



**UNIVERSIDAD FRANZ TAMAYO**

**FACULTA DE TECNOLOGIA**

**CARRERA: INGENIERIA EN SISTEMAS**

## **DEFENSA HITO 3**

- **Nombre Completo:** Julio Marco Medrano
- **Asignatura:** BASE DE DATOS II
- **Carrera:** INGENIERÍA DE SISTEMAS
- **Paralelo:** BDA (1)
- **Docente:** Lic. William R. Barra Paredes
- **fecha:** 02/11/2019

COCHABAMBA-BOLIVIA

2019

# BASE DE DATOS II

## LENGUAJE PROCEDURAL

### INTRODUCCIÓN

Ejercicios de funciones de agregación y lenguaje procedural introducción.

---

#### Resolver los siguientes ejercicios.

Crear una función que genere la serie **Fibonacci**.

- Crear un función de agregación que obtenga la edad mínima.
- Si la respuesta es un numero par debera de multiplicarlo \*2, si la respuesta es impar deberá de multiplicarlo por 3. Entonces dado ese valor multiplicado deberá de generar la serie Fibonacci hasta ese valor.
- Ejem si la respuesta es 30 deberá de multiplicarlo \*2 teniendo así 60. Entonces la serie deberá ser hasta ese valor
- Debe de utilizar WHILE o REPEAT.

Crear una función no recibe parámetros (Utilizar un solo WHILE- REPEAT o LOOP).

- La función debe de retornar un texto TEXT como respuesta.
- Crear una función que obtenga la edad mínima de los AUTORES.
- 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 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,
```

- 
- Retornar la nueva cadena concatenada.

Crear una función que reciba un parámetro VARCHAR(20) .

---

- 
- El nombre de la función deberá ser **getVocalesOrder()**
  - En donde la función deberá de recibir una **cadena** cualesquiera y retorna un Texto de respuesta **TEXT**.
  - Si la cadena fuese **"HOLA TALLER BDA I"** el resultado debería ser **"AAAEIO"**.
  - Retornar la cadena de vocales ordenada.

→ Crear una función que recibe 2 parámetros **VARCHAR(30), INTEGER**.

- El nombre de la función deberá ser **getLetrasMultiplos()**
- En donde la función deberá de recibir una **cadena** cualesquiera y un un valor numérico cualesquiera, la función retorna un Texto de respuesta **TEXT**
- Si la cadena fuese **"HOLA BDA"** y el numero fuese **2**, el resultado debería ser **"OABA"**.
- Retornar la cadena nueva formada con los múltiplos de 3.

→ 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 I"** y la segunda cadena fuese **"GESTION 2019"**.
- La nueva cadena debería ser **"TLLRDB-GSTN 2019"**.
- La nueva cadena es resultado de la concatenación de todos los valores distintos a las vocales.
- Retornar la nueva cadena concatenada.

Crear una función que recibe 2 parámetros **INTEGER, INTEGER**.

- La función debe de retornar un texto **TEXT** como respuesta.
- Crear una función que obtenga la edad promedio de los usuarios.
- Si el valor retornado de la función **es mayor a 20**. Deberá de formar una cadena nueva separada por una **","** **descendente** con los múltiplos del primer parámetro.
- Si el valor retornado de la función **es menor o igual a 20**. Deberá de formar una cadena nueva separada por una **","** **ascendente** con los múltiplos del segundo parámetro.
- Retornar la nueva cadena.

---

## RESOLUCION DE LOS EJERCICIOS

```
#####
#####Ejercicio 1 Serie Fibonacci#####
DROP FUNCTION serieFibonacci;
CREATE FUNCTION serieFibonacci (num INTEGER) RETURNS text
BEGIN
    DECLARE cont integer default 0;
    declare x integer default 1;
    declare y integer default 1;
    DECLARE aux integer default 0;
    DECLARE result TEXT DEFAULT '';
    WHILE cont < num DO
        SET aux=x+y;
        SET result=concat(result, aux, ',');
        SET x=y;
        SET y=aux;

        SET cont=cont+1;
    end while;
RETURN result;
end;

SELECT serieFibonacci(10);

#####Funcion de agregacion edad minima#####
use biblioteca;

create function getMinEdadUser() returns integer
begin
    declare minEdad integer default 0;
    select min(us.edad) into minEdad
    from users as us;

    return minEdad;

end;
SELECT getminEdad();

#####funcion
resultante#####
DROP FUNCTION funcFinal_ej1;
CREATE function funcFinal_ej1() RETURNS TEXT
BEGIN
    DECLARE edadmin integer default getminEdad();
    DECLARE cont integer default 0;
    declare x integer default 1;
    declare y integer default 1;
    DECLARE aux integer default 0;
    DECLARE aux2 integer default 0;
    DECLARE result TEXT DEFAULT '';
    IF (getminEdad()%2=0)
        THEN
```

---

---

```

        SET aux2= edadadmin*2;
    ELSE
        SET aux2= edadadmin*3;
    end if;
WHILE cont < aux2 DO
    SET aux=x+y;
    SET result=concat(result, aux,',');
    SET x=y;
    SET y=aux;

    SET cont=cont+1;
end while;
RETURN result;

end;

SELECT funcFinal_ej1();

#####Ejercicio2#####

ALTER TABLE autor ADD COLUMN edad INTEGER;
INSERT INTO autor (name, nacionality,edad)
VALUES ('autor_name_1', 'Bolivia',26),
       ('autor_name_2', 'Argentina',38),
       ('autor_name_3', 'Mexico',45),
       ('autor_name_4', 'Paraguay',50);
CREATE FUNCTION edadMinAutores() RETURNS TEXT
BEGIN
    declare minEdad integer default 0;
    DECLARE cont integer default 0;
    DECLARE resp TEXT default '';
    select min(au.edad) into minEdad
    from autor as au;
    IF (minEdad % 2=0)
    THEN
    while cont <= minEdad do
        set resp = CONCAT(resp,cont,', ');
        set cont = cont +2;
    end while;
    ELSE
    WHILE minEdad > 0 do
        set resp = CONCAT(resp,minEdad,', ');
        set minEdad = minEdad -2;
    end while;
    end if;
    RETURN resp;
end;

SELECT edadMinAutores();

```

---

---

```
#####Ejercicio 3#####
#####
#####
DROP FUNCTION getVocalesOrder;
CREATE FUNCTION getVocalesOrder(cadena varchar(20)) RETURNS text
BEGIN
DECLARE resp text default '';
DECLARE character char (1) DEFAULT '';
DECLARE n integer DEFAULT char_length(trim(cadena));
DECLARE x INTEGER DEFAULT 1;
WHILE x <= n DO
    SET character = SUBSTR(cadena, x, 1);

    IF character = 'a' or character = 'e' or character = 'i' or character = 'o' or
character = 'u'
        THEN
            SET resp = concat(resp,character );
        END IF;
        SET x = x + 1;
end while;
RETURN resp;
end;

SELECT getVocalesOrder('hola taller bda i');
```

```
#####Ejercicio 4#####
#####
```

```
#####Ejercicio 5#####
#####
DROP FUNCTION quitarVocales;
CREATE FUNCTION quitarVocales(cadena1 varchar(20), cadena2
varchar(20))RETURNS TEXT
BEGIN
declare resp text default ' ';
declare character char(1) default ' ';
declare cadena3 TEXT default concat(cadena1,cadena2);
declare n integer default char_length(trim(cadena3));

declare x integer default 1;

SET cadena3 = CONCAT(cadena3,cadena1,cadena2);

WHILE x<=n DO
SET character = SUBSTR(cadena3, x, 1);

IF character != 'a' and character != 'e' and character != 'i' and character !=
'o' and character != 'u'
    THEN
        SET resp = concat(resp,character);
    END IF;
    SET x = x + 1;
    #SET y= y+1;
```

---

---

```

    END WHILE;
    return resp;
end;

SELECT quitarVocales('taller dba i-', 'gestion 2019');

#####Ejercicio 6#####
#####
use biblioteca;
CREATE FUNCTION promUsuarios() RETURNS INTEGER
BEGIN
    DECLARE promedio int DEFAULT 0;
    SELECT avg(us.edad) into promedio
    FROM users AS us;
    RETURN promedio;

end;

DROP FUNCTION numAscDesc;

CREATE FUNCTION numAscDesc(num1 int, num2 int) RETURNS TEXT
BEGIN
    DECLARE resp TEXT DEFAULT '';
    DECLARE promEdad INTEGER DEFAULT promUsuarios();
    DECLARE i INTEGER DEFAULT num1;
    DECLARE j INTEGER DEFAULT num2;
    IF (promEdad > 20)
    THEN
        repeat
            IF (num1%i=0)
            THEN
                SET resp=CONCAT(resp, 'Descendente:', i, ', ');
            end if;
            SET i=i-1;
        until i<0 end repeat;
    ELSE
        REPEAT
            IF (num2%j=0)
            THEN
                SET resp=CONCAT(resp, 'Ascendente:', j, ', ');
            end if;
            set j=j+1;
        until j > num2 end repeat;
    end if;
    RETURN resp;

end;

SELECT numAscDesc(20,10);

```

---