TAREA HITO3 BASE DE DATOS 2

TAREA HITO3

— BASE DE DATOS II

UNIFRANZ SEDE EL ALTO HITO3 ESTUDIANTE: ISRAEL ALEJANDRO ZAMBRANA

1. Defina que es lenguaje procedural en MySQL.

R. 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.
R. Las funciones son 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? Explique sobre el nombre, el return, parámetros, etc.

R. MySQL viene incluido con una serie de funciones integradas. Las funciones integradas son simplemente funciones que ya vienen implementadas en el servidor MySQL. Estas funciones nos permiten realizar diferentes tipos de manipulaciones en los datos. Las funciones integradas se pueden clasificar básicamente en las siguientes categorías más utilizadas.

Funciones de cadenas – operan en tipos de datos de cadena

Funciones numéricas : opere en tipos de datos numéricos

Funciones de fecha : operan en tipos de datos de fecha

Funciones agregadas : opere en todos los tipos de datos anteriores y produzca conjuntos de resultados resumidos.

Otras funciones : MySQL también admite otros tipos de funciones incorporadas, pero limitaremos nuestra lección a las funciones nombradas anteriormente únicamente.

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

```
DELIMITER //
 CREATE FUCNTION holaMundo() RETURNS VARCHAR(20)
   BEGIN
RETURN ‘HolaMundo’;
END //
```

Eliminamos la función

```
DROP FUNCTION IF EXISTS holaMundo
```

modificamos la función

```
ALTER FUCNTION holaMundo() RETURNS VARCHAR(20)
```

5. Para qué sirve la función CONCAT y como funciona en MYSQL
 ○ ¿Crear una función que muestre el uso de las función CONCAT?
○ La función debe concatenar 3 cadenas.

R. **CONCAT es** una **función de** cadena compatible con **MySQL para** combinar o unir dos o más cadenas y devolverlas como un solo valor. El nombre **CONCAT** proviene del verbo concatenación, **que** significa unir 2 o más entidades juntas.

```
Select CONCAT(Nombre, ' ', Apellidos) As Nombre From usuarios;
```

6. Para qué sirve la función SUBSTRING y como funciona en MYSQL
  - ¿Crear una función que muestre el uso de las función SUBSTRING?
  - La función recibe un nombre completo. ■ INPUT: Ximena Condori Mar
- La función solo retorna el nombre. ■ OUTPUT: Ximena

R. La **función de** subcadena **de MySQL** se utiliza **para** extraer una subcadena o una parte **de** la cadena contra la cadena **de** entrada. Como sugiere el nombre, la **función Substring** opera en una cadena **de** entrada y devuelve una subcadena más pequeña contra las opciones especificadas.

SELECT SUBSTRING('funciones', 3);
-> nciones

7. Para qué sirve la función STRCMP y como funciona en MYSQL
○ ¿Crear una función que muestre el uso de las función STRCMP?

```
SELECT First_Name, Last_Name, STRCMP(First_Name, Last_Name) AS Cmp_Value
FROM StudentDetails
```

○ La función debe comparar 3 cadenas. Y deberá determinar si dos de ellas son iguales. 4

R. La **función STRCMP**() en **MySQL** se usa **para** comparar dos strings. Si ambas strings son iguales, devuelve 0, si el primer argumento **es** más pequeño **que** el segundo según el orden definido, devuelve -1 y devuelve 1 cuando el segundo **es** más pequeño **que** el primero.

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?
R. **La** función **CHAR_LENGTH**() en MySQL se usa 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("SQL Tutorial in geeksforgeeks") AS LengthOfString

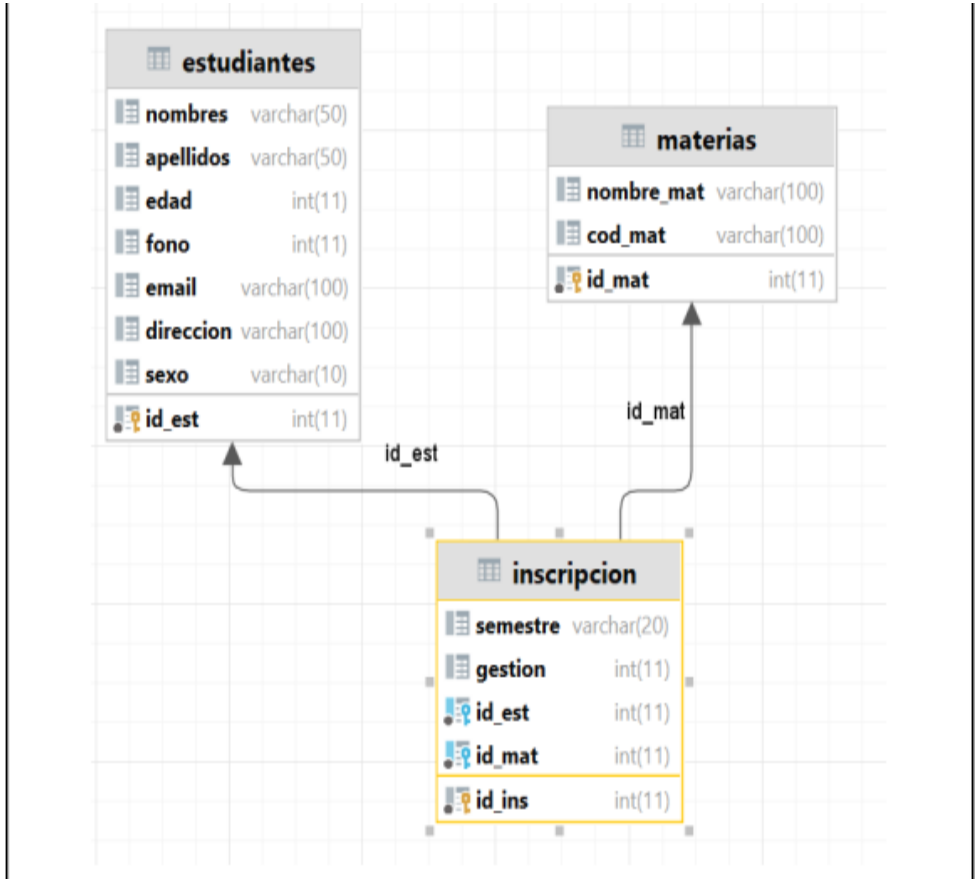
La **función LOCATE**() en **MySQL** se usa **para** encontrar la ubicación de una substring en una string. Devolverá la ubicación de la primera aparición de la substring en la string. Si la substring no está presente en la string, devolverá 0.
SELECT LOCATE('g', 'geeksforgeeks', 3) AS MatchLocation;

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

R. Las funciones de agregación son las que vienen por defento en la base de daos de mysql y de otros programas de base de datos y una **funcione** definida **por el usuario** (UDF) es un modo de extender **MySQL con** una nueva función que funciona como una función nativa de **MySQL** tal como ABS() o CONCAT() . function_name es el nombre que debe usarse en comandos SQL para invocar la función.

10.¿Busque y defina a qué se referirá cuando se habla de parámetros de entrada y salida en MySQL?
R. Los procedimientos almacenados pueden recibir y devolver información; para ello se emplean parámetros, de entrada y salida, respectivamente.
Veamos los primeros. Los parámetros de entrada posibilitan pasar información a un procedimiento.
Para que un procedimiento almacenado admita parámetros de entrada se deben declarar variables como parámetros al crearlo.

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

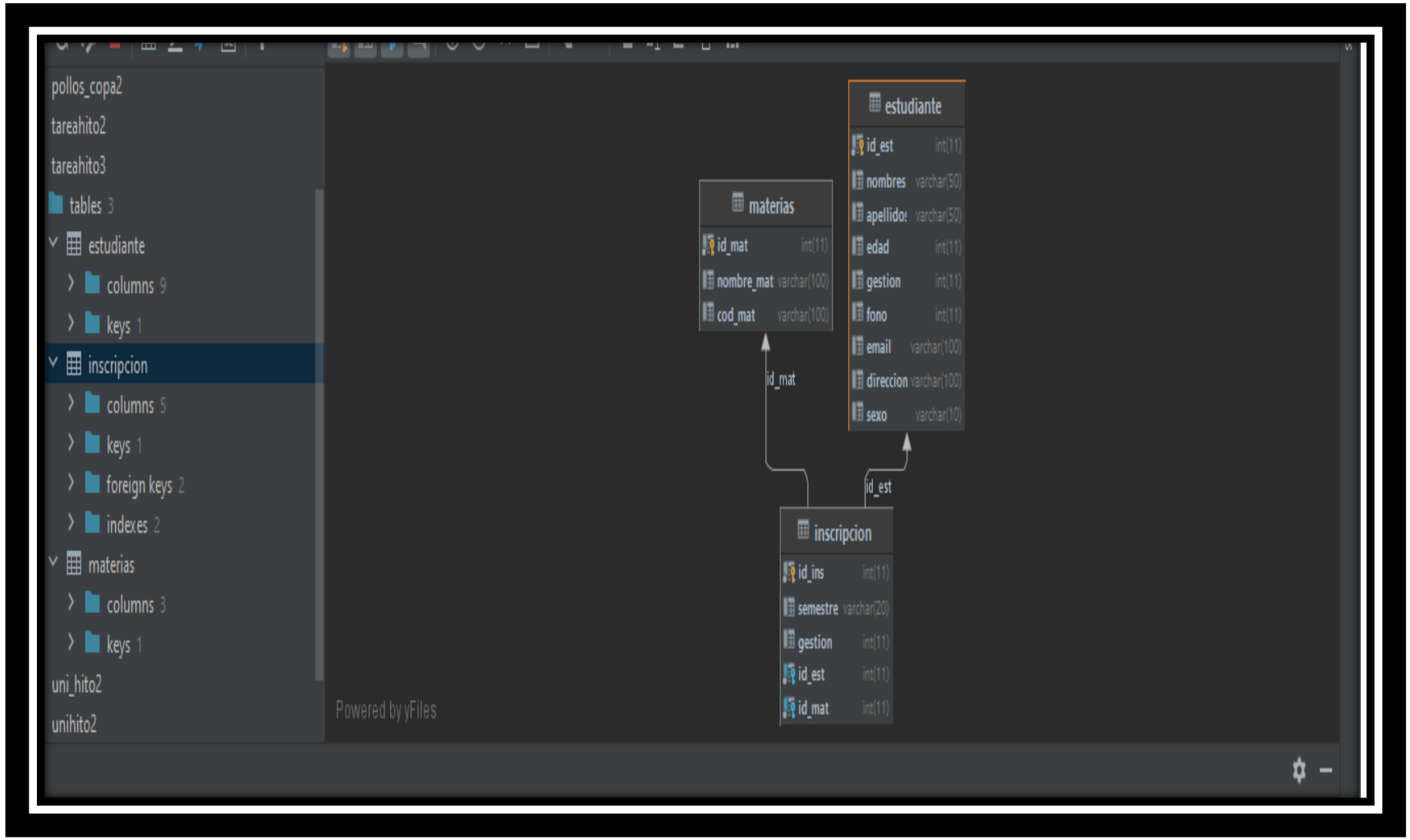
DATOS TABLA MATERIAS

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

YA CREE LA BASE DE DATOS DEL HITO3 CON SUS REGISTROS

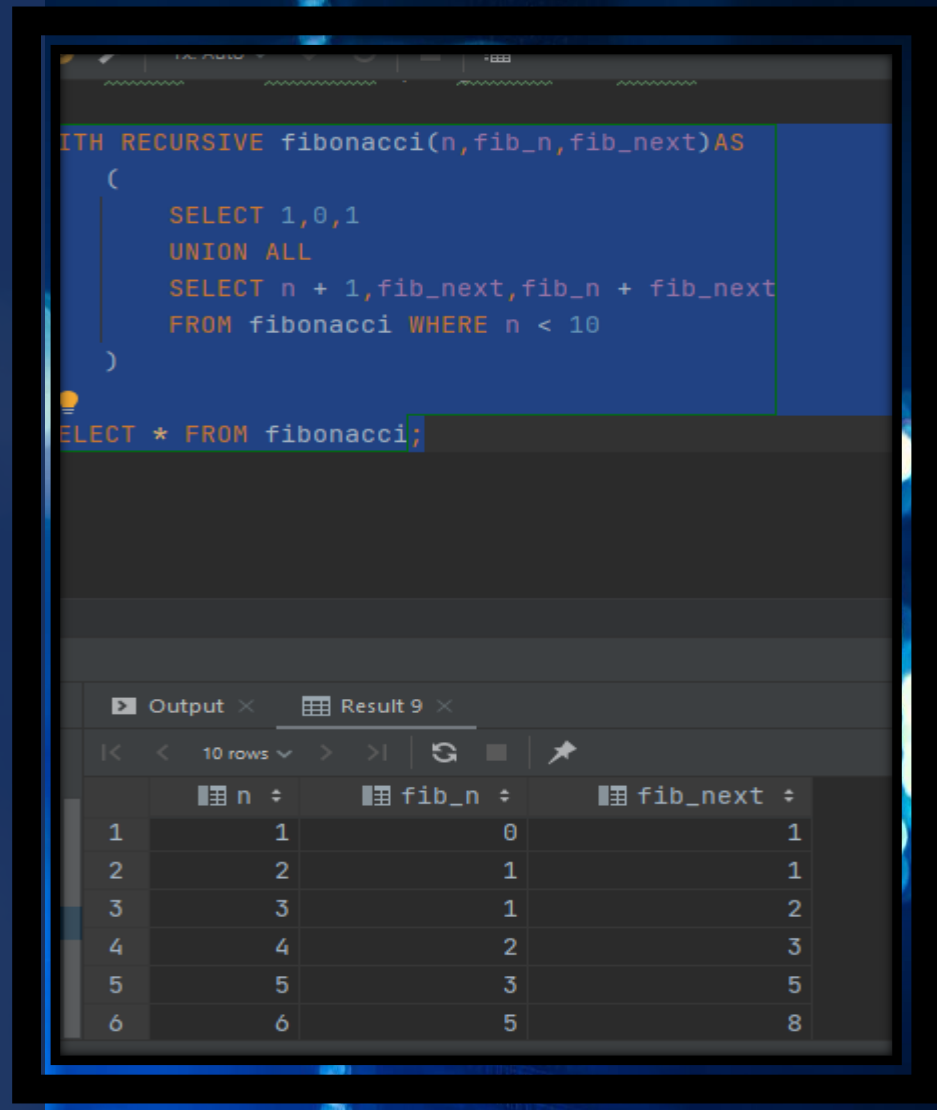


```sql
CREATE DATABASE tareaHito3;
USE tareaHito3;

CREATE TABLE estudiante
(
id_est INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombres VARCHAR(50),
apellidos VARCHAR(50),
edad INTEGER,
gestion INTEGER,
fono INTEGER,
email VARCHAR(100),
direccion VARCHAR(100),
sexo VARCHAR(10)
);

CREATE TABLE inscripcion
(
id_ins INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
semestre VARCHAR(20),
gestion INTEGER,
id_est INT NOT NULL,
id_mat INT NOT NULL,
 FOREIGN KEY (id_est) REFERENCES estudiante (id_est),
 FOREIGN KEY (id_mat) REFERENCES materias (id_mat)
);
CREATE TABLE materias
(
 id_mat INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombre_mat VARCHAR(100),
cod_mat VARCHAR(100)
);
```

```sql
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Miguel', 'Gonzales Veliz' , 20, 2832115,'miguel@gmail.com' , 'Av. 6 de Agosto' , 'masculino' );
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Sandra', 'Mavir Uria' , 25, 2832116, 'sandra@gmail.com' ,'Av. 6 de Agosto' , 'femenino');
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Joel', 'Adubiri Mondar' , 30, 2832117, 'joel@gmail.com' ,'Av. 6 de Agosto' , 'masculino' );
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Andrea', 'Arias Ballesteros' , 21, 2832118,'andrea@gmail.com' , 'Av. 6 de Agosto' ,
'femenino');
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('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');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Urbanismo y Diseno' , 'ARQ-102');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Dibujo y Pintura Arquitectonico' , 'ARQ-103');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Matematica discreta' , 'ARQ-104');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Fisica Basica' , 'ARQ-105');


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

12.CREAR UNA FUNCIÓN QUE GENERE LA SERIE FIBONACCI.

```
WITH RECURSIVE
fibonacci(n,fib_n,fib_next)AS
    (
        SELECT 1,0,1
        UNION ALL
        SELECT n + 1,fib_next,fib_n +
fib_next
        FROM fibonacci WHERE n < 10
    )

SELECT * FROM fibonacci;
```



GRACIAS