Triggers In SQL

A `trigger` is a stored procedure in database which is automatically invoked whenever any special event occurs in the database. The event can be any event including INSERT, UPDATE and DELETE.

For eg: If you want to perfom a task after a record is inserted into the table then we can make use of `triggers`

Syntax for creating triggers

```
create trigger [trigger_name]
[before | after]
{insert | update | delete}
on [table_name]
[for each row | for each column]
[trigger_body]
```

`create trigger [trigger_name]` : Creates or replaces an existing trigger with the trigger_name.

`[before | after]` : Now we can specify when our trigger will get fired. It can be before updating the database or after updating the database.

Generally , `before` triggers are used to validate the data before storing it into the database.

`{insert | update | delete}` : Now, we specify the `DML operation` for which our trigger should get fired .

`on [table_name]` : Here, we specify the name of the table which is associated with the trigger.

`[for each row]` : This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.

`[for each column]` : This specifies a column-level trigger, i.e., the trigger will be executed after the specified column is affected.

`[trigger_body]` : Here, we specify the operations to be performed once the trigger is fired.

Show Trigger

If you want to see all the triggers that are present in your database.

```
show triggers in database_name;
```

Drop Trigger

if you no longer want your trigger then you may delete it.

```
drop trigger trigger_name;
```

Example :

Let us consider we have our database named `library`. Consider a scenario where we want a trigger which is fired everytime any particular book is inserted into the `books` table . The `trigger` should add the logs of all the books that are inserted into the `books` table.

We have created two tables :

1. `books` : It will store all the books available in the library
2. bookrecord : It will generate a statement a log for the inserted book

```
  Select * from library.books;
```

```
+----------+---------------+
| book_id  | book_name     |
+----------+---------------+
|      |       |
|      |       |
+----------+---------------+
```

Here, `book_id` is an auto-incremental field.

```
Select * from library.bookrecord;
```

```
+----------+---------------+-----------+
| SRNO     | bookid        | statement |
+----------+---------------+-----------+
|          |               |           |
|          |       |           |
+----------+---------------+-----------+
```

Here, `SRNO` is an auto-incremental field.

Now, we will create our trigger on the `books` table

```
create trigger library.addstatement
after insert
on library.books
for each row
insert into library.bookrecord(bookid,statement) values
(NEW.book_id,concat('New book named ',NEW.book_name,"  added at
",curdate()));
```

In MySQL, `NEW` is used to access the currently inserted row. We are inserting the log for the currently inserted book in our database.

Now we will insert a book and wait for the output.

```
insert into library.books(book_name) values ("Harry Potter and the Goblet
of fire");
```

Output for `books`:

```
+----------+-------------------------------------------------+
| book_id  | book_name                                       |
+----------+-------------------------------------------------+
|    1     |  Harry Potter and the Goblet of fire    |
|      |                 |
+----------+-------------------------------------------------+
```

Output for `bookrecord`:

```
+----------+--------------
+-------------------------------------------------------------------------------
+
| SRNO     | bookid        | statement                          |
+----------+--------------
+-------------------------------------------------------------------------------
+
|   1      |      1        |     New book named Harry Potter and the
Goblet of fire  added at 2021-10-22      |
|          |       |                                    |
+----------+--------------
+-------------------------------------------------------------------------------
+
```

See. it worked!!

Conclusion:

Here, you learnt what are triggers and how you create them. You can create different types of triggers based on your needs and requirements.