Nikit Gokhe

Class – Comp D1

Roll No. 224024

GR No. 21810522

**ASSIGNMENT NO.06**

**Aim:**

Database Trigger (Row level and Statement level triggers, Before and After Triggers):Write a database trigger on Employee 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 to a new table when the Employee table is updated.Employee(employee no, employee name, join\_date, designation, salary).

**Objective:**

To understand, create and test the working of trigger on employee table before updating or deleting a record.

**Theory:** In programs, sometimes it is required to execute certain code followed by certain events and this requirement can be achieved in PL/SQL through triggers.Triggers are stored programs that are fired automatically when some event occurs. The code to be fired can be defined as per the requirement.Triggers can be classified based on the following parameters.

**Classification based on the timing**

* + **BEFORE Trigger**: It fires before the specified event has occurred.
  + **AFTER Trigger**: It fires after the specified event has occurred.
  + **INSTEAD OF Trigger:** A special type.

**Classification based on the level**

* + **STATEMENT level Trigger**: It fires one time for the specified event statement.
  + **ROW level Trigger:** It fires for each record that got affected in the specified event. (only for DML)

**Classification based on the Event**

* + **DML Trigger:** It fires when the DML event is specified (INSERT/UPDATE/DELETE)
  + **DDL Trigger:** It fires when the DDL event is specified (CREATE/ALTER)
  + **DATABASE Trigger:** It fires when the database event is specified (LOGON/LOGOFF/STARTUP/SHUTDOWN)

**The syntax for creating a trigger is −**

CREATE [OR REPLACE ] TRIGGER trigger\_name

{BEFORE | AFTER | INSTEAD OF }

{INSERT [OR] | UPDATE [OR] | DELETE}

[OF col\_name]

ON table\_name

[REFERENCING OLD

AS o NEW AS n]

[FOR EACH ROW]

WHEN (condition)

DECLARE

Declaration-statements

BEGIN

Executable-statements

EXCEPTION

Exception-handling-statements

END;

**Input/Program:**

Create table employee (no int, name varchar(40), join\_date date, designation varchar(40), salary int);

Create table old\_employee (no int, name varchar(40), join\_date date, designation varchar(40), salary int);

Insert into employee values (1, abc, 2019-01-01, pqr, 20000);

Insert into employee values(1, def,2019-01-02, xyz, 20000);

Select \* from employee;

Delimiter //

create trigger update\_emp

-> before update on employee

-> for each row

-> begin

-> insert into old\_employee values (old.no, old.name, old.join\_date, old.designation, old.salary);

-> end; //

Delimiter //

create trigger update\_emp

-> before delete on employee

-> for each row

-> begin

->insert into old\_employee values (old.no, old.name, old.join\_date, old.designation, old.salary);

-> end; //

Update employee set no=2 where name=”def”;

Update employee set salary=30000 where no=2;

Delete from employee where no=2;

Select \* from employee;

Select \* from old\_employee;

**Output:**

Select \* from employee;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Join\_date** | **Designation** | **Salary** |
| 1 | abc | 2019-01-01 | pqr | 20000 |
| 1 | def | 2019-01-02 | xyz | 20000 |

After update operation:

Select \* from employee;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Join\_date** | **Designation** | **Salary** |
| 1 | abc | 2019-01-01 | pqr | 20000 |
| 2 | def | 2019-01-02 | xyz | 30000 |

After delete operation:

Select \* from employee;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Join\_date** | **Designation** | **Salary** |
| 1 | abc | 2019-01-01 | pqr | 20000 |

Select \* from old\_employee;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Join\_date** | **Designation** | **Salary** |
| 1 | def | 2019-01-02 | xyz | 20000 |
| 2 | def | 2019-01-02 | xyz | 20000 |
| 2 | def | 2019-01-02 | xyz | 30000 |

**Conclusion:** Hence we have tested the working of triggers in MySQL.

|  |  |  |
| --- | --- | --- |
| **Signature** | **Date** | **Remarks** |
|  |  |  |