**TRIGGERS**

**EXPT NO: 10 DATE:13/12/22**

**AIM:**  To study actions and triggers in SQL

**THEORY**

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

It can also be defined as 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](https://www.javatpoint.com/sql-tutorial) 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](https://www.javatpoint.com/mysql-insert), [update](https://www.javatpoint.com/mysql-update), or [delete statement](https://www.javatpoint.com/mysql-delete).

**Statement-Level Trigger:** It is a trigger, which is fired once for each event that occurs on a table regardless of how many rows are inserted, updated, or deleted.

**Note:** MySQL doesn’t provide support for statement-level triggers. It only provides support for row-level triggers only

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

* 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.
* 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:**

1. [**Before Insert**](https://www.javatpoint.com/mysql-before-insert-trigger)**:** It is activated before the insertion of data into the table.
2. [**After Insert**](https://www.javatpoint.com/mysql-after-insert-trigger)**:** It is activated after the insertion of data into the table.
3. [**Before Update**](https://www.javatpoint.com/mysql-before-update-trigger)**:** It is activated before the update of data in the table.
4. [**After Update**](https://www.javatpoint.com/mysql-after-update-trigger)**:** It is activated after the update of the data in the table.
5. [**Before Delete**](https://www.javatpoint.com/mysql-before-delete-trigger)**:** It is activated before the data is removed from the table.
6. [**After Delete**](https://www.javatpoint.com/mysql-after-delete-trigger)**:** It is activated after the deletion of data from the table.

**QUERIES**

1) Create 4 tables with attributes mentioned in the brackets:

table1(a1), table2(a2), table3(a3), table4(a4, b4)

Create a trigger on table1 to insert, delete, update tables table2, table3,table4 respectively. Whenever we insert a value in table1(a1) the same value should be inserted in table2; the number is present in table3, it should be deleted from table3 and if the number is present in table4(a4) then b4 should be updated(incremented by 1).

2) Create a table borrower\_count with attributes cardno, number of books borrowed.

Whenever we insert an entry in the book\_loans table respective changes must be done to the newly created table.

**CONCLUSION**

The concept of Triggers in SQL was studied