
Triggers

What is a Trigger in SQL

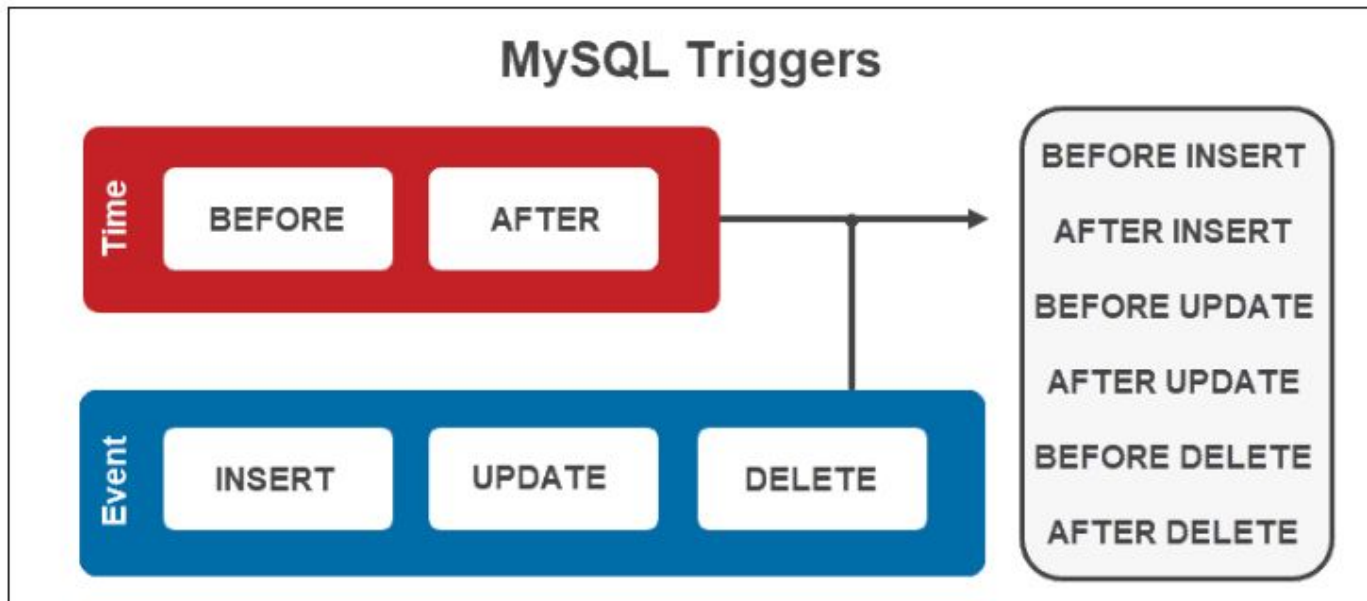
Trigger is a statement that a system executes automatically when there is any modification to the database.

Purpose of Trigger

- Enforce business rules.
 - Validate input data.
 - Write to other files for audit trail purposes
 - Query from other files for cross-referencing purposes
 - Replicate data to different files to achieve data consistency
-

Types of Trigger

1. **Time.** BEFORE or AFTER a specific row event.
2. **Event.** INSERT, UPDATE or DELETE.



Example - Before Insert

- Let us take a Course Management.
 - We want the grades updated based on marks obtained in different subjects
 - Business Rules specify the grading method.
 - Create a trigger to automatically compute the grade when update of marks done
-

Example - Before Insert

Table : Student

```
v sibhapvv_mcte student  
id : int(11)  
name : varchar(100)  
# sub1 : int(11)  
# sub2 : int(11)  
# total : int(11)  
# perc : int(11)  
grade : varchar(25)
```

Key words 'old' and 'new'

| Trigger Event | OLD | NEW |
|---------------|-----|-----|
| INSERT | No | Yes |
| UPDATE | Yes | Yes |
| DELETE | Yes | No |

Example - Before Update - Syntax

```
CREATE TRIGGER `updt_grade` BEFORE UPDATE ON `student`  
FOR EACH ROW  
BEGIN
```

```
SET NEW.total = NEW.sub1+NEW.sub2;  
SET NEW.perc = NEW.total/2;  
IF NEW.perc >=80 THEN  
SET NEW.grade = 'Distinction';  
ELSEIF NEW.perc>=60 AND NEW.perc<80 THEN  
SET NEW.grade = 'Alpha';  
ELSE SET NEW.grade = 'Qualified';  
END IF;
```

```
END
```

Example - Before Insert - Syntax

Marks for sub1 cannot exceed 90.

When inserting a new record with marks exceeding 90 , an alert should be flashed.

Syntax

```
IF NEW.sub1 > 90 THEN  
  SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Marks for RDBM  
cannot exceed 90';  
END IF;
```

Example - Before Delete - Syntax

Before deleting a record , copy it to a backup table

Syntax

```
insert into student_bu
    (id,name,sub1,sub2)
values(old.id,old.name, old.sub1, old.sub2)
```

Before update trigger

```
CREATE TRIGGER `updt_grade` BEFORE UPDATE ON `student`  
FOR EACH ROW BEGIN
```

```
SET NEW.total = NEW.sub1+NEW.sub2;  
SET NEW.perc = NEW.total/2;  
IF NEW.perc >=90 THEN  
SET NEW.grade = 'EXCELLENT';  
ELSEIF NEW.perc>=75 AND NEW.perc<90 THEN  
SET NEW.grade = 'VERY GOOD';  
ELSEIF NEW.perc>=60 AND NEW.perc<75 THEN  
SET NEW.grade = 'GOOD';  
ELSEIF NEW.perc>=40 AND NEW.perc<60 THEN  
SET NEW.grade = 'AVERAGE';  
ELSE SET NEW.grade = 'Qualified';  
END IF;  
END
```

Before Insert Trigger

```
CREATE TRIGGER `bi_rdbm` BEFORE INSERT ON
`student`
FOR EACH ROW IF NEW.sub1 > 90 THEN
SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT =
'Marks for RDBM cannot exceed 90';
END IF
```

Example - Before Delete - Syntax

```
CREATE TRIGGER `bd_backup` BEFORE DELETE ON  
`student`  
FOR EACH ROW insert into student_bu  
    (id,name,sub1,sub2)  
    values(old.id,old.name, old.sub1, old.sub2)
```

Example - After Delete - Syntax

Requirement : once a record is deleted , make a count of remaining records along with a time stamp.

Create a table called stu_log with two fields ; time and count

Syntax

```
CREATE TRIGGER `ad_count` AFTER DELETE ON `student`  
FOR EACH ROW BEGIN  
insert into stu_log (time,count) values (now(),(select count(name) from  
student));  
END
```

Example - After Update - Syntax

Requirement : once a record is updated , remake the list of recommended students

Create a table called recommended with two fields ; name and perc

Syntax

```
CREATE TRIGGER `au_recom` AFTER UPDATE ON `student`  
FOR EACH ROW BEGIN  
delete from recommended;  
insert into recommended (name,perc) (select name,perc from student  
where grade='very good');  
END
```

Example - After Insert - Syntax

Requirement : once a record is inserted , check if dob filled. If not take that name to another table reminder for giving a reminder to fill dob

Create a table called reminders with two fields ; member_id and msg

Syntax

```
CREATE TRIGGER `ai_dob_check` AFTER INSERT ON `student`  
FOR EACH ROW BEGIN  
  IF NEW.dob='0000-00-00' THEN  
    INSERT INTO reminders(member_id, msg)  
    VALUES(new.id,CONCAT('Hi ', NEW.name, ', please update your  
date of birth.));  
  END IF;
```
