Department of Computer Science & Engineering,
CS246 – Database Management Systems Lab

Triggers

Overview

- Introduction
- Example 01
- Example 02
- Triggers & Stored Procedures
- Multiple triggers on same table
- System information
- Nested triggers
- Recursive triggers

Introduction

Introduction - 01

- Triggers are also referred to as event-condition-action (ECA) rules
- Triggers are involved only when certain conditions specified by the database programmer occur.
- Unlike procedure/function, triggers need not be called explicitly. They are invoked based on INSERT, UPDATE, DELETE operations performed by database programmers
- A trigger automatically executes a set of actions (SQL statements) in response to certain events on a table or view.
- These events can include INSERT, UPDATE, DELETE operations or even DDL statements like ALTER, CREATE, and DROP.

Introduction- 02

- Components of a trigger
- **Event**: The type of operation that triggers the execution of the trigger (e.g., INSERT, UPDATE, DELETE).
- **Trigger Type**: Specifies when the trigger is executed relative to the triggering event. Common types include BEFORE and AFTER triggers.
- **Triggering Table**: The table on which the trigger is defined and against which the triggering event occurs.
- **Trigger Body**: The set of SQL statements that are executed when the trigger is fired.

Trigger example - 01

Table 1

T1C1
1
2
3
4

Table 2

T2C1
abc
abd
abe
abf

Table 3

T3C1
2024-07-11
2024-09-12
2024-09-13
2024-10-14

Table 4

T4C1
100.00
120.25
140.50
160.75

Before Inserting a record "5" into Table1 Should

1. Insert a record into **Table 2**
2. Update record in **Table 3**
3. Delete record from **Table 4**

Trigger example - 01

```
DELIMITER //

CREATE TRIGGER iud_1
BEFORE INSERT
ON Table1
FOR EACH ROW
BEGIN
  // increment T2C1
  INSERT INTO Table2(T2C1) VALUES ('abg');
  UPDATE Table3 SET T3C1='2023-05-15' WHERE T3C1 like '%-08-%';
  DELETE FROM Table4 WHERE T4C1=140.50;

END //

DELIMITER ;
```

Trigger example - 02

```
CREATE TABLE account(acc_num INT,
            amount DECMILA(10,2));

DELIMITER //

CREATE TRIGGER ins_sum
BEFORE INSERT ON account
FOR EACH ROW
SET @sum = @sum + NEW.amount;

DELIMITER ;
```

```
MySQL> SET @sum = 0;
MySQL> INSERT INTO account VALUES
(137,14.98),\
(141,1937.50),\
(97,-100.00);

MySQL> SELECT @sum as 'Total Amount';

MySQL> DROP TRIGGER ins_sum;
```

Trigger example - 03

```
CREATE TRIGGER upd_check
BEFORE UPDATE
ON account
FOR EACH ROW
BEGIN
  IF NEW.amount < 0
  THEN
    SET NEW.amount = 0;
  ELSEIF NEW.amount > 100
  THEN
    SET NEW.amount = 100;
  END IF;

END;//

DELIMITER ;
```

Triggers & Stored Procedures

Triggers & Stored Procedures

```
DELIMITER //

CREATE PROCEDURE p1()
BEGIN

  SELECT 'p1 called from trigger';

END //

DELIMITER ;
```

```
DELIMITER //

CREATER TRIGGER call_p1
BEFORE INSERT
ON Table1
FOR EACH ROW
BEGIN

  CALL p1();

END //

DELIMITER ;
```

Triggers & Limitations

- If a BEFORE trigger fails, the operation on the corresponding row is not performed.
- A BEFORE trigger is activated by the attempt to insert or modify the row, regardless of whether the attempt subsequently succeeds.
- An AFTER trigger is executed only if any BEFORE triggers and the row operation execute successfully.
- An error during either a BEFORE or AFTER trigger results in failure of the entire statement that caused trigger invocation.

Multiple Triggers On Same Table

Trigger example - 02

```
DELIMITER //

CREATE TRIGGER iud_2
BEFORE INSERT
ON Table1
FOR EACH ROW
FOLLOWS iud_1
BEGIN
  // incrementa from abf
  DELETE FROM Table2 where T2C1='abc';
  INSERT INTO Table3(T3C1) VALUES('16-Nov';
  DELETE FROM Table4 WHERE T4C1=140.50;

END //

DELIMITER ;
```

Thank You!