MySQL CREATE PROCEDURE

?

Summary: in this tutorial, you will learn step by step how to the MySQL `CREATE PROCEDURE` statement to create new stored procedures.

MySQL `CREATE PROCEDURE` statement

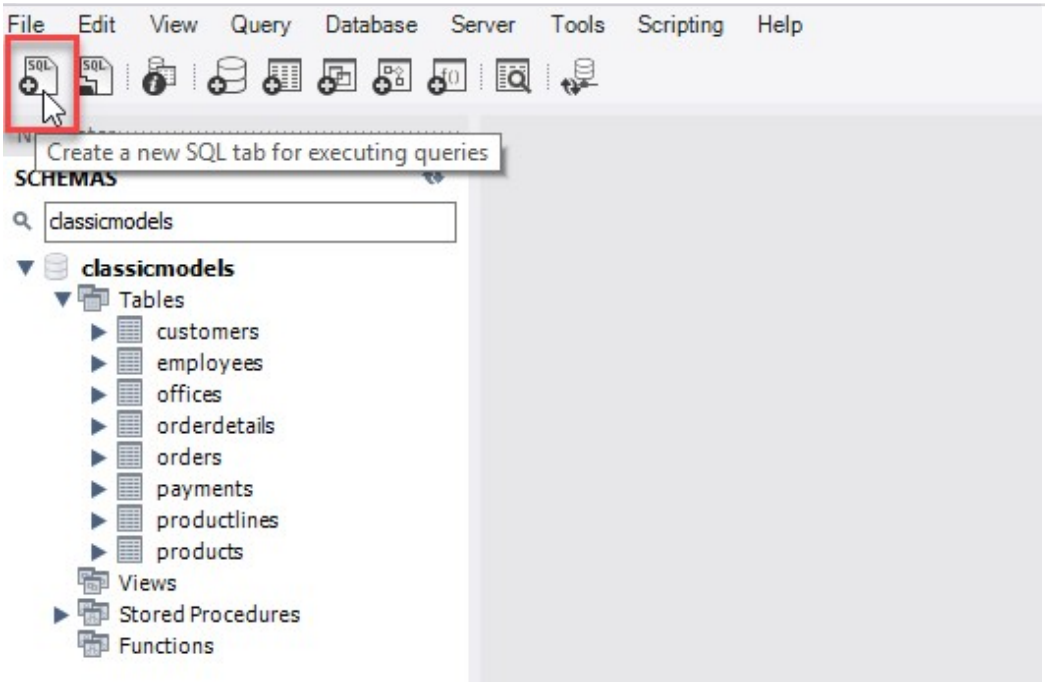
This query returns all products in the `products` table from the sample database.

```
1 SELECT * FROM products;
```

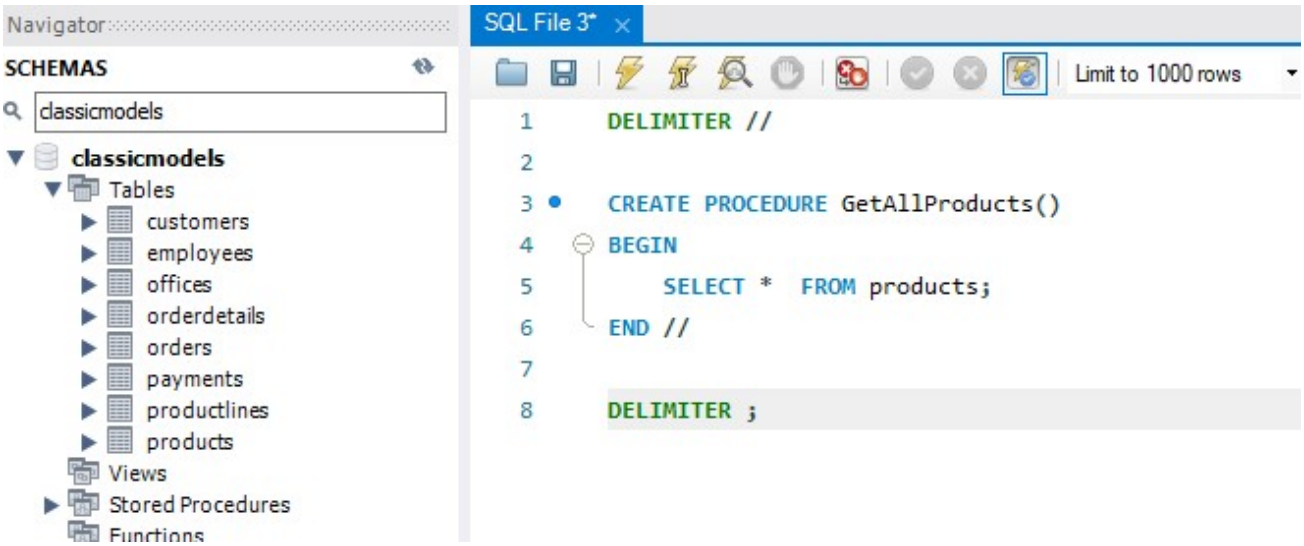
The following statement creates a new stored procedure that wraps the query:

```
1 DELIMITER //
2
3 CREATE PROCEDURE GetAllProducts()
4 BEGIN
5     SELECT *  FROM products;
6 END //
7
8 DELIMITER ;
```

To execute these statements:



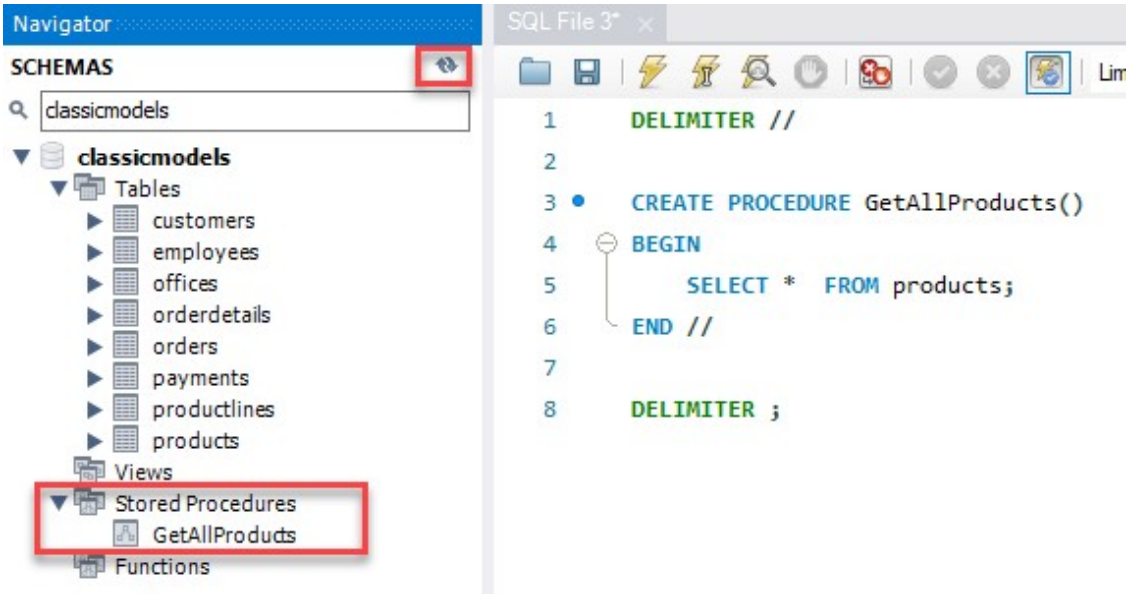
Third, enter the statements in the SQL tab:



Fouth, execute the statements. Note that you can select all statements in the SQL tab (or nothing) and click the **Execute** button. If everything is fine, MySQL will create the stored procedure and save it to the server.

```
5       SELECT *  FROM products;
6   END //
7
8   DELIMITER ;
```

Fifth, check the stored procedure by opening the **Stored Procedures** node. If you don't see the stored procedure, you can click the Refresh button next to the **SCHEMAS** title:



Congratulation! you have successfully created the first stored procedure in MySQL.

Let's examine the syntax of the stored procedure.

The first and last `DELIMITER` commands are not a part of the stored procedure. The first `DELIMITER` command changes the default delimiter to `//` and the last `DELIMITER` command changes the delimiter back to the default one which is semicolon (;).

```
BEGIN
   statements;
END //
```

In this syntax

First, specify the name of the stored procedure that you want to create after the `CREATE PROCEDURE` keywords.

Second, specify a list of comma-separated parameters for the stored procedure in parentheses after the procedure name.

Third, write the code between the `BEGIN END` block. The above example just has a simple `SELECT` statement. After the `END` keyword, you place the delimiter character to end the procedure statement.

Executing a stored procedure

To execute a stored procedure, you use the `CALL` statement:

```
CALL stored_procedure_name(argument_list);
```

In this syntax, you specify the name of the stored procedure after the `CALL` keyword. If the stored procedure has parameters, you need to pass arguments inside parentheses following the stored procedure name.

This example illustrates how to call the `GetAllProducts()` stored procedure:

```
CALL GetAllProducts();
```

Executing this statement is the same as executing an SQL statement:

```
5       SELECT *  FROM products;
6   END //
7
8   DELIMITER ;
9
10  CALL GetAllProducts();
```

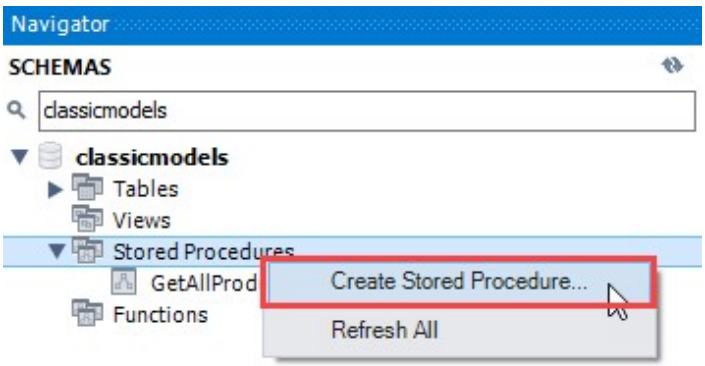
Here is the partial output:

productCode	productName	productLine	productScale	productVendor
S10_1678	1969 Harley Davidson Ultimate Chopper	Motorcycles	1:10	Min Lin Diecast
S10_1949	1952 Alpine Renault 1300	Classic Cars	1:10	Classic Metal Creations
S10_2016	1996 Moto Guzzi 1100i	Motorcycles	1:10	Highway 66 Mini Classics
S10_4698	2003 Harley-Davidson Eagle Drag Bike	Motorcycles	1:10	Red Start Diecast
S10_4757	1972 Alfa Romeo GTA	Classic Cars	1:10	Motor City Art Classics
S10_4962	1962 LanciaA Delta 16V	Classic Cars	1:10	Second Gear Diecast
S12_1099	1968 Ford Mustang	Classic Cars	1:12	Autoart Studio Design
S12_1108	2001 Ferrari Enzo	Classic Cars	1:12	Second Gear Diecast

Creating a stored procedure using the MySQL Workbench wizard

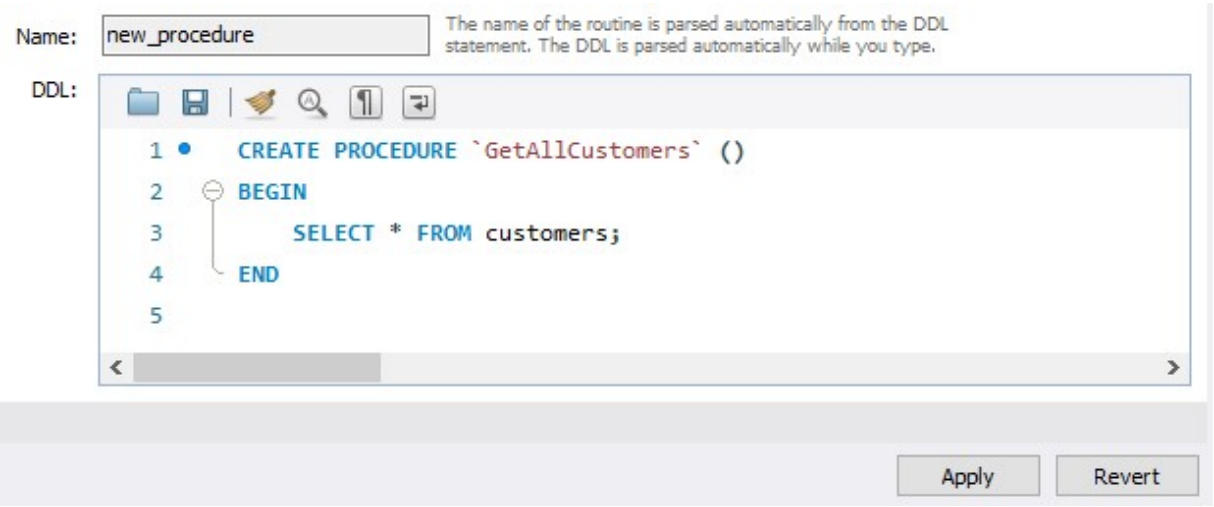
By using the MySQL Workbench wizard, you don't have to take are of many things like delimiters or executing the command to create stored procedures.

First, right-click on the **Stored Procedures** from the **Navigator** and select the **Create Stored Procedure...** menu item.



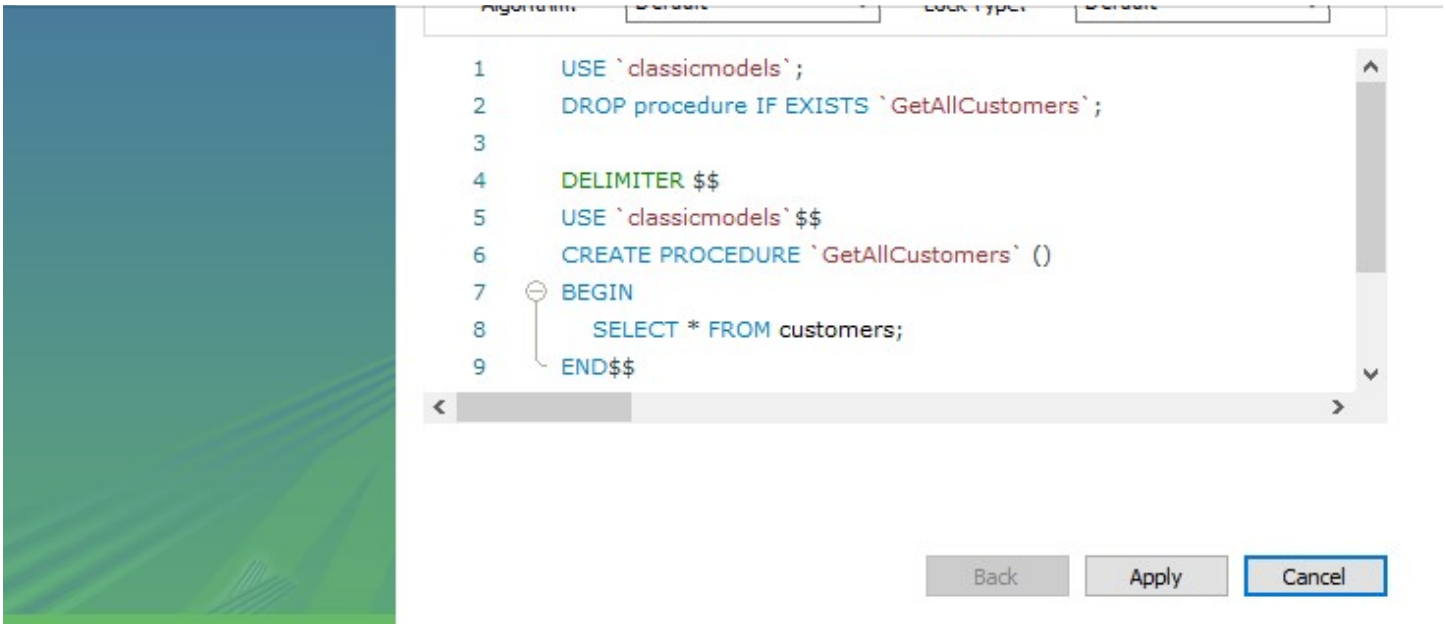
```
1  CREATE PROCEDURE `new_procedure` ()
2  BEGIN
3
4  END
5
```

Second, change the stored procedure's name and add the code between the `BEGIN END` block:

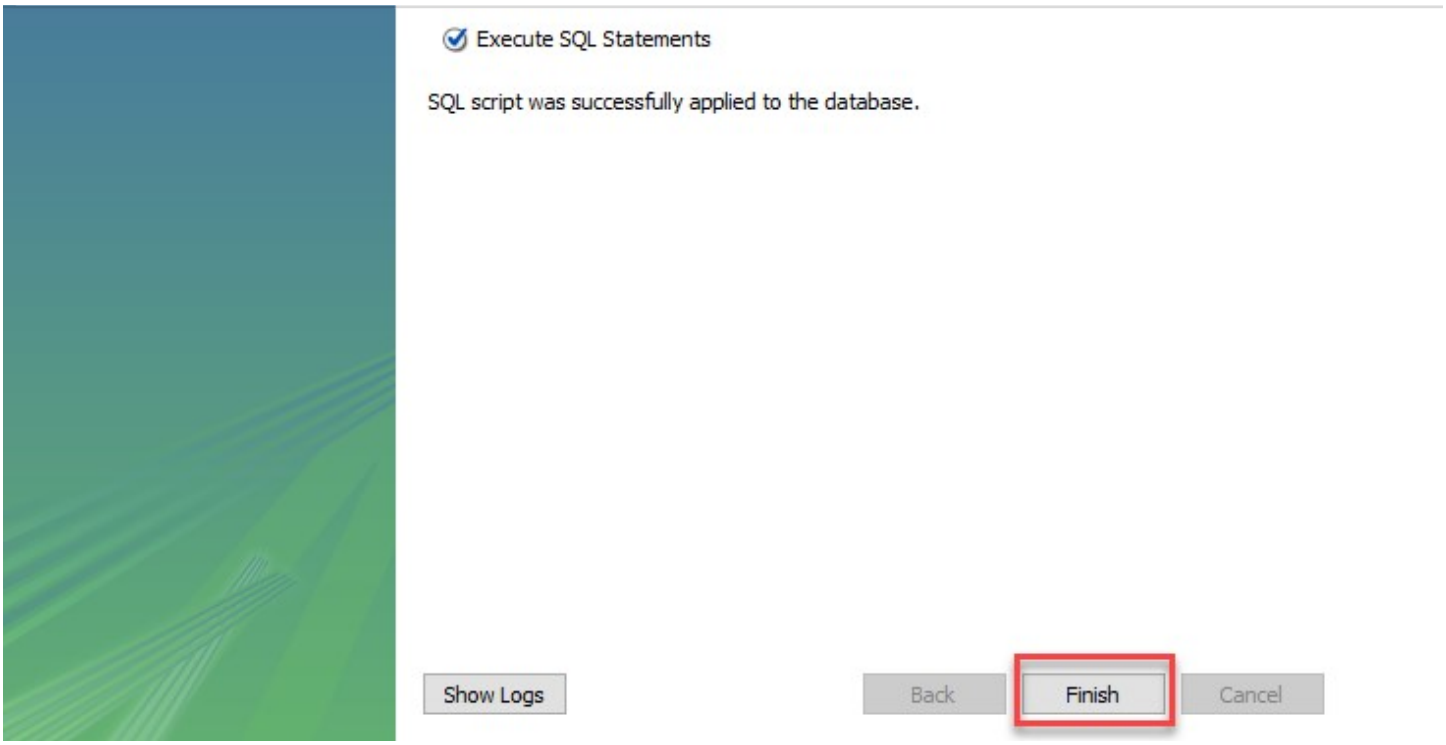


The stored procedure name is `GetAllCustomers()` which returns all rows in the `customers` table from the sample database.

Third, Click the **Apply** button, MySQL Workbench will open a new window for reviewing SQL script before applying it on the database:

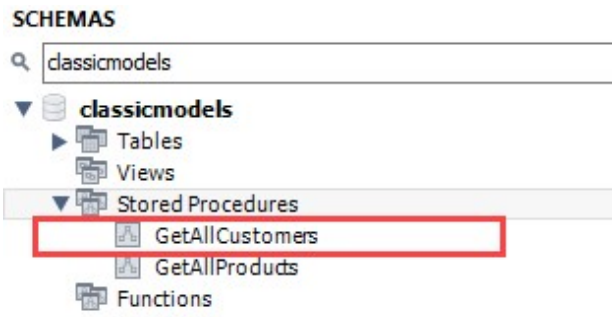


Fourth, Click the **Apply** button to confirm. MySQL Workbench will create the stored procedure:



Fifth, click the **Finish** button to close the window.

Finally, view the stored procedure in the **Stored Procedures** list:



In this tutorial, you have learned how to use the MySQL `CREATE PROCEDURE` statement to create new stored procedures in the database.

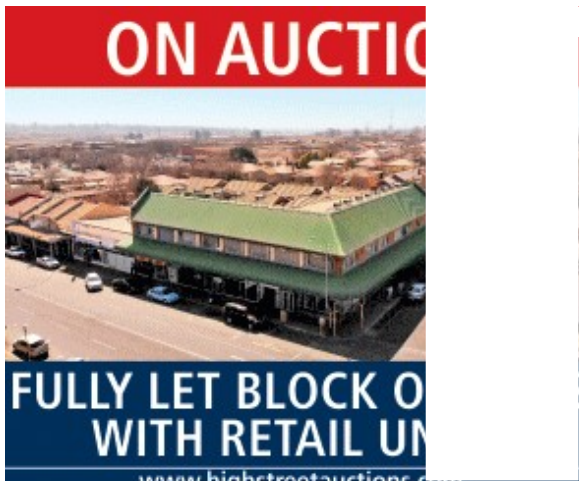
« Previous Tutorial:
Introduction to MySQL Stored Procedures

Next Tutorial:
MySQL Delimiter »

Search this website

MYSQL QUICK START

Load Sample Database



MYSQL STORED PROCEDURES

Getting Started with Stored Procedures

Changing MySQL Delimiter

Creating Stored Procedures

Removing Stored Procedures

Modifying Stored Procedures

Listing Stored Procedures

Variables

Parameters

IF THEN

CASE Statement

LOOP

WHILE

MYSQL PROGRAMMING INTERFACES

PHP MySQL Tutorial

Node.js MySQL Tutorial

Python MySQL Tutorial

Perl MySQL Tutorial

MySQL JDBC Tutorial

RECENT MYSQL TUTORIALS

MySQL Stored Object Access Control

Install MySQL on Ubuntu

Install MySQL CentOS

Getting Started with MySQL

Connect to MySQL Server

MySQL Books

Start MySQL Server

Stop MySQL Server

ABOUT MYSQL TUTORIAL WEBSITE

MySQLTutorial.org is a website dedicated to MySQL database. We regularly publish useful MySQL tutorials to help web developers and database administrators learn MySQL faster and more effectively.

All MySQL tutorials are practical and easy-to-follow, with SQL script and screenshots available. More About Us

SITE LINKS

About Us

Contact Us

Request a Tutorial

Privacy Policy