# **Create Trigger in MySQL**

[**?**](http://liveburst.com/ref/1/)

Summary: in this tutorial, you will learn how to create a trigger in MySQL using the CREATE TRIGGERstatement.

You should follow the [introduction to SQL triggers](http://www.mysqltutorial.org/sql-triggers.aspx) and [trigger implementation in MySQL](http://www.mysqltutorial.org/mysql-trigger-implementation.aspx) first before going forward with this tutorial.

## **MySQL trigger syntax**

In order to create a new trigger, you use the CREATE TRIGGER statement. The following illustrates the syntax of the CREATE TRIGGER statement:

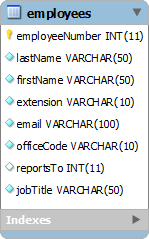
|  |  |
| --- | --- |
| 1  2  3  4  5  6 | CREATE TRIGGER trigger\_name trigger\_time trigger\_event  ON table\_name  FOR EACH ROW  BEGIN  ...  END; |

Let’s examine the syntax above in more detail.

* You put the trigger name after the CREATE TRIGGER statement. The trigger name should follow the naming convention [trigger time]\_[table name]\_[trigger event], for example before\_employees\_update*.*
* Trigger activation time can be BEFORE or AFTER. You must specify the activation time when you define a trigger. You use the BEFORE keyword if you want to process action prior to the change is made on the table and AFTER if you need to process action after the change is made.
* The trigger event can be INSERT, UPDATE or DELETE. This event causes the trigger to be invoked. A trigger only can be invoked by one event. To define a trigger that is invoked by multiple events, you have to define multiple triggers, one for each event.
* A trigger must be associated with a specific table. Without a table trigger would not exist therefore you have to specify the table name after the ON keyword.
* You place the SQL statements between BEGIN and END block. This is where you define the logic for the trigger.

## **MySQL trigger example**

Let’s start creating a trigger in MySQL to log the changes of the employees table.



First, [create a new table](http://www.mysqltutorial.org/mysql-create-table/) named employees\_audit to keep the changes of the employee table. The following statement creates the employee\_audit table.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | CREATE TABLE employees\_audit (  id INT AUTO\_INCREMENT PRIMARY KEY,  employeeNumber INT NOT NULL,  lastname VARCHAR(50) NOT NULL,  changedat DATETIME DEFAULT NULL,  action VARCHAR(50) DEFAULT NULL  ); |

Next, create a BEFORE UPDATE trigger that is invoked before a change is made to the employees table.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | DELIMITER $$  CREATE TRIGGER before\_employee\_update  BEFORE UPDATE ON employees  FOR EACH ROW  BEGIN  INSERT INTO employees\_audit  SET action = 'update',  employeeNumber = OLD.employeeNumber,  lastname = OLD.lastname,  changedat = NOW();  END$$  DELIMITER ; |

Inside the body of the trigger, we used the OLD keyword to access employeeNumber and lastnamecolumn of the row affected by the trigger.

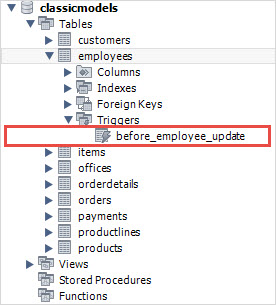
Notice that in a trigger defined for [INSERT](http://www.mysqltutorial.org/mysql-insert-statement.aspx), you can use NEW keyword only. You cannot use the OLDkeyword. However, in the trigger defined for [DELETE](http://www.mysqltutorial.org/mysql-delete-statement.aspx), there is no new row so you can use the OLDkeyword only. In the [UPDATE](http://www.mysqltutorial.org/mysql-update-data.aspx) trigger, OLD refers to the row before it is updated and NEW refers to the row after it is updated.

Then, to view all triggers in the current database, you use SHOW TRIGGERS statement as follows:

|  |  |
| --- | --- |
| 1 | SHOW TRIGGERS; |

MySQL SHOW TRIGGERS example

In addition, if you look at the schema using MySQL Workbench under the employees > triggers, you will see the before\_employee\_update trigger as shown in the screenshot below:



After that, update the employees table to check whether the trigger is invoked.

|  |  |
| --- | --- |
| 1  2  3  4  5 | UPDATE employees  SET  lastName = 'Phan'  WHERE  employeeNumber = 1056; |

Finally, to check if the trigger was invoked by the UPDATE statement, you can query the employees\_audit table using the following query:

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT  \*  FROM  employees\_audit; |

The following is the output of the query:

MySQL Trigger log

As you see, the trigger was really invoked and it inserted a new row into the employees\_audit table.

In this tutorial, you have learned how to create a trigger in MySQL. We also showed you how to develop a trigger to audit the changes of the employees table.