MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	**Class:**
Semester/Year:	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: Part A-08**

PART: A) ASSIGNMENT NO: 08

AIM: **Database Trigger** (All Types: Row level and Statement level triggers, Before and After Triggers).

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To learn the concept of procedural language.
- To learn various Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Database: MySQL/ Oracle 11g Database.

THEORY:

PL/SQL Trigger

- A database trigger is a stored procedure that automatically executes whenever an event occurs. The event may be insert-delete-update operations.
- Trigger is invoked by Oracle engine automatically whenever a specified event occurs.
- Trigger is stored into database and invoked repeatedly, when specific condition match.
- Triggers could be defined on the table, view, schema, or database with which the event is associated.
- A procedure is executed explicitly from another block via a procedure call with passing arguments,
- While a trigger is executed (or fired) implicitly whenever the triggering event (DML: INSERT, UPDATE, or DELETE) happens, and a trigger doesn't accept arguments.
- Triggers are written to be executed in response to any of the following events.

- ✓ A database manipulation (DML) statement (DELETE, INSERT, or UPDATE).
- ✓ A database definition (DDL) statement (CREATE, ALTER, or DROP).
- ✓ A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

- **Type of Triggers**

✓ **BEFORE Trigger:** BEFORE trigger execute before the triggering DML statement (INSERT, UPDATE, DELETE) execute. Triggering SQL statement is may or may not execute, depending on the BEFORE trigger conditions block.

✓ **AFTER Trigger:** AFTER trigger execute after the triggering DML statement (INSERT, UPDATE, DELETE) executed. Triggering SQL statement is execute as soon as followed by the code of trigger before performing Database operation.

✓ **ROW Trigger:** ROW trigger fire for each and every record which are performing INSERT, UPDATE, DELETE from the database table. If row deleting is define as trigger event, when trigger file, deletes the five rows each times from the table.

✓ **Statement Trigger:** Statement trigger fire only once for each statement. If row deleting is define as trigger event, when trigger file, deletes the five rows at once from the table.

- **Syntax of Trigger**

```
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 | FOR EACH STATEMENT [ WHEN Condition ]
        WHEN (condition)
DECLARE
        Declaration-statements
BEGIN
        Executable-statements
EXCEPTION
        Exception-handling-statements
END;
```

- ✓ **CREATE [OR REPLACE] TRIGGER trigger_name**: It creates or replaces an existing trigger with the trigger_name.
- ✓ **{BEFORE | AFTER | INSTEAD OF} :** This specifies when the trigger would be executed. The INSTEAD OF clause is used for creating trigger on a view.
- ✓ {**INSERT [OR] | UPDATE [OR] | DELETE}:** This specifies the DML operation.
- ✓ **[OF col_name]:** This specifies the column name that would be updated.
- ✓ **[ON table_name]:** This specifies the name of the table associated with the trigger.
- ✓ **[REFERENCING OLD AS o NEW AS n]:** This allows you to refer new and old values for various DML statements, like INSERT, UPDATE, and DELETE.
- ✓ **[FOR EACH ROW]:** This specifies a row level trigger, i.e., the trigger would be executed for each row being affected. Otherwise the trigger will execute just once when the SQL statement is executed, which is called a table level trigger.
- ✓ **WHEN (condition):** This provides a condition for rows for which the trigger would fire. This clause is valid only for row level triggers

IMPLEMENTATION:

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.

Note: Instructor will Frame the problem statement for writing PL/SQLblock for all types of Triggers in line with above statement.

CONCLUSION:

QUESTIONS:

1. What is a trigger?
2. What are Benefits of Triggers?
3. What are Row triggers and Statement triggers?
4. Why are we using Before and After triggers?
5. What is Insert, Update and Delete triggers?