Government College of Engineering, Jalgaon
(An Autonomous Institute of Government of Maharashtra)

Name : Dhas Prietysudha Machhindranath **Semester :** V **PRN :** 2141018
Class : T. Y. B.Tech Computer **Academic Year :** 2023-24 **Subject :** CO307U
Course Teacher : Mr. Vinit Kakde
Date of Performance : **Date of Completion :**

Practical no. 7

Aim : Create triggers for the given situations and update the values of the required variables using PL\SQL.

Theory:

Trigger in SQL

In this article, you will learn about the trigger and its implementation with examples.

A Trigger in Structured Query Language is a set of procedural statements which are executed automatically when there is any response to certain events on the particular table in the database. Triggers are used to protect the data integrity in the database.

In SQL, this concept is the same as the trigger in real life. For example, when we pull the gun trigger, the bullet is fired.

To understand the concept of trigger in SQL, let's take the below hypothetical situation:

Suppose Mahendra is the human resource manager in a multinational company. When the record of a new employee is entered into the database, he has to send the 'Congrats' message to each new employee. If there are four or five employees, Mahendra can do it manually, but if the number of new Employees is more than the thousand, then in such condition, he has to use the trigger in the database.

Thus, now Mahendra has to create the trigger in the table, which will automatically send a 'Congrats' message to the new employees once their record is inserted into the database.

The trigger is always executed with the specific table in the database. If we remove the table, all the triggers associated with that table are also deleted automatically.

In Structured Query Language, triggers are called only either before or after the below events:

1. INSERT Event: This event is called when the new row is entered in the table.
2. UPDATE Event: This event is called when the existing record is changed or modified in the table.
3. DELETE Event: This event is called when the existing record is removed from the table.

Types of Triggers in SQL

Following are the six types of triggers in SQL:

1. AFTER INSERT Trigger
   This trigger is invoked after the insertion of data in the table.

2. AFTER UPDATE Trigger
   This trigger is invoked in SQL after the modification of the data in the table.

3. AFTER DELETE Trigger
   This trigger is invoked after deleting the data from the table.

4. BEFORE INSERT Trigger
   This trigger is invoked before the inserting the record in the table.

5. BEFORE UPDATE Trigger
   This trigger is invoked before the updating the record in the table.

6. BEFORE DELETE Trigger
   This trigger is invoked before deleting the record from the table.

Syntax of Trigger in SQL

1. CREATE TRIGGER Trigger_Name
2. [BEFORE | AFTER] [Insert | Update | Delete]
3. ON [Table_Name]
4. [FOR EACH ROW | FOR EACH COLUMN]
5. AS
6. Set of SQL Statement

In the trigger syntax, firstly, we have to define the name of the trigger after the CREATE TRIGGER keyword. After that, we have to define the BEFORE or AFTER keyword with anyone event.
Then, we define the name of that table on which trigger is to occur.
After the table name, we have to define the row-level or statement-level trigger.
And, at last, we have to write the SQL statements which perform actions on the occurring of event.

Program and Output:

```
%%sql

SET @total_pets = 0;
SET @total_owners_income = 0;
```

```
### Create a trigger

The trigger should be called before every insert to the Pets table. The total number of Pets should be recorded in the variable total_pets defined above. It should be increased by one every time the trigger is called
```

```
%%sql
 CREATE TRIGGER total_pet_increase
 AFTER INSERT
```

```
ON Pets
FOR EACH ROW
BEGIN
  SET @total_pets = @total_pets + 1;
END;

insert into Pets values(1, 'Mou', 8, 'street 2', 'jalgaon', 425002, 'maharashtra', 'cat');
insert into Pets values(2, 'Hitu', 15, 'street 2', 'jalgaon', 425002, 'maharashtra', 'cat');

select @total_pets;
```

```
 * mysql+mysqlconnector://newuser:***@localhost:3306/CS350
0 rows affected.
1 rows affected.
t[31]:
 @total_pets
           2
```

Create another trigger

The trigger should be called before every insert to the Owners table. It should update the total_owners_income variable so that it contains the cumulative income of all pet owners after update.

```
%%sql
create trigger total_owner_income_increaser
before insert
on Owners
for each row
BEGIN
  SET @total_owners_income = @total_owners_income + new.AnnualIncome;
END;

INSERT INTO Owners(OID , Name , Street , City , ZipCode , State , Age , AnnualIncome) VALUES
(1, "Hasan Jamil", "Hampshire", "Moscow", 94110, "Idaho", 40, 100000);
INSERT INTO Owners(OID , Name , Street , City , ZipCode , State , Age , AnnualIncome) VALUES
(2, "Ronald Knuth", "Hampshire", "Moscow", 94110, "Idaho", 40, 1000);
INSERT INTO Owners(OID , Name , Street , City , ZipCode , State , Age , AnnualIncome) VALUES
(3, "Reginal Knuth", "Hampshire", "Moscow", 94110, "Idaho", 40, 1000);
select @total_owners_income;
```

```
 * mysql+mysqlconnector://newuser:***@localhost:3306/CS350
0 rows affected.
1 rows affected.
1 rows affected.
[47]:  @total_owners_income
                        102000
```

Conclusion:

This way we created triggers to update the variables @total_pets and @total_owners_income as per required conditions.

Questions and Answers:

1. **What is a trigger in SQL?**

Ans. A trigger is a special type of stored procedure that is automatically executed in response to certain events, such as the insertion or update of a row in a table. Triggers can be used to enforce business rules, validate data, and perform other actions in response to changes in the database.

2. **How are triggers activated in SQL?**

Ans. Triggers are activated in response to certain events, such as the insertion or update of a row in a table. These events are specified when the trigger is created using the AFTER, BEFORE, or INSTEAD OF keywords. For example, a trigger created with the AFTER INSERT clause will be activated after a new row is inserted into the table.

3. **What are the different types of triggers in SQL?**

Ans. There are two types of triggers in SQL: row-level triggers and statement-level triggers. Row-level triggers are activated once for each row affected by an insert, update, or delete statement. Statement-level triggers are activated once for each statement, regardless of the number of rows affected.

4. **Can triggers be nested in SQL?**

Ans. Yes, triggers can be nested in SQL. This means that a trigger can be activated by another trigger. However, there is a limit to the number of levels of trigger nesting that is allowed, and this limit is set by the database server.

5. **How do I disable or enable a trigger in SQL?**

Ans. To disable a trigger in SQL, you can use the DISABLE TRIGGER statement. For example:
DISABLE TRIGGER trigger_name ON table_name;
To enable a disabled trigger, you can use the ENABLE TRIGGER statement. For example:
ENABLE TRIGGER trigger_name ON table_name;

Course Teacher

Mr. Vinit Kakde