**Batch:C1 Roll No.:16010122221**

**Experiment / assignment / tutorial No. 6**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title: Queries based on Triggers** |

**Objective:** To be able to use trigger on table.

**Expected Outcome of Experiment:**

CO 3 : Use SQL for Relational database creation, maintenance and query processing

**Books/ Journals/ Websites referred:**

1. Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g.Black book, Dreamtech Press

2. www.db-book.com

3. Korth, Slberchatz, Sudarshan : “Database Systems Concept”, 5th Edition , McGraw

Hill

4. Elmasri and Navathe,”Fundamentals of database Systems”, 4th Edition,PEARSON

Education.

**Resources used:** Postgresql

**Theory**

**Triggers** are database call-back functions, which are automatically performed/invoked when a specified database event occurs.

**Triggers** can be specified to fire

* Before the operation is attempted on a row (before constraints are checked and the INSERT, UPDATE or DELETE is attempted)
* After the operation has completed (after constraints are checked and the INSERT, UPDATE, or DELETE has completed)
* Instead of the operation (in the case of inserts, updates or deletes on a view)

The basic syntax of creating a trigger is as follows −

CREATE TRIGGER trigger\_name [BEFORE|AFTER|INSTEAD OF] event\_name

ON table\_name

[

-- Trigger logic goes here....

];

event\_name could be INSERT, DELETE, UPDATE, and TRUNCATE database operation on the mentioned table table\_name. You can optionally specify FOR EACH ROW after table name.

The following is the syntax of creating a trigger on an UPDATE operation on one or more specified columns of a table as follows −

CREATE TRIGGER trigger\_name [BEFORE|AFTER] UPDATE OF column\_name

ON table\_name

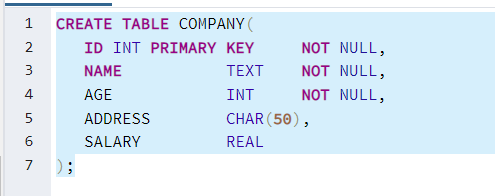
[

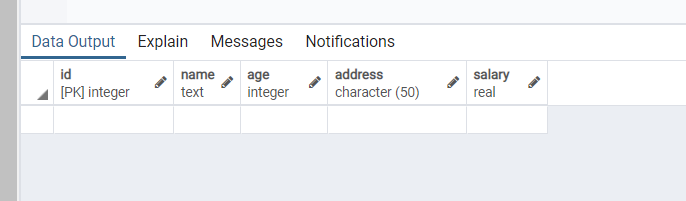
-- Trigger logic goes here....

];

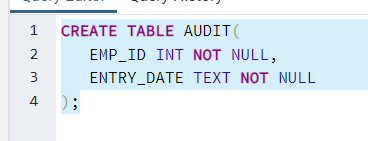
**Implementation Screenshots (Problem Statement, Query and Screenshots of Results):**

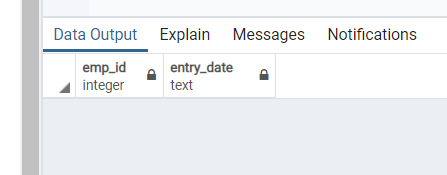
Created a table COMPANY



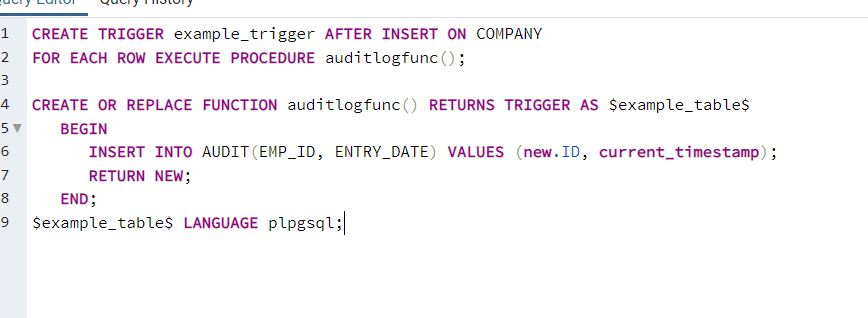


Created table audit :

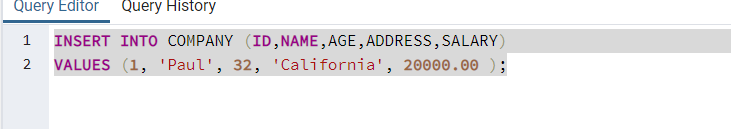


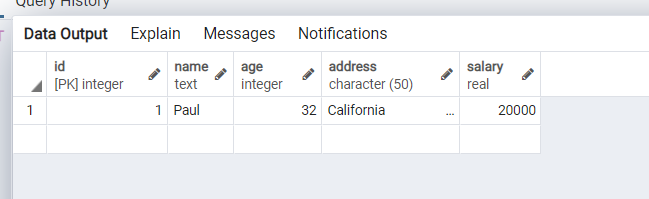


Created a trigger on COMPANY table

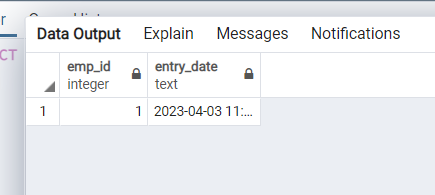


Added values to COMPANY

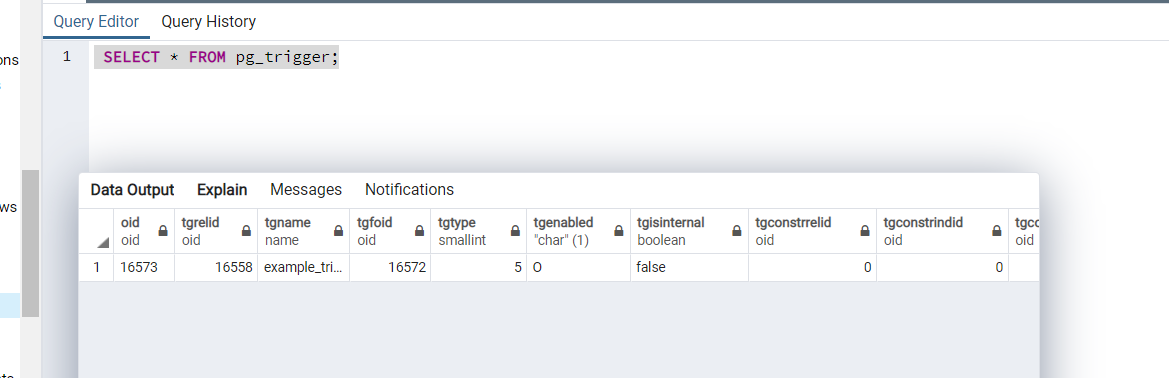




The audit table gets updated :



Listing all triggers :



# Conclusion:

# The given exp on triggers were learned and implemented successfully.

# Post Lab Questions:

# Write a trigger to count number of new tuples inserted using each insert statement.

# CREATE TRIGGER count\_inserts AFTER INSERT ON table\_name

# FOR EACH ROW

# BEGIN

# SET @num\_inserts = @num\_inserts + 1;

# END;

# Trigger is special type of \_\_\_\_\_\_\_\_\_\_ procedure.

# a) Stored

# b) Function

# c) View

# d) Table

# Triggers can be enabled or disabled with the \_\_\_\_\_\_\_\_ statement.

# a) ALTER TABLE statement

# b) DROP TABLE statement

# c) DELETE TABLE statement

# d) None of the mentioned