ADVANCED DBMS LAB

TRIGGER

Submitted By:

Jobin Joseph

S2 RMCA-B

Roll no:01

AIM

Create a Trigger for employe 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 `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>

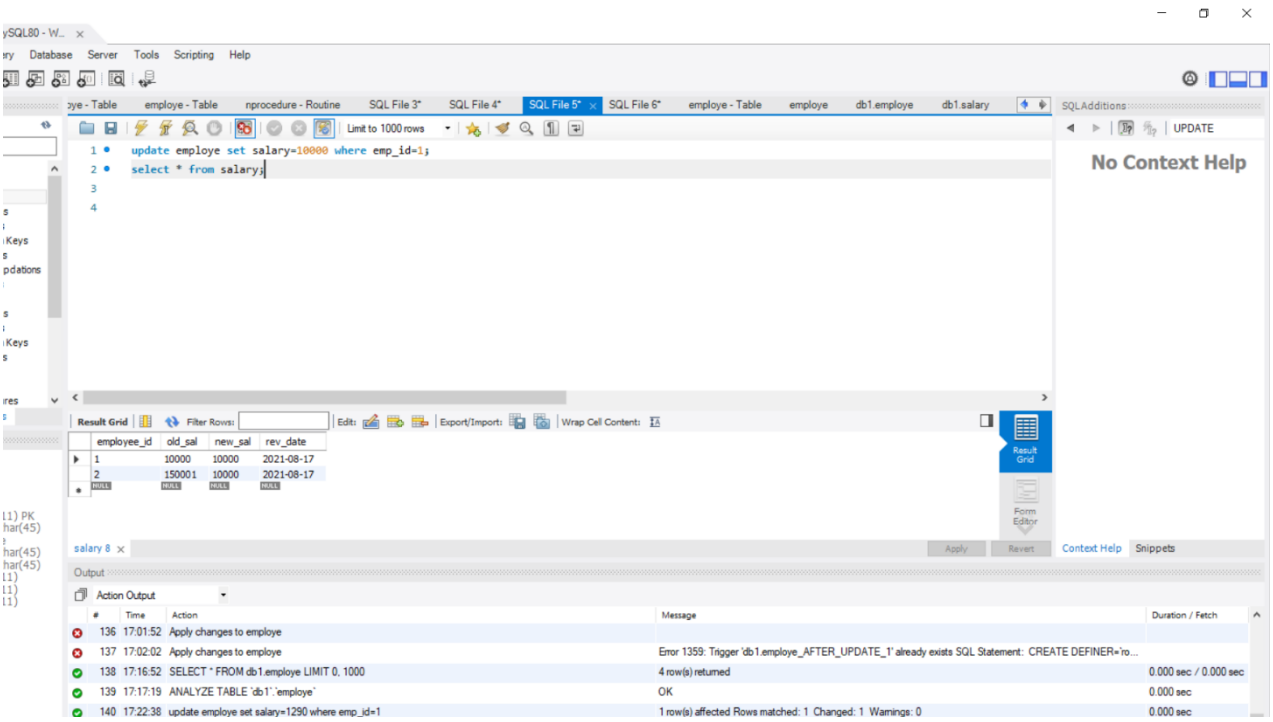
```
CREATE TABLE `salary` (
 `employee_id` int(11) NOT NULL,
 `old_sal` int(11) DEFAULT NULL,
 `new_sal` int(11) DEFAULT NULL,
 `rev_date` date DEFAULT NULL,
 PRIMARY KEY (`employee_id`)
);
```

sql>

```
 CREATE DEFINER=`root`@`localhost` TRIGGER
`db1`.`personal_updations_AFTER_UPDATE_1`
AFTER UPDATE ON `employe`
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 employe set salary=234569 where emp_id=1;
select * from salary;



AIM

Create a Trigger for employe table it will update another table personal_updations 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_updations` (
 `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`)
);
```

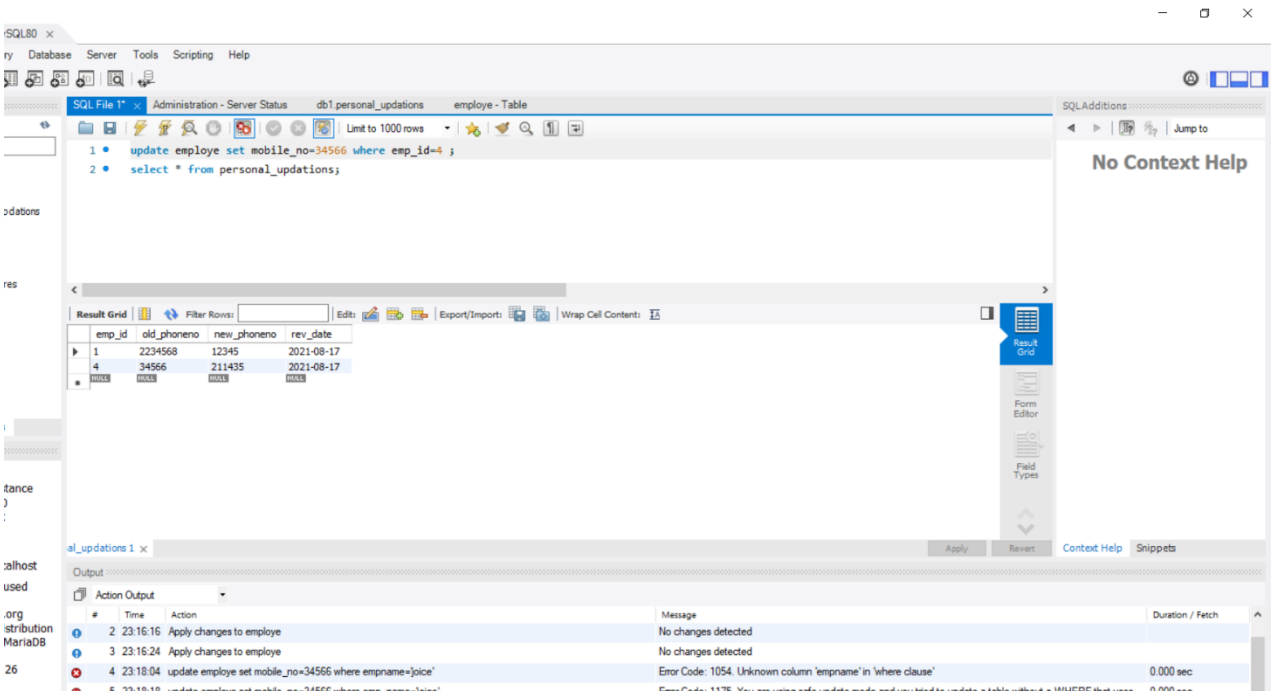
sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER
`db1`.`personal_updations_AFTER_UPDATE`
AFTER UPDATE ON `employe`
FOR EACH ROW
BEGIN
if(new.mobile_no != old.mobile_no)
 then
```

INSERT INTO personal_updations (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_updations;



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_updations` (
 `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;