# MySQL Stored Procedures (Create, List, Alter, & Drop)

Introduction

MySQL stored procedures group multiple tasks into one and save the task on the server for future use.

Stored procedures simplify database management and reduce network traffic. For example, issuing a query to the MySQL server processes the query and returns the results. Using stored procedures saves the queries on the server so they can be executed later.

In this tutorial, you will learn to create, list, alter, and drop stored procedures.

Prerequisites

* MySQL Server and MySQL Workbench installed
* A MySQL user account with root privileges

## What Are Stored Procedures in MySQL?

MySQL stored procedures are pre-compiled SQL statements stored in a database. They are subroutines containing a name, a parameter list, and SQL statements.

All [relational database systems](https://phoenixnap.com/kb/what-is-a-relational-database) support stored procedures and do not require any additional runtime-environment packages.

## How to Use Stored Procedures?

To invoke stored procedures, you can use the CALL statement or other stored procedures. The first time a stored procedure is invoked, MySQL looks it up in the database catalog, compiles the code, places it in the cache memory, and executes it.

Subsequent runs in the same session execute stored procedures from the cache memory, making them extremely useful for repetitive tasks.

Stored procedures make use of parameters to pass values and customize results. Parameters are used to specify the columns in a table in which the query operates and returns results.

Stored procedures can also include the IF, CASE, and LOOP control flow statements that procedurally implement the code.

### Create Stored Procedure

Create a stored procedure in two ways:

1. Use MySQL Shell

Use the following syntax to create a stored procedure in MySQL:

DELIMITER //

CREATE PROCEDURE procedure\_name ( IN | OUT | INOUT parameter\_name parameter\_datatype (length), … )

BEGIN

SQL statements

END //

DELIMITER ;

**By default, the syntax is associated with the database in use, but you can also use the syntax for another database by specifying the database name in the following way:** database\_name.procedure\_name.

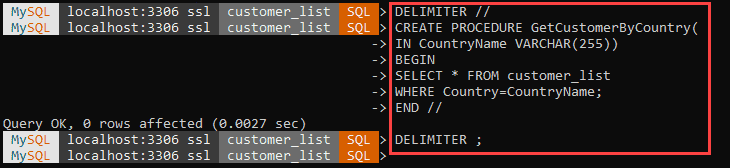
Here, the first DELIMITER argument sets the default delimiter to //, while the last DELIMITER argument sets it back to the semicolon ;. To use multiple statements, specify different delimiters like $$.

The procedure name comes after the CREATE PROCEDURE argument. After the procedure name, use parenthesis to specify the parameters to use in the procedure, the name of the parameter, the data type, and data length. Separate each parameter with a comma.

The parameter modes are:

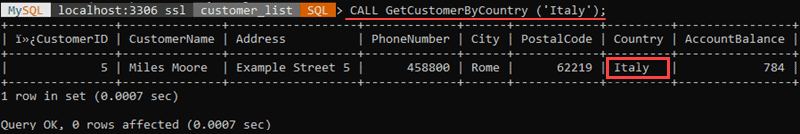
* IN – Use to pass a parameter as input. When it is defined, the query passes an argument to the stored procedure. The value of the parameter is always protected.
* OUT – Use to pass a parameter as output. You can change the value within the stored procedure, and the new value is passed back to the calling program.
* INOUT – A combination of IN and OUT parameters. The calling program passes the argument, and the procedure can modify the INOUT parameter, passing the new value back to the program.

For example:



Execute the stored procedure by calling it:

CALL procedure\_name;



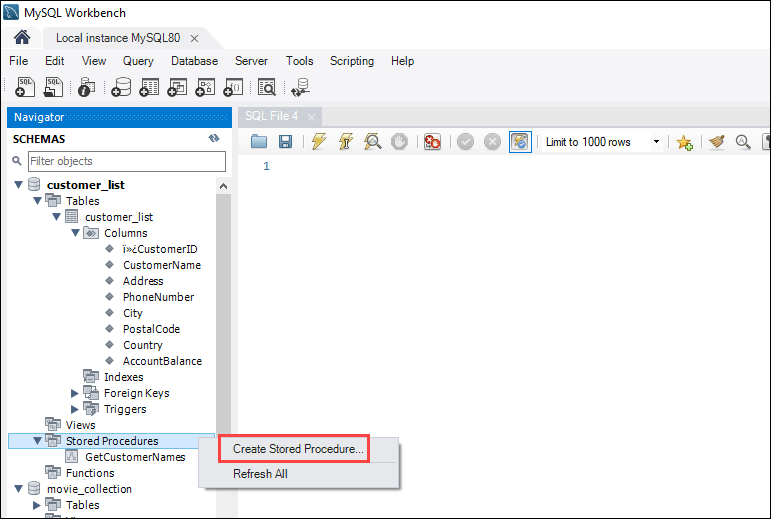
The query returns results for the stored procedure.

2. Use MySQL Workbench

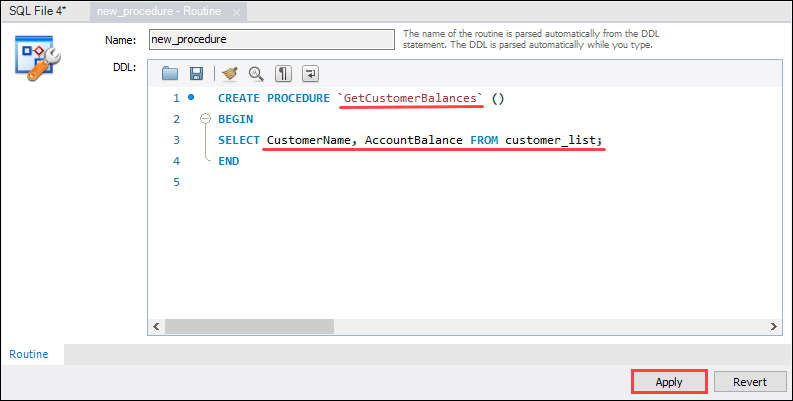
Another way to create a stored procedure is to use the [MySQL Workbench Wizard](https://phoenixnap.com/kb/install-get-started-mysql-workbench-on-ubuntu). The wizard is intuitive and simplifies the process since you do not have to place delimiters or worry about the format.

Follow these steps:

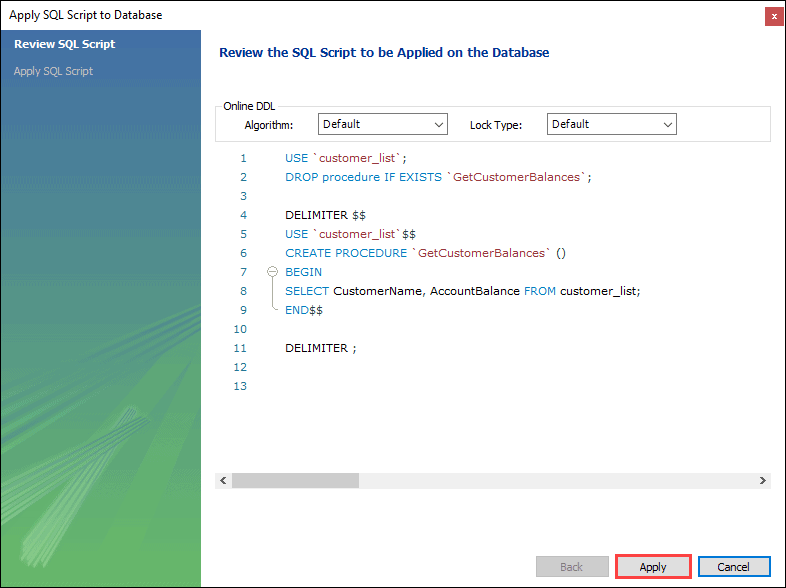
Step 1: Right-click Stored Procedures in the Navigator window of MySQL Workbench and choose Create Stored Procedure… to start the wizard.



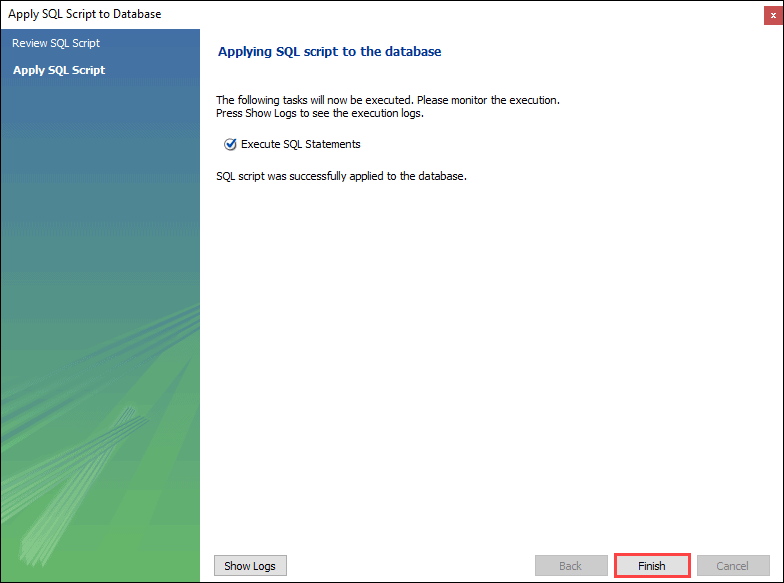
Step 2: Specify the procedure name and enter the code within the BEGIN … END block.



Step 3: Review the code and click Apply.

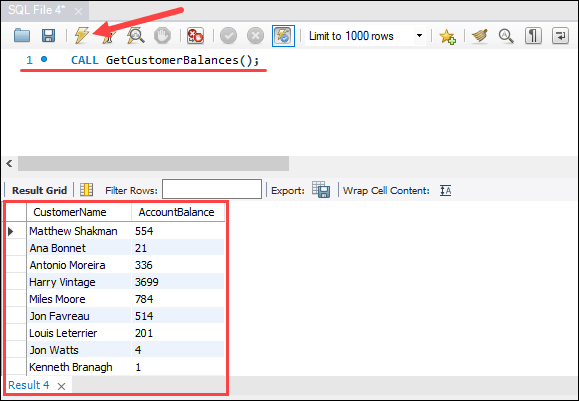


Step 4: Confirm execution by clicking Apply and create the procedure by clicking Finish.



Step 5: Execute the procedure to see if it works. Create a new SQL tab for executing queries.

Step 6: CALL the procedure in the SQL tab and click Execute.



If no errors return, MySQL executes the stored procedure and displays the results.

### List Stored Procedures

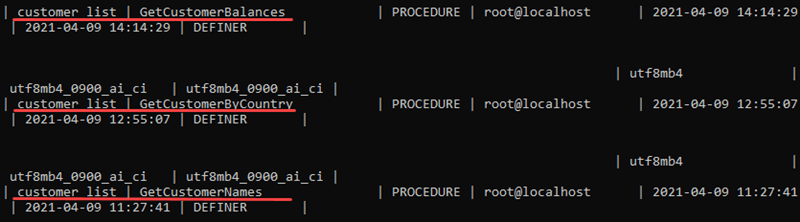
There are three ways to see a list of all stored procedures:

1. Use MySQL Shell

To get a list of all stored procedures you have access to, including their characteristics, use the following syntax:

SHOW PROCEDURE STATUS [LIKE 'pattern' | WHERE search\_condition]

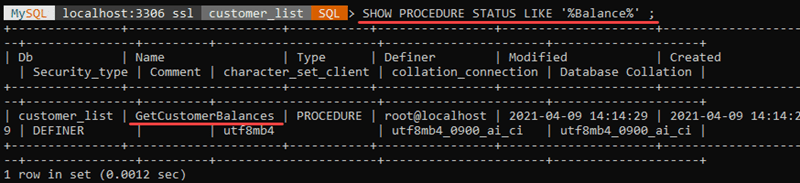
The SHOW PROCEDURE STATUS statement returns a lengthy output. The statement displays the names and characteristics of stored procedures that you have access to on the server.



Scroll through the output to find the procedures currently on the server.

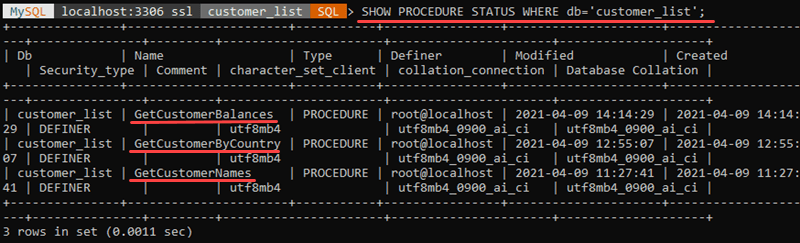
The LIKE argument finds stored procedures containing a specific word in their name. Use % to replace any number of characters, including zero.

For example:



The WHERE argument allows you to list stored procedures only in a particular database.

For example:



In this example, the statement returns only the stored procedures for the ’customer\_list’ database.

2. Use Data Dictionary

The information\_schema database contains a table called routines, which has information on stored procedures and functions related to all databases on the current MySQL server.

Use the following syntax to see all stored procedures for a database:

SELECT

routine\_name

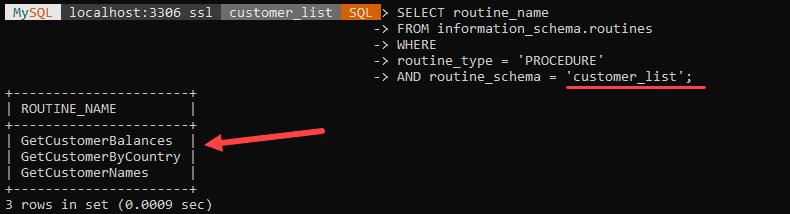
FROM

information\_schema.routines

WHERE

routine\_type = 'PROCEDURE'

AND routine\_schema = 'database\_name';

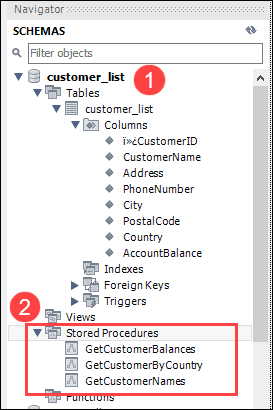


3. Use MySQL Workbench

For a GUI approach to viewing stored procedures, use MySQL Workbench. Follow these steps to see stored procedures:

Step 1: Double-click the database you want to use in the Navigator section.

Step 2: Expand the Stored Procedures drop-down item.



This item shows all stored procedures for the current database.

### Alter Stored Procedure

To alter a stored procedure means to change the characteristics of a procedure. There is no statement in MySQL for modifying the parameters or the body of a stored procedure. To change parameters or the body, drop the stored procedure and create a new one.

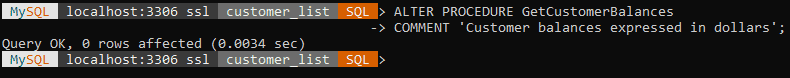
Alter a stored procedure in two ways:

1. Use MySQL Shell

Change a procedure characteristic by using the ALTER PROCEDURE statement. For example, we can add a comment to a procedure we created previously. The syntax is:

ALTER PROCEDURE procedure\_name

COMMENT 'Insert comment here';

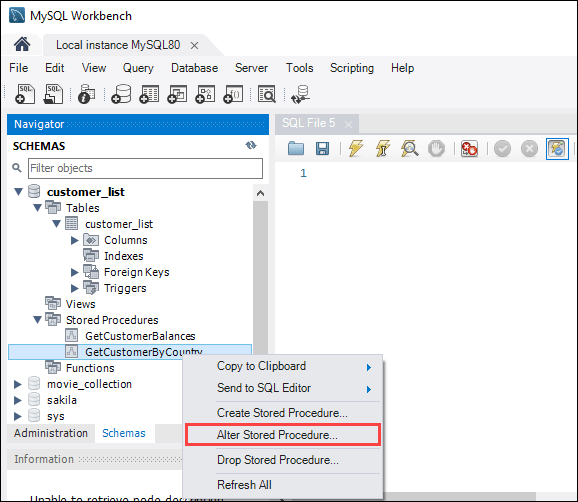


2. Use MySQL Workbench

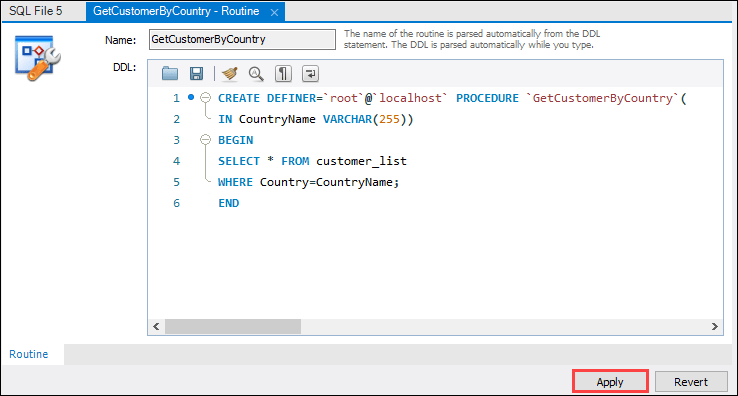
MySQL Workbench GUI allows users to alter a stored procedure where users can add parameters or change the code. MySQL Workbench drops the existing stored procedure and creates a new one after the changes have been made.

Follow these steps:

Step 1: In the Navigator section, right-click the stored procedure you want to modify. Select the Alter Stored procedure… item.

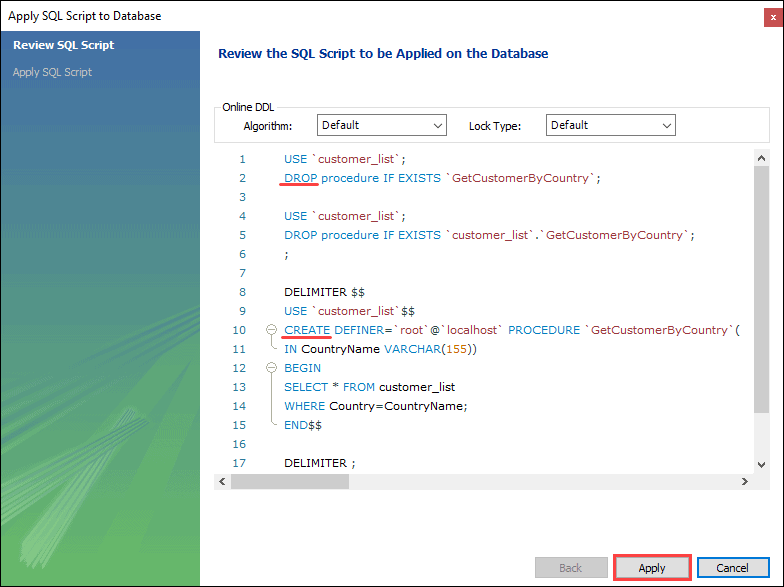


Step 2: When the tab opens, make the desired changes to the existing stored procedure and click Apply.



Step 3: An SQL Script review window appears showing the process – dropping the existing stored procedure and creating a new one containing the changes.

Click Apply and then Finish in the next window to execute the script.



### Drop Stored Procedure

To drop (delete) a procedure:

1. Use MySQL Shell

Delete a stored procedure from the server by using the DROP PROCEDURE statement.

The basic syntax is:

DROP PROCEDURE [IF EXISTS] stored\_procedure\_name;

The IF EXISTS parameter drops the stored procedure only if it exists on the server. Enter the name of the stored procedure in place of the stored\_procedure\_name syntax.

For example:



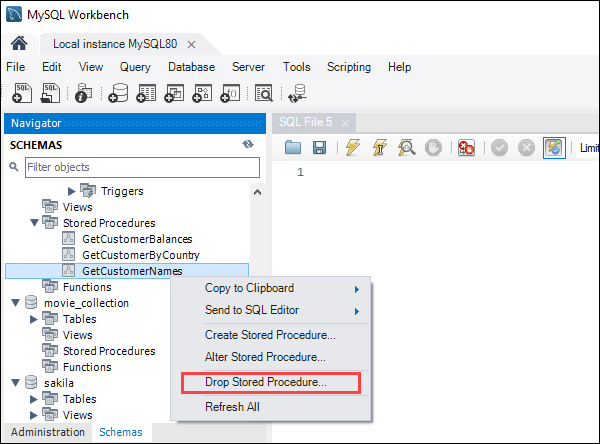
Since there is no procedure named ‘test’ on the server, the output states that 0 rows were affected and that the specified procedure does not exist.

Dropping a non-existing procedure without the IF EXISTS parameter returns an error.

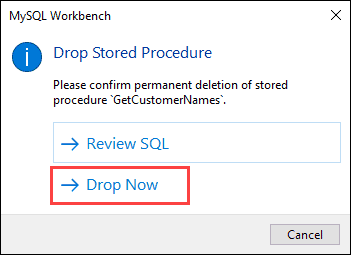
2. Use MySQL Workbench

To drop a stored procedure with MySQL Workbench, follow these steps:

Step 1: Expand the Stored Procedures item in the Navigator section. Right-click the stored procedure you want to delete and choose Drop Stored Procedure… in the context menu.



Step 2: In the confirmation window, click Drop Now to delete the stored procedure.



This action permanently deletes the procedure.

## MySQL Stored Procedures Advantages and Disadvantages

Stored procedures have several advantages and disadvantages as they tailor to specific needs. Below are some of the advantages and disadvantages.

### Advantages of Using Stored Procedures

The advantages of stored procedures are:

Network Traffic Reduction

Stored procedures help reduce the network traffic between applications and MySQL Server by keeping all the programming logic on the server. Instead of sending multiple query results across the network, apps send only the procedure name and the parameter input.

Improved Security

The database administrator grants apps privileges to call and access only specific stored procedures without giving them direct access to tables. Stored procedures help prevent script injection attacks since input parameters are treated as values and not as executable code.

Centralized Business Logic

Stored procedures encapsulate the business logic reusable by multiple applications. That helps reduce duplicating that same logic in many different applications and makes the database more consistent.

### Disadvantages of Using Stored Procedures

The disadvantages of stored procedures are:

Resource Usage

Using many stored procedures and logical operations causes the memory and [CPU usage](https://phoenixnap.com/kb/check-cpu-usage-load-linux) to increase significantly for every connection.

No Portability

It is not easy to port stored procedures written in a specific language from one installation to another. Relying on a stored procedure also ties the user to a particular database.

Troubleshooting and Testing

MySQL does not provide utilities for testing and debugging stored procedures, so it can be difficult to debug them. Developing and maintaining stored procedures require extensive knowledge. This is a challenge for new developers and results in added maintenance costs.

Conclusion

After reading this article, you know what stored procedures are and when to use them. You also know how to create, modify, see all available stored procedures, and delete the ones you no longer need.

<https://phoenixnap.com/kb/mysql-stored-procedure>