Name :- Rushikesh Kazbhazi Palve
Roll No. 31258

Date : / /20 | Page No :

Assignment No. A7

DOP :- 01-10-2021 DOS :- 30-11-2021

Title :- Database Triggers : (All Types : Row level and statement level triggers, Before and After Triggers.)

Problem Definition :-

Write a database trigger on Library table. The system should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library_Audit table.

Objectives :-

(i) Understand the concept of row level and statement level triggers.
(ii) Understand the concept of triggers initiated against event.

Outcomes :-

After completion of assignment, students will be able to understand the concept of row level and statement level triggers.

Scanned by TapScanner

ROLL NO. 131258 | Date: / /20 | Page No: | (2)

Theory :-

DATABASE TRIGGERS :-

A database trigger is a PL/SQL program unit, which gets fired automatically whenever the data event such as DML or DDL system event. Triggers are associated with a specific table and are fired automatically whenever the table gets manipulated in a predefined way. The act of executing a trigger is called as firing a trigger.

Triggers are similar to procedures in that they are named PL/SQL blocks with declarative, executable and exception handling sections. But the difference is a procedure is executed explicitly from another block via a procedure call but a trigger is executed implicitly whenever the triggering event happens. A procedure can pass arguments but trigger doesn't accept arguments.

A database trigger has following components :-

1.) A triggering Event
2.) A triggering Constraint
3.) A triggering Action.

Trigger Categories :-

Triggers are categorized in various ways :-

1) Trigger type
2) Triggering time
3) Triggering Event.

Scanned by TapScanner

Trigger types :-

There are two types of triggers -

1.] **Statement Trigger :** A statement trigger is a trigger in which the trigger action is executed once for the manipulation operation that fired the trigger.

2.] **Row Trigger :** A row trigger is a trigger in which the trigger action is performed repeatedly for each row of the table that is affected by the manipulation operation that frees the trigger.

Triggering Time :-

Triggers can specify the time of trigger action.

1] **Before the triggering event :**
The trigger action is performed before the operation that frees the trigger is executed. The trigger is used when execution of operation depends on trigger action.

2] **After the triggering event :**
The trigger action is performed after the operation that frees the trigger is executed. This trigger is used when triggering action depends on the execution of operation -

Scanned by TapScanner

Triggering Events :-

Triggering events are the DML operations. These operations are insert, update and delete. When these operations are performed on a table, the trigger which is associated with the operation is fired.

Triggering events divide triggers into three types :-

1) DELETE TRIGGER
2) UPDATE TRIGGER
3) INSERT TRIGGER

General syntax for creation of Trigger :

```
Create [or replace] TRIGGER <trigger_name>
<BEFORE | AFTER>
DELETE | [OR] INSERT | [OR] UPDATE [OF <column1>,
                                    [,<column2>...]
ON <table_name>
[for each row [when <condition>]]
Begin
.....  .....
.....
End ;
```

Where,

Trigger_name : trigger_name is the name of the trigger

Scanned by TapScanner

Table_name :- is the table name for which the trigger is defined.

Trigger_condition :- The trigger condition in the when clause, if present is evaluated first. The body of the trigger is executed only when this condition evaluates to true.

Dropping Trigger :-

Suppose you want to drop triggers then the syntax is

Syntax :- Drop trigger trigger_name;

Enabling and Disabling Triggers :-

The triggers can be disabled without dropping them. When the trigger is disabled, it is still exists in data dictionary but never fired, To disable triggers, use alter command.

Syntax :-

Alter TRIGGER triggername DISABLE/ ENABLE;

For all triggers on a particular table

Syntax :- Alter TRIGGER triggername (DISABLE/ ENABLE) all triggers;

Scanned by TapScanner

ROLL NO. 31258

: Date : 1/120 | Page No :

Conclusion :-

Thus, we understood the concept of row level and statement level triggers. And also, understood the concept of triggers initiated against event.

OUTPUT :-

```
/*
Problem statement :-
   Write a database trigger on Library table. The System should keep track of the
   records that are being updated or deleted. The old value of updated or deleted
   records should be added in Library_Audit table.
*/

mysql> CREATE TABLE Library(
    -> stu_roll_no INT PRIMARY KEY,
    -> stu_name VARCHAR(20),
    -> book_name VARCHAR(20),
    -> date_of_issue DATE,
    -> price INT
    -> );

mysql> CREATE TABLE Library_Audit(
    -> Action_Performed VARCHAR(20),
    -> DateAndTime DATETIME,
    -> stu_roll_no INT REFERENCES Library(stu_roll_no),
    -> stu_name VARCHAR(20),
    -> book_name VARCHAR(20),
    -> date_of_issue DATE,
    -> price INT
    -> );

mysql> INSERT INTO Library VALUES
    -> (1, 'Vidyut', 'DBMS', '2021-07-21', 300),
    -> (2, 'Pratap', 'CNS', '2021-05-21', 500),
    -> (3, 'Kailash', 'SPOS', '2021-07-10', 400),
    -> (4, 'Mukund', 'OOP', '2021-08-16', 250),
    -> (5, 'Girish', 'DSA', '2021-06-26', 650),
    -> (6, 'Neeraj', 'OOP', '2021-01-01', 330),
    -> (7, 'Prashant', 'SPOS', '2021-02-18', 540),
    -> (8, 'Raj', 'CNS', '2021-01-31', 540),
    -> (9, 'Hari', 'DBMS', '2021-04-04', 820),
    -> (10, 'Aditya', 'PPL', '2021-03-07', 430);

mysql> DELIMITER $$
mysql> CREATE TRIGGER trig_library_insert
    -> AFTER INSERT ON library
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO Library_Audit
    -> VALUES ("INSERT", NOW(), NEW.stu_roll_no, NEW.stu_name, NEW.book_name,
NEW.date_of_issue, NEW.price);
    -> END $$

mysql> DELIMITER $$
mysql> CREATE TRIGGER trig_library_update
    -> AFTER UPDATE ON library
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO Library_Audit
    -> VALUES ("UPDATE", NOW(), OLD.stu_roll_no, OLD.stu_name, OLD.book_name,
OLD.date_of_issue, OLD.price);
    -> END $$
```

Scanned by TapScanner

```
mysql> DELIMITER $$
mysql> CREATE TRIGGER trig_library_delete
    -> BEFORE DELETE ON library
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO Library_Audit
    -> VALUES ("DELETE", NOW(), OLD.stu_roll_no, OLD.stu_name, OLD.book_name,
OLD.date_of_issue, OLD.price);
    -> END $$



mysql> DELIMITER ;
mysql> SELECT * FROM Library;
```

stu_roll_no	stu_name	book_name	date_of_issue	price
1	Vidyut	DBMS	2021-07-21	300
2	Pratap	CNS	2021-05-21	500
3	Kailash	SPOS	2021-07-10	400
4	Mukund	OOP	2021-08-16	250
5	Girish	DSA	2021-06-26	650
6	Neeraj	OOP	2021-01-01	330
7	Prashant	SPOS	2021-02-18	540
8	Raj	CNS	2021-01-31	540
9	Hari	DBMS	2021-04-04	820
10	Aditya	PPL	2021-03-07	430

```
mysql> UPDATE Library SET stu_name = "Tushar" WHERE stu_name = "Pratap";

mysql> SELECT * FROM Library_Audit;
```

Action_Performed	DateAndTime	stu_roll_no	stu_name	book_name	date_of_issue	price
UPDATE	2021-10-02 01:50:55	2	Pratap	CNS	2021-05-21	500

```
mysql> SELECT * FROM Library;
```

stu_roll_no	stu_name	book_name	date_of_issue	price
1	Vidyut	DBMS	2021-07-21	300
2	Tushar	CNS	2021-05-21	500
3	Kailash	SPOS	2021-07-10	400
4	Mukund	OOP	2021-08-16	250
5	Girish	DSA	2021-06-26	650
6	Neeraj	OOP	2021-01-01	330
7	Prashant	SPOS	2021-02-18	540
8	Raj	CNS	2021-01-31	540
9	Hari	DBMS	2021-04-04	820
10	Aditya	PPL	2021-03-07	430

```
mysql> DELETE FROM Library WHERE stu_name = "Tushar";

mysql> SELECT * FROM Library_Audit;
```

Action_Performed	DateAndTime	stu_roll_no	stu_name	book_name	date_of_issue	price
UPDATE	2021-10-02 01:50:55	2	Pratap	CNS	2021-05-21	500
DELETE	2021-10-02 01:52:25	2	Tushar	CNS	2021-05-21	500

```
mysql> SELECT * FROM Library;
```

Scanned by TapScanner

stu_roll_no	stu_name	book_name	date_of_issue	price
1	Vidyut	DBMS	2021-07-21	300
3	Kailash	SPOS	2021-07-10	400
4	Mukund	OOP	2021-08-16	250
5	Girish	DSA	2021-06-26	650
6	Neeraj	OOP	2021-01-01	330
7	Prashant	SPOS	2021-02-18	540
8	Raj	CNS	2021-01-31	540
9	Hari	DBMS	2021-04-04	820
10	Aditya	PPL	2021-03-07	430

```
mysql> INSERT INTO Library VALUES(15, 'Ajay', 'PPS', '2021-08-26', 700);

mysql> SELECT * FROM Library_Audit;
```

Action_Performed	DateAndTime	stu_roll_no	stu_name	book_name	date_of_issue	price
UPDATE	2021-10-02 01:50:55	2	Pratap	CNS	2021-05-21	500
DELETE	2021-10-02 01:52:25	2	Tushar	CNS	2021-05-21	500
INSERT	2021-10-02 01:53:48	15	Ajay	PPS	2021-08-26	700

```
mysql> SELECT * FROM Library;
```

stu_roll_no	stu_name	book_name	date_of_issue	price
1	Vidyut	DBMS	2021-07-21	300
3	Kailash	SPOS	2021-07-10	400
4	Mukund	OOP	2021-08-16	250
5	Girish	DSA	2021-06-26	650
6	Neeraj	OOP	2021-01-01	330
7	Prashant	SPOS	2021-02-18	540
8	Raj	CNS	2021-01-31	540
9	Hari	DBMS	2021-04-04	820
10	Aditya	PPL	2021-03-07	430
15	Ajay	PPS	2021-08-26	700

Scanned by TapScanner