

[HOME](#)[START HERE](#)[BASIC](#)[ADVANCED](#) ▼[FUNCTIONS](#) ▼[INTERFACES](#) ▼[TIPS](#)[TRYIT](#)

Python MySQL – Call Stored Procedures in Python

?

Summary: in this tutorial, you will learn how to call MySQL stored procedures in Python by using MySQL Connector/Python API.

Before we start

If you are not familiar with MySQL stored procedures or want to review it as a refresher, you can follow the [MySQL stored procedures tutorial](#).

We will [create two stored procedures](#) for the demonstration in this tutorial.

The following stored procedure returns all books with authors information from [books](#) and [authors](#) tables:

```
1 USE python_mysql;
2
3 DELIMITER $$
4
5 CREATE PROCEDURE find_all()
6 BEGIN
7     SELECT
8         title,
```

[HOME](#)[START HERE](#)[BASIC](#)[ADVANCED](#) ▾[FUNCTIONS](#) ▾[INTERFACES](#) ▾[TIPS](#)[TRYIT](#)

```
16 ORDER BY title;
17 END$$
18
19 DELIMITER ;
```

The `find_all()` stored procedure has a `SELECT` statement with `INNER JOIN` clauses that retrieve title, isbn and author's full name from `books` and `authors` tables.

Calling the `find_all()` stored procedure, you will get the following result set:

```
1 CALL find_all();
```

	title	isbn	author
▶	A History of Nebraska	1239609581078	Clinton York
	Anemones of the British Coast	1233540471995	Francis Tugwell
	Ask to Embla poem-cycle	1237417184084	Randolph Henry Ash
	Bacon Death	1236766330719	Marsha Patterson
	Bel and the Dragon	1239374496485	Herbert Methley
	Bel and the Dragon	123828863494	Herbert Methley
	Breakfast First	1236432913317	Samuel Humber
	Cassandra verse drama	1235260611012	Randolph Henry Ash
	Chidiok Tichbourne	1230468662299	Randolph Henry Ash
	Cromwell verse drama	1239653041219	Randolph Henry Ash

The second stored procedure called `find_by_isbn()` is used to find a book by its ISBN as follows:

```
1 DELIMITER $$
2
3 CREATE PROCEDURE find_by_isbn(
4     IN p_isbn VARCHAR(13),
5     OUT p_title VARCHAR(255)
6 )
7 BEGIN
8     SELECT title
9     INTO p_title
10    FROM books
11    WHERE isbn = p_isbn;
12 END$$
```

[HOME](#)[START HERE](#)[BASIC](#)[ADVANCED](#) ▼[FUNCTIONS](#) ▼[INTERFACES](#) ▼[TIPS](#)[TRYIT](#)

for example:

```
1 CALL find_by_isbn('1235927658929',@title);
2 SELECT @title;
```

Here is the output:

```
1 +-----+
2 | @title |
3 +-----+
4 | Debatable Land Between This World and the Next |
5 +-----+
6 1 row in set (0.00 sec)
```

Calling stored procedures from Python

To call a stored procedure in Python, you follow the steps below:

1. Connect to the database by creating a new `MySQLConnection` object.
2. Instantiate a new `MySQLCursor` object from the `MySQLConnection` object by calling the `cursor()` method.
3. Call `callproc()` method of the `MySQLCursor` object. You pass the stored procedure's name as the first argument of the `callproc()` method. If the stored procedure requires parameters, you need to pass a list as the second argument to the `callproc()` method. In case the stored procedure returns a result set, you can invoke the `stored_results()` method of the `MySQLCursor` object to get a list iterator and iterate this result set by using the `fetchall()` method.
4. Close the cursor and database connection as always.

The following example demonstrates how to call the `find_all()` stored procedure in Python and output the result set.

```
1 from mysql.connector import MySQLConnection, Error
2 from python_mysql_dbconfig import read_db_config
3
```

[HOME](#)[START HERE](#)[BASIC](#)[ADVANCED ▾](#)[FUNCTIONS ▾](#)[INTERFACES ▾](#)[TIPS](#)[TRYIT](#)

```
11     cursor.callproc('find_all')
12
13     # print out the result
14     for result in cursor.stored_results():
15         print(result.fetchall())
16
17 except Error as e:
18     print(e)
19
20 finally:
21     cursor.close()
22     conn.close()
23
24
25 if __name__ == '__main__':
26     call_find_all_sp()
```

The following example shows you how to call the `find_by_isbn()` stored procedure.

```
1 from mysql.connector import MySQLConnection, Error
2 from python_mysql_dbconfig import read_db_config
3
4
5 def call_find_by_isbn():
6     try:
7         db_config = read_db_config()
8         conn = MySQLConnection(**db_config)
9         cursor = conn.cursor()
10
11         args = ['1236400967773', 0]
12         result_args = cursor.callproc('find_by_isbn', args)
13
14         print(result_args[1])
15
16 except Error as e:
17     print(e)
18
19 finally:
20     cursor.close()
21     conn.close()
22
23
```

[HOME](#)[START HERE](#)[BASIC](#)[ADVANCED](#) ▼[FUNCTIONS](#) ▼[INTERFACES](#) ▼[TIPS](#)[TRYIT](#)

second element of the args list (0) is just a placeholder to hold the `p_title` parameter.

The `callproc()` method returns a list (`result_args`) that contains two elements: the second element (`result_args[1]`) holds the value of the `p_title` parameter.

In this tutorial, you have learned how to call stored procedures in Python by using `callproc()` method of the `MySQLCursor` object.

Was this tutorial helpful?

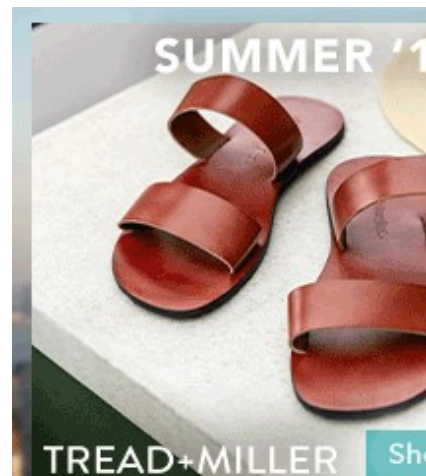


Previous Tutorial:

[Python MySQL – Delete Data from a Table](#)

Next Tutorial:

[Python MySQL – Read & Update BLOB in MySQL Database](#)



[HOME](#)[START HERE](#)[BASIC](#)[ADVANCED](#) ▼[FUNCTIONS](#) ▼[INTERFACES](#) ▼[TIPS](#)[TRY IT](#)[Connect to MySQL Server](#)[Download MySQL Sample Database](#)[Load Sample Database](#)

Why Choose Postgres



In this whitepaper, we'll explore where PostgreSQL makes sense and some places where it's not the ideal choice.

PostgreSQL is popular with developers for its numerous features like logical replication, simple read balancing, commit timestamp, and full-text search support for JSON.

[Learn More](#)

PYTHON MYSQL TUTORIAL

[Python MySQL Getting Started](#)[Python MySQL Connect](#)[Python MySQL Select](#)[Python MySQL Insert](#)[Python MySQL Update](#)[Python MySQL Delete](#)[Python MySQL BLOB](#)[Python MySQL Stored Procedures](#)

[HOME](#)[START HERE](#)[BASIC](#)[ADVANCED](#) ▾[FUNCTIONS](#) ▾[INTERFACES](#) ▾[TIPS](#)[TRYIT](#)[Perl MySQL Tutorial](#)[MySQL JDBC Tutorial](#)[OTHERS](#)[MySQL Cheat Sheet](#)[MySQL Resources](#)[MySQL Books](#)[Getting Started with MySQL](#)[Connect to MySQL Server](#)[How To Lock User Accounts in MySQL](#)[How To Unlock User Accounts in MySQL Server](#)[Restart MySQL Server](#)[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](#)

[HOME](#)

[START HERE](#)

[BASIC](#)

[ADVANCED](#) ▼

[FUNCTIONS](#) ▼

[INTERFACES](#) ▼

[TIPS](#)

[TRYIT](#)
