**Batch: A1 Roll No.: 1911004**

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

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

|  |
| --- |
| **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

[

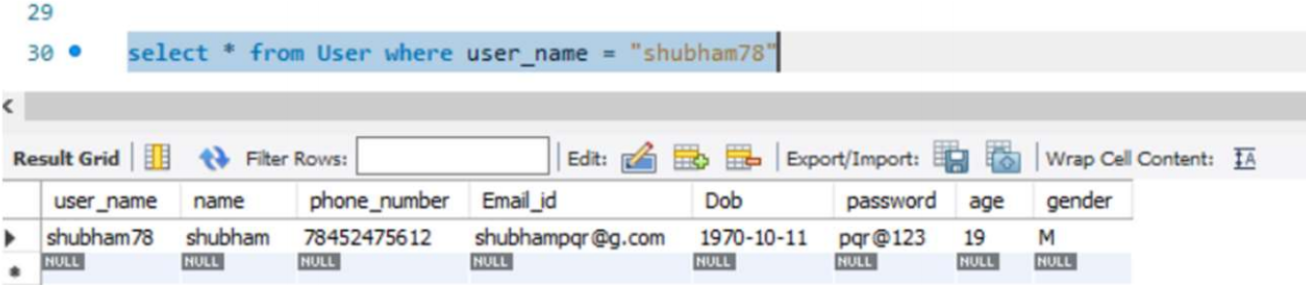
-- Trigger logic goes here....

];

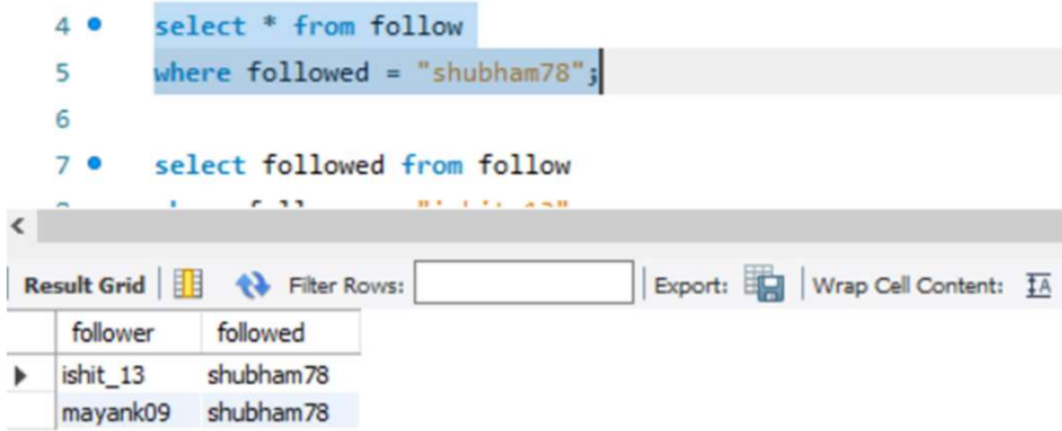
**Implementation Screenshots (Problem Statement, Query and Screenshots of Results):**

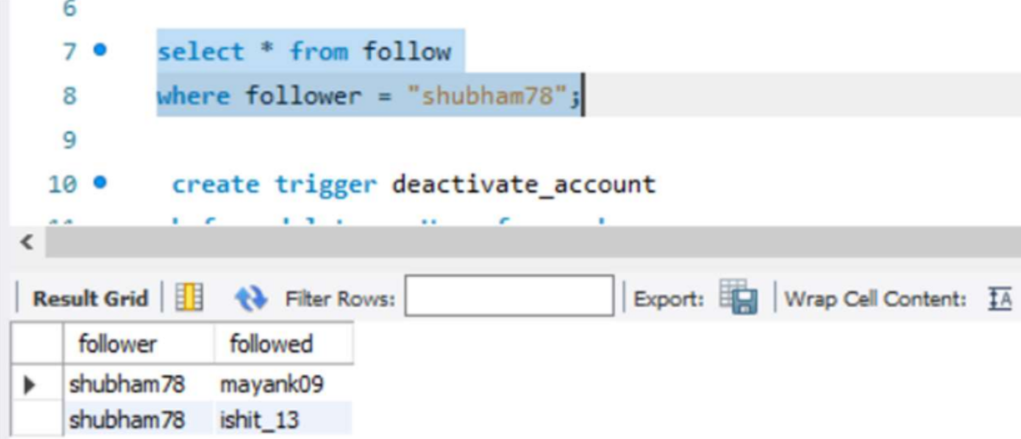
1. **Enter details of a user in the User table**

insert into User(user\_name,name,phone\_number,Email\_id,Dob,password,age,gender)values("shubham78","shubham","78452475612","shubhampqr@g.com","1970-10-11","pqr@123","19","M");



1. **Adding some followers and following some users from the selected user.**





1. **Create trigger for storing user details into another table when user wants to deactivate.**

create trigger deactivate\_account before delete on User for each row

insert into deactivated select \* from User where user\_name = old.user\_name ;

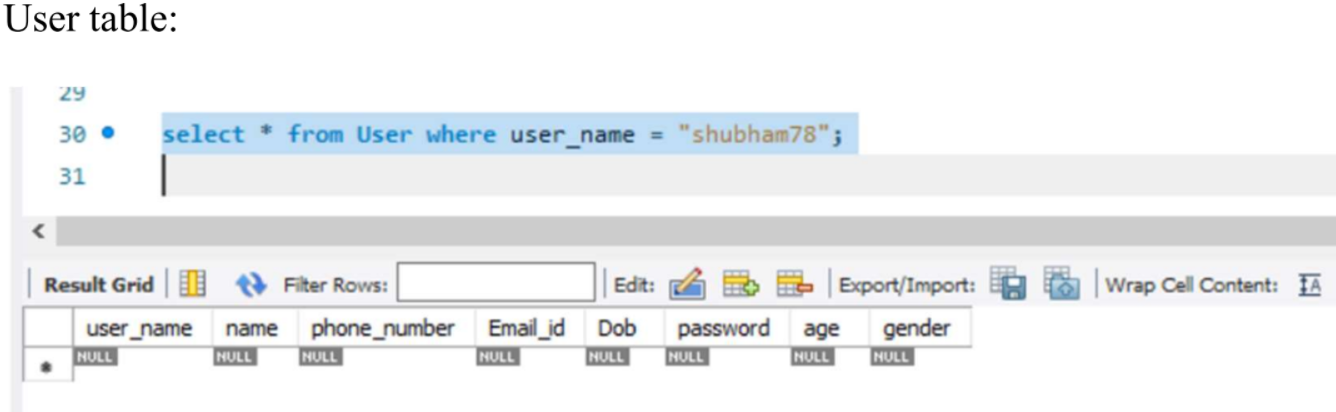
1. **Create trigger to store all instances of the user = “shubham78” in the follow table to another table.**

create trigger upd\_follow before delete on User for each row

insert into deactivate\_data select \* from follow where follower = old.user\_name or followed = old.user\_name;

1. **Delete the user shubham78.**

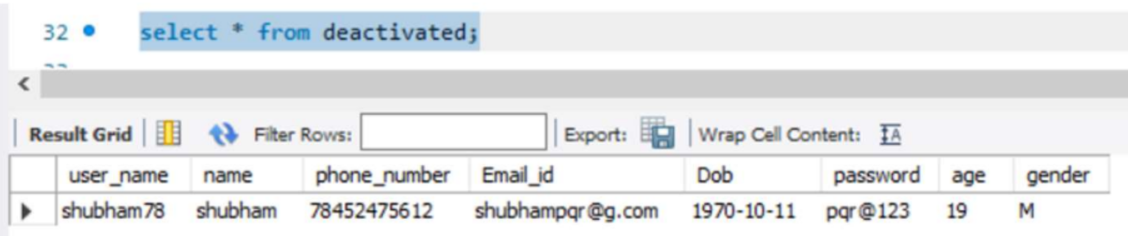
delete from User where user\_name = "shubham78";

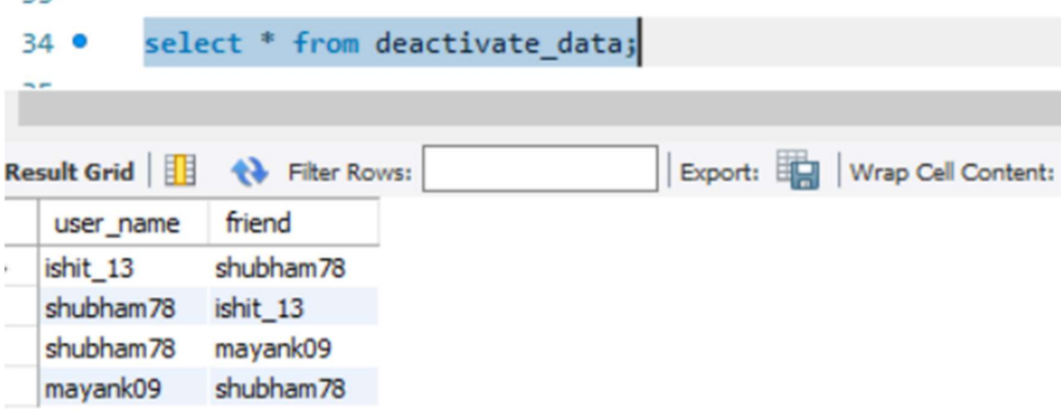






The details have been moved to another tables





Hence we created 2 triggers to store the data of deactivated users in another table and also to remove the information from friends and following table but to keep them in another table to be returned to main table when the user re-activates the account.

# Conclusion:

# In this experiment we successfully implemented a trigger function. We learnt about various different kinds of triggers.

# Post Lab Questions:

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

# Declare count int

# Set count=0;

# delimiter $$

# CREATE TRIGGER Count\_tupples

# AFTER INSERT ON employee

# FOR EACH ROW

# BEGIN

# SET count = count + 1;

# END; $$

# delimiter;

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

# a) Stored

# b) Function

# c) View

# d) Table

# Ans => a) Stored: A trigger is a special kind of store procedure that executes in response to certain action on the table like insertion, deletion or updation of data.

# 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

# Ans => a) ALTER TABLE Statement : You can also use the All keyword instead of a trigger name to enable/disable all of the triggers on a table in question.