

ADVANCED DBMS

LABS

TRIGGER

TOM JOSEPH

MCA-1

S2

ROLL.NO 36

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 updating

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

step 6: stop

PROGRAM

sql>

```
CREATE TABLE `employee` (  
  `empno` 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 (`empno`)  
);
```

Sql>

```
CREATE TABLE `salary` (  
  `empno` int(11) NOT NULL,  
  `oldsalary` int(11) DEFAULT NULL,  
  `newsalary` int(11) DEFAULT NULL,  
  `revdate` date DEFAULT NULL  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER  
`class`.`personal_updates_AFTER_UPDATE_1`  
AFTER UPDATE ON `employee`  
FOR EACH ROW  
BEGIN  
  if(new.salary != old.salary)  
  then  
    INSERT INTO salary (empno,oldsalary,newsalary ,revdate) values  
    (new.empno,old.salary,new.salary,sysdate());  
  END if;  
end;
```

sql>
update employee set salary=5000 where empno=1;
select * from salary;

The screenshot shows the SQL Developer interface. On the left is the 'Navigator' pane with a tree view of schemas. The 'class' schema is expanded, showing 'Tables' and 'salary'. The 'salary' table is selected. The main pane shows 'Query 1' with the following SQL statements:

```
1 • update employee set salary=5000 WHERE empno=1;  
2 • SELECT * FROM employee;  
3 • select * from salary
```

The 'Result Grid' shows the results of the third query. It has columns: empno, oldsalary, newsalary, and revdate. The first row shows empno=1, oldsalary=9000, newsalary=5000, and revdate=2021-08-26 18:00:37. The second row shows NULL for all columns.

empno	oldsalary	newsalary	revdate
1	9000	5000	2021-08-26 18:00:37
NULL	NULL	NULL	NULL

AIM

Create a Trigger for employee 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 `employee` (  
  `empno` 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 (`empno`)  
);
```

Sql>

```
CREATE TABLE `personalupdatation` (  
  `empno` varchar(11) NOT NULL,  
  `old_phoneno` int(11) DEFAULT NULL,  
  `new_phoneno` int(11) DEFAULT NULL,  
  `revdate` date DEFAULT NULL,  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER  
`class`.`personal_updates_AFTER_UPDATE`  
AFTER UPDATE ON `employee`  
FOR EACH ROW  
BEGIN  
if(new.mobile_no != old.mobile_no)  
then  
INSERT INTO personalupdatation  
(empno,oldphone,newphone,revdate) values  
(new.empno,old.mobile_no,new.mobile_no,sysdate());  
END if; end;
```

sql>

update employee set mobile_no=5678901234 where emp_id=4 ;

select * from personalupdation;

MySQL Workbench

MySQLDBConnection1 - War... x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- bloodstore
- book
- classess
 - Tables
 - Views
 - Stored Procedures
 - Functions
- class
 - Tables
 - employee
 - personalupdation
 - salary
 - student
 - Views
 - Stored Procedures
 - Functions
- datas

Administration Schemas

Information

Table: personalupdation

Columns:

- empno varchar(45)
- oldphone varchar(45)
- newphone varchar(45)
- revdate varchar(45)

Query 1 employee employee - Table

Limit to 1000 rows

```
1 • SELECT * FROM class.employee;  
2 • update employee set mobile_no="5678901234" where empno=4;  
3 • select * from personalupdation;
```

Automatic context help is disabled for the current caret position

Result Grid

empno	oldphone	newphone	revdate
4	5678643	5678901234	2021-08-28 16:45:12

personalupdation6 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message
9	16:43:48	SELECT * FROM class.employee LIMIT 0, 1000	5 row(s) returned
10	16:44:05	update employee set mobile_no="5678901234" where empno=4	Error Code: 1054. Unknown column 'oldphone' in 'field list'
11	16:45:00	Apply changes to employee	Changes applied
12	16:45:06	Apply changes to employee	No changes detected
13	16:45:12	update employee set mobile_no="5678901234" where empno=4	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
14	16:45:16	select * from personalupdation LIMIT 0, 1000	1 row(s) returned

Object Info Session

AIM

Create a Trigger for employee 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 `employee` (  
  `empno` 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 (`empno`)  
);
```

Sql>

```
CREATE TABLE `promotoin` (  
  `empno` int(11) NOT NULL,  
  `old_designation` varchar(11) DEFAULT NULL,  
  `new_designation` varchar(11) DEFAULT NULL,  
  `revdate` date DEFAULT NULL,  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost`  
TRIGGER `classes`.`employee_AFTER_UPDATE_1`  
AFTER UPDATE ON `employee`  
FOR EACH ROW  
BEGIN  
  if(new.designation != old.designation)  
  then  
    INSERT INTO promotoin (empno,old_designation,new_designation,revdate) values  
    (new.empno,old.designation,new.designation,sysdate());  
  END if;  
end;
```

sql>

update employee set designation='cleark' where empno=2 ;

select * from promotoin;

MySQL Workbench

MySQLDBConnection1 - War... x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- bloodstore
- book
- classess
 - Tables
 - employee
 - personal_updatoin
 - promotoin
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - salaryupdated
 - student
- Views
- Stored Procedures

Administration Schemas

Information

Table: promotoin

Columns:

empno	varchar(45)
old_designation	varchar(45)
new_designation	varchar(45)
revdate	varchar(45)

Query 1 x

Limit to 1000 rows

```
1 update employee set designation='cleark' where empno=2 ;
2 select * from promotoin;
3
```

Result Grid

	empno	old_designation	new_designation	revdate
▶	2	analyst	cleark	2021-08-26 18:54:11

Form Editor

Field Types