



Practical 8

Title : Trigger in SQL

Aim : Demonstration of trigger in MySQL in MySQL workbench.

Theory : A trigger in SQL is a set of instructions that are automatically executed in response to specific events, such as inserting, updating or deleting data in a table. Triggers allow you to perform specific actions, such as updating other tables, sending notifications, or logging events, in response to data changes in your database. Triggers are useful for enforcing data integrity constraints, implementing complex business rules, & maintaining data consistency across multiple tables. They are executed automatically, which helps simplify database management & ensure that your data remains accurate & consistent.

Syntax

```
CREATE TRIGGER triggerName  
{AFTER | BEFORE} {INSERT | UPDATE | DELETE}  
ON tableName  
FOR EACH ROW  
BEGIN  
    -- trigger actions go here  
END;
```



Where,

- CREATE TRIGGER is the statement used to create a new trigger.
- triggerName is the name of the trigger being created.
- AFTER / BEFORE specifies when the trigger should be executed, either after or before the data change event.
- INSERT / UPDATE / DELETE specifies the type of data change event that will trigger execution of the trigger.
- ON tableName specifies the table on which the trigger should be applied.
- FOR EACH ROW specifies that the trigger should be executed for each row affected by the data change event.
- BEGIN / END delimit the block of statements that make up the trigger action. The statements within this block define the actions that will be performed when the trigger is executed.
- CREATE OR REPLACE TRIGGER is the statement used to create or replace a trigger.

DROP TRIGGER

Syntax

DROP TRIGGER [IF EXISTS] triggerName;

- IF EXISTS is an optional clause that will cause the statement to return a warning instead of an error if the trigger does not exist.



Row Level Trigger

A row level trigger in SQL is a type of trigger that is executed once for each row affected by a data modification operation.

Statement Level Trigger

A statement level trigger in SQL is a type of trigger that is executed once for each statement that modifies data in table.

Trigger Time : BEFORE & AFTER

Trigger Event : INSERT, UPDATE & DELETE.

Nested Trigger

A nested trigger in SQL is a trigger that is executed as a result of the execution of another trigger. Nested triggers are triggered in response to the execution of an outer trigger & can perform additional actions in response to the data changes that triggered the outer trigger.

The use of nested triggers can help to implement complex business logic & data integrity constraints. However, they can also increase the complexity of your database & make it more difficult to maintain, so it is important to use them judiciously & only when necessary.

The syntax for creating a nested trigger in SQL is similar to the syntax for creating a regular trigger, but with the addition of the CREATE TRIGGER statement within the trigger action block of the outer trigger.



Syntax (Example) :

```
CREATE TRIGGER outerTrigger
AFTER INSERT ON outerTable
FOR EACH ROW
BEGIN
    -- outer trigger action
    CREATE TRIGGER innerTrigger
    AFTER INSERT ON innerTable
    FOR EACH ROW
    BEGIN
        -- inner trigger action
    END;
END;
```

Conclusion: In conclusion, Triggers in SQL allows you to automate actions in response to data modification events in a database. Triggers can be set to execute AFTER, BEFORE or INSTEAD OF data modification operations, & can be set at either the row-level or the statement level depending on the nature of the action being automated.