1. Crea un procedimiento almacenado llamado proc1 que sin tener parámetros de entrada devuelva la suma de 2+2.

DELIMITER //

CREATE PROCEDURE suma ()

BEGIN

SELECT 2 + 2

END //

DELIMITER ;

2. Crea un procedimiento almacenado llamado proc2 que introduzca un valor de tipo entero. Dicho valor se restará a 5.

DELIMITER //

CREATE PROCEDURE proc2 (in valor integer)

BEGIN

SELECT valor - 5

END //

DELIMITER ;

3. Crea un procedimiento almacenado llamado proc3 que devuelva la version actual de mysql que se está usando. Usa un parámetro OUT tipo Varchar(25).

DELIMITER //

Drop procedure if exists proc3//

CREATE PROCEDURE proc 3 (out ver varchar(25))

BEGIN

SELECT versión() into ver;

END //

DELIMITER ;

Call proc3(@version);

Select @version;

4. Crea un procedimiento almacenado llamado proc4 que incremente en uno el parámetro pasado como argumento y además lo muestre por pantalla. Inicializa el parámetro a 10.

DELIMITER //

Drop procedure if exists proc4//

CREATE PROCEDURE proc4 (inout num integer)

BEGIN

SELECT num1 = num1 + 1

END //

DELIMITER ;

Set @numero = 10;

Call proc4 (@numero);

Select @numero;

5. Crea un procedimiento almacenado llamado proc5 que muestre la versión de MySQL que se está usando y también la fecha de hoy.

DELIMITER //

Drop procedure if exists proc5//

CREATE PROCEDURE proc5 (out version varchar(25), out fecha date)

BEGIN

SELECT versión() into version;

SELECT currentDate() into fecha;

END //

DELIMITER ;

Call proc5(@version, @fecha);

Select @version, @fecha;

6. Crea un procedimiento almacenado llamado proc6 que muestre el año de la fecha actual.

DELIMITER //

Drop procedure if exists proc6//

CREATE PROCEDURE proc6 (out fecha int)

BEGIN

SELECT year(current\_date()) into fecha;

END //

DELIMITER ;

Call proc6(@año);

Select @año;

7. Crea un procedimiento almacenado llamado proc7 que muestre el año de la fecha actual y que además incremente en uno el parámetro pasado como argumento. Inicialízalo a 5.

DELIMITER //

Drop procedure if exists proc7//

CREATE PROCEDURE proc7 (out fecha int, inout param int)

BEGIN

SELECT year(current\_date()) into fecha;

Set param = param + 1;

END //

DELIMITER ;

Set @param = 5

Call proc7(@fecha, @param);

Select @fecha, @param;

8. Crea un procedimiento almacenado llamado proc8 en el que declaremos una variable NUM1, le demos un valor de inicialización por defecto y el procedimiento nos devuelva dicho valor.

DELIMITER //

Drop procedure if exists proc8//

CREATE PROCEDURE proc8 (out num int)

BEGIN

DECLARE num1 integer default 5;

Set num = num1;

END //

DELIMITER ;

Call proc8(@num);

Select @num;

9. Crea un procedimiento almacenado llamado proc9 en el que declaremos dos variables de tipo entero y nos devuelva el resto de su división hasta con dos decimales.

DELIMITER //

Drop procedure if exists proc9//

CREATE PROCEDURE proc9 (in num1 int, in num2 int, out división int)

BEGIN

Set @división = mod(num1, num2);

END //

DELIMITER ;

Call proc9(@num1, @num2, @division);

Select @division;

10. Crea un procedimiento almacenado llamado proc10 que dada una cadena, devuelva la misma cadena con pero en mayúsculas.

DELIMITER //

Drop procedure if exists proc10//

CREATE PROCEDURE proc10 (inout cadena varchar(25))

BEGIN

SELECT upper(cadena) into cadena;

END //

DELIMITER ;

Call proc10(@cadena);

Select @cadena;

11. Crea un procedimiento almacenado llamado proc11 que dada una cadena, devuelva la longitud de dicha cadena.

DELIMITER //

Drop procedure if exists proc11//

CREATE PROCEDURE proc11 (in cadena varchar(25), out long int)

BEGIN

SELECT lenght(cadena) into long;

END //

DELIMITER ;

Set @cadena = “Hola”

Call proc11(@cadena, @long);

Select @long;

12. Crea un procedimiento almacenado llamado proc12 que dada una cadena, devuelva la misma cadena con las tres primera letras en mayúsculas.

DELIMITER //

Drop procedure if exists proc12//

CREATE PROCEDURE proc12 (in cadena varchar(25), out cadenaFormat varchar(25))

BEGIN

SELECT concat(upper(left(cadena, 3)), lower(right(cadena, (lenght(cadena-4))))) into cadenaFormat;

END //

DELIMITER ;

Call proc12(“una cadena de texto”, @nuevaCadena);

Select @nuevaCadena;

13. Crea un procedimiento almacenado llamado proc13 que calcule el área de una circunferencia. El parámetro pasado como entrada será el radio y debe ser de tipo FLOAT.

DELIMITER //

Drop procedure if exists proc13//

CREATE PROCEDURE proc13 (in radio float, out area float)

BEGIN

SELECT 2 \* pi() \* radio \* radio into area;

END //

DELIMITER ;

Call proc13(@radio, @area);

Select @area;

14. Crea un procedimiento almacenado llamado proc14 que reciba dos cadenas de entrada; un nombre y un apellido y devuelva la concatenación de ellos.

DELIMITER //

Drop procedure if exists proc14//

CREATE PROCEDURE proc14 (in nombre varchar(25), in apellido varchar(25), out concatenacion varchar(51))

BEGIN

SELECT concat(nombre, ‘ ’, apellido) into concatenacion;

END //

DELIMITER ;

Call proc14(@nombre, @apellido, @result);

Select @result;