What is trigger

A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. Some uses for triggers are to perform checks of values to be inserted into a table or to perform calculations on values involved in an update.

A trigger is defined to activate when a statement inserts, updates, or deletes rows in the associated table. These row operations are trigger events. For example, rows can be inserted by INSERT or LOAD DATA statements, and an insert trigger activates for each inserted row. A trigger can be set to activate either before or after the trigger event. For example, you can have a trigger activate before each row that is inserted into a table or after each row that is updated.

MySQL Trigger

A trigger in MySQL is a set of SQL statements that reside in a system catalog. It is a special type of stored procedure that is invoked automatically in response to an event. Each trigger is associated with a table, which is activated on any DML statement such as INSERT, UPDATE, or DELETE.

A trigger is called a special procedure because it cannot be called directly like a stored procedure. The main difference between the trigger and procedure is that a trigger is called automatically when a data modification event is made against a table. In contrast, a stored procedure must be called explicitly.

Generally, triggers are of two types according to the SQL standard: row-level triggers and statement-level triggers.

Row-Level Trigger: It is a trigger, which is activated for each row by a triggering statement such as insert, update, or delete. For example, if a table has inserted, updated, or deleted multiple rows, the row trigger is fired automatically for each row affected by the insert, update, or delete statement.

Important

MySQL triggers activate only for changes made to tables by SQL statements. This includes changes to base tables that underlie updatable views. Triggers do not activate for changes to tables made by APIs that do not transmit SQL statements to the MySQL Server. This means that triggers are not activated by updates made using the NDB API.

Triggers are not activated by changes in INFORMATION_SCHEMA or performance_schema tables. Those tables are actually views and triggers are not permitted on views.

Why we need/use triggers in MySQL?

We need/use triggers in MySQL due to the following features:

Triggers help us to enforce business rules.
Triggers help us to validate data even before they are inserted or updated.

Triggers help us to keep a log of records like maintaining audit trails in tables.

SQL triggers provide an alternative way to check the integrity of data.

Triggers provide an alternative way to run the scheduled task.

Triggers increases the performance of SQL queries because it does not need to compile each time the query is executed.

Triggers reduce the client-side code that saves time and effort.

Triggers help us to scale our application across different platforms.

Triggers are easy to maintain.

Types of Triggers in MySQL?

We can define the maximum six types of actions or events in the form of triggers:

Before Insert: It is activated before the insertion of data into the table.

After Insert: It is activated after the insertion of data into the table.

Before Update: It is activated before the update of data in the table.

After Update: It is activated after the update of the data in the table.

Before Delete: It is activated before the data is removed from the table.

After Delete: It is activated after the deletion of data from the table.

When we use a statement that does not use INSERT, UPDATE or DELETE query to change the data in a table, the triggers associated with the trigger will not be invoked.