

A decorative graphic on the left side of the slide, consisting of white lines and circles on a dark teal background, resembling a circuit board or a stylized tree structure.

CLASE 12

PROCEDIMIENTOS ALMACENADOS

PROCEDIMIENTOS

¿Qué es un Procedimiento?

- Un procedimiento almacenado en MySQL es una porción de código que se puede guardar y reutilizar.

¿Para qué se emplea?

- Es útil cuando se repite la misma tarea en varias ocasiones

¿Necesita parámetros?

- Si. Acepta datos como parámetros para interactuar con ellos

¿Cómo es formato?

```
delimiter //  
create procedure procedimiento(in varEntrada tipo,out varSalida tipo)  
begin  
    select * from tabla; *  
end//  
delimiter ;
```

SINTAXIS

```
delimiter //  
create procedure procedimiento(in varEntrada tipo,out varSalida tipo)  
begin  
    select * from tabla; *  
end//  
delimiter ;
```

Delimitadores: // indican a MySql que inicia una porción de código que debe ejecutarse como una unidad

Nombre del procedimiento: nombre por el que se invocará al procedimiento

IN: indica el nombre del parámetro de entrada o el valor que recibirá para ser procesado.

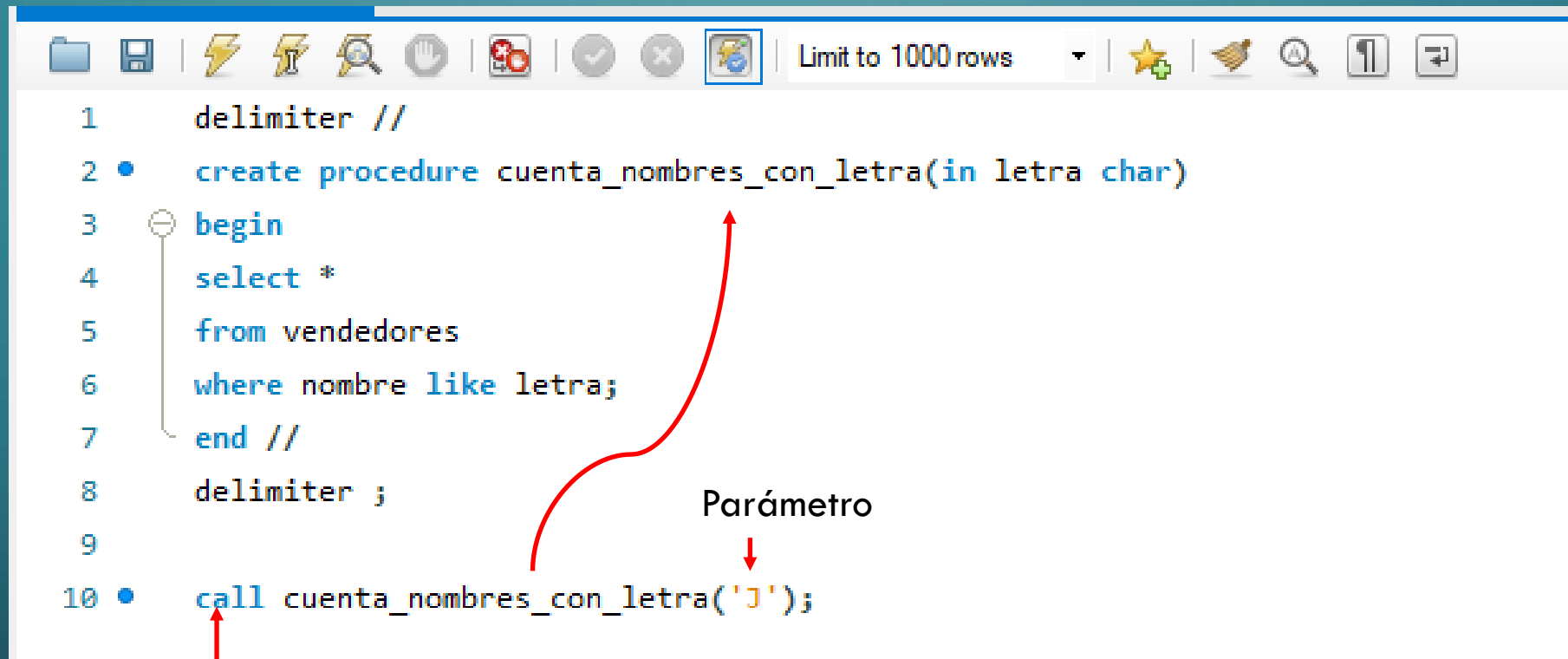
TIPO: indica el tipo de dato del parámetro

OUT: Parámetro de salida indica el nombre de la variable o parámetro donde se almacenará el resultado devuelto por la ejecución del código.

BEBING – END: Palabras reservadas para indicar el inicio y fin de la transacción o código a ejecutar.

; Delimitador punto y coma. Es que usa por defecto MySql para indicar al motor la ejecución de un código

CODIFICANDO UN PROCEDIMIENTO



The screenshot shows a SQL IDE window with a toolbar at the top. The toolbar includes icons for file operations (folder, save, lightning bolt, magnifying glass, hand), execution (play, stop, refresh), and viewing (Limit to 1000 rows, star, brush, magnifying glass, vertical bar, and a button with a right arrow). The main area contains SQL code with line numbers 1 through 10. A red arrow points from the word 'call' on line 10 to the text 'CALL: Ejecuta la llamada al procedimiento' below the code. Another red arrow points from the parameter 'J' in the procedure call to the word 'Parámetro' above it.

```
1  delimiter //  
2  • create procedure cuenta_nombres_con_letra(in letra char)  
3  begin  
4  select *  
5  from vendedores  
6  where nombre like letra;  
7  end //  
8  delimiter ;  
9  
10 • call cuenta_nombres_con_letra('J');
```

CALL: Ejecuta la llamada al procedimiento

Parámetro

EJEMPLO

```
-- Clase 12-Procedimientos.sql
delimiter //
create procedure cuenta_nombres_con_letra(in letra char)
begin
select *
from vendedores
where nombre like concat('%',letra,'%');
end //
delimiter;

call cuenta_nombres_con_letra('J');
```

EJEMPLO CON VARIABLES

```
-- Ejemplo 2
delimiter //
create procedure cuenta_nombres_con_letra(in letra char, out vnum int)
begin
select count(*) into vnum
from vendedores
where nombre like concat('%',letra,'%');
end //
delimiter;
```



Variable de salida

Ejecutar el procedimiento y mostrar el contenido de la variable

EJECUTAR CON DISTINTOS PARAMETROS

```
call cuenta_nombres_con_letra(letra, @cant_a);  
call cuenta_nombres_con_letra(letra, @cant_e);  
call cuenta_nombres_con_letra(letra, @cant_j);  
call cuenta_nombres_con_letra(letra, @cant_v);  
  
select @cant_a, @cant_e, @cant_j, @cant_v;
```

EJERCICIO

- Construya un procedimiento que:
 1. Reciba el código de cliente y verifique si ha realizado al menos una compra.
 2. Si ha realizado compras muestre "Cliente activo" en caso contrario escriba "Cliente no activo"