



BASE DE DATOS II
TAREA HITO 3

ESTUDIANTE: BRITTANY IBLING
MARINO QUISPE

CODIGO: SIS13181271

DOCENTE: LIC. WILLIAM BARRA
PAREDES

CARRERA: INGENIERIA DE SISTEMAS
SEMESTRE: 3

FECHA: 19-05-202

MANEJO DE CONCEPTOS.

1.- DEFINA QUE EL LENGUAJE PROCEDURAL EN MYSQL.

LENGUAJES PROCEDURAL ES, EL USUARIO DA ÓRDENES PARA QUE SE REALICEN LAS TAREAS PERTINENTES CON EL OBJETIVO DE RECUPERAR LOS DATOS REQUERIDOS. ES LA BASE DEL LENGUAJE DE CONSULTA SQL

2.- DEFINA QUE ES UNA FUNCIÓN EN MYSQL.

MEDIANTE LAS FUNCIONES QUE NOS PROVEE EL MOTOR O MEDIANTE FUNCIONES DEFINIDAS POR NOSOTROS, PODEMOS TAMBIÉN MANIPULAR LOS DATOS ANTES DE RECUPERARLOS O GUARDARLOS.

3.- ¿QUÉ COSAS CARACTERÍSTICAS DEBE DE TENER UNA FUNCIÓN? EXPLIQUE SOBRE EL NOMBRE, EL RETURN, PARAMETROS, ETC.

LAS FUNCIONES SON UNO DE LOS BLOQUES DE CONSTRUCCIÓN FUNDAMENTALES EN MYSQL. UNA FUNCIÓN EN MYSQL ES SIMILAR A UN PROCEDIMIENTO — UN CONJUNTO DE INSTRUCCIONES QUE REALIZA UNA TAREA O CALCULA UN VALOR, PERO PARA QUE UN PROCEDIMIENTO CALIFIQUE COMO FUNCIÓN, DEBE TOMAR ALGUNA ENTRADA Y DEVOLVER UNA SALIDA DONDE HAY ALGUNA RELACIÓN OBVIA ENTRE LA ENTRADA Y LA SALIDA. UNA FUNCIÓN ES UN CONJUNTO DE SENTENCIAS DE FORMA SIMILAR A UN PROCEDIMIENTO ALMACENADO PERO COMO DIFERENCIA SOLO DEBE RETORNAR UN VALOR SIMPLE (INT, VARCHAR, FLOAT ETC.) UNA FUNCIÓN TIENE UN NOMBRE, ACEPTA PARÁMETROS SOLO DE ENTRADA (NO HAY QUE ANTECEDER LA PALABRA CLAVE IN) Y RETORNA UN VALOR OBLIGATORIAMENTE. LUEGO A UNA FUNCIÓN A DIFERENCIA DE UN PROCEDIMIENTO ALMACENADO SE LO PUEDE LLAMAR DESDE UNA SENTENCIA SELECT.

4.- ¿CÓMO CREAR, MODIFICAR Y CÓMO ELIMINAR UNA FUNCIÓN? ADJUNTE UN EJEMPLO DE SU USO.

```
CREATE OR ALTER FUNCTION MIN_SUELDO(@SUELDO VARCHAR(50))
RETURNS INTEGER AS
BEGIN
    DECLARE @minsueldo INTEGER = 0;

    SET @minsueldo = (
        SELECT MIN(emp.sueldo) AS sueldo_minimo_empleados
        FROM Empleado AS emp
        WHERE emp.puesto = @SUELDO
    );

    RETURN @minsueldo
END;
SELECT dbo.MIN_SUELDO('Urología.');
```

DROP FUNCTION MIN_SUELDO;

5.- PARA QUÉ SIRVE LA FUNCIÓN CONCAT Y COMO FUNCIONA EN MYSQL
CONCAT ES UNA FUNCIÓN DE CADENA COMPATIBLE CON MYSQL
PARA COMBINAR O UNIR DOS O MÁS CADENAS Y DEVOLVERLAS COMO UN SOLO
VALOR.

- ¿CREAR UNA FUNCIÓN QUE MUESTRE EL USO DE LA FUNCIÓN CONCAT?
- LA FUNCIÓN DEBE CONCATENAR 3 CADENAS.

```
create or replace function doscadenas(cad1 text, cad2 text)
returns text
begin
    declare uno text default '';
    declare parametro int default 0;
    declare respuesta text default '';
    set uno = (concat(cad1, ' ', cad2));
    set parametro = (char_length(uno));
    if parametro > 10
    then
        set respuesta = (concat(uno, ' ', parametro));
        return respuesta;
    else
        return cad2;
    end if;
end;

select doscadenas('DBA II', '2');
```


6.- PARA QUÉ SIRVE LA FUNCIÓN SUBSTRING Y COMO FUNCIONA EN 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.

- ¿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

```
create function subCadena(cadena varchar(20), position
integer)
returns text
begin
declare subCadena text default "";
set subCadena = substr(cadena, position);
return subCadena;
end;
select subCadena('dba II', 5);
```

7.- PARA QUÉ SIRVE LA FUNCION STRCMP Y COMO FUNCIONA EN MYSQL

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.

- ¿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 compareStrings(cad1 text, cad2 text)
returns text
begin
    declare sonIguales int default -1;
    declare response text default "";
    set sonIguales = strcmp(cad1, cad2);
    if sonIguales = 0
    then
        set response = 'Cadenas iguales';
    else
        set response = 'Cadenas distintas';
    end if;
    return response;
end;

select compareStrings('dba ii', 'dba Ili');
```

8.- PARA QUÉ SIRVE LA FUNCIÓN CHAR_LENGTH Y LOCATE Y COMO FUNCIONA EN MYSQL

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. 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.

○ ¿CREAR UNA FUNCIÓN QUE MUESTRE EL USO DE AMBAS FUNCIONES?

```
create or replace function doscadenas(cad1 text, cad2 text)
returns text
begin
    declare uno text default '';
    declare parametro int default 0;
    declare respuesta text default '';
    set uno = (concat(cad1, ' ', cad2));
    set parametro = (char_length(uno));
    if parametro > 10
    then
        set respuesta = (concat(uno, ' ', parametro));
        return respuesta;
    else
        return cad2;
    end if;
end;

select doscadenas('DBA II', '2');
```


9. ¿CUAL ES LA DIFERENCIA ENTRE LAS FUNCIONES DE AGREGACIÓN Y FUNCIONES CREADOS POR EL DBA? ES DECIR FUNCIONES CREADAS POR EL USUARIO.

LAS FUNCIONES DE AGREGACIÓN EN SQL NOS PERMITEN EFECTUAR OPERACIONES SOBRE UN CONJUNTO DE RESULTADOS, PERO DEVOLVIENDO UN ÚNICO VALOR AGREGADO PARA TODOS ELLOS.

DBA IMPLEMENTA, DA SOPORTE Y GESTIONAR BASES DE DATOS CORPORATIVAS, CREAR Y CONFIGURAR BASES DE DATOS RELACIONALES.

10.¿BUSQUE Y DEFINA A QUÉ SE REFERIRÁ CUANDO SE HABLA DE PARÁMETROS DE ENTRADA Y SALIDA EN MYSQL?

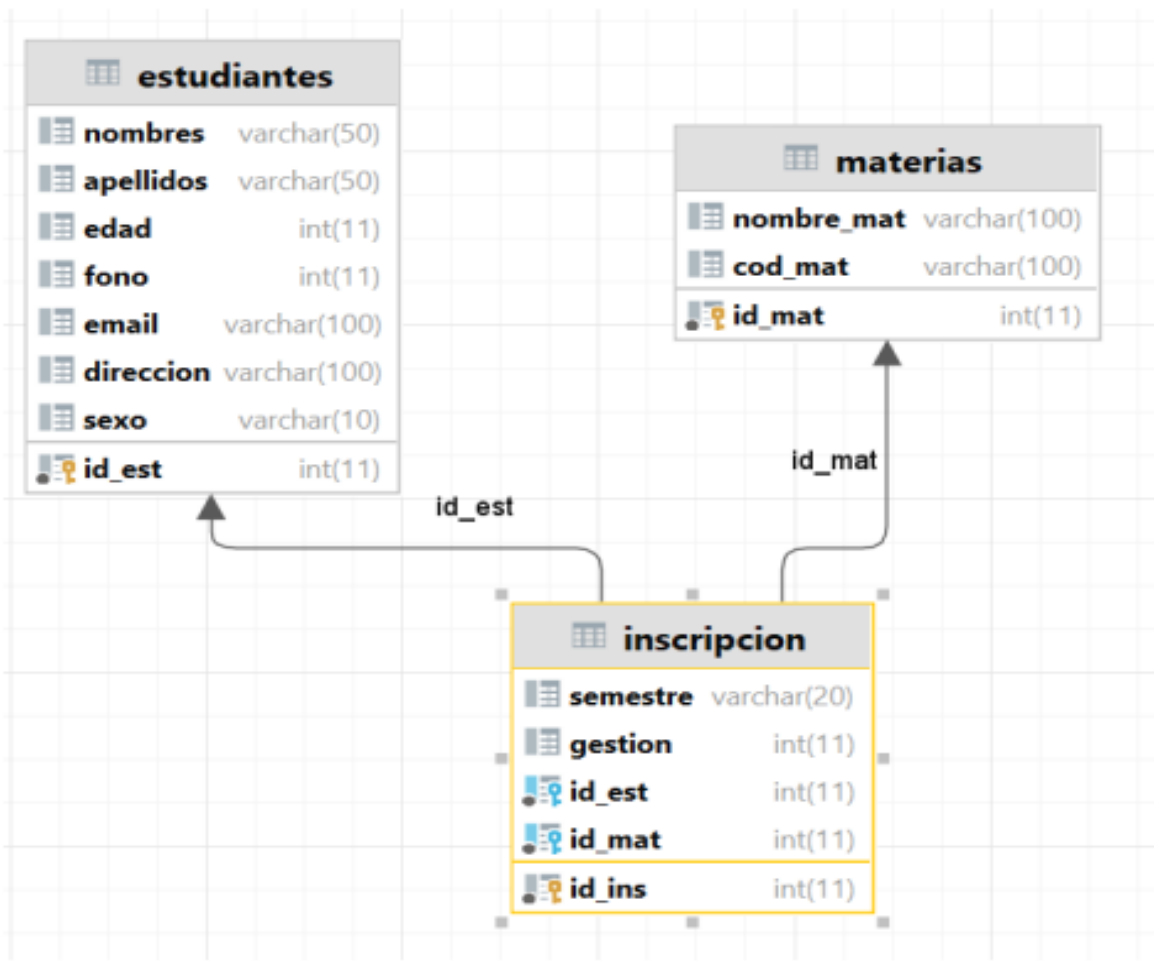
○ ES DECIR IN INOUT, ETC

ENTRADA: SE INDICAN PONIENDO LA PALABRA RESERVADA IN DELANTE DEL NOMBRE DEL PARÁMETRO. ESTOS PARÁMETROS 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.



AGREGAMOS REGISTROS A LAS TABLAS

```
console_1 [EXAMENH4]
console [@proyectoFINAL] x console_1 [IlblingHito4] x console_4 [BrittanyHito2@local]

[2] 1 of 24
of 13

1 CREATE DATABASE hito3;
2 USE hito3;
3
4 CREATE TABLE estudiantes
5 (
6     id_est INTEGER IDENTITY(1,1) PRIMARY KEY NOT NULL,
7     nombres VARCHAR(50),
8     apellidos VARCHAR(50),
9     edad INTEGER,
10    fono INTEGER,
11    email VARCHAR(100),
12    direccion VARCHAR(100),
13    sexo VARCHAR(10)
14 );
```

```
console [@proyectoFINAL] x console_1 [IlblingHito4] x console_4 [BrittanyHito2@local]

16 CREATE TABLE materias
17 (
18     id_mat INTEGER IDENTITY(1,1) PRIMARY KEY NOT NULL,
19     nombre_mat VARCHAR(100),
20     cod_mat VARCHAR(100)
21 );
22 CREATE TABLE inscripcion
23 (
24     id_ins INTEGER IDENTITY(1,1) PRIMARY KEY NOT NULL,
25     id_est INT NOT NULL,
26     id_mat INT NOT NULL,
27     semestre VARCHAR(20),
28     gestion INTEGER,
29     FOREIGN KEY (id_est) REFERENCES estudiantes (id_est),
30     FOREIGN KEY (id_mat) REFERENCES materias (id_mat)
31 );
```



```
37 ✓ INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
38     VALUES ('Miguel', 'Gonzales Veliz', 20, 2_832_115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino');
39 INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
40     VALUES ('Sandra', 'Mavir Uria', 25, 2_832_116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino');
41 INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
42     VALUES ('Joel', 'Aduviri Mondar', 30, 2_832_117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino');
43 INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
44     VALUES ('Andrea', 'Arias Ballesteros', 21, 2_832_118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino');
45 INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
46     VALUES ('Santos', 'Montes Valenzuela', 24, 2_832_119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
47
```

```
46     VALUES ('Santos', 'Montes Valenzuela', 24, 2_832_119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
47
48 INSERT INTO materias (nombre_mat, cod_mat) VALUES ('Introduccion a la Arquitectura', 'ARQ-101');
49 INSERT INTO materias (nombre_mat, cod_mat) VALUES ('Urbanismo y Diseno', 'ARQ-102');
50 INSERT INTO materias (nombre_mat, cod_mat) VALUES ('Dibujo y Pintura Arquitectonico', 'ARQ-103');
51 INSERT INTO materias (nombre_mat, cod_mat) VALUES ('Matematica discreta', 'ARQ-104');
52 INSERT INTO materias (nombre_mat, cod_mat) VALUES ('Fisica Basica', 'ARQ-105');
53 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (1, 1, '1er Semestre', 2018);
54 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (1, 2, '2do Semestre', 2018);
55 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (2, 4, '1er Semestre', 2019);
56 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (2, 3, '2do Semestre', 2019);
57 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (3, 3, '2do Semestre', 2020);
58 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (3, 1, '3er Semestre', 2020);
59 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (4, 4, '4to Semestre', 2021);
60 INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (5, 5, '5to Semestre', 2021);
61
```

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

| WHERE | | | | | | ORDER BY | | | | | |
|-------|--------|--------|--------|--------------|---------|----------|--|--|--|--|--|
| | id_ins | id_est | id_mat | semestre | gestion | | | | | | |
| 1 | 1 | 1 | 1 | 1er Semestre | 2018 | | | | | | |
| 2 | 2 | 1 | 2 | 2do Semestre | 2018 | | | | | | |
| 3 | 3 | 2 | 4 | 1er Semestre | 2019 | | | | | | |
| 4 | 4 | 2 | 3 | 2do Semestre | 2019 | | | | | | |
| 5 | 5 | 3 | 3 | 2do Semestre | 2020 | | | | | | |
| 6 | 6 | 3 | 1 | 3er Semestre | 2020 | | | | | | |
| 7 | 7 | 4 | 4 | 4to Semestre | 2021 | | | | | | |
| 8 | 8 | 5 | 5 | 5to Semestre | 2021 | | | | | | |

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

1.2.- 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,

```
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;
```


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:

```
create or replace function paramint(parametro int)
returns text
begin

    if parametro > 50000
    then
        return 'PLATINIUM';
    end if;
if parametro >= 10000 and parametro <= 50000
then
return 'GOLD';
end if;
if parametro < 10000
then
return 'SILVER';
end if;
end;

select paramint(50008);
```