Search this website…

MySQL Triggers Implementation

Summary: in this tutorial, you will learn about **MySQL triggers implementation**. In addition, we will show you how MySQL stores trigger definitions and the limitations of triggers in MySQL.

Introduction to MySQL triggers

In MySQL, a trigger is a set of SQL statements that is invoked automatically when a change is made to the data on the associated table. A trigger can be defined to be invoked either before or after the data is changed by INSERT, UPDATE or DELETE statements. MySQL allows you to define maximum six triggers for each table.

- `BEFORE INSERT` – activated before data is inserted into the table.
- `AFTER INSERT` - activated after data is inserted into the

MySQL Tutorial Newsletter

Sign up to receive updates about MySQL Tutorials and find out what's new in MySQL.

Enter your email address...

table.

- `BEFORE UPDATE` – activated before data in the table is updated.
- `AFTER UPDATE` - activated after data in the table is updated.
- `BEFORE DELETE` – activated before data is removed from the table.
- `AFTER DELETE` – activated after data is removed from the table.

When you use a statement that makes change to the table but does not use `INSERT`, `DELETE` or `UPDATE` statement, the trigger is not invoked. For example, the `TRUNCATE` statement removes the whole data of a table but does not invoke the trigger associated with that table.

There are some statements that use the `INSERT` statement behind the scenes such as `REPLACE` statement and `LOAD DATA` statement. If you use these statements, the corresponding triggers associated with the tables if available will be invoked.

Triggers defined for a table must have a unique name. You can have the same trigger name that defines for different tables but it is not recommended. In practice, the names of triggers follow the following naming convention:

```
(BEFORE | AFTER)_tableName_(INSERT| UPDATE | DELETE)
```

MySQL Triggers Storage

MySQL stores triggers in a data directory e.g., `/data/classicmodels/` with the files named `tablename.TRG` and `triggername.TRN` :

- The `tablename.TRG` file maps the trigger to the corresponding table.
- the `triggername.TRN` file contains the trigger definition.

MySQL Quick Start

Install MySQL Database Server

Download MySQL Sample Database

Load Sample Database

MySQL Programming Interfaces

PHP MySQL Tutorial

Perl MySQL Tutorial

Other Tutorials

MySQL Full-Text Search

MySQL Cheat Sheet

MySQL Books

MySQL Hosting

MySQL Resources

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 fast and use MySQL effectively.

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

About Us

MySQL is trademark of Oracle Corp.

Recent MySQL Tutorials

MySQL Export Table to CSV

Raising Error Conditions with SIGNAL / RESIGNAL Statements

MySQL Error Handling in Stored Procedures

MySQL Stored Function

Import CSV File Into MySQL Table

MySQL Natural Sorting with ORDER BY Clause

Modifying MySQL Events

Working with MySQL Scheduled Event

PHP MySQL BLOB

PHP MySQL Transaction

Site Links

About Us

Contact Us

Request a Tutorial

Privacy Policy

You can back up the MySQL triggers by copying the trigger files to the backup folder. You can also backup the triggers using the *mysqldump* tool.

MySQL Trigger Limitations

MySQL triggers have all features in standard SQL however there are some limitations that you should know before using them in your applications.

MySQL triggers cannot:

- Use `SHOW` , `LOAD DATA` , `LOAD TABLE` , `BACKUP DATABASE`, `RESTORE` , `FLUSH` and `RETURN` statements.
- Use statements that commit or rollback implicitly or explicitly such as `COMMIT` , `ROLLBACK` , `START TRANSACTION` , `LOCK/UNLOCK TABLES` , `ALTER` , `CREATE` , `DROP` , `RENAME` , etc.
- Use prepared statements such as `PREPARE` , `EXECUTE` , etc.
- Use dynamic SQL statements.
- Call a stored procedure or stored function.

In this tutorial, we have shown you how triggers are implemented in MySQL. We also discussed about trigger's storage as well as trigger's limitations in MySQL.

Related Tutorials

- Introduction to SQL Trigger
- Create Trigger in MySQL
- Managing Trigger in MySQL

Share this:

Share 0 Like 4 Tweet 0

« Previous Tutorial:
Introduction to SQL Trigger

Next Tutorial: »