



DEFENSA HITO 3 BASE DE DATOS II

ESTUDIANTE



**JHONATAN DAVID
ALANOCÁ BLANCO**



MANEJO DE CONCEPTOS

DATA

1. DEFINA QUE ES LENGUAJE PROCEDURAL EN MYSQL.

Permite escribir programas que contienen instrucciones de control de flujo y estructuras de programación, como bucles, condicionales, funciones y procedimientos almacenados. En otras palabras, se trata de una extensión del lenguaje SQL que permite crear código más complejo y sofisticado para realizar operaciones más avanzadas en la base de datos.

2. DEFINA QUE ES UNA FUNCIÓN EN MYSQL.

Una función es un subprograma que realiza una tarea específica y devuelve un valor.

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

- Debe devolver un valor.
- Debe tener argumentos de entrada.
- Debe ser determinista o no determinista.
- Debe ser reutilizable.
- Debe ser fácil de entender y mantener.

```
7   CREATE FUNCTION nombre_funcion (argumento1 tipo_dato, argumento2 tipo_dato)
8     RETURNS tipo_dato_retorno
9     BEGIN
10       -- Cuerpo de la función
11       -- Declaración de variables
12       -- Lógica de negocio
13       -- Return del valor calculado
14
15     END;
```

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

Para crear una función se utiliza el comando "CREATE FUNCTION"

```
7  #EJERCICIO 4
8  #MOSTRAR LOS APELLIDOS DE LAS PERSONAS
9  CREATE OR REPLACE FUNCTION RETORNARAPELLIDO (CADENA TEXT)
10 RETURNS TEXT
11 BEGIN
12     DECLARE RESP TEXT DEFAULT '';
13     DECLARE CONT INT DEFAULT LOCATE(' ',CADENA) ;
14     SET RESP = SUBSTRING(CADENA,CONT);
15     RETURN RESP;
16 END;
17 SELECT RETORNARAPELLIDO( CADENA: 'JHONATAN ALANCA BLANCO ')
```

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

Para modificar se utiliza el comando "ALTER FUNCTION" o tambien se puede usar el "CREATE OR REPLACE".

```
7  #EJERCICIO 4
8  #MOSTRAR LOS APELLIDOS DE LAS PERSONAS
9  CREATE OR REPLACE FUNCTION RETORNARAPELLIDO (CADENA TEXT)
10 RETURNS TEXT
11 BEGIN
12     DECLARE RESP TEXT DEFAULT '';
13     DECLARE CONT INT DEFAULT LOCATE(' ',CADENA) ;
14     SET RESP = SUBSTRING(CADENA,CONT);
15     RETURN RESP;
16 end;
17 SELECT RETORNARAPELLIDO( CADENA: 'JHONATAN ALANOCA BLANCO')
```

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

Para eliminar una función se utiliza el comando "DROP FUNCTION"

82

83

```
DROP FUNCTION RETORNARAPELIDO;
```

664771

204381

5. PARA QUÉ SIRVE LA FUNCION CONCAT Y COMO FUNCIONA EN MYSQL

La función CONCAT sirve para concatenar.

- ¿Crear una función que muestre el uso de la función CONCAT?
- La función debe concatenar 3 cadenas.

```
4  create function manejo_de_concat(cadena1 text,cadena2 text,cadena3 text)
5    returns text
6    begin
7      declare resp text default '';
8      set resp = concat(cadena1,cadena2,cadena3);
9      return resp;
10     end;
11     select manejo_de_concat( cadena1: 'manejo' , cadena2: 'de' , cadena3: 'concepto' );
```

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

Permite extraer una subcadena de una cadena.

- ¿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

```
13 | create function manejo_de_substring(cadena text)
14 | returns text
15 | begin
16 |     declare resp text default '';
17 |     declare cont int default locate(' ',cadena);
18 |     set resp = substring(cadena,1,cont);
19 |     return resp;
20 | end;
21 | select manejo_de_substring( cadena: 'ximena condori mar');
```

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

Sirve para comparar cadenas de texto y devuelve un valor numerico 0 si es verdad y un valor numerico si no es verdad.

- ¿Crear una función que muestre el uso de las función STRCMP?
- La función debe comparar 3 cadenas. Y deberá determinar si dos de ellas son iguales.

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

```
23  create or replace function manejo_de_strcmp(cadena1 text,cadena2 text,cadena3 text)
24    returns text
25  begin
26    declare resp text default '';
27    if strcmp(cadena1,cadena2)=0 then
28      set resp = '2 cadenas son iguales';
29    else if strcmp(cadena2,cadena3)=0 then
30      set resp = '2 cadenas son iguales';
31    else if strcmp(cadena1,cadena3)=0 then
32      set resp = '2 cadenas son iguales';
33
34      else set resp = 'ninguno es igual';
35    end if;
36  end if;
37  end if;
38  return resp;
39 end;
40 select manejo_de_strcmp( cadena1: 'bda' , cadena2: 'bd' , cadena3: 'bda' );
```

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

CHAR_LENGTH= CUENTA EL NUMERO DE CARACTERES QUE HAY DENTRO DE UNA CADENA

LOCATE= BUSCA UNA PALABRA DESEADA EN LA CADENA

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

DATA ANALYSIS

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

```
41  ↴#8
42  create or replace function manejo_de_locate(cadena text, letra char)
43  returns text
44  begin
45      declare resp text default '';
46      declare cont int default 0;
47      declare nVeces int default 0;
48      declare puntero char;
49      if locate(letra,cadena) > 0 then
50          while cont <= char_length(cadena) do
51              set puntero = substring(cadena,cont,1);
52              if puntero = letra then
53                  set nVeces = nVeces + 1 ;
54              end if;
55              set cont = cont + 1;
56          end while;
57          set resp = concat('en la cadena ',cadena,' la letra ',letra,' se repite ',nVeces);
58      end if;
59      return resp;
60  end;
61  select manejo_de_locate( cadena: 'cadena' , letra: 'a' )
```

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

las funciones de agregación son funciones predefinidas que sirven para realizar cálculos sobre un conjunto de valores, mientras que las funciones creadas por el usuario son funciones personalizadas que se pueden crear para realizar operaciones específicas en la base de datos.

Las funciones de agregación son funciones que ya existen en MySQL, mientras que las funciones creadas por el usuario son funciones que se pueden definir según las necesidades específicas del usuario.

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

Cuando se habla de parámetros de entrada y salida nos referimos a los valores que se pasan a una función al momento de ejecutarla (parámetros de entrada) y los valores que la función devuelve después de su ejecución (parámetros de salida). IN se usa para definir un parámetro de entrada en una función, y INOUT se usa para definir un parámetro de entrada y salida en una función,



A photograph of a person sitting on a concrete bench, wearing a light-colored long-sleeved shirt and blue jeans. They are looking down at a small electronic device, possibly a smartphone or a small tablet, which they are holding in their hands. The background shows a brick wall and some foliage.

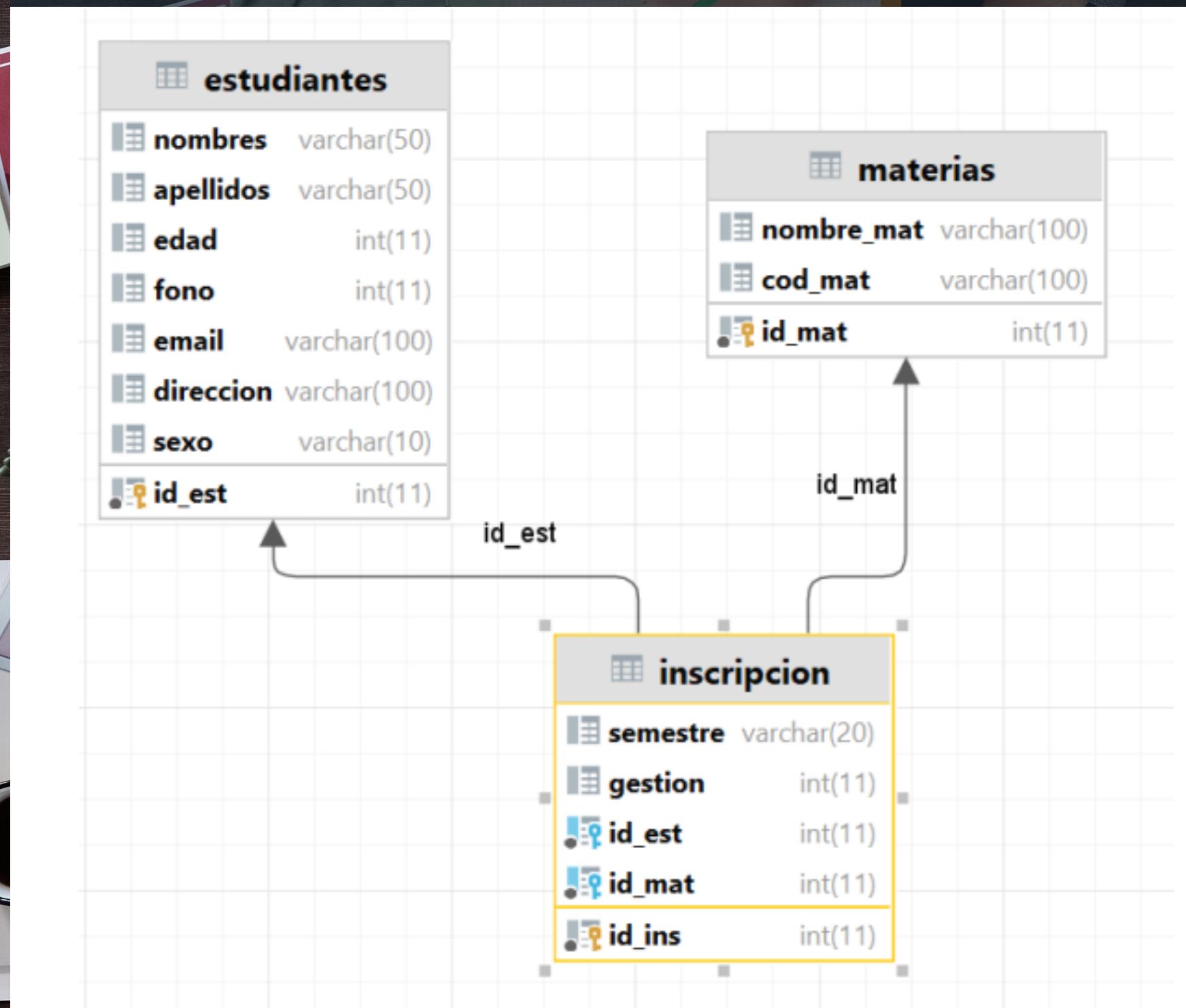
PARTE PRACTICA



A large, stylized graphic of the word "DATA" in white, overlaid on a dark background. The background is filled with various digital icons and symbols, including a globe, a circuit board, a smartphone, a laptop, a cloud, a heart rate monitor, a bar chart, and an envelope. The overall theme is data science, technology, and digital communication.

DATA

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 Urias	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	Introducción a la Arquitectura	ARQ-101
2	Urbanismo y Diseño	ARQ-102
3	Dibujo y Pintura Arquitectónico	ARQ-103
4	Matemática discreta	ARQ-104
5	Física Básica	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.

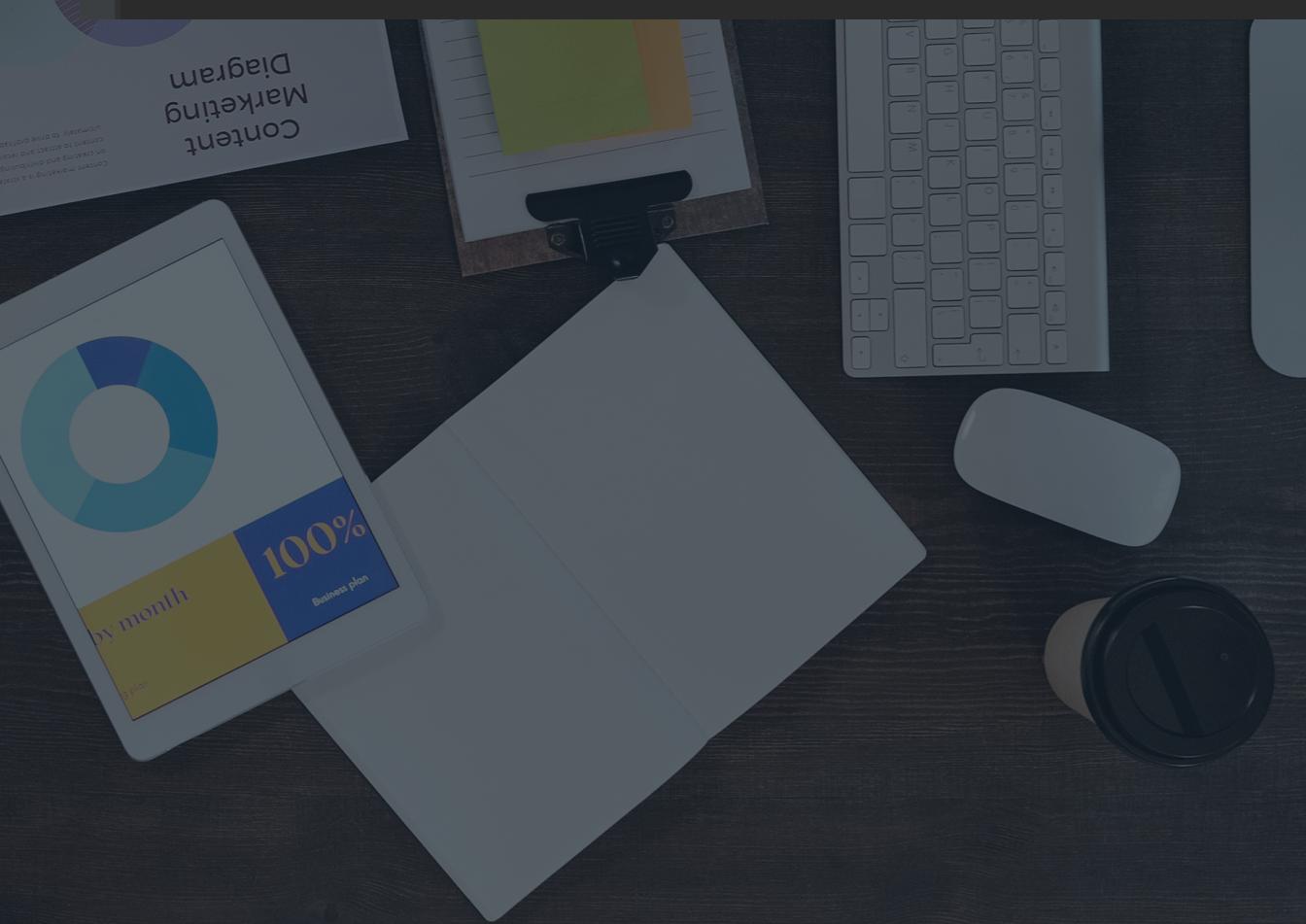
- La función recibe un límite(number)
- La función debe de retornar una cadena.
- Ejemplo para n=7. OUTPUT: 0, 1, 1, 2, 3, 5, 8,
- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento

12. CREAR UNA FUNCIÓN QUE GENERE LA SERIE FIBONACCI.

```
#12 GENERAR LA SERIE FIBONACCI
CREATE OR REPLACE FUNCTION FIBO(LIMITE INT)
RETURNS TEXT
BEGIN
    DECLARE RESP TEXT DEFAULT '';
    DECLARE a int default 0;
    DECLARE b int default 1;
    DECLARE c int default 0;
    DECLARE cont int default 0;
    while cont < LIMITE do
        SET resp = concat(resp,c,',');
        SET a = b;
        SET b = c;
        SET c = a + b;
        SET cont = cont + 1;
    end while;
    return resp;
end;
select FIBO( LIMITE: 7);
```

FIBO(7)

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



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.

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.

```
'paresImpares0'  
1 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24,
```

14. CREAR UNA FUNCIÓN NO RECIBE PARÁMETROS (UTILIZAR WHILE, REPEAT O LOOP).

- Si la edad mínima es IMPAR mostrar descendente mente todos los impares hasta el valor 0

```
'paresImpares0'  
1 25, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1,
```

- Retornar la nueva cadena concatenada.
- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.
- Nota: Esta función está llamando a otra función, considere eso.

14. CREAR UNA FUNCIÓN NO RECIBE PARÁMETROS (UTILIZAR WHILE, REPEAT O LOOP).

```
CREATE OR REPLACE FUNCTION EDAD_MINIMA()
RETURNS INT
BEGIN
    DECLARE RESP INT DEFAULT 0;
    SELECT MIN(est.Edad) INTO RESP
    FROM Estudiante AS est;
    RETURN RESP;
end;
select EDAD_MINIMA();
```

14. CREAR UNA FUNCIÓN NO RECIBE PARÁMETROS (UTILIZAR WHILE, REPEAT O LOOP).

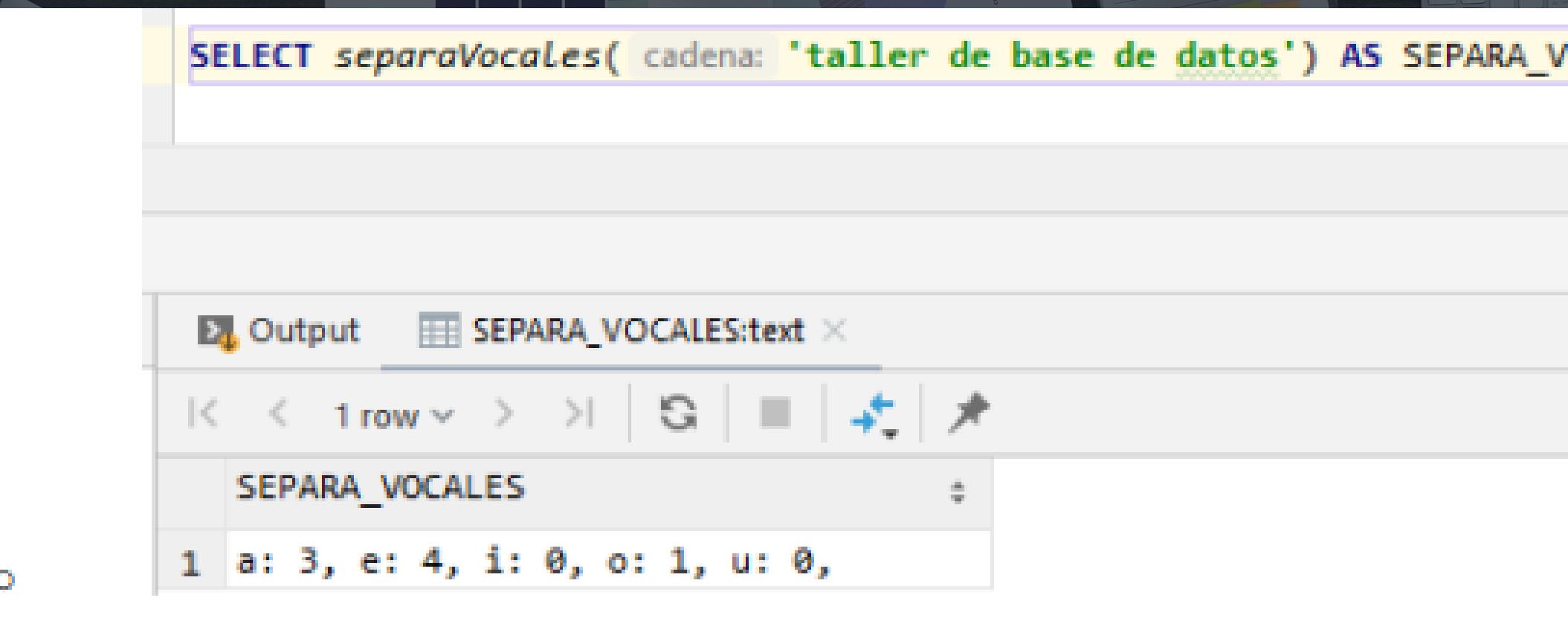
```
CREATE OR REPLACE FUNCTION PAR_IMPAR()
RETURNS TEXT
BEGIN
    DECLARE RESP TEXT DEFAULT '';
    DECLARE EDAD INT DEFAULT EDAD_MINIMA();
    DECLARE PAR INT DEFAULT 0;
    DECLARE IMPAR INT DEFAULT EDAD;
    IF EDAD MOD 2 = 0 THEN
        WHILE PAR<=EDAD DO
            SET RESP = CONCAT(RESP,PAR,',');
            SET PAR = PAR + 2;
        end while;
    ELSE IF EDAD MOD 2 = 1 THEN
        WHILE IMPAR >= 1 DO
            SET RESP = CONCAT(RESP,IMPAR,',');
            SET IMPAR = IMPAR - 2;
        end while;
    end if;
    end if;
    RETURN RESP;
end;
SELECT PAR_IMPAR();
```

The screenshot shows the MySQL Workbench interface with the following details:

- Output Tab:** Shows the results of the query `SELECT PAR_IMPAR();`. The result is a single row containing the string: `0,2,4,6,8,10,12,14,16,18,20,`.
- Table:** A table titled `PAR_IMPAR():text` is displayed with one row.
- Row Number:** The row is numbered 1.
- Column Headers:** The column header is `PAR_IMPAR():text`.

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 MySQL Workbench session window. The SQL editor contains the following query:

```
SELECT separaVocales( cadena: 'taller de base de datos' ) AS SEPARA_VOCALES;
```

The results pane shows the output of the query:

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

- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.

15. CREAR UNA FUNCIÓN QUE DETERMINA CUANTAS VECES SE REPITE LAS VOCALES.

```
CREATE OR REPLACE FUNCTION VOCALES(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;
    end while;
    set RESP = concat('a: ',CONTa,' e: ',CONTe,' i: ',CONTi,' o: ',CONTo,' u: ',CONTu);
    return RESP;
end;
```

```
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142 ✓
143
```

```
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;
```

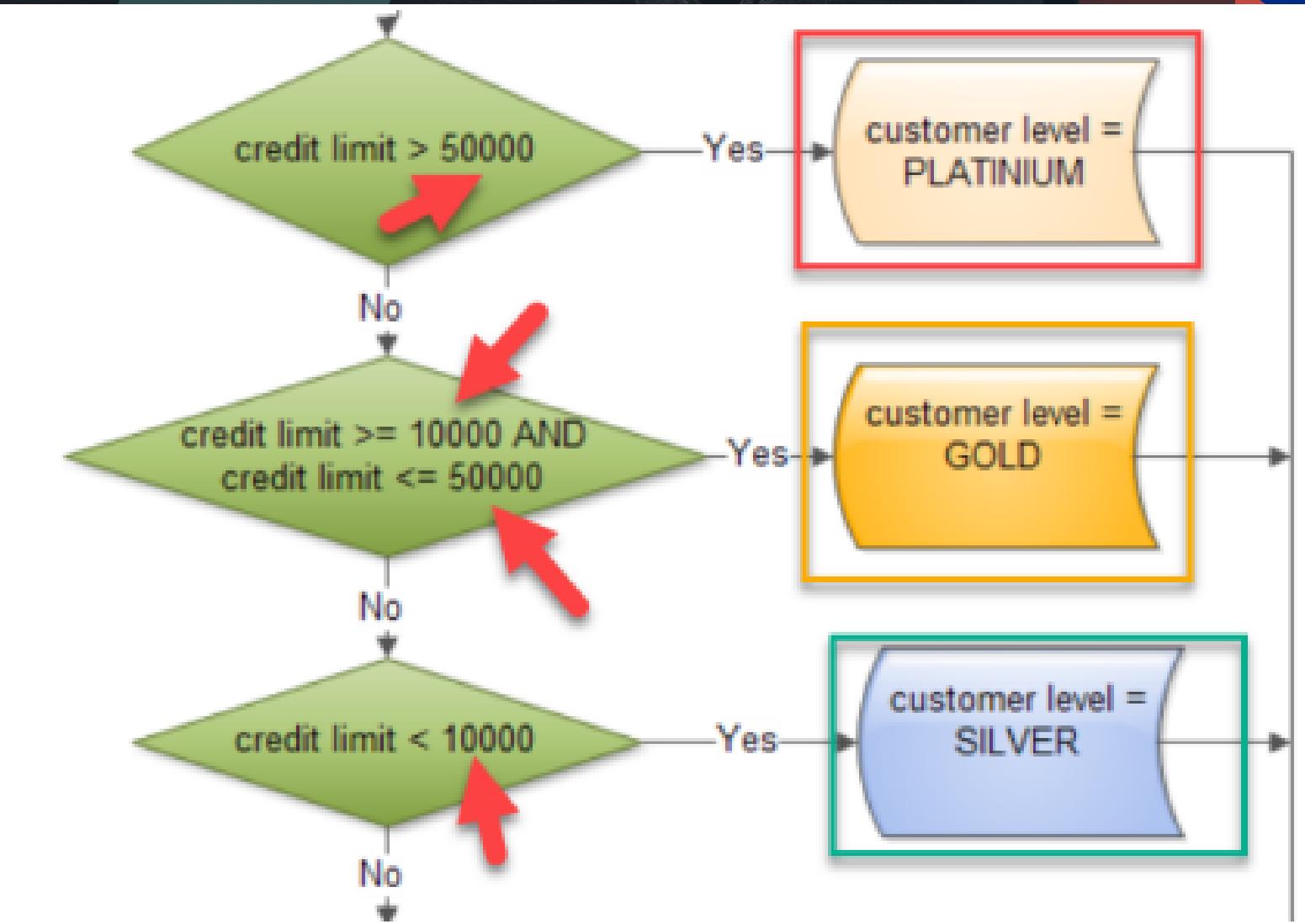
```
Select VOCALES( CADENA: 'taller de base de datos');
```

Output VOCALES('taller de base de datos'):text

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

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 PLATINUM.
- 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.
- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.



16. CREAR UNA FUNCIÓN QUE RECIBE UN PARÁMETRO INTEGER.

```
CREATE OR REPLACE FUNCTION CREDITO(CREDIT_NUMBER INT)
RETURNS TEXT
BEGIN
    DECLARE RESP TEXT DEFAULT '';
    CASE
        WHEN CREDIT_NUMBER > 50000 THEN SET RESP='PLATINUM';
        WHEN CREDIT_NUMBER >= 10000 AND CREDIT_NUMBER <=50000 THEN SET RESP='GOLD';
        WHEN CREDIT_NUMBER < 10000 AND CREDIT_NUMBER >=0 THEN SET RESP='SILVER';
    END CASE;
    RETURN RESP;
end;
```

```
SELECT CREDITO(1000);
```

Output CREDITO(1000):text

	CREDITO(1000)
1	SILVER

3 53 ms

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.

17. CREAR UNA FUNCIÓN QUE RECIBE 2 PARÁMETROS VARCHAR(20), VARCHAR(20).

```
#17 ELIMINAR VOCALES
CREATE OR REPLACE FUNCTION SINVOCALES(CADENA1 TEXT,CADENA2 TEXT)
RETURNS TEXT
BEGIN
    DECLARE RESP TEXT DEFAULT '';
    DECLARE CADENA TEXT DEFAULT CONCAT(CADENA1,'-',CADENA2);
    DECLARE PUNTERO char;
    DECLARE CONT INT DEFAULT 1;
    WHILE CONT <=CHAR_LENGTH(CADENA) DO
        SET PUNTERO = SUBSTRING(CADENA,CONT,1);
        IF PUNTERO = 'a' OR PUNTERO = 'e' OR PUNTERO = 'i' OR PUNTERO = 'o' OR PUNTERO = 'u' then
            SET CONT = CONT +1;
        ELSE IF PUNTERO = ' ' THEN
            SET CONT = CONT +1;
            SET RESP = CONCAT(RESP,PUNTERO,' ');
        ELSE
            SET CONT = CONT +1;
            SET RESP = CONCAT(RESP,PUNTERO);
        end if;
        end if;
    end WHILE;
    return RESP;
end;
SELECT SINVOCALES( CADENA1: 'TALLER DBA II', CADENA2: 'GESTION 2023');
SELECT SINVOCALES( CADENA1: 'BASE DE DATOS', CADENA2: 'ESTRUCTURA DE DATOS');
```

17. CREAR UNA FUNCIÓN QUE RECIBE 2 PARÁMETROS VARCHAR(20), VARCHAR(20).

SINVOCALES('TALLER D...', 'GESTION 2023'):text

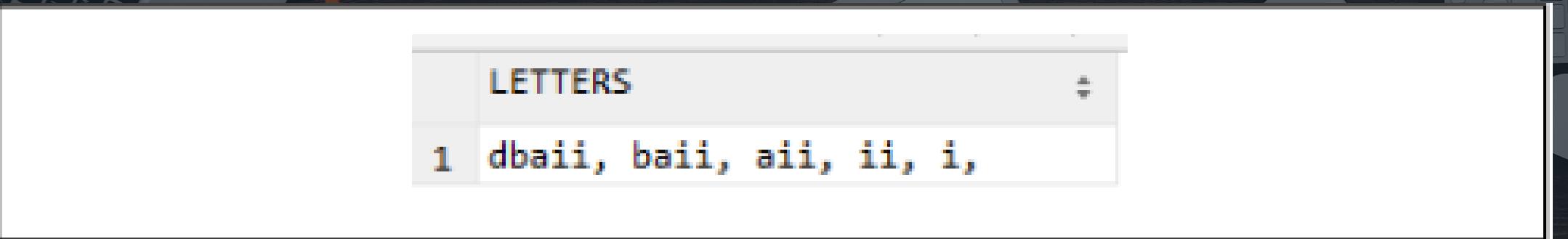
|< < 1 row > >|

1 TLLR DB -GSTN 2023

1 TLLR DB - GSTN 2023

18. CREAR UNA FUNCIÓN QUE RECIBA UN PARÁMETRO TEXT

- 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:



The screenshot shows a database interface with a results grid. The first column is labeled 'LETTERS' and contains the value '1'. The second column contains the repeated string 'dbaii, baii, aii, ii, i,'. This visualizes how the REPEAT function works by concatenating the string 'aii' five times.

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

- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento

18. CREAR UNA FUNCIÓN QUE RECIBA UN PARÁMETRO TEXT

```
181  #18
182  CREATE OR REPLACE FUNCTION REDUCCION(CADENA TEXT)
183  RETURNS TEXT
184  BEGIN
185      DECLARE RESP TEXT DEFAULT '';
186      DECLARE PUNTERO TEXT DEFAULT '';
187      DECLARE CONT INT DEFAULT 0;
188      REPEAT
189          SET PUNTERO = SUBSTRING(CADENA,CONT);
190          SET RESP = CONCAT(RESP,PUNTERO,',');
191          SET CONT = CONT +1;
192      until CONT >CHAR_LENGTH(CADENA)
193          end repeat;
194      return RESP;
195  END;
196  SELECT REDUCCION( CADENA: 'BDAII');
197
```

```
195 END;
196 ✓ SELECT REDUCCION( CADENA: 'BDAII');
197
198
```

Output REDUCCION('BDAII'):text

	REDUCCION('BDAII')
1	,BDAII,DAII,AII,II,I,