# What are Triggers?

This lesson provides an introduction to triggers and discusses the benefits and trade-offs of using them.

What are Triggers?

Triggers are MySQL programs that are executed automatically in response to an event. They are associated with tables. More specifically, a trigger is invoked in response to events that change the contents of a table like the **INSERT**, **UPDATE**, and **DELETE** statements. Triggers can be invoked before or after these events take place.
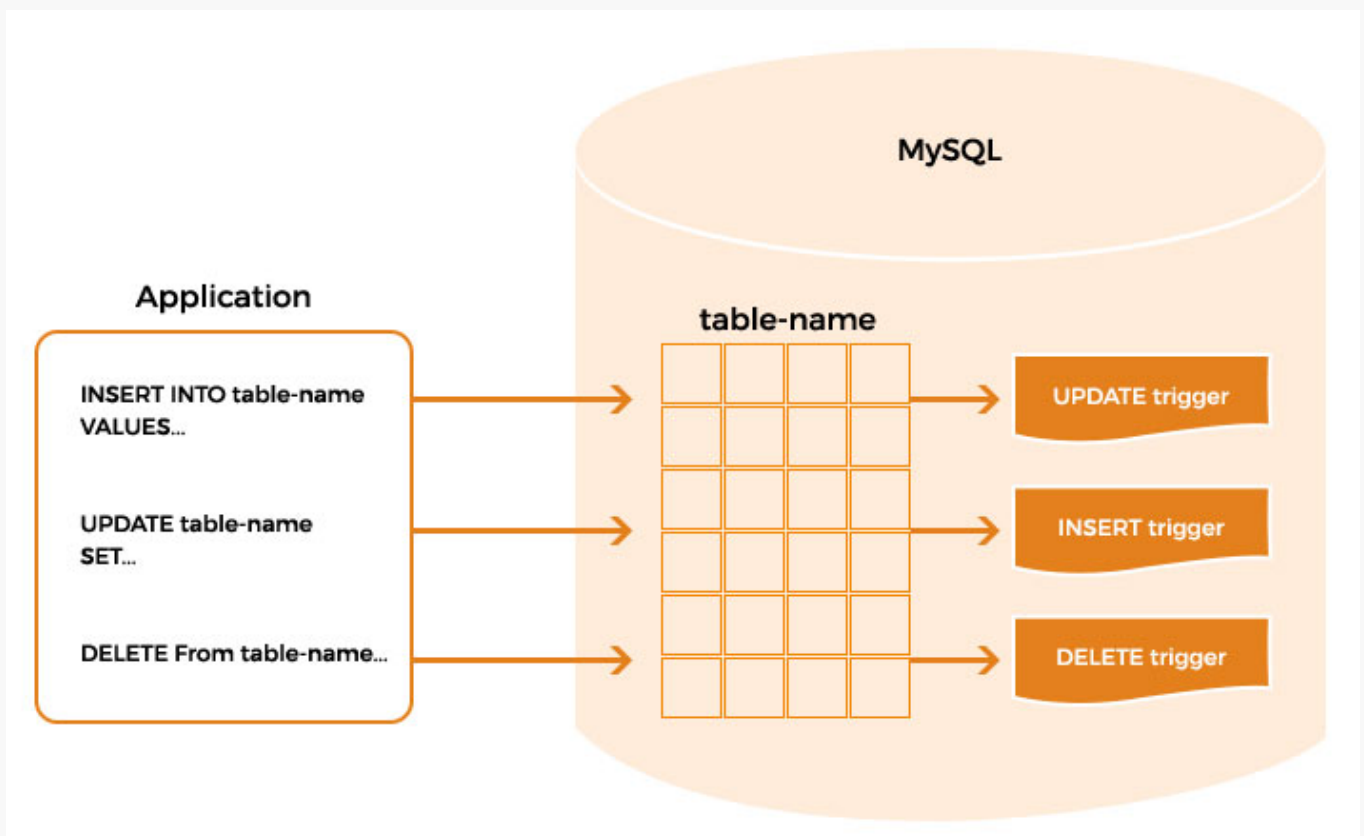
The SQL standard specifies two types of triggers; row-level triggers and statement-level triggers. The former are invoked when a row is inserted, updated or deleted. Say, if a query updates 25 rows, then a row-level trigger will be invoked as many times. The latter type of triggers are invoked once per statement even if the statement targets 25 rows. MySQL only provides support for row-level triggers.

Triggers offer a number of advantages. They prevent invalid DML statements from executing and are used to handle errors from the database layer by ensuring data integrity. Triggers provide an alternative to **scheduled events**. Scheduled events in MySQL are invoked at a specified time whereas triggers are invoked whenever a DML statement is executed. Triggers are also useful for auditing purposes and keeping a log of changes made to a table.
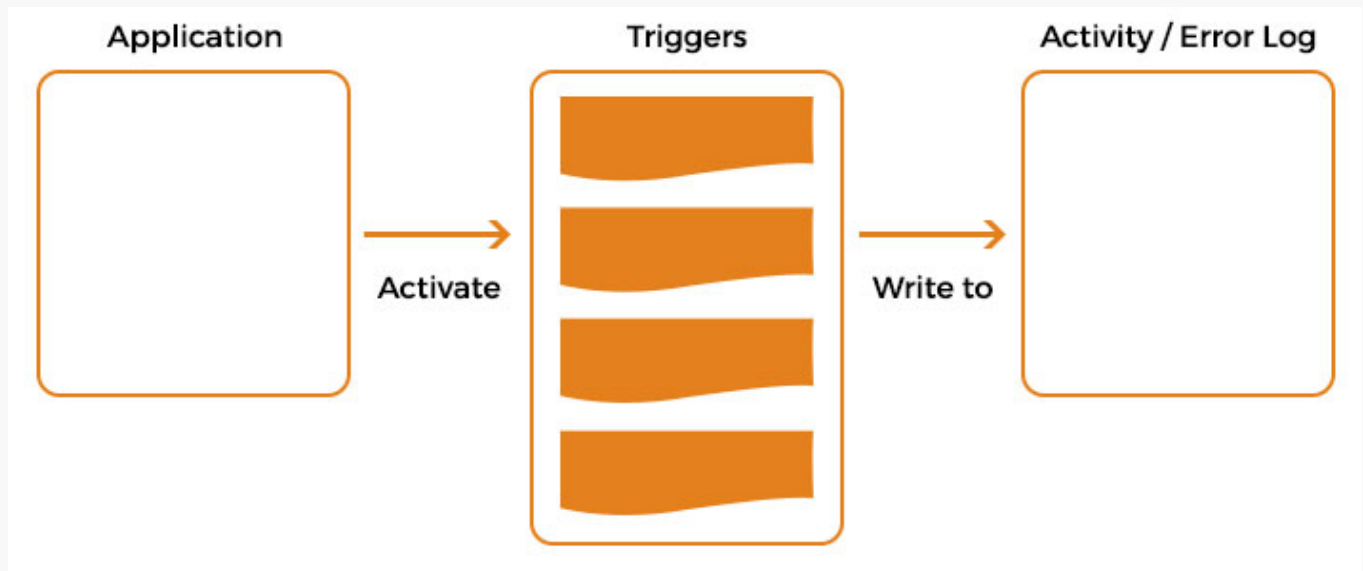
On the flip-side, the limitations of triggers include an increased overhead on MySQL server. They are also difficult to debug because they run automatically at database layer and it is not possible to apply breakpoints and see the step-by-step execution. In cases where the number of events occurring per second is high, triggers may incur a significant overhead. Another limitation is that triggers do not provide all validations which are needed at the application layer.

Triggers can only be associated with three statements; **INSERT**, **UPDATE** and **DELETE**. If any other statement makes similar changes to the data in the table, the associated triggers will not be executed. For example. Instead of **DELETE**, if **TRUNCATE** statement is used to remove all rows from a table, the trigger associated with **DELETE** statement will not be invoked.