

DEFENSA HITO 3

BASE DE DATOS II

EVER TICONA HUALLPA

SIS10031434

TRANSFORMA
TU FORMA
DE APRENDER



UNIFRANZ
UNIVERSIDAD PRIVADA
FRANZ TALEDO
Internacionalízate

INNOVACIÓN
EN EDUCACIÓN



Tabla De Contenido:

PARTE TEORICA

En donde se encuentra un conglomerado de preguntas relacionadas a BASES DE DATOS RELACIONALES.

PARTE PRACTICA

En donde deberá realizar y crear funciones manejando el motor de base de datos MySQL manejando el concepto del lenguaje procedural.





PARTE TEORICA

Defina que es lenguaje procedural en MySQL


Conjunto de instrucciones SQL, más una serie de estructuras de control que pueden almacenarse en el servidor.

Defina que es una función en MySQL

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

¿Qué cosas características debe de tener una función?

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



```
CREATE OR REPLACE FUNCTION ejemplo(cadena TEXT)
RETURNS TEXT
BEGIN
end;

SELECT ejemplo( cadena: 'dba III');
```

The diagram features a dark blue background with a glowing network of white nodes and lines. Three white arrows point from the network towards the SQL code block on the right. The code block is dark grey with orange text. The first arrow points to the function name 'ejemplo' in the first line. The second arrow points to the parameter 'cadena' in the first line. The third arrow points to the function name 'ejemplo' in the last line.

¿Cómo crear,
modificar y cómo
eliminar una función?

CREA:

```
CREATE FUNCTION cualquier_nombre_para_llamar_a_una_funcion
```

MODIFICA:

```
CREATE OR REPLACE FUNCTION nombre()
```

ELIMINA:

```
DROP FUNCTION comparacion;
```

Para qué sirve la
función CONCAT y
como funciona en
MYSQL

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

```
CREATE OR REPLACE FUNCTION EJEMPLO_CONCAT(cadena text, cadena2 text, cadena3 text)
returns text
begin
    return (CONCAT(cadena, cadena2, cadena3));
end;

select EJEMPLO_CONCAT( cadena: 'EVER', cadena2: 'TICONA', cadena3: 'HUALLPA') AS CONCAT ;
```

```
■ CONCAT
1 EVERTICONAHUALLPA
```

Para qué sirve la función
SUBSTRING y como
funciona en MYSQL

Para qué sirve la función
SUBSTRING y como
funciona en MYSQL

```
CREATE OR REPLACE FUNCTION EJEMPLO_SUBSTRING(cadena TEXT)
RETURNS TEXT
BEGIN
    RETURN (SUBSTRING(cadena, 1, 4));
end;

SELECT EJEMPLO_SUBSTRING( cadena: 'EVER TICONA HUALLPA');
```

```
■ `EJEMPLO_SUBSTRING('EVER TICONA HUALLPA')`
1 EVER
```

Para qué sirve la función 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(cadena1 text,cadena2 text,cadena3 text)
returns text
begin
    if STRCMP(cadena1,cadena2) = 0 OR STRCMP(cadena1,cadena3) = 0
    then
        return 'cadenas iguales';
    else
        return 'cadenas diferentes';
    end if;
end;

select consulta( cadena1: 'DBA III', cadena2: 'DBA I', cadena3: 'DBA III');
```

```
`consulta('DBA III','DBA I','DBA III')`
```

```
1 cadenas iguales
```

Para qué 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 datos',car);
    return position;
end;

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

```
`pregunta('II','Base de datos II')`
```

```
1 Posicion:15 N de datos: 19
```


¿Cual es la diferencia entre las funciones de agresión y funciones creados por el DBA?

- AGREGACION son aquellas 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.

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

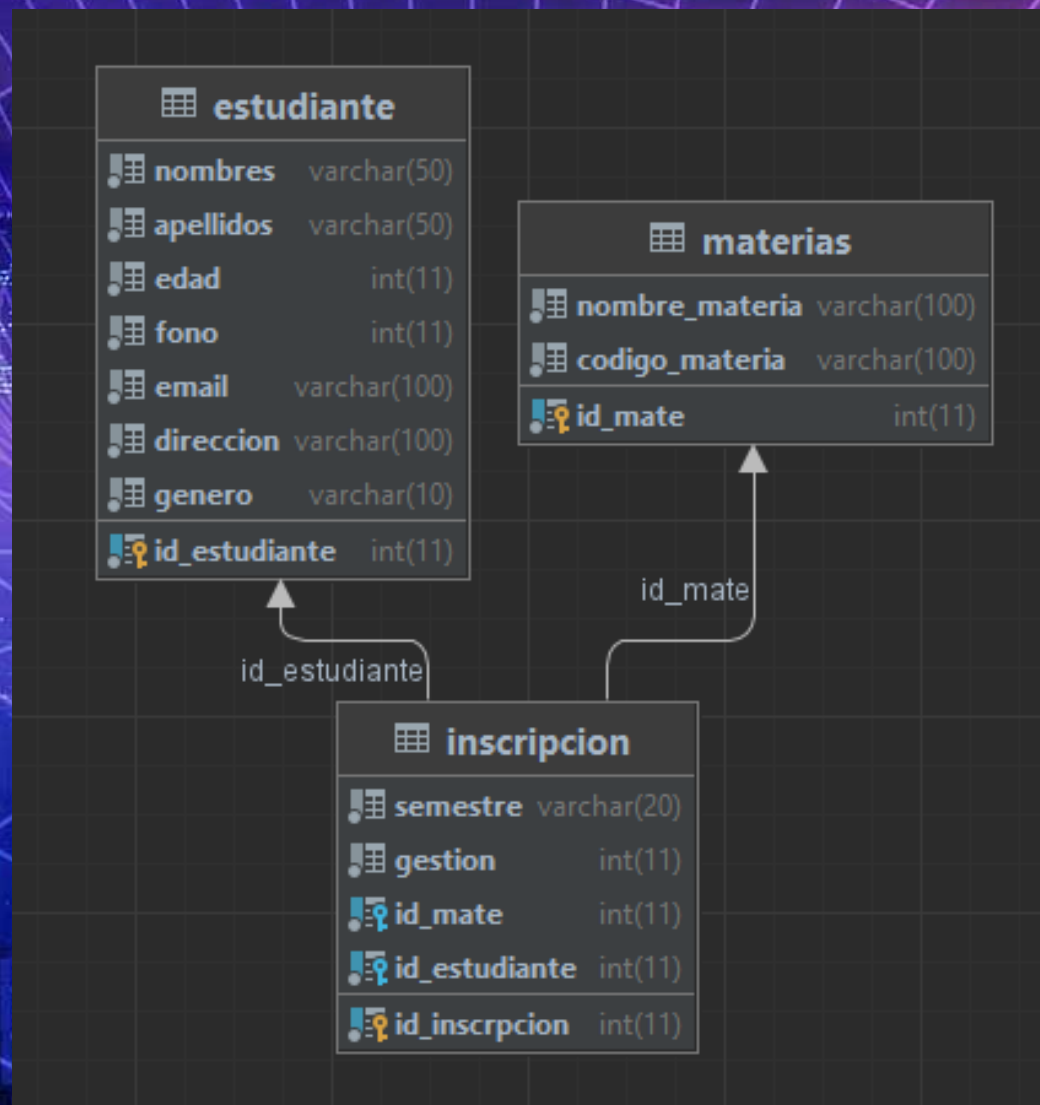
ENTRADA O INPUT refiere a la introducción de datos.

SALIDA O OUTPUT refiere a la muestra de los mismo.



PARTE PRACTICA

Crear la siguiente Base de datos y sus registros



CREACION DE LA DBA

```
CREATE DATABASE HITO_3;  
USE HITO_3;
```

El comando CREATE TABLE es el que genera la DBA.

Comando USE para trabajar en el mismo.

PARA LA CREACION DE TABLAS:

- Definimos a los PRIMARY KEY.
- El comando FOREIGN KEY es utilizado para relacionar las tablas mediante los PRIMARY KEY.
- NOT NULL para no tener columnas sin registro.

AUTO_INCREMENT para generar automáticamente el registro de la columna.

CREACION DE LAS TABLAS

```
CREATE TABLE estudiante(  
  id_estudiante INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  nombres VARCHAR(50) NOT NULL,  
  apellidos VARCHAR(50) NOT NULL,  
  edad INTEGER NOT NULL,  
  fono INTEGER NOT NULL,  
  email VARCHAR(100) NOT NULL,  
  direccion VARCHAR(100) NOT NULL,  
  genero VARCHAR(10) NOT NULL  
);
```

```
CREATE TABLE materias(  
  id_mate INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  nombre_materia VARCHAR(100) NOT NULL,  
  codigo_materia VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE inscripcion(  
  id_inscripcion INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  semestre VARCHAR(20) NOT NULL,  
  gestion INTEGER NOT 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)  
);
```

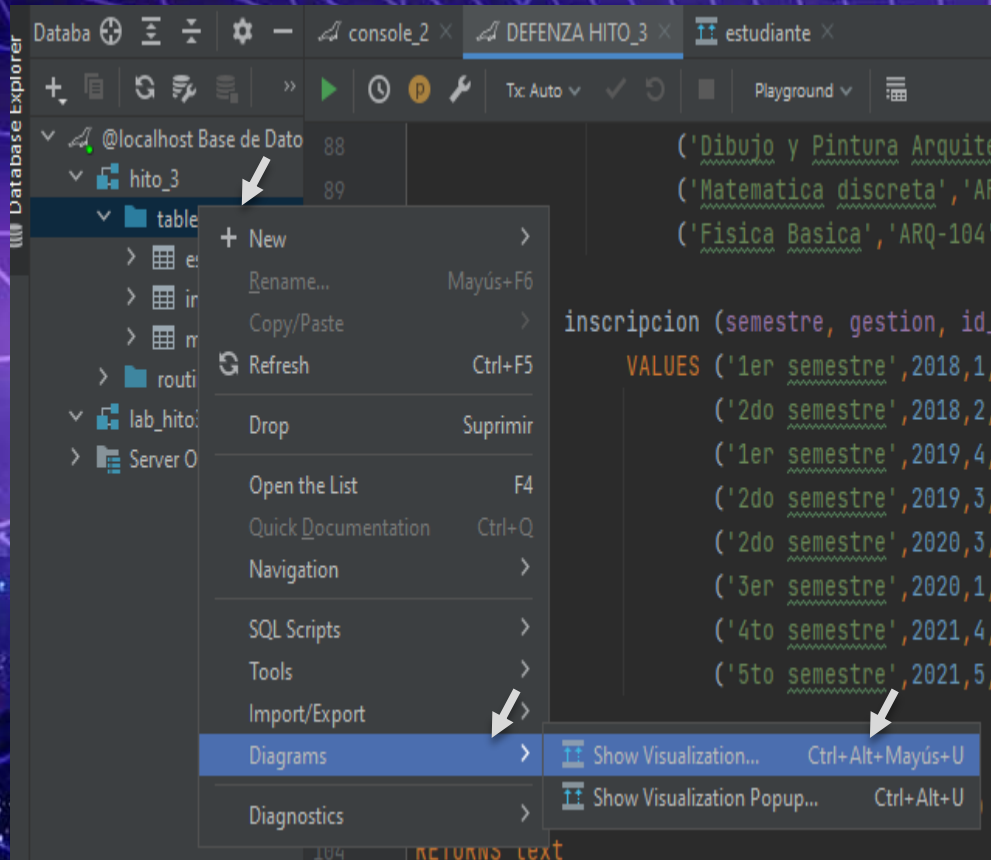

LLENADO DE LAS TABLAS

```
INSERT INTO estudiante (nombres,apellidos,edad,fono,email,direccion,genero,id_estudiante)
VALUES ('Miguel',' Gonzales Veliz',20,2831115,'miguel@gmail.com','Av. 6 de agosto','masculino',1),
('Sandra','Mavir Uria',25,2832116,'sandra@gmail.com','Av. 6 de agosto','femeino',2),
('joel','Adubiri Mondar',30,2832117,'joel@gmail.com','Av. 6 de agosto','masculino',3),
('Andrea','Arias Bellasteros',21,2832118,'andrea@gmail.com','Av. 6 de agosto','femenino',4),
('Santos','Montes Valenzuela',24,2832119,'santos@gmail.com','Av. 6 de agosto','masculino',5);
```

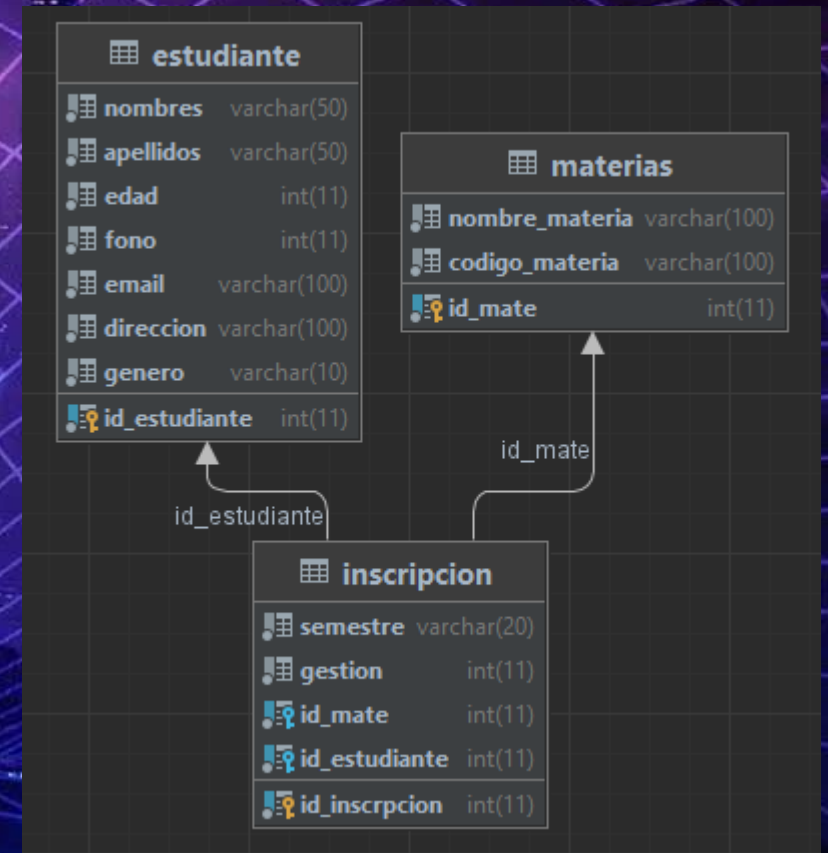
```
INSERT INTO materias (nombre_materia, codigo_materia)
VALUES ('Introduccion a la arquitectura','ARQ-101'),
('Urbanismo y Diseño','ARQ-102'),
('Dibujo y Pintura Arquitectonico','ARQ-103'),
('Matematica discreta','ARQ-104'),
('Fisica Basica','ARQ-104');
```

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


LLENADO DE LAS TABLAS



Haciendo un clic derecho sobre la carpeta de tablas, vaya y seleccione **DIAGRAMS** y la opción **SHOW VISUALIZATION** para poder ver el diagrama.



Crear una función que genere la serie Fibonacci

```
CREATE OR REPLACE FUNCTION FIBONACCI(LIMIT INT)
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 <= LIMIT 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 FIBONACCI( LIMIT: 7);
```

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


Crear una variable global a nivel BASE DE DATOS

```
SET @USER='Ever';  
  
CREATE FUNCTION VARIABLE_GLOBAL()  
RETURNS TEXT  
BEGIN  
    RETURN @USER;  
end;  
  
SELECT VARIABLE_GLOBAL() AS USER;
```

USER	
1	Ever

Crear una función no recibe parámetros (Utilizar WHILE, REPEAT o LOOP)

```
CREATE OR REPLACE FUNCTION MIN_EDAD()  
RETURNS INT  
BEGIN  
    RETURN  
    (  
        SELECT MIN(edad)  
        FROM estudiante AS ES  
    );  
end;  
SELECT MIN_EDAD();
```

```
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( X: MIN_EDAD()) AS PAR ;
```

```
SELECT SERIE_EDAD( X: 23) AS INPAR;
```

PAR

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

INPAR

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

Crear una función que determina cuantas veces se repite las vocales

```
CREATE OR REPLACE FUNCTION separandovocales(par1 text)
RETURNS TEXT
BEGIN
    DECLARE x INT DEFAULT 1;
    DECLARE response TEXT DEFAULT '';
    DECLARE letra CHAR DEFAULT '';
    DECLARE limite INT DEFAULT char_length(par1);
    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 <= limite DO
        SET letra = substring(par1, 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;
        END IF;
        SET x = x + 1;
    END WHILE;
    SET response = concat('A: ', a, ', E: ', e, ', I: ', i, ', O: ', o, ', U: ', u);
    RETURN response;
END;
```

```
SELECT separandovocales( par1: 'taller de base de datos') as SEPARA_VOCALES;
```

SEPARA_VOCALES

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

Crear una función que recibe un parámetro INTEGER

```
create or replace function TIPO_DE_USUARIO(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 TIPO_DE_USUARIO( parametro: 5000) AS VERIFICAR_TIPO_DE_USUARIO;
```

VERIFICAR_TIPO_DE_USUARIO

1 SILVER

Crear una función que reciba un parámetro TEXT

```
CREATE OR REPLACE FUNCTION descomponer(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 descomponer( x: 'BBAII');
```

```
`descomponer('BBAII')`  
1 I , II , AII , BAI , BBAI ,
```




GRACIAS POR SU ATENCION!



everticon123@gmail.com

facebook Ever Ticona



WhatsApp +591 60614566



**INNOVACIÓN
EN EDUCACIÓN**