

**20MCA132 OBJECT ORIENTED**  
**PROGRAMMING LAB**

**ASSIGNMENT**

SUBMITTED BY

VIVIN V. ABRAHAM  
R MCA-2020-S2  
ROLL NO : 42

SUBMITTED TO ,

Lisha Miss

# **Question:1**

## **AIM**

Create a Trigger for employee table it will update another table salary while updating values

## **OBJECTIVE**

To develop and execute a Trigger for After update/Delete/Insert operations on a table

## **PROCEDURE**

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

## **PROGRAM**

**sql>**

```
CREATE TABLE `employee` (  
  `emp_id` int(11) NOT NULL,  
  `emp_name` varchar(45) DEFAULT NULL,  
  `dob` date DEFAULT NULL,  
  `address` varchar(45) DEFAULT NULL,  
  `designation` varchar(45) DEFAULT NULL,  
  `mobile_no` int(11) DEFAULT NULL,  
  `dept_no` int(11) DEFAULT NULL,  
  `salary` int(11) DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

**Sql>**

```
CREATE TABLE `salary` (  
  `employee_id` int(11) NOT NULL,  
  `old_sal` int(11) DEFAULT NULL,  
  `new_sal` int(11) DEFAULT NULL,  
  `rev_date` date DEFAULT NULL,  
  PRIMARY KEY (`employee_id`)  
);
```

**sql>**

```
CREATE DEFINER=`root`@`localhost` TRIGGER `db1`.`personal_updatations_AFTER_UPDATE_1`  
AFTER UPDATE ON `employee`  
FOR EACH ROW  
BEGIN  
  if(new.salary != old.salary)  
  then  
    INSERT INTO salary (employee_id,old_sal,new_sal,rev_date) values  
    (new.emp_id,old.salary,new.salary,sysdate());  
  END if;  
end;
```

**sql>**

```
update employee set salary=234569 where emp_id=1;  
select * from salary;
```

MySQL Workbench

Local instance MySQL80 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

db1

Tables

employee

Columns

Indexes

Foreign Keys

Triggers

personal\_updates

promotions

salary

Columns

Indexes

Foreign Keys

Triggers

student

Views

Stored Procedures

Administration Schemas

Information

Table: employee

Columns:

- emp\_id int(11) PK
- emp\_name varchar(45)
- dob date
- address varchar(45)
- designation varchar(45)
- mobile\_no int(11)
- dept\_no int(11)
- salary int(11)

Result Grid

employee_id	old_sal	new_sal	rev_date
1	10000	10000	2021-08-17
2	15000	10000	2021-08-17

salary 8 x

Apply Revert Context Help Snippets

Output

#	Time	Action	Message	Duration / Fetch
136	17:01:52	Apply changes to employee		
137	17:02:02	Apply changes to employee	Error 1359: Trigger 'db1.employee_AFTER_UPDATE_1' already exists SQL Statement: CREATE DEFINER='ro...	
138	17:16:52	SELECT * FROM db1.employee LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
139	17:17:19	ANALYZE TABLE 'db1'.employee	OK	0.000 sec
140	17:22:38	update employee set salary=1290 where emp_id=1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
141	17:22:38	select * from salary LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

30°C Rain showers 05:23 PM 17-08-2021

# Question:2

## AIM

Create a Trigger for employe table it will update another table personal\_updates while updating values

## OBJECTIVE

To develop and execute a Trigger for Before and After update/Delete/Insert operations on a table

## PROCEDURE

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

## PROGRAM

sql>

```
CREATE TABLE `employe` (  
  `emp_id` int(11) NOT NULL,  
  `emp_name` varchar(45) DEFAULT NULL,  
  `dob` date DEFAULT NULL,  
  `address` varchar(45) DEFAULT NULL,  
  `designation` varchar(45) DEFAULT NULL,  
  `mobile_no` int(11) DEFAULT NULL,  
  `dept_no` int(11) DEFAULT NULL,  
  `salary` int(11) DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

Sql>

```
CREATE TABLE `personal_updates` (  
  `emp_id` int(11) NOT NULL,  
  `old_phoneno` int(11) DEFAULT NULL,  
  `new_phoneno` int(11) DEFAULT NULL,  
  `rev_date` date DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER `db1`.`personal_updates_AFTER_UPDATE`  
AFTER UPDATE ON `employe`  
FOR EACH ROW  
BEGIN  
  if(new.mobile_no != old.mobile_no)  
  then  
    INSERT INTO personal_updates (emp_id,old_phoneno,new_phoneno,rev_date) values  
    (new.emp_id,new.mobile_no,old.mobile_no,sysdate());  
  END if;  
end;
```

sql>

```
update employe set mobile_no=34566 where emp_id=4 ;
```

```
select * from personal_updates;
```

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigation: Administration - Server Status db1 personal\_updates employee - Table

SQL File 1: x

```
1 • update employee set mobile_no=34566 where emp_id=4 ;
2 • select * from personal_updates;
```

Limit to 1000 rows

Result Grid

emp_id	old_phoneno	new_phoneno	rev_date
1	2234568	12345	2021-08-17
4	34566	211435	2021-08-17

Information

Connection Details

Name: Local instance  
Host: MySQL80  
Port: localhost  
Port: 3306  
Login User: root  
Current User: root@localhost  
SSL cipher: SSL not used

Server

Product: mariadb.org binary distribution  
Version: 10.4.19-MariaDB  
Connector Version: C++ 8.0.26

Output

#	Time	Action	Message	Duration / Fetch
2	23:16:16	Apply changes to employee	No changes detected	
3	23:16:24	Apply changes to employee	No changes detected	
4	23:18:04	update employee set mobile_no=34566 where empname='joice'	Error Code: 1054. Unknown column 'empname' in 'where clause'	0.000 sec
5	23:18:18	update employee set mobile_no=34566 where emp_name='joice'	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses...	0.000 sec
6	23:19:00	update employee set mobile_no=34566 where emp_id=4	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
7	23:19:00	select * from personal_updates LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

24°C AQI 89 11:19 PM 17-08-2021

# Question:3

## AIM

Create a Trigger for employe table it will update another table promotions while updating values

## OBJECTIVE

To develop and execute a Trigger for Before and After update/Delete/Insert operations on a table

## PROCEDURE

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

## PROGRAM

sql>

```
CREATE TABLE `employe` (  
  `emp_id` int(11) NOT NULL,  
  `emp_name` varchar(45) DEFAULT NULL,  
  `dob` date DEFAULT NULL,  
  `address` varchar(45) DEFAULT NULL,  
  `designation` varchar(45) DEFAULT NULL,  
  `mobile_no` int(11) DEFAULT NULL,  
  `dept_no` int(11) DEFAULT NULL,  
  `salary` int(11) DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

Sql>

```
CREATE TABLE `personal_updatations` (  
  `emp_id` int(11) NOT NULL,  
  `old_phoneno` int(11) DEFAULT NULL,  
  `new_phoneno` int(11) DEFAULT NULL,  
  `rev_date` date DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER `db1`.`employe_AFTER_UPDATE_1`  
AFTER UPDATE ON `employe`  
FOR EACH ROW  
BEGIN  
  if(new.designation != old.designation)  
  then  
    INSERT INTO promotions (emp_id,old_designation,new_designation,rev_date) values  
    (new.emp_id,new.designation,old.designation,sysdate());  
  END if;  
end;
```

sql>

```
update employe set designation='clk' where emp_id=4 ;
```

select \* from promotions;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Administration - Server Status db1 personal\_updates employee - Table

SQL File 1: x

```
1 • update employee set designation='ck' where emp_id=4 ;
2 • select * from promotions;
```

Limit to 1000 rows

Result Grid

emp_id	old_designation	new_designation	rev_date
1	rcb	ceo	2021-08-17
4	ck	ceo	2021-08-17

Filter Rows: [ ] Edit: Export/Import: Wrap Cell Content: [ ]

Result Grid Form Editor Field Types

promotions 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
4	23:18:04	update employee set mobile_no=34566 where empname='joice'	Error Code: 1054. Unknown column 'empname' in 'where clause'	0.000 sec
5	23:18:18	update employee set mobile_no=34566 where emp_name='joice'	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses...	0.000 sec
6	23:19:00	update employee set mobile_no=34566 where emp_id=4	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
7	23:19:00	select * from personal_updates LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
8	23:23:11	update employee set designation='ck' where emp_id=4	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
9	23:23:11	select * from promotions LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Connection Details

Name: Local instance  
Host: MySQL80  
Host: localhost  
Port: 3306  
Login User: root  
Current User: root@localhost  
SSL cipher: SSL not used

Server

Product: mariadb.org binary distribution  
Version: 10.4.19-MariaDB  
Connector: C++ 8.0.26

SELECT Syntax:

SELECT is used to retrieve rows selected from one or more tables, and can include [UNION](#) statements and subqueries. See [union](#), and [subqueries](#).

The most commonly used clauses of SELECT statements are these:

- Each **select\_expr** indicates a column that you want to retrieve. There must be at least one **select\_expr**.
- table\_references** indicates the table or tables from which to retrieve rows. Its syntax is described in [join](#).
- SELECT supports explicit partition selection using the **PARTITION** with a list of

Context Help Snippets

24°C Light rain 11:23 PM 17-08-2021