**EXPERIMENT 8**

**AIM**

To have familiarize with trigger functions

**Question**

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: Carry out the operation on the table to check for trigger execution.

step 6: Stop

**PROGRAM**

Create Table employee:-

CREATE TABLE `employe` (

`emp\_id` int 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 DEFAULT NULL,

`dept\_no` int DEFAULT NULL,

`salary` int DEFAULT NULL,

PRIMARY KEY (`emp\_id`)

);

Create Table salary:-

CREATE TABLE `salary` (

`employee\_id` int NOT NULL,

`old\_sal` int DEFAULT NULL,

`new\_sal` int DEFAULT NULL,

`rev\_date` date DEFAULT NULL,

PRIMARY KEY (`employee\_id`)

);

Sql>

CREATE DEFINER=`root`@`localhost` TRIGGER

`employee\_db`.`employe\_AFTER\_UPDATE` 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());

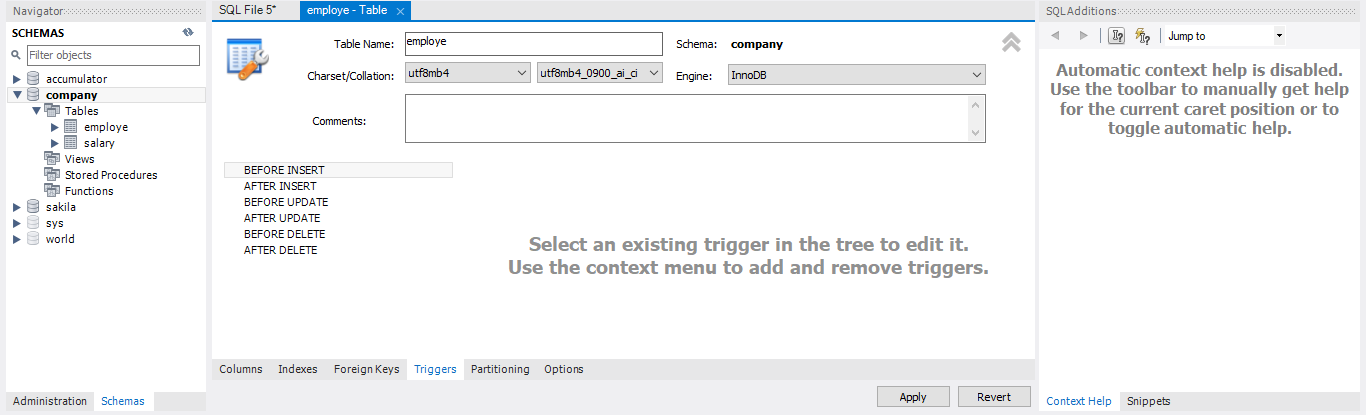
END if;

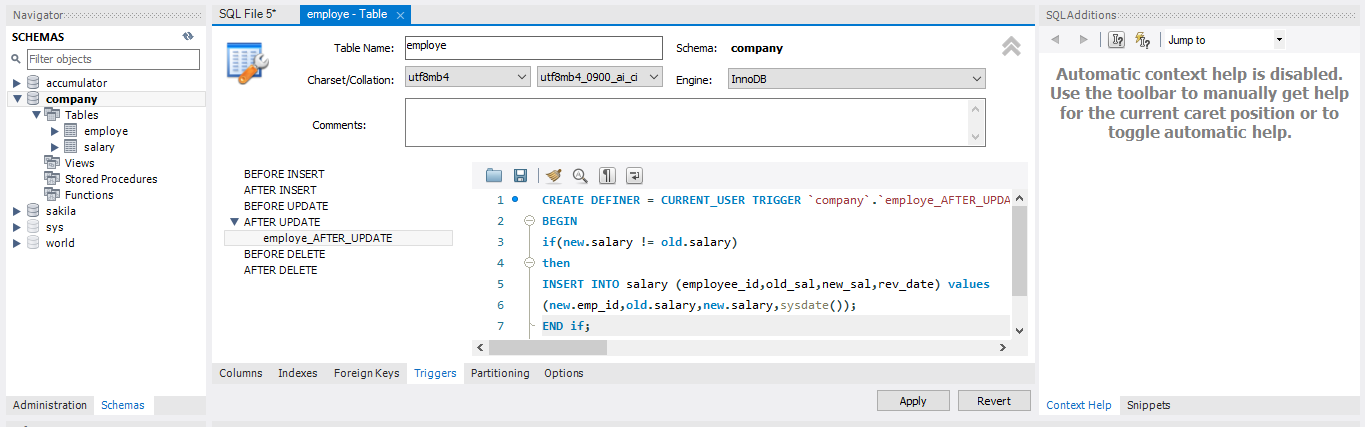
END

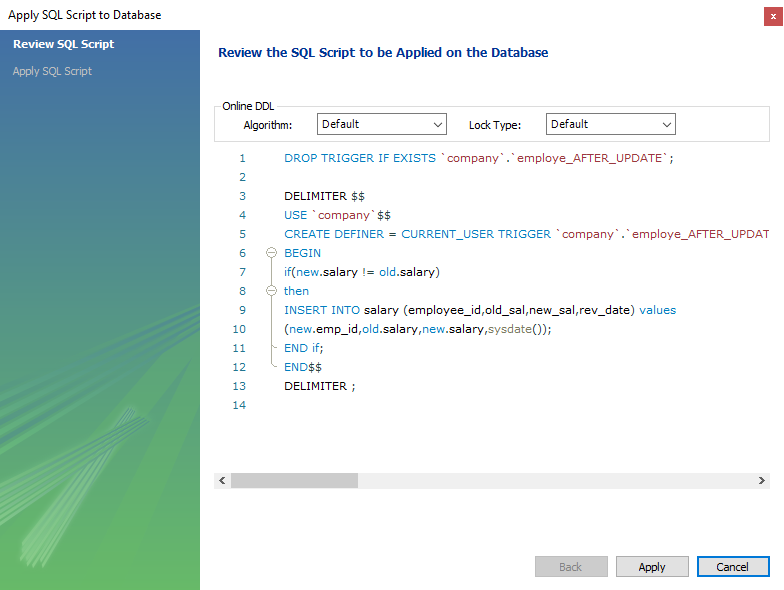
Sql>

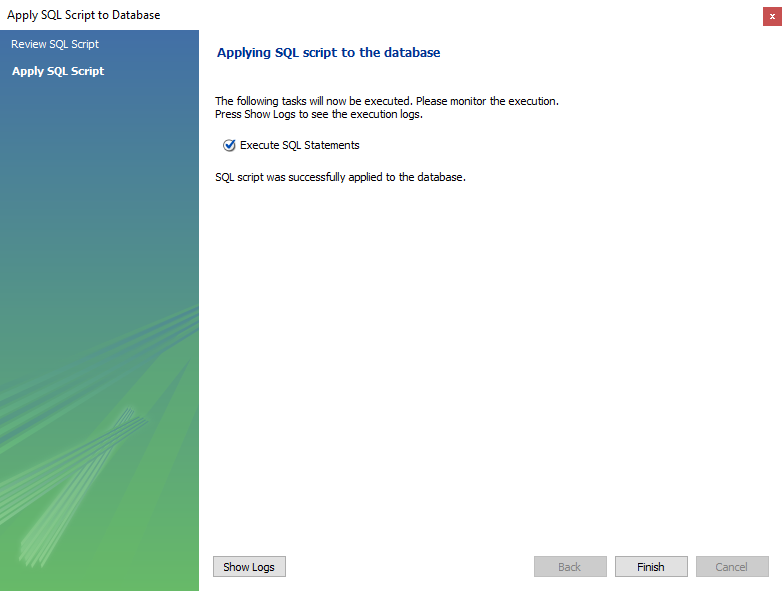
update employe set salary=234569 where emp\_id=1;

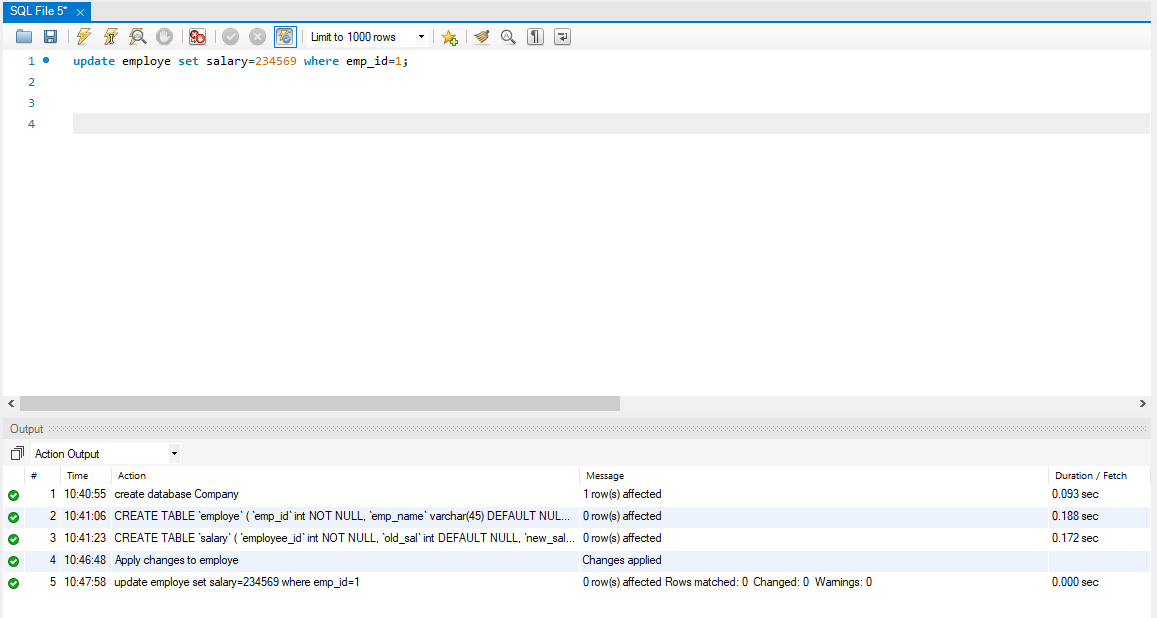
**OUTPUT:-**

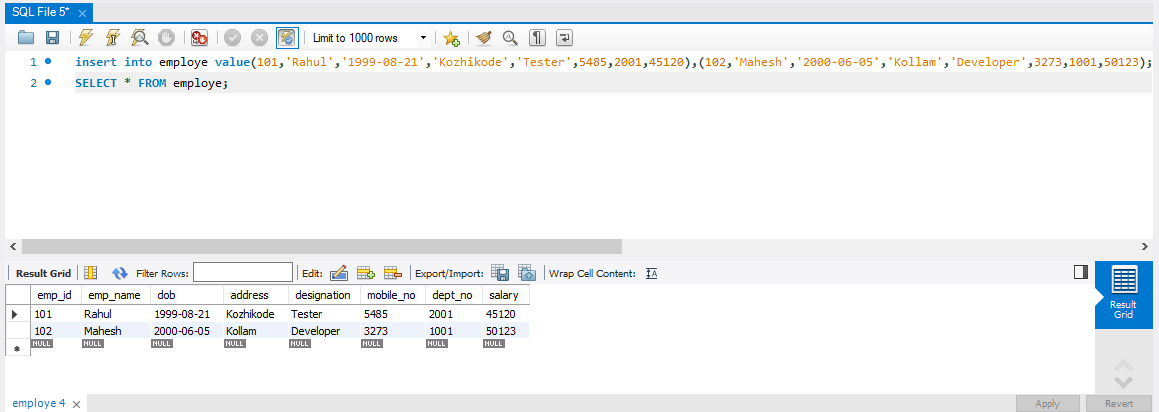
****

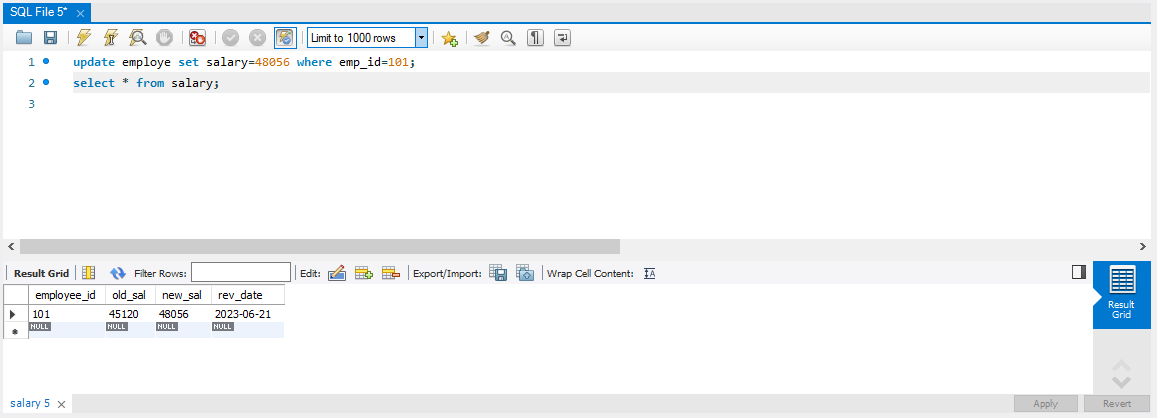
****

****

****

****

****

****

**EXPERIMENT 9**

DELIMITER //

CREATE PROCEDURE WhoBorrow()

BEGIN

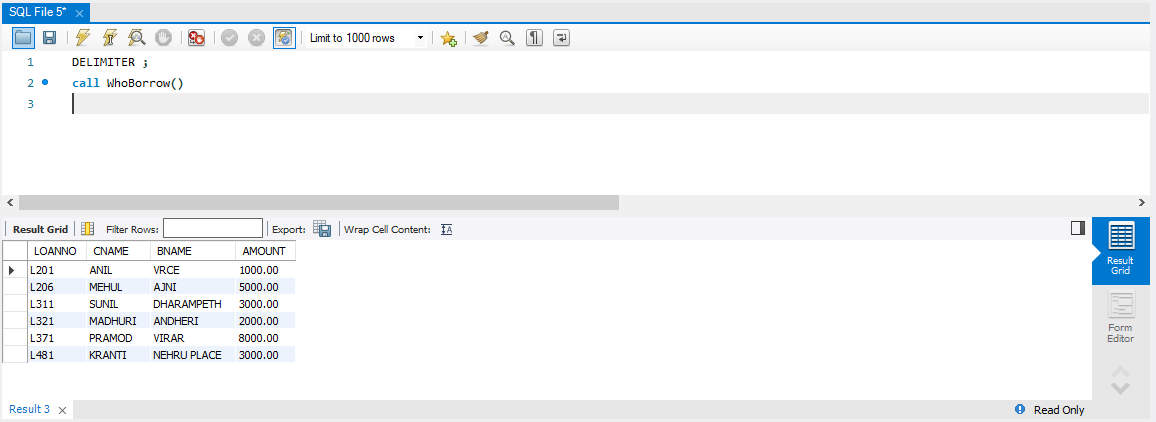
SELECT \* FROM BORROW;

END//

DELIMITER ;

call WhoBorrow()

**OUTPUT:-**



DELIMITER //

CREATE PROCEDURE `SP\_Sum`(

IN `n1` INT,

IN `n2` INT,

OUT `result` INT

)

BEGIN

Set result = n1 + n2;

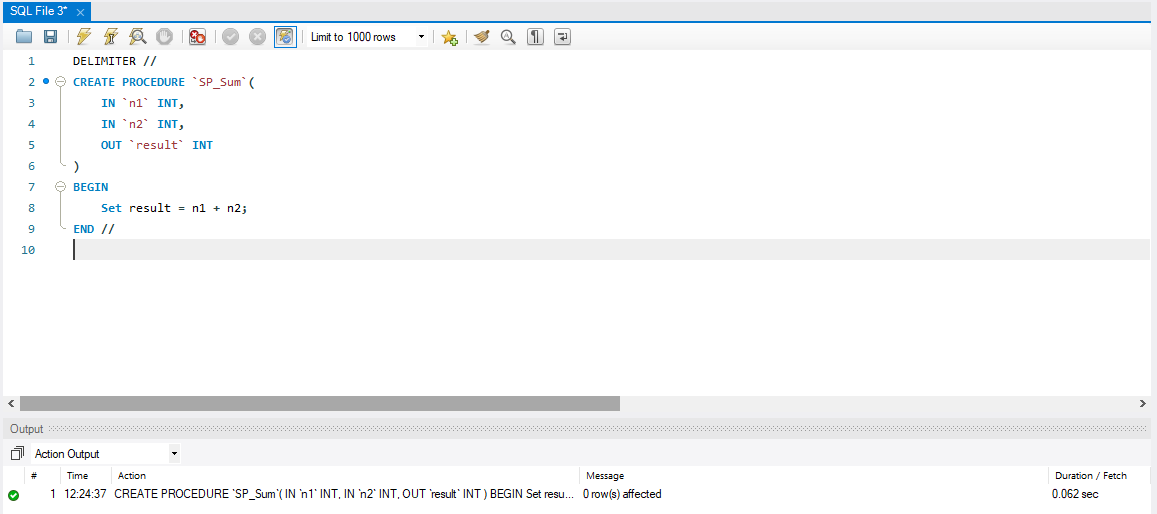
END //

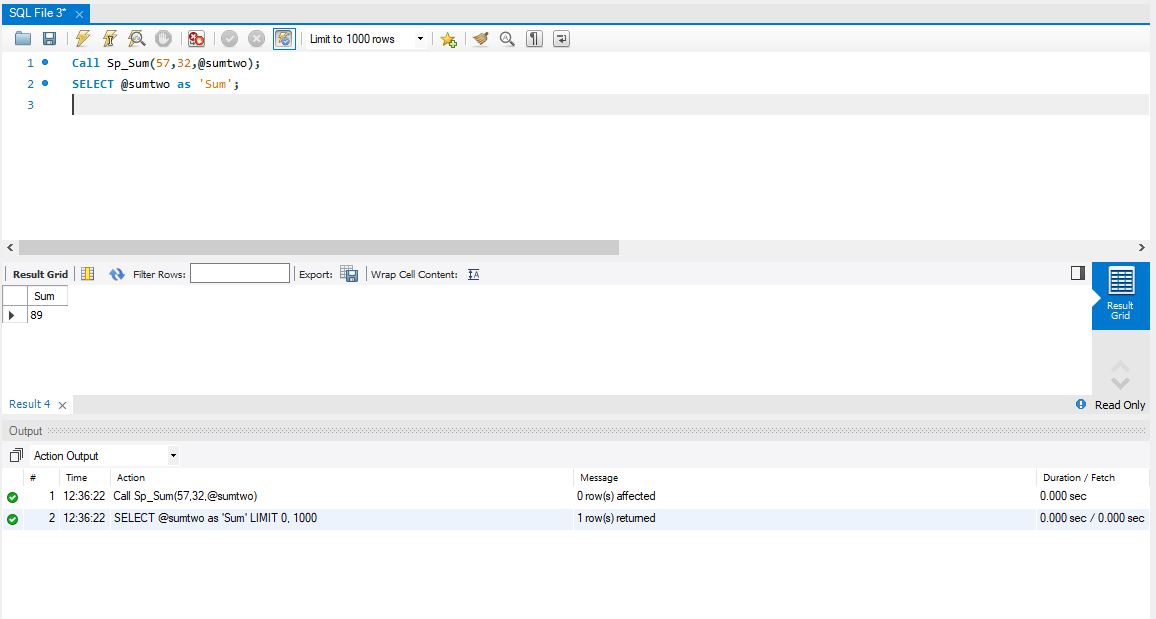
DELIMITER;

Call Sp(57,32,@sumtwo);

SELECT @sumtwo;

**OUTPUT:-**





**EXPERIMENT 10**

**AIM:**

CREATE TABLE account (acct\_num INT, amount DECIMAL(10,2))

1. Create a trigger which acts as an accumulator, summing the values inserted into one of the columns of the table.

2. define an UPDATE trigger that checks the new value to be used for updating each row, and modifies the value to be within the range from 0 to 100.

**PROCEDURE:**

1. create database accumulator;
2. create table account (acct\_num INT, amount DECIMAL(10,2));
3. create trigger insum

before insert

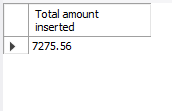
on account

for each row

set @sum = @sum + new.amount;

1. set @sum = 0;
2. insert into account values(101, 1541),(102,256),(103,5478.56);
3. select @sum as 'Total amount inserted';

**OUTPUT:-**

’

1. delimiter //

create trigger upd\_check

before update

on account

for each row

Begin

if new.amount < 0 then

set new.amount = 0;

elseif new.amount > 100 then

set new.amount = 100;

end if;

end;//

1. alter table account add primary key(acct\_num);
2. update account set amount = 123 where acct\_num = 101;
3. select \* from account;
4. update account set amount = -526 where acct\_num = 102;

**OUTPUT:-**

