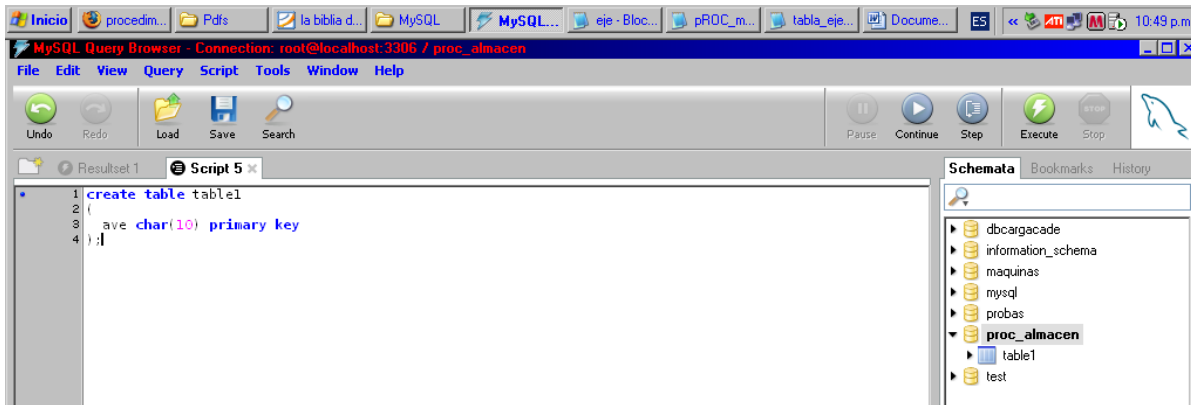


Creación De Procedimientos Almacenados En Mysql

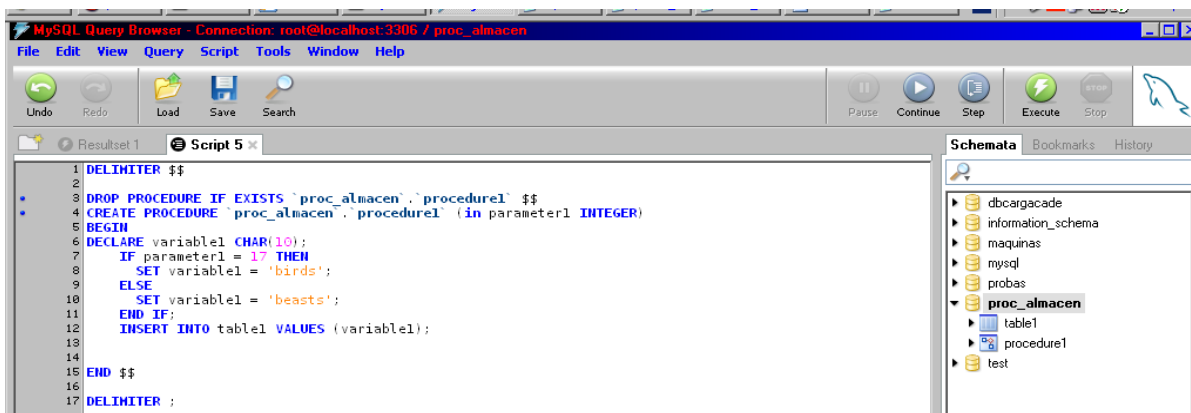
Se crea una tabla que servirá como **test** en la prueba de los procedimientos almacenados en mysql



Ver archivo:

tabla_eje.txt

Se introduce el procedimiento almacenado bajo la siguiente sintaxis:



Ver archivo:

pROC_mYsql

DELIMITER : por lo regular el manejador no puede interpretar la diferencia entre un query normal, y la ejecución de un procedimiento almacenado, una función o un trigger, por lo cual es necesario añadir un delimitador que le permita conocer el fin del procedimiento almacenado.

DROP PROCEDURE IF EXISTS `proc_almacen`.`procedure1` \$\$ Esta sentencia SQL elimina de la base de datos al procedimiento, antes de crearlo.

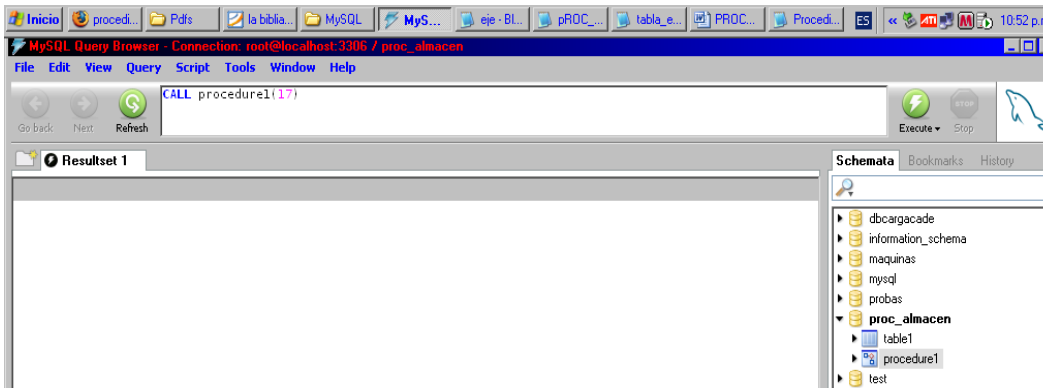
CREATE PROCEDURE `proc_almacen`.`procedure1` (in parameter1 INTEGER)
CREATE PROCEDURE ` nombre_base_de_Datos `.` nombre_proc_almacenado` (in parámetros [...])
Se usan las palabras CREATE PROCEDURE para crear el procedimiento, seguido del nombre de la base de datos entre `` , separada de ella por un punto e igual entre `` el nombre del procedimiento creado y entre paréntesis los parámetros que recibe.

BEGIN ... [y]... END \$\$ estas dos líneas encierran al cuerpo del procedimiento marcan el inicio y el fin de la codificación del mismo.

DECLARE; esta sentencia sirve para declara una variable local al procedimiento

Creación De Procedimientos Almacenados En Mysql

LLAMADA AL PROCEDIMIENTO ALMACENADO

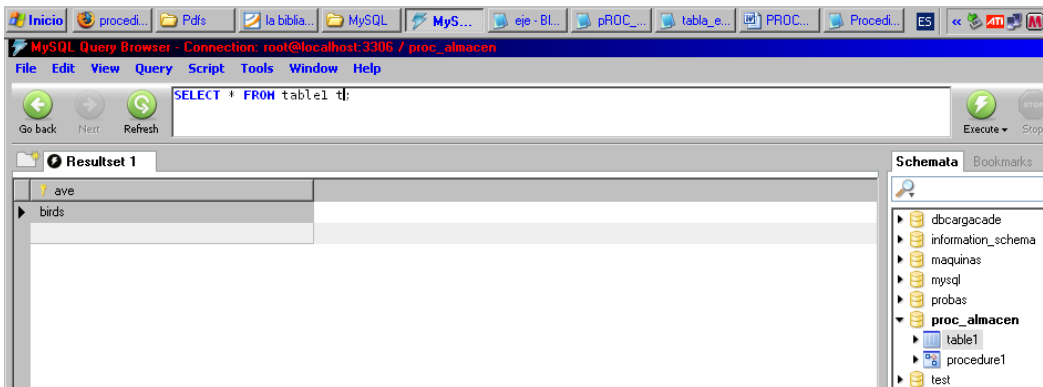


Para llamar al procedimiento almacenado, simplemente se ejecuta un query, en el que se usa la SIG. Sintaxis:

call [nombre](parámetros);

Si analizamos el cuerpo del procedimiento almacenado del ejemplo, podremos darnos cuenta que es bastante sencillo de comprender, al llamar al procedimiento, se le tiene que pasar un parámetro de tipo entero.

*En el cuerpo del procedimiento, se hace un control de flujo a través de una sentencia **IF**, si el parámetro introducido es **17**, se modifica a la variable declarada asignándole '**birds**', en caso de no cumplir esta condición la variable adquiere el valor de '**beast**', finalmente se hace una inserción a la tabla1 pasando el valor de la variable modificada.*

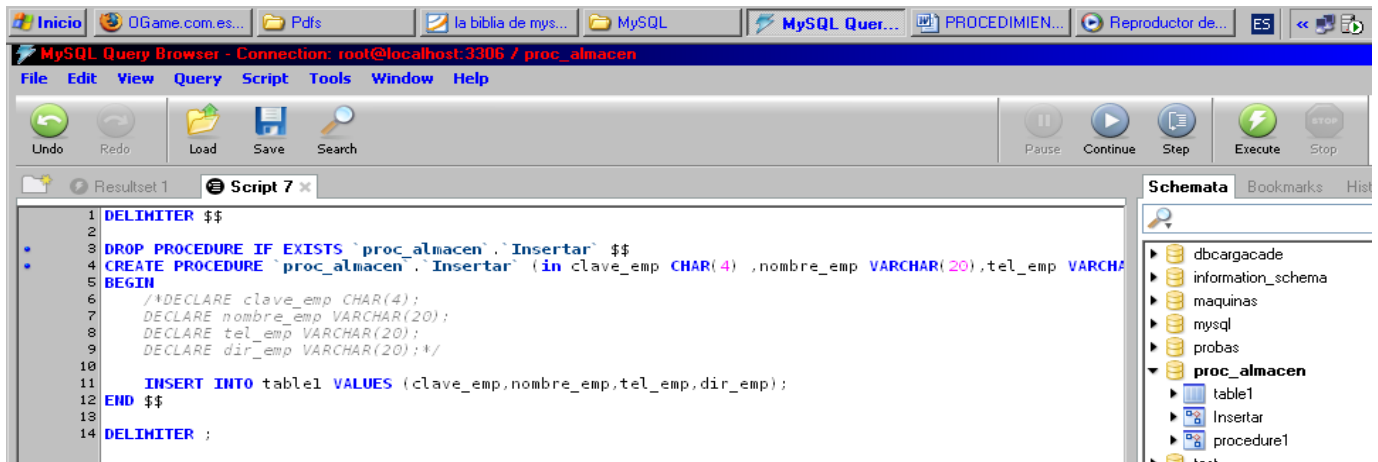


Resultados del query:

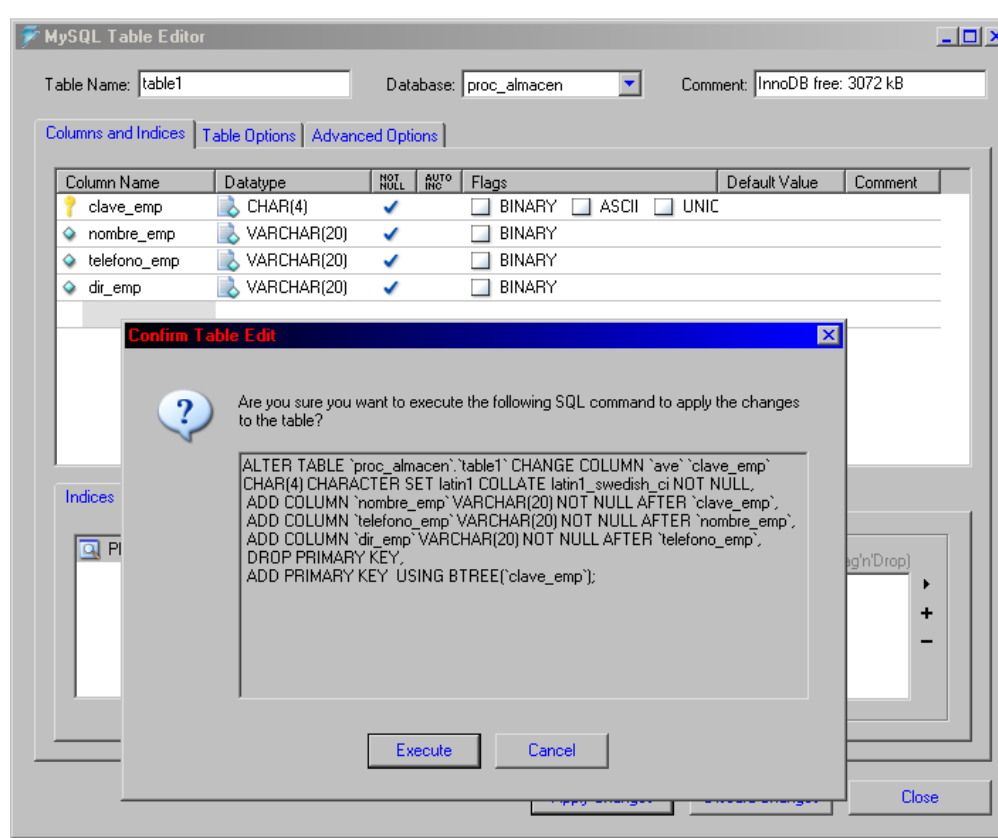
En esta consulta podemos ver, como efectivamente el procedimiento almacenado alteó a la tabla (tabla1), añadiendo un registro en ella, procedemos a hacer la llamada del procedimiento, pueden pasar dos cosas, si introducimos como parámetro otra vez el numero 17, nos mandará un error, pues existirá una duplicidad de llaves, pero si por el contrario insertamos un valor diferente a 17, el control de flujo llevará a realizar en la tabla el registro 'beast'.

Creación De Procedimientos Almacenados En Mysql

Otro Ejemplo utilizando Procedimientos Almacenados...

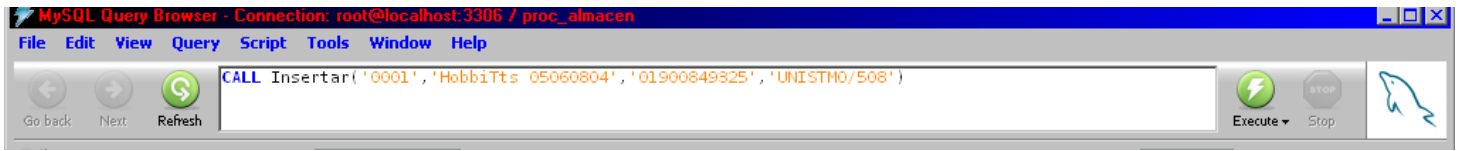


Ojo: para poder hacer la llamada de este procedimiento almacenado deberá de crear una nueva tabla con las columnas que se ocupan en la inserción, de lo contrario este procedimiento no surtirá ningún efecto sobre la base de datos.

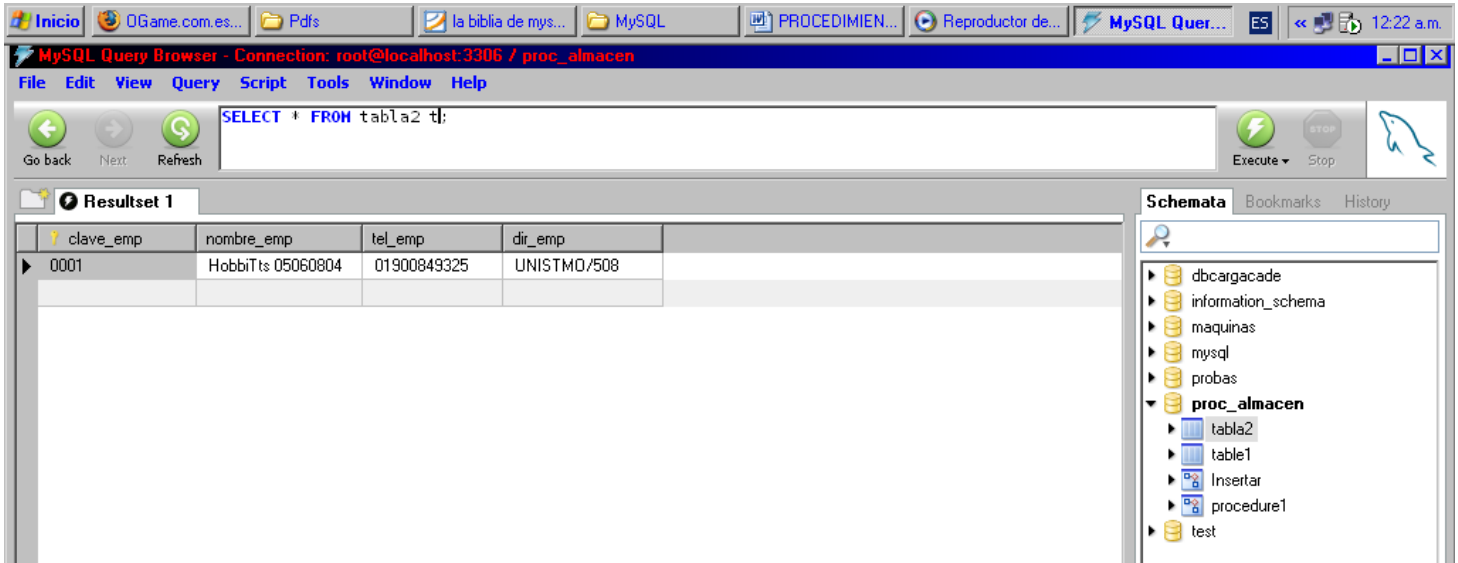


Creación De Procedimientos Almacenados En Mysql

Llamada al procedimiento Almacenado



Muestra los resultados en la tabla:



Ojo: de no querer modificar la primera tabla puedes crear la tabla dos usando el script “tabla2”.