

EVALUACION PROCESUAL HITO 3

BASE DE DATOS II

CARLOS DANIEL FLORES PAUCARA

3ER SEMESTRE

2023

MANEJO DE CONCEPTOS

1.- DEFINA QUE ES LENGUAJE PROCEDURAL EN MYSQL

❖ El lenguaje procedural es la capacidad que tiene el servidor MySQL para crear y utilizar procedimientos almacenados y funciones definidas por el usuario, estos procedimientos, funciones pueden ser llamados desde cualquier lugar dentro de la base de datos MySQL estos pueden recibir parámetros de entrada y devolver valores de salida.

2.- DEFINA QUE ES UNA FUNCION EN MySQL

La función en MySQL es un Objeto de programación que puede ser definido por el usuario y utilizarlo para realizar operaciones específicas donde estos pueden recibir uno o varios valores de entrada los cuales serán procesados y devolverán un resultado.

3.- QUE COSAS CARACTERISTICAS DEBE TENER UNA FUNCION EXPLIQUE? EL NOMBRE, EL RETURN, PAREMETROS, ETC.

```
create or replace function Quita_Letra_1en1( cadena1
varchar(50) )
returns text
begin
    declare resp text default ' ';
    declare puntero varchar(50);
    declare cont int default char_length ( cadena1 );
    repeat
        set puntero = substr( cadena1 , -cont , cont );
        set resp = concat( resp , puntero , ' ' );
        set cont = cont-1;
    until cont = 0 end repeat;
    return resp;
end;
```

Nombre: el nombre debe de describir el proceso que ara la función.

Parámetros: son el o los valores de entrada que serán procesados dentro de la función, estos tiene que tener un tipo de dato ya sea cadena o numérico.

Returns: este es donde pondremos el tipo de dato que devolverá esta función será numérico o cadena.

Procesos: dentro de este se cuenta las variables que declaremos o los bucles que usemos, condónales, etc., están todos los procedimientos de la función las cuales devolverán un valor de salida.

Return: este es el encargado de devolver dicho valor de salida donde de acuerdo al tipo de valor que nosotros le hallamos dado en el return devolverá un numérico o una cadena.

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

Para crear o hacer modificaciones en una función utilizamos `create or replace function`, donde `create` lo utilizamos para crear si la función no existe y `replace` para reemplazar si la función existe.

```
create or replace function Apellidos(Parametros varchar  
(50))  
returns text  
begin  
    declare resp text default ' ';  
    return resp;  
end;
```

Para eliminar una función utilizamos `drop function` y el nombre de dicha función.


```
Drop function Apellidos;
```

5.-¿PARA QUE SIRVE LA FUNCION CONCAT Y COMO FUNCIONA EN MySQL

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

La función **CONCAT(parámetros)**, la funcion concat sirve par unir una o varias cadenas, números en una sola cadena.

```
create or replace function CONCATENADOR(cad1  
varchar(50),cad2 varchar(50),cad3 varchar(50),cad4  
varchar(50))  
returns text  
begin  
    declare resp text default '';  
    set resp = concat(cad1,cad2,cad3,cad4);  
    return resp;  
end;  
  
select CONCATENADOR(' CARLOS ',' DANIEL ','  
FLORES ',' PAUCARA ');
```



The screenshot shows a MySQL query window with the following SQL statement: `CONCATENADOR(' CARLOS ',' DANIEL ',' FLORES ',' PAUCARA ')`. The result is displayed in a table with one row and one column, showing the concatenated string: `CARLOS DANIEL FLORES PAUCARA`.

1 CARLOS DANIEL FLORES PAUCARA

6.-Para qué sirve la función SUBSTRING y como funciona en MYSQL

=¿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

La funcion **SUBSTRING** sirve para hacer cortes dentro de una cadena que ingresemos y tiene esta forma **substr(cadena,posición,cantidad)**, la posición es en donde se encuentra la parte que deseamos cortar de la cadena si ingresamos un numero positivo contara de izquierda a derecha y si ingresamos un numero negativo ira de derecha a izquierda., la cantidad es cuantas letras queremos que se mantenga.

```
create or replace function utilizando_Substring(cad1 varchar(50))
returns text
begin
  declare resp text default '';
  set resp = substr(cad1,1,6);
  return resp;

end;

select utilizando_Substring('XIMENA CONDORI MAR');
```

```
'utilizando_Substring('XIMENA CONDORI MAR')'
1 XIMENA
```

7.-Para qué sirve la función STRCMP y como funciona en MYSQL

-¿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.

La función strcmp() sirve para comparar dos cadenas sin importar si es mayúscula o minúscula, este retornara 1 o -1 si las dos cadenas no son iguales y retornara 0 si los dos valores son iguales.

```
create or replace function Utilizando_Strcmp(cad1 varchar(50),cad2
varchar(50),cad3 varchar(50))
returns text
begin
    declare resp text default ' ';
    if(strcmp(cad1,cad2)=0 or strcmp(cad1,cad3)=0 or strcmp(cad2,cad3)=0)
then
        set resp = 'Si hay son iguales dos cadenas';
    else
        set resp = 'No hay igualdad entre las cadenas';
    end if;
    return resp;
end;
select Utilizando_Strcmp('CARLOS','CARLO','carlos');
```

```
`Utilizando_Strcmp('CARLOS','CARLO','carlos')`
```

```
1 Si hay son iguales dos cadenas
```

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?

La función `char_length(cadena)` la utilizamos para contar la cantidad de letras que se encuentran dentro de una cadena incluido los vacíos

La función `locate(buscamos,cadena,posicion)` la utilizamos para ver en que posición se encuentra lo que buscamos, la posición es desde donde nosotros queremos que busque.

```
create or replace function Utilizando_Locate_Charlength(Buscar varchar(50),cad1 varchar(50))
returns text
begin
    declare resp text default ' ';
    declare cont text default '0';
    declare pos text default '0';
    set pos = locate(Buscar,cad1);
    if(pos>0) then
        set cont = char_length(Buscar);
        set resp = concat('La palabra que buscas se encuentra en la posicion : ',pos,' y es de ', cont, ' palabras');
    end if;

    return resp;
end;
select Utilizando_Locate_Charlength('Flores','Carlos Daniel Flores Paucara');
```

```
1 `Utilizando_Locate_Charlength('Flores','Carlos Daniel Flores Paucara')`
1 La palabra que buscas se encuentra en la posicion : 15 y es de 6 palabras
```


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 agresión son las que viene con el servidor de MySQL, como puede ser count(), sum, max, etc.

Las funciones creados por el DBA son las funciones que los usuarios creamos para una tarea en específico.

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

Parametros de entrada: estos son los datos que ingresan una función ya sea numérico o tipo cadena, Iso caules pasan por el procedimiento de la función a al que le damos estos valores.

Salida: son esos datos resultantes de la función tras haber recibido uno o varios parámetros el cual se almacena en una variable de tipo cadena o de tipo numéricos.



PARTIE

PRACTICA

11.-CREAR LA SIGUIENTE BASE DE DATOS Y SUS REGISTROS.

```
create database procesual_hito3;  
use procesual_hito3;
```

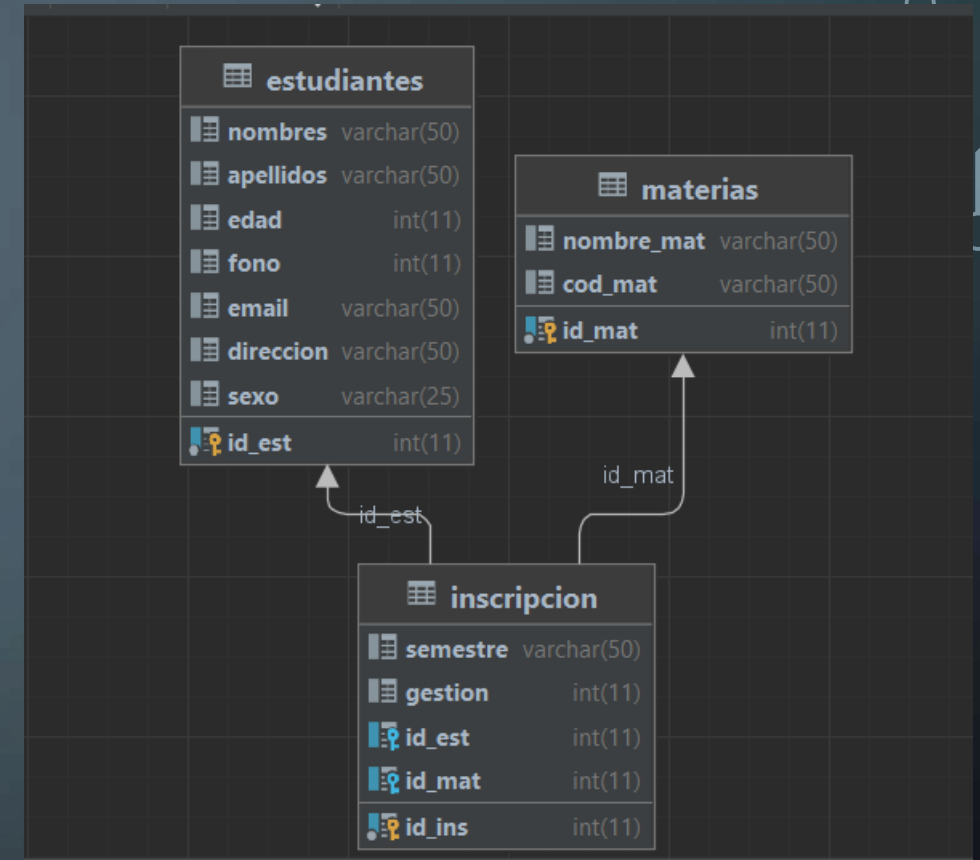
```
create table Estudiantes(  
  
    id_est int auto_increment primary  
key,  
    nombres varchar(50),  
    apellidos varchar(50),  
    edad int,  
    fono int,  
    email varchar(50),  
    direccion varchar(50),  
    sexo varchar(25)  
);
```

```
insert into Inscripcion(semestre,  
gestion, id_est, id_mat)  
values ('1er Semestre',2018,1,1),  
('2do Semestre',2018,1,2),  
('1er Semestre',2019,2,4),  
('2do Semestre',2019,2,3),  
('2do Semestre',2020,3,3),  
('3er Semestre',2020,3,1),  
('4to Semestre',2021,4,4),  
('5to Semestre',2021,5,5);
```

```
create table Inscripcion(  
  
    id_ins int auto_increment primary key,  
    semestre varchar(50),  
    gestion int,  
    id_est int,  
    id_mat int,  
  
    foreign key (id_est) references  
Estudiantes(id_est),  
    foreign key (id_mat) references  
Materias(id_mat)  
);
```

```
create table Materias(  
  
    id_mat int auto_increment primary  
key,  
    nombre_mat varchar(50),  
    cod_mat varchar(50)  
);
```

```
insert into Estudiantes(nombres, apellidos, edad, fono, email, direccion, sexo)  
values ('Miguel','Gonzales Veliz',20,2832115,'miguel@gmail.com','Av. 6 de Agosto','masculino'),  
('Sandra','Mavir Uria',25,2832115,'sandra@gmail.com','Av. 6 de Agosto','femenino'),  
('Joel','Aubiri Mondar',30,2832115,'joel@gmail.com','Av. 6 de Agosto','masculino'),  
('Andrea','Arias Ballesteros',21,2832115,'andrea@gmail.com','Av. 6 de Agosto','femenino'),  
('Santos','Montes Valenzuela',24,2832115,'santos@gmail.com','Av. 6 de Agosto','masculino');
```



12. CREAR UNA FUNCIÓN QUE GENERE LA SERIE FIBONACCI.

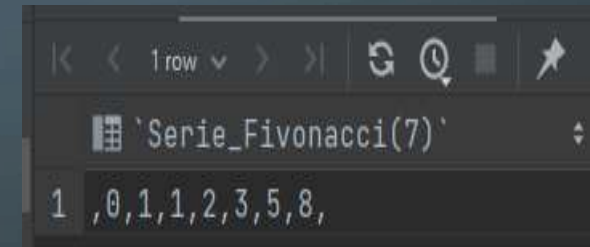
- ❖ La función recibe un límite(number)
- ❖ La función debe de retornar una cadena.
- ❖ Ejemplo para n=7. OUTPUT: 0, 1, 1, 2, 3, 5, 8

```
create or replace function Serie_Fivonacci(numb int)
returns TEXT
begin
    declare respuesta text default '';
    declare cont int default 1;
    declare numR int default 0;
    declare num1 int default 0;
    declare num2 int default 1;

    while(numb >= cont) do
        set respuesta = concat(respuesta,num1,',');
        set numR = num1+num2;
        set num1 = num2 ;
        set num2 = numR ;
        set cont = cont+1;

    end while;
    return respuesta;
end;

select Serie_Fivonacci(7);
```



The screenshot shows a database query result window. At the top, there are navigation icons and a dropdown menu showing '1 row'. Below this, the query 'Serie_Fivonacci(7)' is entered. The result is displayed in a table with one row containing the string '1,0,1,1,2,3,5,8,'.

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

13.- CREAR UNA VARIABLE GLOBAL A NIVEL BASE DE DATOS.

- ❖ o Crear una función cualquiera.
- ❖ o La función debe retornar la variable global.

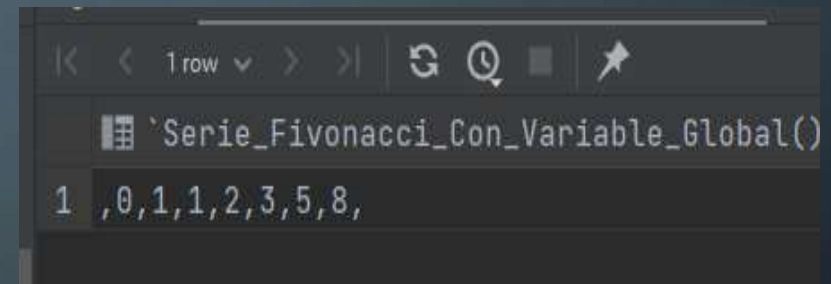
```
create or replace function Serie_Fivonacci_Con_Variable_Global()
returns text
begin
  declare resp text default ',';
  declare cont int default 1;
  declare numR int default 0;
  declare num1 int default 0;
  declare num2 int default 1;

  while(@limit >= cont) do
    set resp = concat(resp,num1,',');
    set numR = num1+num2;
    set num1 = num2 ;
    set num2 = numR ;
    set cont = cont+1;

  end while;
  return resp;
end;

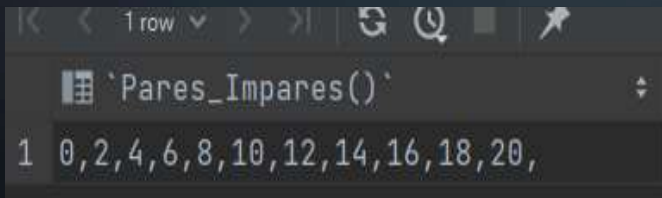
set @limit = 7;

select Serie_Fivonacci_Con_Variable_Global();
```



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

- ❖ o 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).
- ❖ o Si la edad mínima es PAR mostrar todos los pares empezando desde 0 a este ese valor de la edad mínima
- ❖ Si la edad mínima es IMPAR mostrar descendentemente todos los impares hasta el valor 0.



The screenshot shows a database interface with a table named 'Pares_Impares()'. The table has one row containing the sequence of even numbers from 0 to 20: '0,2,4,6,8,10,12,14,16,18,20,'.

1 0,2,4,6,8,10,12,14,16,18,20,

```
create or replace function Pares_Impares()
returns text
begin
  declare resp text default '';
  declare par int default 0;
  declare impar int default EdadMnima();
  if EdadMnima() % 2 = 0 then
    repeat
      if( par % 2 = 0 ) then
        set resp = concat( resp,par,', ' );
      end if;
      set par = par+1;
    until par > EdadMnima() end repeat;
  else
    impares: LOOP
      if impar<0 then
        leave impares;
      end if;

      if impar % 2 != 0 then
        set resp= concat( resp ,impar,', ' );
      end if;

      set impar = impar - 1;

    iterate impares;
  end loop;
end if;
return resp;
end;
```

```
create or replace function
EdadMnima()
returns int
begin
  declare resp int default 0;

  select min(edad) into resp from
Estudiantes;

  return resp;
end;

select Pares_Impares();
```

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.

```
create or replace function Separa_Vocales(Cad
varchar(100))
returns TEXT
begin
    declare resp text default '';
    declare num int default char_length(Cad);
    declare conA int default 0;
    declare conE int default 0;
    declare conI int default 0;
    declare conO int default 0;
    declare conU int default 0;
    if (locate('a',Cad)>0
        or locate('e',Cad)>0 or locate('i',Cad)>0 or
        locate('o',Cad)>0
        or locate('u',Cad)>0) then
        while (num > 0) do
            if substr(Cad,num,1) like '%a' then
                set conA = conA +1;
            end if;
            if substr(Cad,num,1) like '%e' then
                set conE = conE +1;
            end if;
```

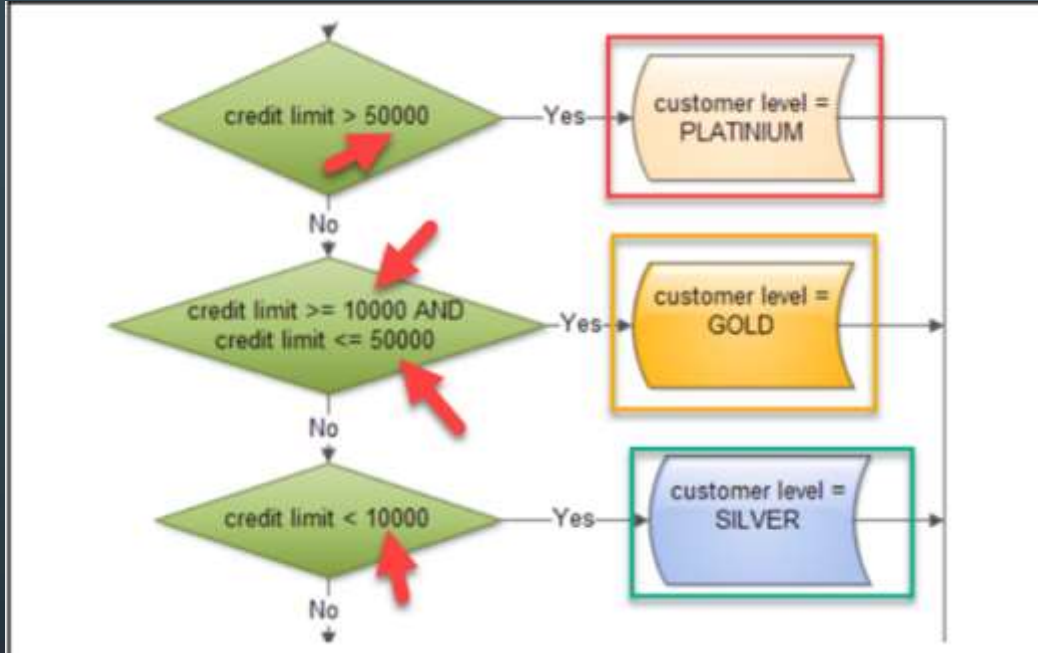
```
            if substr(Cad,num,1) like '%i' then
                set conI = conI +1;
            end if;
            if substr(Cad,num,1) like '%o' then
                set conO = conO +1;
            end if;

            if substr(Cad,num,1) like '%U' then
                set conU = conU + 1;
            end if;
            set num = num - 1;
        end while;
        set resp = concat('a: ',conA , ' e: ',conE, ' i: ',conI , ' o:
',conO,' u: ',conU);
    else
        set resp = 'No hay Vocales en la Cadena que ingreso';
    end if;
    return resp;
end;

select Separa_Vocales('Taller de base de datos');
```

```
'Separa_Vocales('Taller de base de datos')'
1 a: 3 e: 4 i: 0 o: 1 u: 0
```

16.-CREAR UNA FUNCIÓN QUE RECIBE UN PARÁMETRO INTEGER.



```
create or replace function Customer_level(Money integer)
returns text
begin
  declare resp text default '';
  if Money > 0 then
    case
      when Money > 50000 then set resp = 'PLATINIUM';
      when Money >= 10000 and Money <= 50000 then
        set resp = 'GOLD';
      when Money < 10000 then set resp = 'SILVER';
    end case;
  else
    set resp = concat('-',Money, ' BS');
  end if;
  return resp;
end;

select Customer_level(50000);
```

Para resolver debe de utilizar la instrucción CASE - WHEN.

The screenshot shows a database query result for the function `Customer_level(50000)`. The result is a single row with the value '1 GOLD'.

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.

```
create or replace function Eliminar_Vocales(cad1 varchar(20),cad2 varchar(20))
returns TEXT
begin
  declare resp text default "";
  declare puntero char;
  declare cont int default 1;
  declare nuevacadena varchar (100) default concat(cad1,'-',cad2);
  WHILE (cont <= char_length(nuevacadena)) do
    set puntero = substr(nuevacadena,cont,1);
    if(puntero != 'a' and puntero != 'e' and puntero != 'i' and puntero != 'o'
      and puntero != 'u')then
      set resp = concat(resp,puntero);
      if(puntero = '-') then
        set resp = concat(resp,' ');
      end if;
    end if;
    set cont = cont+1;
  end while;
  return resp;
end;
```

```
select Eliminar_Vocales('TALLER DBA II','GESTION 2023');
```

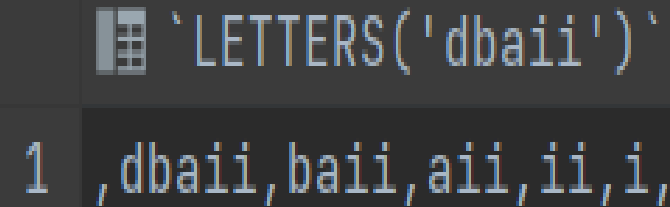
```
'Eliminar_Vocales('TALLER DBA II','GESTION 2023')'
```

```
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.

```
create or replace function LETTERS(cad1
varchar(50))
returns text
begin
  declare resp text default '';
  declare puntero varchar(50);
  declare cont int default
char_length(cad1);
  repeat
    set puntero = substr(cad1,-cont,cont);
    set resp = concat(resp,puntero,',');
    set cont = cont-1;
  until cont = 0 end repeat;
  return resp;
end;
select LETTERS('dbaii');
```



The screenshot shows a SQL query result in a dark-themed interface. The query is `'LETTERS('dbaii')'`. The result is displayed in a table with one row and one column. The value in the table is `,dbaii,baii,aii,ii,i,`.

1 ,dbaii,baii,aii,ii,i,