

Ingenieria de Sistemas

Hito 3

Victor Hugo Quispe Torrez



# Base de Datos II

# Parte Conceptual

1. Defina que es lenguaje procedural en MySQL.

2. Defina que es una función en MySQL.

Los lenguajes procedurales están fundamentados en la utilización de variables para almacenar valores y en la realización de operaciones con los datos almacenados

Una función en MySQL es una rutina creada para tomar unos parámetros, procesarlos y retornar en un salida. Deben retornar en un valor con algún tipo de dato definido

# Parte Conceptual

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

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

```
CREATE OR REPLACE function edad_minima()
  RETURNS INT
  BEGIN
    DECLARE resp int;
    Select Min(est.edad)
    into resp
    from estudiante as est;
    return resp;
  END;
```

```
CREATE OR REPLACE FUNCTION usuariosGestion2017_2018()
returns int
begin
  declare cont int;
  SELECT count(bk.id_book) into cont
  FROM book as bk
  inner join prestamos p on bk.id_book = p.id_book
  WHERE p.fec_prestamo like '%2017%' or p.fec_prestamo like '%2018%';
  return cont;
end;
```

```
DROP function usuariosGestion2017_2018;
```

# Parte Conceptual

## 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.

## 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.
- La función solo retorna el nombre

Sirve para unir dos o mas cadenas utilizando el commando select para llamar a esta funcion

```
SELECT CONCAT('Hola ', 'Soy estudiante de ', 'DBA II') ;
```

```
SUBSTRING(str, pos) SUBSTRING(str FROM pos) SUBSTRING(str, pos, len)  
SUBSTRING(str FROM pos FOR len)
```

```
CREATE OR REPLACE FUNCTION nombres(cad1 text)  
RETURNS TEXT  
BEGIN  
    DECLARE posi int default Locate(' ', cad1);  
    return SUBSTRING(cad1, 1, posi);  
End;  
  
SELECT nombres( cad1: 'Victor Quispe Torrez');
```

# Parte Conceptual

## 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.

```
CREATE OR REPLACE FUNCTION compararCad(cad1 text, cad2 text, cad3 text)
RETURNS text
BEGIN
    DECLARE Resp text default '';
    If STRCMP(cad1,cad2)=0 or STRCMP(cad2,cad3)=0 or strcmp(cad1,cad3)=0 THEN
        SET Resp='Dos de ellas son iguales';
    ELSE
        SET Resp='Ninguno es igual';
    end if;
    return Resp;
end;

Select compararCad( cad1: 'HERO', cad2: 'AVPO', cad3: 'POA');
```

## 8. Para qué sirve la función CHAR\_LENGTH y LOCATE y como funciona en MYSQL

- ¿Crear una función que muestre el uso de ambas funciones?

```
CREATE OR REPLACE FUNCTION nom(cad1 text,pos int)
RETURNS TEXT
BEGIN
    DECLARE posi int default char_length(cad1);
    DECLARE RESP TEXT default '';
    DECLARE num int default LOCATE(' ',cad1);
    if posi<=pos then
        SET RESP=Concat(SUBSTRING(cad1,1,num),'Cadena aceptada ');
    ELSE
        SET RESP='Cadena no aceptada';
    end if;

    return RESP;
End;

SELECT nom( cad1: 'ada mio', pos: 7);
```

# Parte Conceptual

9. ¿Cual es la diferencia entre las funciones de agregacion y funciones creados por el DBA? Es decir funciones creadas por el usuario.

Las funciones de agregacion son aquellas que vienen ya definidas en el editor MySql, mientras que las funciones creadas por el usuario son aquellas que actuan segun el comportamiento que les da el usuario y retornaran el tipo de valor que se les asigne.

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

Entrada: Se indican poniendo la palabra reservada IN delante del nombre del parámetro. Estos no pueden cambiar su valor dentro del procedimiento, es decir, cuando el procedimiento finalice estos parámetros tendrán el mismo valor que tenían cuando se hizo la llamada al procedimiento.

Salida: Se indican poniendo la palabra reservada OUT delante del nombre del parametro. Estos parámetros cambian su valor dentro del procedimiento. Cuando se hace la llamada al procedimiento empiezan con un valor inicial y cuando finaliza la ejecución del procedimiento pueden terminar con otro valor diferente.

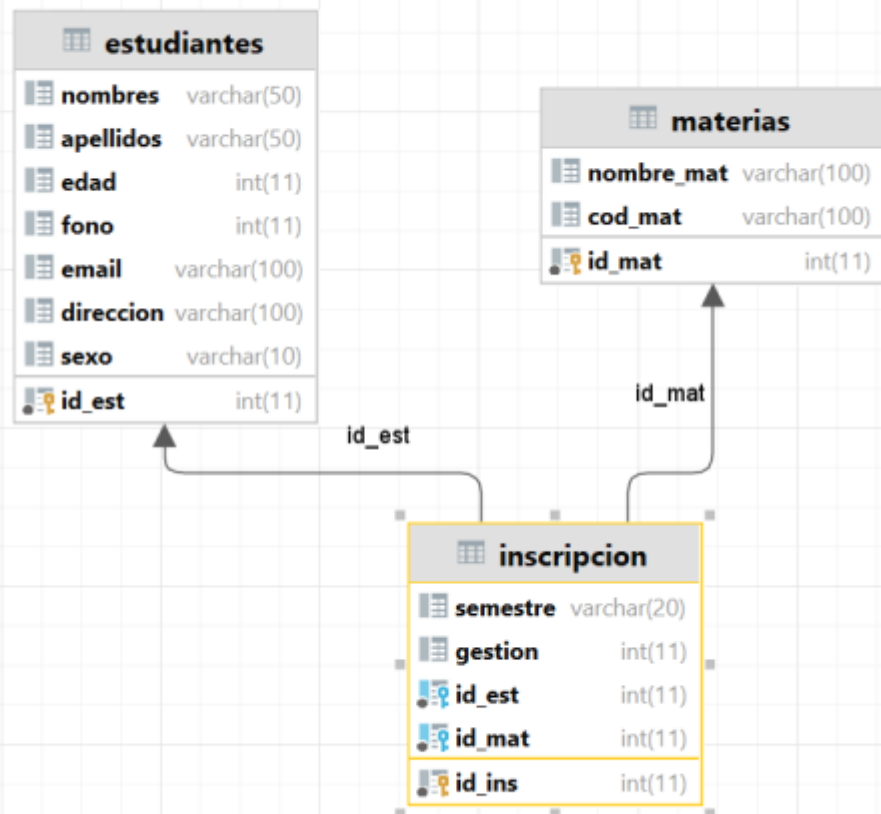


# Parte Practica





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



### DATOS TABLA ESTUDIANTES

id_est	nombres	apellidos	edad	fono	email	direccion	sexo
1	Miguel	Gonzales Veliz	20	2832115	miguel@gmail.com	Av. 6 de Agosto	masculino
2	Sandra	Mavir Uria	25	2832116	sandra@gmail.com	Av. 6 de Agosto	femenino
3	Joel	Adubiri Mondar	30	2832117	joel@gmail.com	Av. 6 de Agosto	masculino
4	Andrea	Arias Ballesteros	21	2832118	andrea@gmail.com	Av. 6 de Agosto	femenino
5	Santos	Montes Valenzuela	24	2832119	santos@gmail.com	Av. 6 de Agosto	masculino

### DATOS TABLA MATERIAS

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

### DATOS TABLA INSCRIPCION

id_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.

- 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,**

```
CREATE OR REPLACE FUNCTION fibonacci(numeroF int)
    RETURNS TEXT
    BEGIN
        DECLARE num1 int default 0;
        DECLARE num2 int default 1;
        DECLARE num3 int default 0;
        DECLARE cont int default 0;
        DECLARE resp text default '';
        SET resp=CONCAT(num1,',',num2);
        SET cont=2;
        While cont<numeroF do

            SET num3=num1+num2;
            Set num1=num2;
            Set num2=num3;
            SET resp=Concat(resp,',',num3);
            Set cont=cont+1;
        end while;

        return resp;
    end;
SELECT fibonacci( numeroF: 4);
```

```
`fibonacci(4)`
1 0,1,1,2
```

### 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.**
- 

- Crear una variable global de nombre LIMIT.
- Este valor debe almacenar un valor entero.
  - Ejemplo, **LIMIT = 7**
  - OUTPUT: **0,1,1,2,3,5,8**
- Crear una función que genere la serie **fibonacci** hasta ese valor LIMIT.
  - Note que el valor LIMIT debe ser usado en la función
  - La función no recibe ningún parámetro.

```
SET @LIMIT=7;
```

```
CREATE OR REPLACE FUNCTION fibonacciV2()  
  RETURNS TEXT  
  BEGIN  
    DECLARE num1 int default 0;  
    DECLARE num2 int default 1;  
    DECLARE num3 int default 0;  
    DECLARE cont int default 0;  
    DECLARE resp text default '';  
    SET resp=CONCAT(num1,',',num2);  
    SET cont=2;  
    While cont<@LIMIT do  
  
      SET num3=num1+num2;  
      Set num1=num2;  
      Set num2=num3;  
      SET resp=Concat(resp,',',num3);  
      Set cont=cont+1;  
    end while;  
  
    return resp;  
  end;  
SELECT fibonacciV2();
```

```
mysql> `fibonacciV2()`
```

```
1 0,1,1,2,3,5,8
```

#### 14. Crear una función no recibe parámetros (Utilizar WHILE, REPEAT o 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 MÍNIMA).
- 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,
```

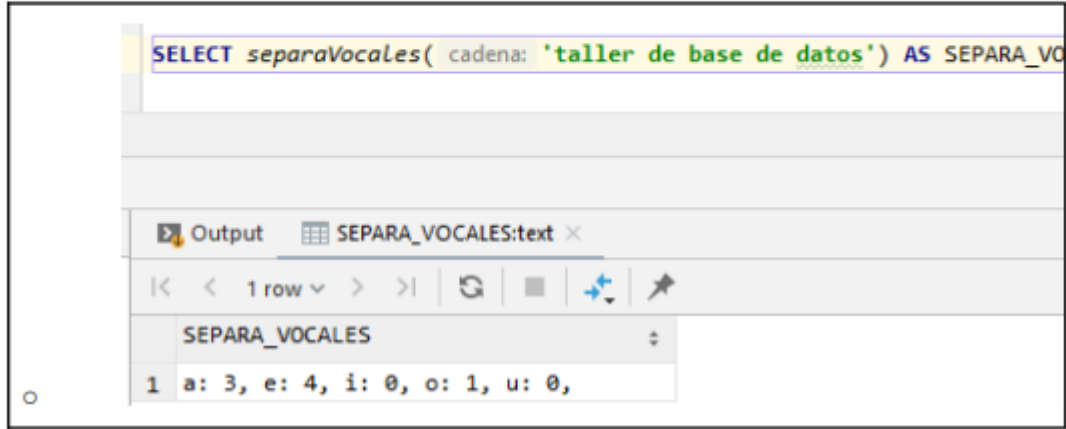
```
CREATE OR REPLACE function edad_minima()  
  RETURNS INT  
  BEGIN  
    DECLARE resp int;  
    Select Min(est.edad)  
    into resp  
    from estudiante as est;  
    return resp;  
  END;
```

```
CREATE OR REPLACE FUNCTION mostrarParImpar()  
  RETURNS TEXT  
  BEGIN  
    DECLARE contP int default 0;  
    DECLARE contI int default edad_minima();  
    DECLARE resp text default '';  
    DECLARE cont int default 0;  
    WHILE cont<=edad_minima() do  
      if edad_minima()%2=0 THEN  
        SET resp=concat(resp,contP,',');  
        SET contP=contP+2;  
      ELSE  
        SET resp=concat(resp,contI,',');  
        SET contI=contI-2;  
      end if;  
      Set cont=cont+2;  
    end WHILE ;  
    return resp;  
  end;  
SELECT mostrarParImpar();
```

```
`mostrarParImpar()`  
1 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,
```

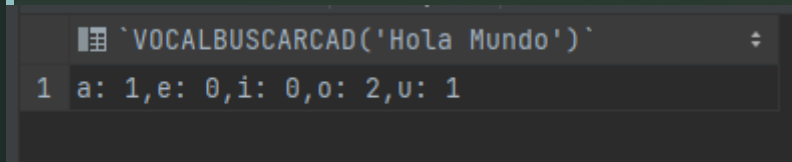
## 15. Crear una función que determina cuantas veces se repite las vocales.

- 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.



The screenshot shows a SQL query in a text editor: `SELECT separaVocales( cadena: 'taller de base de datos') AS SEPARA_VO`. Below the query, the output is displayed in a table with the title "SEPARA\_VOCALES". The table has one row with the following data: "1 a: 3, e: 4, i: 0, o: 1, u: 0,".

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



The screenshot shows a SQL query in a text editor: ``VOCALBUSCARCAD('Hola Mundo')``. Below the query, the output is displayed in a table with the following data: "1 a: 1, e: 0, i: 0, o: 2, u: 1".

1	a: 1, e: 0, i: 0, o: 2, u: 1
---	------------------------------

```
CREATE OR REPLACE FUNCTION VOCALBUSCARCAD (CADENA TEXT)
RETURNS TEXT
BEGIN
    DECLARE RESP TEXT DEFAULT '';
    DECLARE CONT INT DEFAULT 0;
    DECLARE CONTA INT DEFAULT 0;
    DECLARE CONTE INT DEFAULT 0;
    DECLARE CONTI INT DEFAULT 0;
    DECLARE CONTO INT DEFAULT 0;
    DECLARE CONTU INT DEFAULT 0;
    DECLARE PUNTERO CHAR;

    WHILE CONT <= CHAR_LENGTH(CADENA) DO
        SET PUNTERO = SUBSTRING(CADENA,CONT,1);
        IF PUNTERO = 'a' THEN
            SET CONTA = CONTA + 1;
        end if;
        IF PUNTERO='e' THEN
            SET CONTE=CONTE+1;
        end if;
        IF PUNTERO='i' THEN
            SET CONTI=CONTI+1;
        end if;
        IF PUNTERO='o' THEN
            SET CONTO=CONTO+1;
        end if;
        IF PUNTERO='u' THEN
            SET CONTU=CONTU+1;
        end if;
        SET CONT =CONT+1;
    END WHILE;

    SET RESP=CONCAT('a: ',CONTA,',e: ',CONTE,',i: ',CONTI,',o: ',CONTO,',u: ',CONTU);

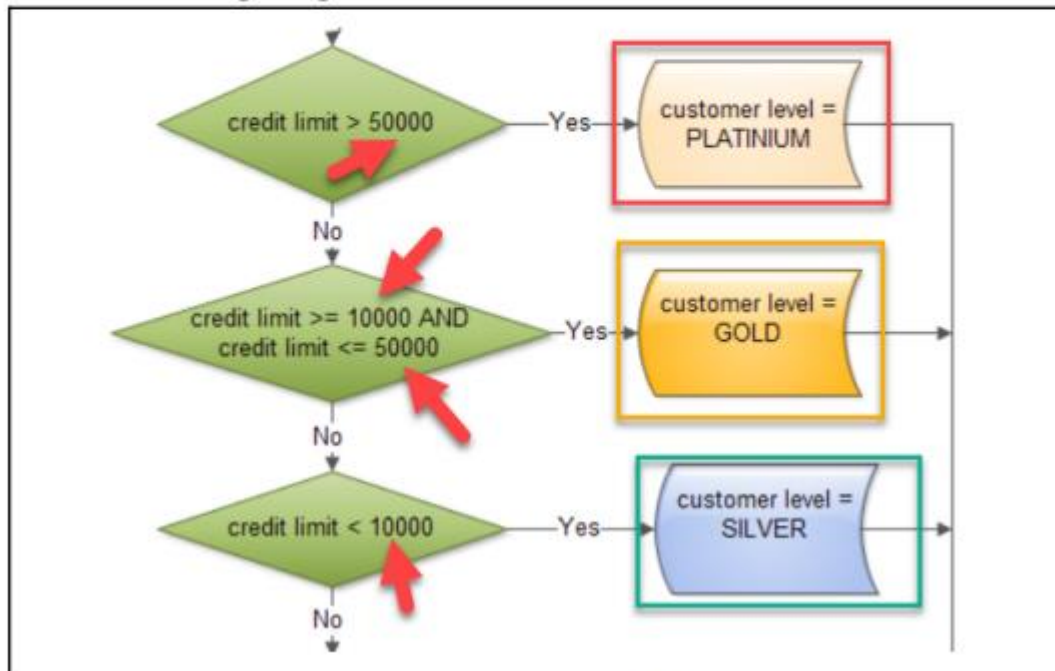
    RETURN RESP;
end;

SELECT VOCALBUSCARCAD( CADENA: 'Hola Mundo');
```

## 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 PLATINUM, GOLD o SILVER en base al valor del credit\_number.
- Considere la imagen siguiente:



- Para resolver debe de utilizar la instrucción **CASE - WHEN**.

```
CREATE OR REPLACE FUNCTION TipoCredito(creditNumber int)
RETURNS TEXT
BEGIN
    DECLARE TipoCliente TEXT DEFAULT '';
    CASE
        WHEN creditNumber > 50000 then SET TipoCliente = 'PLATINIUM';
        WHEN creditNumber between 10000 and 50000 then SET TipoCliente = 'GOLD';
        WHEN creditNumber < 10000 then SET TipoCliente = 'SILVER';
    END CASE;

    RETURN TipoCliente;
end;

SELECT TipoCredito(creditNumber: 15000);
```

#Ejercicio 17

```
CREATE OR REPLACE FUNCTION VOCALBUSCAR (CADENA VARCHAR(20), CADENA1 VARCHAR(20))
```

Output TipoCredito(15000):text mostrarParImpar():text

1 row

`TipoCredito(15000)`

1 GOLD



17. Crear una función que recibe 2 parámetros **VARCHAR(20)**, **VARCHAR(20)**.

- La función debe de retornar un texto **TEXT** como respuesta.
- Si las cadenas fueran **"TALLER DBA II"** y la segunda cadena fuese **"GESTION 2023"**.
- La nueva cadena debería ser **"TLLR DB -GSTN 2023"**.
- La nueva cadena es resultado de la concatenación de todos los valores distintos a las vocales.
- Retornar la nueva cadena concatenada.

```
CREATE OR REPLACE FUNCTION VOCALBUSCAR (CADENA VARCHAR(20),CADENA1 VARCHAR(20))
RETURNS TEXT
BEGIN
    DECLARE RESP TEXT DEFAULT '';
    WHILE char_length(CADENA)>0 Do
        If substring(CADENA,1,1) NOT IN ('a','e','i','o','u') THEN
            SET RESP=Concat(RESP,SUBSTRING(CADENA,1,1));
        end if;
        SET CADENA=SUBSTRING(CADENA,2);
    end while;
    WHILE char_length(CADENA1)>0 DO
        If substring(CADENA1,1,1) NOT IN ('a','e','i','o','u') THEN
            SET RESP=Concat(RESP,SUBSTRING(CADENA1,1,1));
        end if;
        SET CADENA1=SUBSTRING(CADENA1,2);
    end while;
    return RESP;
end;
SELECT VOCALBUSCAR( CADENA: 'Taller dba II', CADENA1: 'Gestion 2023');
```

```
`VOCALBUSCAR('Taller dba II','Gestion 2023')`
1 Tllr db Gstn 2023
```



- 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:

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

```
CREATE OR REPLACE FUNCTION letras(CADENA TEXT)
RETURNS TEXT
)BEGIN
    DECLARE RESP TEXT DEFAULT '';
    DECLARE punt TEXT DEFAULT '';
    DECLARE cont INT DEFAULT 0;
) REPEAT
    SET punt = SUBSTRING(CADENA,cont);
    SET RESP = CONCAT(RESP,punt,',');
    SET cont = cont +1;
until cont >CHAR_LENGTH(CADENA)
) end repeat;
return RESP;
)END;

SELECT letras( CADENA: 'VICTOR');
```

```
`letras('VICTOR')`
1 ,VICTOR,ICTOR,CTOR,TOR,OR,R,
```

**Gracias por su atención**