K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch: A2 **Roll No.: 1911027**

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):

Problem Statement : - Online pharmacy management system where user can order medicines from a particular pharmacy by comparing prices from different pharmacies available. And also checking availability of medicines in the pharmacy.

Trigger 1: - If any customer is making transaction of amount greater than 1000 then that customer will be given discount of 5% of the total amount and that amount will be deducted from total cost.

Trigger query : -

```
CREATE TRIGGER discount BEFORE INSERT ON transactions
FOR EACH ROW
BEGIN
IF NEW.total_cost>1000 THEN
        SET NEW.total_cost=NEW.total_cost*0.95;
END IF;
END
```

Results : - The total cost for the particular transactions is 2000 (>1000) so this customer will be getting discount of 5%. So after executing below query a new transaction is added to the transaction table with 5 % discount on total cost.

insert into Transactions values(201,3,14,1,1,2000,'NonePhrama',35,'Kritarth','Jain');

After executing above query : - Transactions table : -

Select * From transactions;

Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
164	1	12	1	3	100	AbcPhrama	33	Nayan	Mandliya
166	2	13	2	1	200	XyzPhrama	34	Hussein	Motiwala
168	3	14	1	1	500	NonePhrama	35	Kritarth	Jain
200	3	14	1	1	5000	NonePhrama	35	Kritarth	Jain
201	3	14	1	1	1900	NonePhrama	35	Kritarth	Jain

For the query given below total cost is less than 1000 so this particular customer will not get any discount.

insert into Transactions values(202,3,14,1,1,100,'NonePhrama',35,'Kritarth','Jain');

After running above query : - Transactions table : -
Select * From transactions;

Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
164	1	12	1	3	100	AbcPhrama	33	Nayan	Mandliya
166	2	13	2	1	200	XyzPhrama	34	Hussein	Motiwala
168	3	14	1	1	500	NonePhrama	35	Kritarth	Jain
200	3	14	1	1	5000	NonePhrama	35	Kritarth	Jain
201	3	14	1	1	1900	NonePhrama	35	Kritarth	Jain
202	3	14	1	1	100	NonePhrama	35	Kritarth	Jain
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trigger 2 : - Calculating total cost from price and quantity of medicines and then updating total cost of transaction into transaction table also decreasing quantity of medicine accordingly from the medicine table.

Trigger Query : -

```
CREATE TRIGGER update_medicine BEFORE INSERT ON transactions
FOR EACH ROW
BEGIN
SET @t=(SELECT max(price) from medicine where Med_ID=NEW.M_ID);
SET NEW.total_cost=NEW.quantity*@t;
UPDATE medicine SET Quantity=Quantity-NEW.quantity WHERE Med_ID=NEW.M_ID;
END
```

Before insert : - Medicine Table : -
Select * from medicine;

Med_ID	Med_Name	Med_Company	Quantity	Price	Med_Type	Pharma_Name	Exp_Date	Phar_ID	Cus_ID
1	Crocin	intel	200	15	headache	AbcPharma	2022-12-12	12	1
2	Paracetemol	AMD	30	100	Antibiotic	XyzPharma	2023-12-12	13	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

After insert into transactions table : - For med id 2 quantity is 30 but after executing query given below quantity is changed to 10 as a customer has ordered 20 packs of medicines and also using the price and quantity of medicine total cost is also calculated.

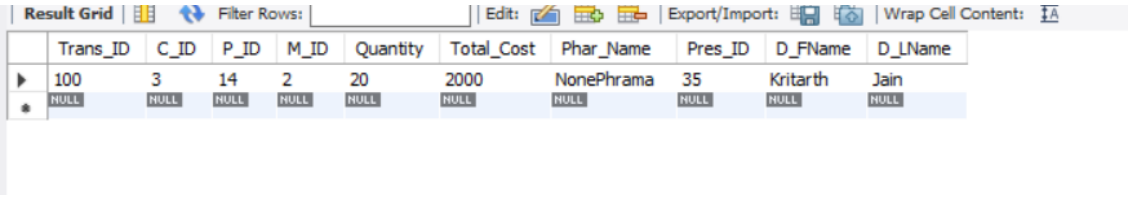
```
insert into Transactions values(100,3,14,2,20,0,'NonePhrama',35,'Kritarth','Jain');
```

Medicine table : -
Select * from medicine;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Med_ID	Med_Name	Med_Company	Quantity	Price	Med_Type	Pharma_Name	Exp_Date	Phar_ID	Cus_ID
1	Crocin	intel	200	15	headache	AbcPharma	2022-12-12	12	1
2	Paracetemol	AMD	10	100	Antibiotic	XyzPharma	2023-12-12	13	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Transactions table : -
Select * from transactions;

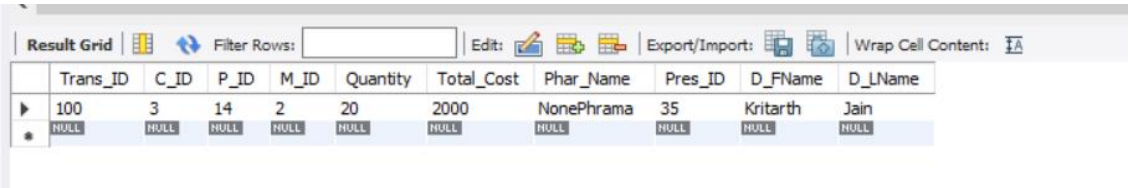


Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
100	3	14	2	20	2000	NonePhrama	35	Kritarth	Jain
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trigger 3 : - Whenever some transactions are retracted then quantity of medicines are updated into medicine table.

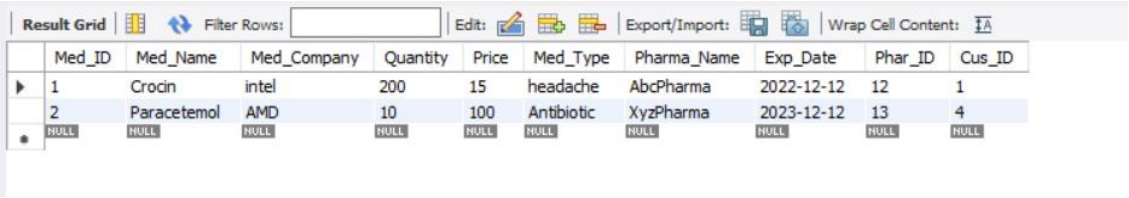
```
CREATE TRIGGER med_update BEFORE DELETE ON transactions
FOR EACH ROW
BEGIN
UPDATE medicine SET Quantity=Quantity+OLD.quantity WHERE Med_ID=OLD.M_ID;
END
```

Before delete : -
Transactions Table : -
Select * from transactions;



Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
100	3	14	2	20	2000	NonePhrama	35	Kritarth	Jain
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Medicine Table : -
Select * from transactions;

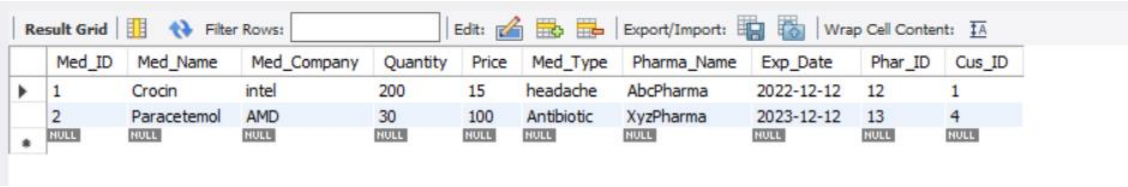


Med_ID	Med_Name	Med_Company	Quantity	Price	Med_Type	Pharma_Name	Exp_Date	Phar_ID	Cus_ID
1	Crocin	intel	200	15	headache	AbcPharma	2022-12-12	12	1
2	Paracetemol	AMD	10	100	Antibiotic	XyzPharma	2023-12-12	13	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

After deletion : - Quantity of medicine with ID 2 is 10 when a customer cancel a particular order with med id 2 then that particular quantity of medicine must be added to medicines quantity in the medicine table.

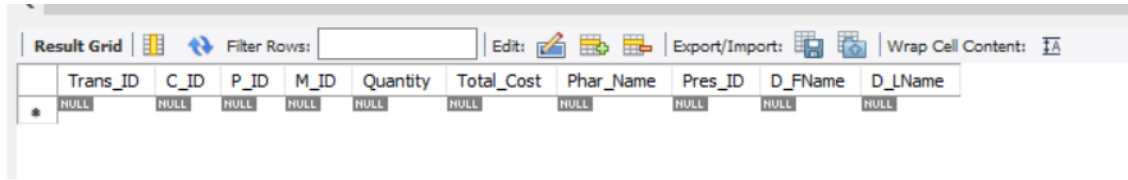
DELETE FROM transactions WHERE Trans_ID=100;

Medicine Table : -
Select * from medicine;



Med_ID	Med_Name	Med_Company	Quantity	Price	Med_Type	Pharma_Name	Exp_Date	Phar_ID	Cus_ID
1	Crocin	intel	200	15	headache	AbcPharma	2022-12-12	12	1
2	Paracetemol	AMD	30	100	Antibiotic	XyzPharma	2023-12-12	13	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Transactions Table : -
Select * from transactions;



Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Conclusion: By performing this experiment understood the concept of triggers in MySql and also successfully implemented triggers on our case study. Also tried to implement various triggers.

Post Lab Questions:

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

ANS)

Declare count int

Set count=0;

```
CREATE TRIGGER Cou_tup

        AFTER INSERT ON employee

FOR EACH ROW

BEGIN

SET count = count + 1;

END;
```

2. Trigger is special type of _____ procedure.

a) Stored
b) Function
c) View
d) Table
ANS) a) Stored

3. 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