[MYSQL TUTORIAL](http://www.mysqltutorial.org/)

* [HOME](http://mysqltutorial.org/)
* [BASIC MYSQL](http://www.mysqltutorial.org/basic-mysql-tutorial.aspx)
* [ADVANCED MYSQL](http://www.mysqltutorial.org/mysql-triggers/create-multiple-triggers-for-the-same-trigger-event-and-action-time/)
  + [MySQL Stored Procedures](http://www.mysqltutorial.org/mysql-stored-procedure-tutorial.aspx)
  + [MySQL Views](http://www.mysqltutorial.org/mysql-views-tutorial.aspx)
  + [MySQL Triggers](http://www.mysqltutorial.org/mysql-triggers.aspx)
  + [MySQL Administration](http://www.mysqltutorial.org/mysql-administration.aspx)
  + [MySQL Full-Text Search](http://www.mysqltutorial.org/mysql-full-text-search.aspx)
  + [MySQL Functions](http://mysqltutorial.org/mysql-functions.aspx)
* [INTERFACES](http://www.mysqltutorial.org/mysql-triggers/create-multiple-triggers-for-the-same-trigger-event-and-action-time/)
  + [MySQL PHP](http://www.mysqltutorial.org/php-mysql/)
  + [MySQL Node.js](http://www.mysqltutorial.org/mysql-nodejs/)
  + [MySQL JDBC](http://www.mysqltutorial.org/mysql-jdbc-tutorial/)
  + [MySQL Python](http://www.mysqltutorial.org/python-mysql/)
  + [MySQL Perl](http://www.mysqltutorial.org/perl-mysql/)
* [TIPS](http://www.mysqltutorial.org/mysqltips.aspx)
* [TRYIT](http://mysqltutorial.org/tryit/)

[Home](http://www.mysqltutorial.org/) / [MySQL Triggers](http://www.mysqltutorial.org/mysql-triggers.aspx) / Create Multiple Triggers For The Same Trigger Event And Action Time

Create Multiple Triggers For The Same Trigger Event And Action Time

**Summary**: in this tutorial, you will learn how to create multiple triggers for the same event and action time in MySQL.

This tutorial is relevant to the MySQL version 5.7.2+. If you have an older version of MySQL, the statements in the tutorial will not work.

Before MySQL version 5.7.2, you can only create one trigger for an event in a table e.g., you can only create one trigger for the BEFORE UPDATE or AFTER UPDATE event. MySQL 5.7.2+ lifts this limitation and allows you to create multiple triggers for the same event and action time in a table. The triggers will activate sequentially when the event occurs.

[The syntax for creating the first trigger](http://mysqltutorial.org/create-the-first-trigger-in-mysql.aspx) remains the same. In case you have multiple triggers for the same event in a table, MySQL will invoke the triggers in the order that they were created. To change the order of triggers, you need to specify FOLLOWS or PRECEDES after the FOR EACH ROW clause.

* The FOLLOWS   option allows the new trigger to activate after the existing trigger.
* The PRECEDES  option allows the new trigger to activate before the existing trigger.

The following is the syntax of creating a new additional trigger with explicit order:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | DELIMITER $$  CREATE TRIGGER  trigger\_name  [BEFORE|AFTER] [INSERT|UPDATE|DELETE] ON table\_name  FOR EACH ROW [FOLLOWS|PRECEDES] existing\_trigger\_name  BEGIN  …  END$$  DELIMITER ; |

MySQL Multiple Triggers Example

Let’s look at an example of creating multiple triggers on the same event and action time in a table to get a better understanding.

We will use the products table in the [sample database](http://mysqltutorial.org/mysql-sample-database.aspx) for the demonstration. Suppose, whenever we change the price of a product (column MSRP ), we want to log the old price in a separate table named price\_logs .

First, we create a new price\_logs table using the [CREATE TABLE statement](http://mysqltutorial.org/mysql-create-table/) as follows:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | CREATE TABLE price\_logs (    id INT(11) NOT NULL AUTO\_INCREMENT,    product\_code VARCHAR(15) NOT NULL,    price DOUBLE NOT NULL,    updated\_at TIMESTAMP NOT NULL DEFAULT               CURRENT\_TIMESTAMP               ON UPDATE CURRENT\_TIMESTAMP,      PRIMARY KEY (id),      KEY product\_code (product\_code),      CONSTRAINT price\_logs\_ibfk\_1 FOREIGN KEY (product\_code)    REFERENCES products (productCode)    ON DELETE CASCADE    ON UPDATE CASCADE  ); |

Second, we create a new trigger that activates when the BEFORE UPDATE event of the products table occurs. The trigger’s name is before\_products\_update :



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | DELIMITER $$    CREATE TRIGGER before\_products\_update     BEFORE UPDATE ON products     FOR EACH ROW  BEGIN       INSERT INTO price\_logs(product\_code,price)       VALUES(old.productCode,old.msrp);  END$$    DELIMITER ; |

Third, we change the price of a product and query the price\_logs table using the following [UPDATE](http://mysqltutorial.org/mysql-update-data.aspx)statement:



|  |  |
| --- | --- |
| 1  2  3 | UPDATE products  SET msrp = 95.1  WHERE productCode = 'S10\_1678'; |



|  |  |
| --- | --- |
| 1 | SELECT \* FROM price\_logs; |

mysql multiple triggers example

It works as expected.

Suppose we want to see not only the old price and when it was changed but also who changed it. We can add additional columns to the price\_logs table. However, for the purpose of multiple triggers demonstration, we will create a new table to store the data of users who made the changes. The name of the new table is user\_change\_logs with the following structure:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | CREATE TABLE user\_change\_logs (    id int(11) NOT NULL AUTO\_INCREMENT,    product\_code varchar(15) DEFAULT NULL,    updated\_at timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP    ON UPDATE CURRENT\_TIMESTAMP,      updated\_by varchar(30) NOT NULL,      PRIMARY KEY (id),      KEY product\_code (product\_code),      CONSTRAINT user\_change\_logs\_ibfk\_1 FOREIGN KEY (product\_code)    REFERENCES products (productCode)    ON DELETE CASCADE ON UPDATE CASCADE  ); |

Now, we create a second trigger that activates on the BEFORE UPDATE event of the products table. This trigger will update the user\_change\_logs table with the data of the user who made the changes. It is activated after the before\_products\_update trigger.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | DELIMITER $$  CREATE TRIGGER before\_products\_update\_2     BEFORE UPDATE ON products     FOR EACH ROW FOLLOWS before\_products\_update  BEGIN     INSERT INTO user\_change\_logs(product\_code,updated\_by)     VALUES(old.productCode,user());  END$$    DELIMITER ; |

Let’s do a quick test.

First, we update the prices of the product using the [UPDATE statement](http://mysqltutorial.org/mysql-update-data.aspx)as follows:



|  |  |
| --- | --- |
| 1  2  3 | UPDATE products  SET msrp = 95.3  WHERE productCode = 'S10\_1678'; |

Second, we query the data from both price\_logs and user\_change\_logs tables:



|  |  |
| --- | --- |
| 1 | SELECT \* FROM price\_logs; |

mysql multiple triggers example 2



|  |  |
| --- | --- |
| 1 | SELECT \* FROM user\_change\_logs; |

mysql multiple triggers: second trigger activates

As you can see, both triggers were activated in the order as expected.

Information On Triggers Order

If you use the SHOW TRIGGERS statement, you will not see the order that triggers activate for the same event and action timing in a table.

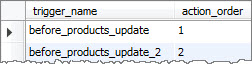


|  |  |
| --- | --- |
| 1 | SHOW TRIGGERS FROM classicmodels; |

To find this information, you need to query the action\_order column in the triggers table of the information\_schema database as follows:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | SELECT      trigger\_name, action\_order  FROM      information\_schema.triggers  WHERE      trigger\_schema = 'classicmodels'  ORDER BY event\_object\_table ,           action\_timing ,           event\_manipulation |



In this tutorial, we have shown you how to create multiple triggers on the same event for a table in MySQL.

Related Tutorials

* [Managing Triggers in MySQL](http://www.mysqltutorial.org/managing-trigger-in-mysql.aspx)
* Was this tutorial helpful ?
* [Yes](http://www.mysqltutorial.org/mysql-triggers/create-multiple-triggers-for-the-same-trigger-event-and-action-time/)   [No](http://www.mysqltutorial.org/mysql-triggers/create-multiple-triggers-for-the-same-trigger-event-and-action-time/)

[Previous Tutorial:Create Trigger in MySQL](http://www.mysqltutorial.org/create-the-first-trigger-in-mysql.aspx)

[Next Tutorial:Managing Triggers in MySQL](http://www.mysqltutorial.org/managing-trigger-in-mysql.aspx)

MYSQL QUICK START

* [What Is MySQL?](http://www.mysqltutorial.org/what-is-mysql/)
* [Install MySQL Database Server](http://www.mysqltutorial.org/install-mysql/)
* [Download MySQL Sample Database](http://www.mysqltutorial.org/mysql-sample-database.aspx)
* [Load Sample Database](http://www.mysqltutorial.org/how-to-load-sample-database-into-mysql-database-server.aspx)

MYSQL TRIGGERS

* [Introduction to SQL Trigger](http://www.mysqltutorial.org/sql-triggers.aspx)
* [MySQL Triggers Implementation](http://www.mysqltutorial.org/mysql-trigger-implementation.aspx)
* [Create Trigger in MySQL](http://www.mysqltutorial.org/create-the-first-trigger-in-mysql.aspx)
* [Managing Triggers in MySQL](http://www.mysqltutorial.org/managing-trigger-in-mysql.aspx)
* [Create Multiple Triggers For The Same Trigger Event And Action Time](http://www.mysqltutorial.org/mysql-triggers/create-multiple-triggers-for-the-same-trigger-event-and-action-time/)
* [Working with MySQL Scheduled Event](http://www.mysqltutorial.org/mysql-triggers/working-mysql-scheduled-event/)
* [Modifying MySQL Events](http://www.mysqltutorial.org/mysql-triggers/modifying-mysql-events/)

MYSQL PROGRAMMING INTERFACES

* [PHP MySQL Tutorial](http://www.mysqltutorial.org/php-mysql/)
* [Node.js MySQL Tutorial](http://www.mysqltutorial.org/mysql-nodejs/)
* [Python MySQL Tutorial](http://www.mysqltutorial.org/python-mysql/)
* [Perl MySQL Tutorial](http://www.mysqltutorial.org/perl-mysql/)
* [MySQL JDBC Tutorial](http://www.mysqltutorial.org/mysql-jdbc-tutorial/)

OTHER TUTORIALS

* [MySQL Administration](http://www.mysqltutorial.org/mysql-administration.aspx)
* [MySQL Full-Text Search](http://www.mysqltutorial.org/mysql-full-text-search.aspx)
* [MySQL Cheat Sheet](http://www.mysqltutorial.org/mysql-cheat-sheet.aspx)
* [MySQL Books and Video Training](http://www.mysqltutorial.org/mysqlbooks.aspx)
* [MySQL Hosting](http://www.mysqltutorial.org/mysql-hosting.aspx)
* [MySQL Resources](http://www.mysqltutorial.org/mysql-resources.aspx)

RECENT MYSQL TUTORIALS

* [MySQL UUID Smackdown: UUID vs. INT for Primary Key](http://www.mysqltutorial.org/mysql-uuid/)
* [MySQL RIGHT JOIN](http://www.mysqltutorial.org/mysql-right-join/)
* [MySQL MINUS](http://www.mysqltutorial.org/mysql-minus/)
* [MySQL Join Made Easy](http://www.mysqltutorial.org/mysql-join/)
* [How To Delete Duplicate Rows in MySQL](http://www.mysqltutorial.org/mysql-delete-duplicate-rows/)
* [How To Find Duplicate Values in MySQL](http://www.mysqltutorial.org/mysql-find-duplicate-values/)
* [MySQL LAST\_DAY Function](http://www.mysqltutorial.org/mysql-last_day/)
* [MySQL EXTRACT Function](http://www.mysqltutorial.org/mysql-extract/)
* [MySQL SYSDATE Function](http://www.mysqltutorial.org/mysql-sysdate/)
* [MySQL CURDATE Function](http://www.mysqltutorial.org/mysql-curdate/)

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](http://www.mysqltutorial.org/about-us/)

SITE LINKS

* [About Us](http://www.mysqltutorial.org/about-us/)
* [Contact Us](http://www.mysqltutorial.org/contact-us/)
* [Request a Tutorial](http://www.mysqltutorial.org/request-mysql-tutorial/)
* [Privacy Policy](http://www.mysqltutorial.org/privacy-policy/)

Copyright © 2018 by www.mysqltutorial.org. All Rights Reserved.