MySQL Triggers

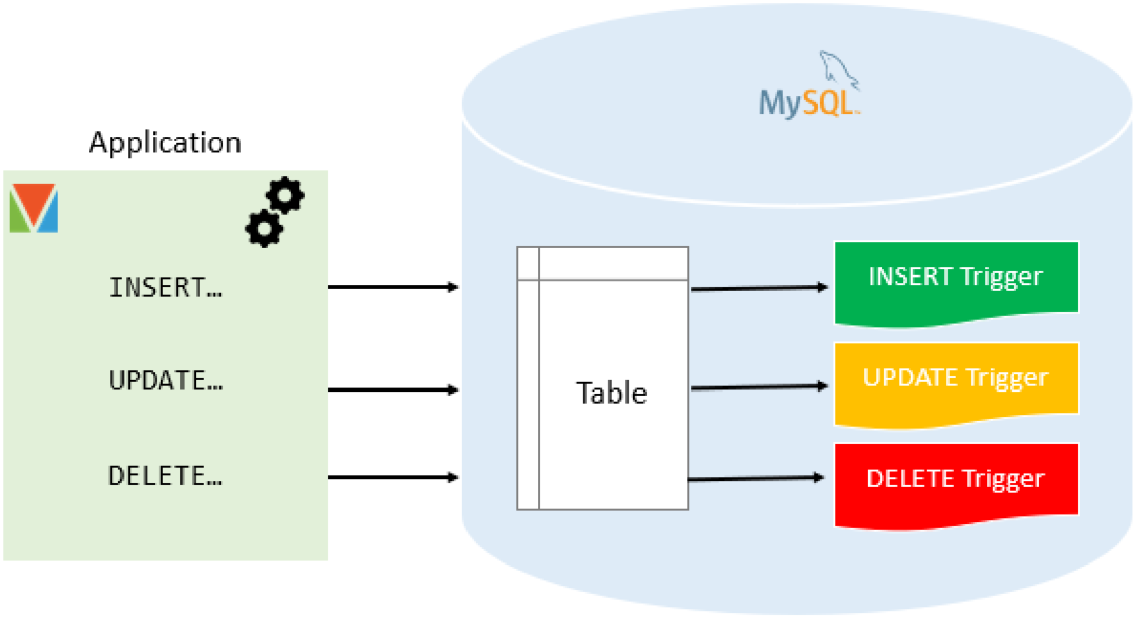
In MySQL, a trigger is a stored program invoked automatically in response to an event such as insert (https://www.mysqltutorial.org/mysql-insert-statement.aspx) , update (https://www.mysqltutorial.org/mysql-update-data.aspx) , or delete (https://www.mysqltutorial.org/mysql-delete-statement.aspx) that occurs in the associated table. For example, you can define a trigger that is invoked automatically before a new row is inserted into a table.

MySQL supports triggers that are invoked in response to the `INSERT (https://www.mysqltutorial.org/mysql-insert-statement.aspx)` , `UPDATE (https://www.mysqltutorial.org/mysql-update-data.aspx)` or `DELETE (https://www.mysqltutorial.org/mysql-delete-statement.aspx)` event.

The SQL standard defines two types of triggers: row-level triggers and statement-level triggers.

- A row-level trigger is activated for each row that is inserted, updated, or deleted. For example, if a table has 100 rows inserted, updated, or deleted, the trigger is automatically invoked 100 times for the 100 rows affected.
- A statement-level trigger is executed once for each transaction regardless of how many rows are inserted, updated, or deleted.

MySQL supports only row-level triggers. It doesn't support statement-level triggers.



Advantages of triggers

- Triggers provide another way to check the integrity of data.
- Triggers handle errors from the database layer.
- Triggers give an alternative way to run scheduled tasks (https://www.mysqltutorial.org/mysql-triggers/working-mysql-scheduled-event/) . By using triggers, you don't have to wait for the scheduled events (https://www.mysqltutorial.org/mysql-triggers/working-mysql-scheduled-event/) to run because the triggers are invoked automatically *before* or *after* a change is made to the data in a table.
- Triggers can be useful for auditing the data changes in tables.

Disadvantages of triggers

- Triggers can only provide extended validations, not all validations. For simple validations, you can use the `NOT NULL (https://www.mysqltutorial.org/mysql-not-null-constraint/)` , `UNIQUE (https://www.mysqltutorial.org/mysql-unique-constraint/)` , `CHECK (https://www.mysqltutorial.org/mysql-check-constraint/)` and `FOREIGN KEY (https://www.mysqltutorial.org/mysql-foreign-key/)` constraints.
- Triggers can be difficult to troubleshoot because they execute automatically in the database, which may not invisible to the client applications.
- Triggers may increase the overhead of the MySQL Server.

Managing MySQL triggers

- Create triggers (https://www.mysqltutorial.org/create-the-first-trigger-in-mysql.aspx) – describe steps of how to create a trigger in MySQL.
- Drop triggers (https://www.mysqltutorial.org/mysql-triggers/mysql-drop-trigger/) – show you how to drop a trigger.
- Create a BEFORE INSERT trigger (https://www.mysqltutorial.org/mysql-triggers/mysql-before-insert-trigger/) – show you how to create a `BEFORE INSERT` trigger to maintain a summary table from another table.
- Create an AFTER INSERT trigger (https://www.mysqltutorial.org/mysql-triggers/mysql-after-insert-trigger/) – describe how to create an `AFTER INSERT` trigger to insert data into a table after inserting data into another table.

- Create a BEFORE UPDATE trigger (https://www.mysqltutorial.org/mysql-triggers/mysql-before-update-trigger/) – learn how to create a `BEFORE UPDATE` trigger that validates data before it is updated to the table.
- Create an AFTER UPDATE trigger (https://www.mysqltutorial.org/mysql-triggers/mysql-after-update-trigger/) – show you how to create an `AFTER UPDATE` trigger to log the changes of data in a table.
- Create a BEFORE DELETE trigger (https://www.mysqltutorial.org/mysql-triggers/mysql-before-delete-trigger/) – show how to create a `BEFORE DELETE` trigger.
- Create an AFTER DELETE trigger (https://www.mysqltutorial.org/mysql-triggers/mysql-after-delete-trigger/) – describe how to create an `AFTER DELETE` trigger.
- Create multiple triggers for a table that have the same trigger event and time (https://www.mysqltutorial.org/mysql-triggers/create-multiple-triggers-for-the-same-trigger-event-and-action-time/) – MySQL 8.0 allows you to define multiple triggers for a table that have the same trigger event and time.
- Show triggers (https://www.mysqltutorial.org/mysql-triggers/mysql-show-triggers/) – list triggers in a database, table by specific patterns.