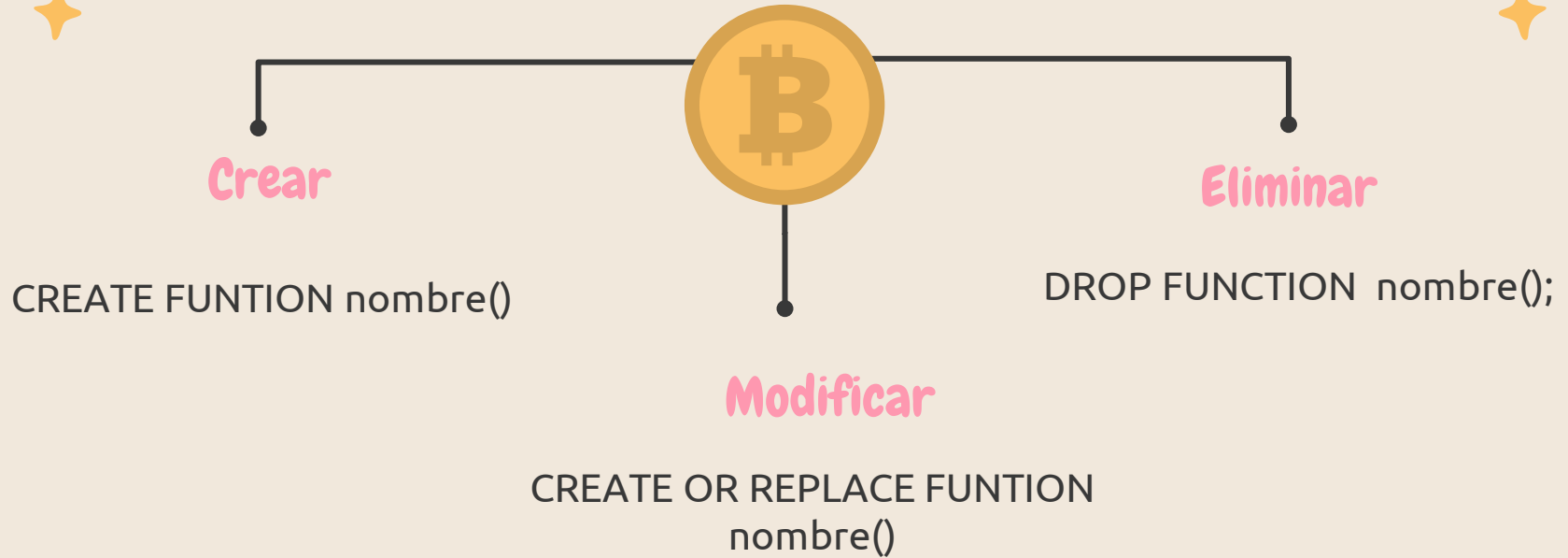
Defina que es una función en MySQL

Es un proceso usando o creado para tomar parámetro y transformarlos en salidas; el proceso es rutinario.

¿Que cosas características debe tener una función?

- La creación de la función.
- Asignación de un nombre y si es requerido, parámetro de entrada.
- Define del tipo de dato que se retorna.
- La selección de la función.

¿Como crear, modificar y como eliminar una funcion?



Manejo de Conceptos

Para que sirve la función **CONCAT** y como funciona en **MYSQL**

CONCAT es utilizado para unir todos de los datos de tipo cadena.

```
create or replace function get_prestados() returns integer
begin
    select COUNT(pres.fec_prestamo)
    from book as bo
    inner join prestamos as pres 1<->1..n on bo.id_book = pres.id_book
    where pres.fec_prestamo = year(2018-10-22);
return year();
end;
```



```
select count(pr.fec_prestamo)
from prestamos as pr
inner join book bo 1..n<->1: on pr.id_book = bo.id_book
where year(pr.fec_prestamo) = 2018;
```

Manejo de Conceptos

Para que sirve la función **SUBSTRING** y como funciona en **MYSQL**

SUBSTRING recupera los elementos que se le indican de una cadena principal (conteo).

```
CREATE OR REPLACE FUNCTION ejemplo(cadena TEXT)
RETURNS TEXT
BEGIN
    RETURN(SUBSTR(cadena, 1, 6));
end;

SELECT ejemplo(cadena: 'Ronald Choque P.');
```

Manejo de Conceptos

★ Para que sirve La funcion STRCMP y como funciona en MYSQL ★

STRCMP es una función de tipo booleano que presenta dos respuestas:

- 0 en TRUE
- 1 o -1 en FALSE.

```
CREATE OR REPLACE FUNCTION consulta(c1 TEXT, c2 TEXT, c3 TEXT)
RETURNS TEXT
BEGIN

    IF STRCMP(c1, c2) = 0 OR STRCMP(c1, c3) = 0
    then
        RETURN 'CADENAS IGUALES';
    else
        RETURN 'CADENAS DIFERENTES';
    end if;
end;

SELECT consulta( c1: 'dba iii', c2: 'dba i', c3: 'dba iii');
```

Manejo de Conceptos

✦ ✦ Para que sirve la función CHAR_LENGTH y LOCATE y como funciona en MYSQL ✦ ✦

- CHAR_LENGTH contabiliza los caracteres de una cadena, devuelve un INT.
- LOCATE realiza la búsqueda de un dato dentro de otro.

```
create or replace function pregunta(p1 text, p2 text)
returns text
begin
    declare position text default '';
    DECLARE con TEXT DEFAULT CONCAT(p1, ' ', p2);
    DECLARE car INT DEFAULT CHAR_LENGTH(con);
    set position = CONCAT('Posicion: ', locate(p1, p2), ' || N de dates: ', car);
    return position;
end;

select pregunta(p1 'II', p2 'Base de Datos II');
```

Manejo de Conceptos

Funciones de agregacion y funciones creados por el DBA?

- AGREGACION son aquellos que funcionan bajo la clausula SELECT, aplicado a un grupo de registros y que devuelven un único valor.
- CUSTOM esta definida por el comando CREATE FUNCTION, su sintaxis y uso esta completamente definida por el USER.

Manejo de Conceptos

Funciones de agregacion y funciones creados por el DBA?

- ENTRADA O INPUT refiere a la introducción de datos.
- SALIDA O OUTPUT refiere a la muestra de los mismos.

```
CREATE OR REPLACE FUNCTION ejemplo(cadena TEXT)
  RETURNS TEXT
BEGIN
  RETURN(SUBSTR(cadena, 1, 6));
end;

SELECT ejemplo(cadena: 'Ronald Choque P.');
```



```
$studio_list = array();  
$result = mysql::query("SELECT DISTINCT(studio) as studio, COUNT(*) as count FROM image WHERE day_id = '$day_id' AND enabled='y' GROUP BY studio");  
while($studio_list = mysql::fetch($shots_result)) {  
    $day_info = metadata::day_info($day->shot_date, $studio_list->studio, "quick");  
    $tmp_studio_list[] = array("studio" => $studio_list->studio, "count" => $studio_list->count, "title" => $day_info->title);  
}  
$day->studio_list = $tmp_studio_list;  
$return[$day->shot_date] = $day;  
  
return $return;
```

```
function day_images_list($date, $studio) {  
    global $global_studio_list;  
    if(!in_array($studio, $global_studio_list)) die("error studio");  
    $date = mysql::escape($date);  
    if(mysql::count("image_date", "shot_date = '$date'") < 1) die("date not found");  
    $studio = intval($studio);
```

```
$studio_list = array();  
  
$result = mysql::query("SELECT image.id as image_id, image.image, image_date WHERE image_date.id=image.day_id AND image_date.shot_date='$date' AND image.enabled='y' AND i");  
while($image = mysql::fetch($result)) {  
    $image->copyright = metadata::get_copyright($image->image_id);  
    $image->models = metadata::get_models($image->image_id);  
    $return[$image->image_id] = $image;  
}
```

1273

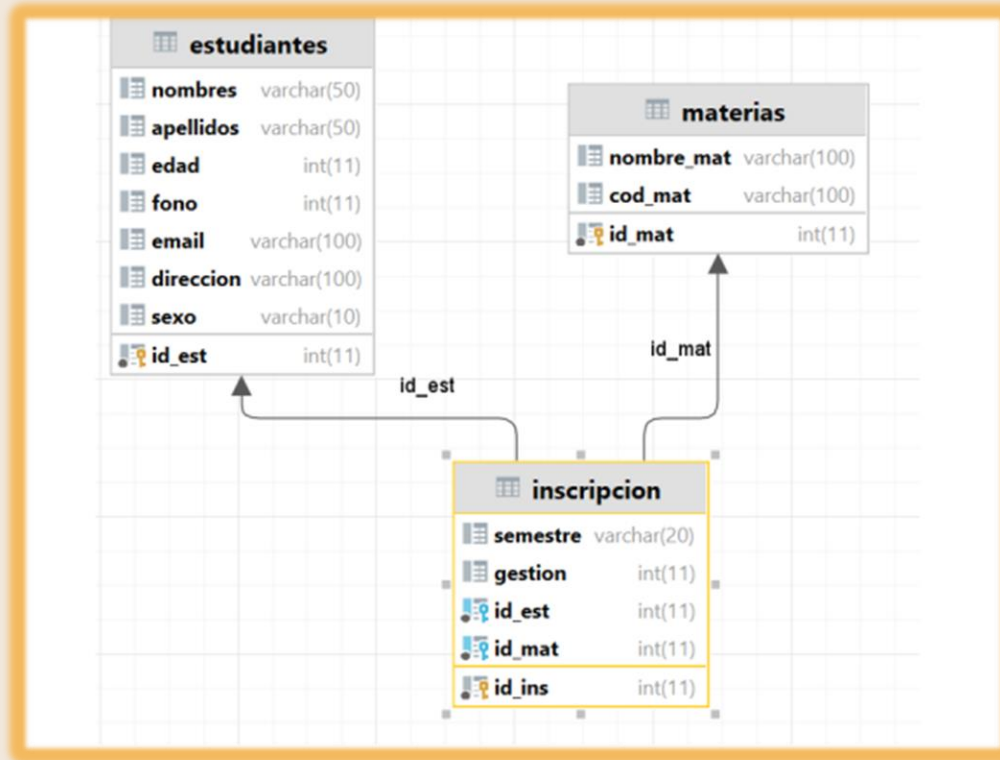
1274

1275

```
function day_list() {
```

11

Crear las siguiente Base de datos y sus registros



11

Crear las siguiente Base de datos y sus registros

```
create database PRACTICA_HIT03;  
USE PRACTICA_HIT03;
```

CREATE TABLE es un commando que genera la base de Datos
USE es un commando para trabajar en el mismo.

CREACION DE TABLAS

Para crear tablas se necesita los comandos

- Definimos a los PRIMARY KEY
- El commando FOREIGN KEY es utilizando para relacionar las tablas mediante los PRIMARY KEY.

CREACION DE LAS TABLAS

```
create table estudiante
(
    id_estudiante integer PRIMARY KEY not null,
    nombres varchar(50),
    apellidos varchar(50),
    edad integer,
    fono integer,
    email varchar(100),
    direccion varchar(100),
    genero varchar(10)
);
```

```
create table materias
(
    id_mate integer primary key not null ,
    nombre_materia varchar(100),
    codigo_materia varchar(100)
);
```

CREACION DE LAS TABLAS

```
create table inscripcion
(
    id_inscrpcion integer primary key not null,
    semestre varchar(20) null ,
    gestion integer null ,
    id_mate integer not null ,
    id_estudiante integer not null ,
    Foreign key (id_mate) references materias (id_mate),
    foreign key (id_estudiante) references estudiante (id_estudiante)
);
```

LLENADO DE LAS TABLAS

```
insert into estudiante (nombres,apellidos,edad,fono,email,direccion,genero,id_estudiante)
values ('Miguel','Gonzales Veliz',20,2_831_115,'miguel@gmail.com','Av. 6 de agosto','masculino',1),
('Sandra','Mavir Uria',25,2_832_116,'sandra@gmail.com','Av. 6 de agosto','femeino',2),
('joel','Adubiri Mondar',30,2_832_117,'joel@gmail.com','Av. 6 de agosto','masculino',3),
('Andrea','Arias Bellasteros',21,2_832_118,'andrea@gmail.com','Av. 6 de agosto','femenino',4),
('Santos','Montes Valenzuela',24,2_832_119,'santos@gmail.com','Av. 6 de agosto','masculino',5);
```

```
insert into materias(id_mate, nombre_materia, codigo_materia) values
(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-104');
```

LLENADO DE LAS TABLAS

```
insert into inscripcion (id_inscripcion, semestre, gestion, id_mate, id_estudiante) VALUES  
(1, '1er semestre', 2018, 1, 1),  
(2, '2do semestre', 2018, 2, 1),  
(3, '1er semestre', 2019, 4, 2),  
(4, '2do semestre', 2019, 3, 2),  
(5, '2do semestre', 2020, 3, 3),  
(6, '3er semestre', 2020, 1, 3),  
(7, '4to semestre', 2021, 4, 4),  
(8, '5to semestre', 2021, 5, 5);
```


12

Crear una funcion que genere la serie Fibonacci

```
CREATE OR REPLACE FUNCTION seriefibonacci(limite INT)
RETURNS TEXT
BEGIN
    DECLARE resp TEXT DEFAULT '';
    DECLARE i INT DEFAULT 0;
    DECLARE siguiente INT DEFAULT 1;
    DECLARE uno INT DEFAULT 0;
    DECLARE dos INT DEFAULT 1;

    WHILE i <= limite DO
        SET i = i + 1;
        IF i <= 1
        THEN
            SET siguiente = i;
        ELSE
            SET siguiente = uno + dos;
            SET uno = dos;
            SET dos = siguiente;
        end if;
        SET resp = CONCAT(resp, siguiente, ' , ');
    end while;
    RETURN resp;
end;

SELECT seriefibonacci(limite 7);
```

Función Ejecución

```
`seriefibonacci(7)`  
1 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 ,
```

13

Crear una variable global a nivel BASE DE DATOS

```
SET @userALL = 'SR.SKYNET';  
  
CREATE OR REPLACE FUNCTION variableglobal()  
RETURNS TEXT  
BEGIN  
    RETURN @userALL;  
end;  
  
SELECT variableglobal() AS ADMIN;
```

FUNCION EJECUTADA

	ADMIN
1	SR.SKYNET

14

Crear una funcion no recibe parametro (Utilizar WHILE, REPEAT o LOOP)

```
SET @LIMT=8;
```

```
CREATE OR REPLACE FUNCTION parametros()
```

```
RETURNS text
```

```
BEGIN
```

```
DECLARE A INT DEFAULT 0;
```

```
DECLARE B INT DEFAULT 1;
```

```
declare i int default 0;
```

```
declare siguiente int default 0;
```

```
declare respuesta text default '';
```

```
while i <= @LIMT DO
```

```
SET i=i+1;
```

```
if i < 1
```

```
then
```

```
set siguiente = i;
```

```
else
```

```
set siguiente=A+B;
```

```
set A=B;
```

```
SET B=siguiente;
```

```
end if;
```

```
set respuesta= concat(respuesta,siguiente, ' , ' ,
```

```
');
```

```
end while ;
```

```
return respuesta;
```

```
END;
```

```
SELECT parametros() AS ADMIN;
```

```
CREATE OR REPLACE FUNCTION SERIE_EDAD(X INT)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
DECLARE A TEXT DEFAULT '';
```

```
REPEAT
```

```
IF X % 2=0
```

```
THEN
```

```
SET A=CONCAT(' , ' ,X,A);
```

```
SET X=X-2;
```

```
ELSE IF X%2=1
```

```
THEN
```

```
SET A= CONCAT(A,X, ' , ' );
```

```
SET X=X-2;
```

```
end if;
```

```
end if;
```

```
until X <=0
```

```
end repeat;
```

```
RETURN A;
```

```
end;
```

```
SELECT SERIE_EDAD(MIN_EDAD());
```

```
SELECT SERIE_EDAD(23);
```

14

Crear una funcion no recibe parametro (Utilizar WHILE, REPEAT o LOOP)

```
CREATE OR REPLACE FUNCTION MIN_EDAD()  
RETURNS INT  
BEGIN  
    RETURN  
    (  
        SELECT MIN(est.edad)  
        FROM estudiantes AS est  
    );  
end;  
  
SELECT MIN_EDAD();
```

FUNCION EJECUTADA

```
SELECT SERIE_EDAD(10, MIN_EDAD());
```

Output: SET: CWD\MIN (CWD\edad) *

'SERIE_EDAD(10,MIN_EDAD())'

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

15

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

Consigna de la practica

```
SELECT separaVocales( cadena: 'taller de base de datos' ) AS SEPARA_VOCALES;
```

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

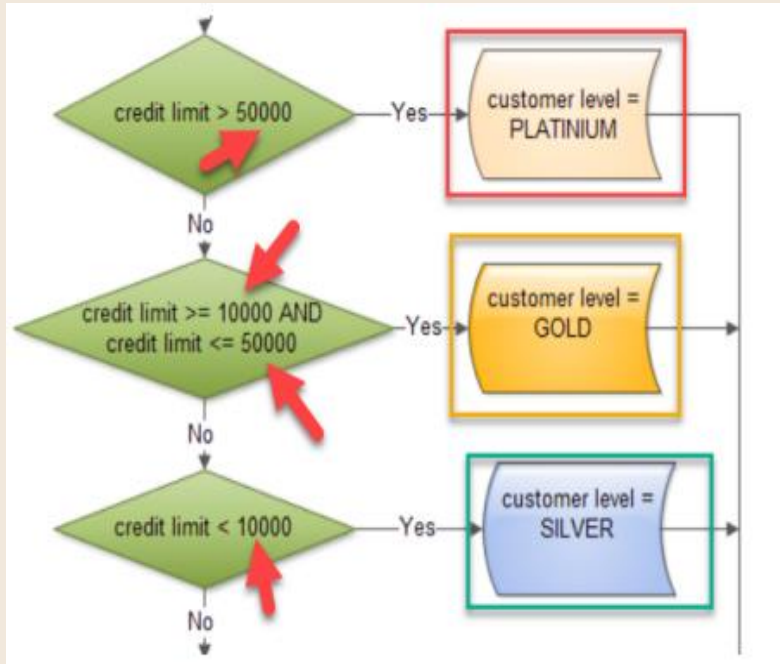
FUNCION EJECUTADA

SEPARA_VOCALES
1 A: 3 ,E: 4 ,I: 0 ,O: 1 ,U: 0

16

Crear una función que recibe un parámetro INTEGER

Consigna de Practica



```
CREATE OR REPLACE FUNCTION tipodeusuario(parametro INT)
RETURNS TEXT
BEGIN
    DECLARE credit_number INT DEFAULT 0;
    DECLARE respuesta TEXT DEFAULT '';
    SET credit_number = parametro;
    CASE
        WHEN credit_number > 50000 THEN SET respuesta = 'PLATINIUM';
        WHEN credit_number >= 10000 AND credit_number <= 50000 THEN SET respuesta = 'GOLD';
        WHEN credit_number < 10000 THEN SET respuesta = 'SILVER';
        else SET respuesta = 'usuario no IDENTIFICADO';
    END CASE;
    RETURN respuesta;
end;

SELECT tipodeusuario( parametro: 5000) AS VERIFICAR_TIPO_USUARIO;
```

FUNCION EJECUTADA

```
VERIFICAR_TIPO_USUARIO
1 SILVER
```

17

Crear una funcion que recibe un parametro ✨

```
create OR REPLACE FUNCTION descomponerc(x TEXT)
returns TEXT
begin
  DECLARE str TEXT DEFAULT '';
  DECLARE legth_concat INT DEFAULT CHAR_LENGTH(x);
  DECLARE limite INT DEFAULT 1;
  DECLARE sub INT DEFAULT legth_concat;

  REPEAT
    IF legth_concat >= limite
    THEN
      SET str = CONCAT(str, substr(x, legth_concat, sub - 1), ', ');
      SET legth_concat = legth_concat - 1;
    end if;
  UNTIL legth_concat <= 0
  end repeat;
  RETURN str;
END;

select descomponerc(x: 'DBAII');
```

LETTERS	
1	dbaii, baii, aii, ii, i,

FUNCION EJECUTADA

```
'descomponerc('DBAII')'
1 I , II , AII , BAI , DBAI ,
```



Gracias por su atencion

eate.iliaaraceli.sarzo.la@unifranz.edu.bo