

EJERCICIO 4. 1 Crear un procedimiento que muestre la tabla de multiplicar de un número que se paso como entrada

nota concat

The image shows two screenshots of a SQL IDE. The top screenshot displays the creation of a stored procedure named `mostrar_tabla_multiplicar`. The procedure takes an integer parameter `numero` and uses a `WHILE` loop to generate a multiplication table for that number. The `concat` function is used to format the output rows. The bottom screenshot shows the execution of the procedure with the argument `8`. The result grid displays the output of the `concat` function for the first row, showing the format `8x10=80`.

```
1 Use tablas
2 Drop procedure mostrar_tabla_multiplicar
3 DELIMITER //
4 CREATE PROCEDURE mostrar_tabla_multiplicar (in numero int)
5 BEGIN
6     DECLARE contador int;
7     set contador = 1;
8
9     WHILE contador <= 10 do
10         select concat(numero, 'x', contador, '=', (numero * contador));
11         SET contador = contador + 1;
12     END while;
13 END//
14 DELIMITER ;
15
16
```

SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 8\* SQL File 7\* SQL File 9\* SQL File 10\* mostrar\_tabla\_multiplicar x

```
1 call tablas.mostrar_tabla_multiplicar(8);
2
```

Result Grid Filter Rows Export: Wrap Cell Content: `concat(numero, 'x', contador, '=', (numero * contador))`

8x10=80
---------

Result 1 Result 2 Result 3 Result 4 Result 5 Result 6 Result 7 Result 8 Result 9 Result 10 x Read Only

EJERCICIO 4. 1 Crear un procedimiento que inserte en una tabla previamente creada de multiplicar de un número que se paso como entrada

```
SQL File 9*  SQL File 10*  SQL File 8*  SQL File 7*  SQL File 9* x  SQL File 10*  insertar_tabla_multiplicar  tabla1
Limit to 1000 rows
1 Use tablas
2 Drop procedure insertar_tabla_multiplicar
3 DELIMITER //
4 CREATE PROCEDURE insertar_tabla_multiplicar (in numero int)
5 BEGIN
6 DECLARE contador int;
7 set contador = 1;
8
9 WHILE contador <= 10 do
10 insert into tabla1 (numero1,numero2,resultado) values (numero, contador,(numero * contador));
11 SET contador = contador + 1;
12 END while;
13
14 END//
15 DELIMITER ;
16
17
```

```
SQL File 9*  SQL File 10*  SQL File 8*  SQL File 7*  SQL File 9*  SQL File 10*  insertar_tabla_multiplicar x  tabla1
Limit to 1000 rows
1 call tablas.insertar_tabla_multiplicar(4);
2
```

```
SQL File 9*  SQL File 10*  SQL File 8*  SQL File 7*  SQL File 9*  SQL File 10*  insertar_tabla_multiplicar  tabla1 x
Limit to 1000 rows
1 SELECT * FROM tablas.tabla1;
```

	numero1	numero2	resultado
4	1	4	
4	2	8	
4	3	12	
4	4	16	
4	5	20	
4	6	24	
4	7	28	
4	8	32	

tabla1 1 x

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Read Only

## EJERCICIO 4. 2

creamos un procedimiento que recibe un número entero y muestra los números anteriores hasta llegar al uno. (El número debe ser mayor que cero).

The image shows two screenshots of the SQL Server Enterprise Manager interface. The top screenshot displays the SQL Editor with the following code:

```
1 DELIMITER //
2 CREATE PROCEDURE restar (in numero int)
3 BEGIN
4
5 WHILE numero >= 1 do
6   select numero;
7   set numero = numero -1;
8 END while;
9
10 END//
```

The bottom screenshot shows the same interface after executing the code. The SQL Editor now contains:

```
1 call tablas.restar(5);
2
```

Below the editor, the 'Result Grid' is visible, showing a single row with the column name 'numero' and the value '1'. The 'Result Grid' tab is selected, and the 'Form Editor' is also visible on the right side of the interface.