**EXPERIMENT 10**

**PRE-LAB**

1. **Analyze the code and tell your observation?**  
     **DECLARE  
    a number(3) := 100;  
    BEGIN  
    IF (a = 50 ) THEN  
    dbms\_output.put\_line('Value of a is 10' );  
    ELSEIF ( a = 75 ) THEN  
    dbms\_output.put\_line('Value of a is 20' );  
    ELSE  
    dbms\_output.put\_line('None of the values is matching');  
    END IF;  
    dbms\_output.put\_line('Exact value of a is: '|| a );  
    END;**

Ans) The Output is

None of the values is matching

Exact value of a is 100

**2. What will be the output of the following code?**

**DECLARE  
 lines dbms\_output.chararr;  
 num\_lines number;  
BEGIN  
 Dbms\_output.enable;  
 dbms\_output.put\_line('Hello!');  
 dbms\_output.put\_line('Hope you are doing well!');  
 num\_lines := 2;  
 dbms\_output.get\_lines(lines, num\_lines);  
 FOR i IN 1..num\_lines LOOP  
 dbms\_output.put\_line(lines(i));  
 END LOOP;  
END;**

**Ans)** Hello Reader

Hope you have enjoyed doing well

2

**3. Consider the following code :−**

**DECLARE  
 -- Global variables  
 num number := 95;  
BEGIN  
 dbms\_output.put\_line('num: ' || num1);  
 DECLARE  
 -- Local variables  
 num number := 195;  
 BEGIN  
 dbms\_output.put\_line('num: ' || num1);  
 END;  
END;**

**What will happen when the code is executed?**

Ans) Not executed , because syntax error.

1. **What would be printed when the following code is executed?**

**DECLARE  
 x NUMBER;  
 BEGIN  
 x := 5;  
 x := 10;  
 dbms\_output.put\_line(-x);  
 dbms\_output.put\_line(+x);  
 x := -10;  
 dbms\_output.put\_line(-x);  
 dbms\_output.put\_line(+x);  
 END;**

Ans) -10

10

10

-10

1. **What will be printed by the following PL/SQL block?**

**DECLARE  
 a number;  
 b number;  
 c number;  
PROCEDURE findMin(x IN number, y IN number, z OUT number) IS  
BEGIN  
IF x < y THEN  
 z:= x;  
 ELSE  
 z:= y;  
END IF;  
END;  
BEGIN  
 a:= 2;  
 b:= 5;  
 findMin(a, b, c);  
 dbms\_output.put\_line(c);  
END;**

Ans) -5

-10

-25

1. **What will be printed by the following PL/SQL block?**

**DECLARE  
 a number;  
PROCEDURE squareNum(x IN OUT number) IS  
BEGIN  
 x := x \* x;  
END;  
BEGIN  
 a:= 5;  
 squareNum(a);  
 dbms\_output.put\_line(a);  
END;**

**Ans) -5**

**-10**

**-25**

1. **When is the pre-defined exception “CASE\_NOT\_FOUND” raised?**

None of the choices in the when clauses of a case statement is selected , and there is no ELSE clause.

**IN-LAB**

**Case Study 1 : TRANSPORT DEPARTMENT**

1. Write a PL/SQL stored procedure to know the current age of customers who are associated with AP transport department.

**delimiter $$**

**create procedure ageOfCustomers()**

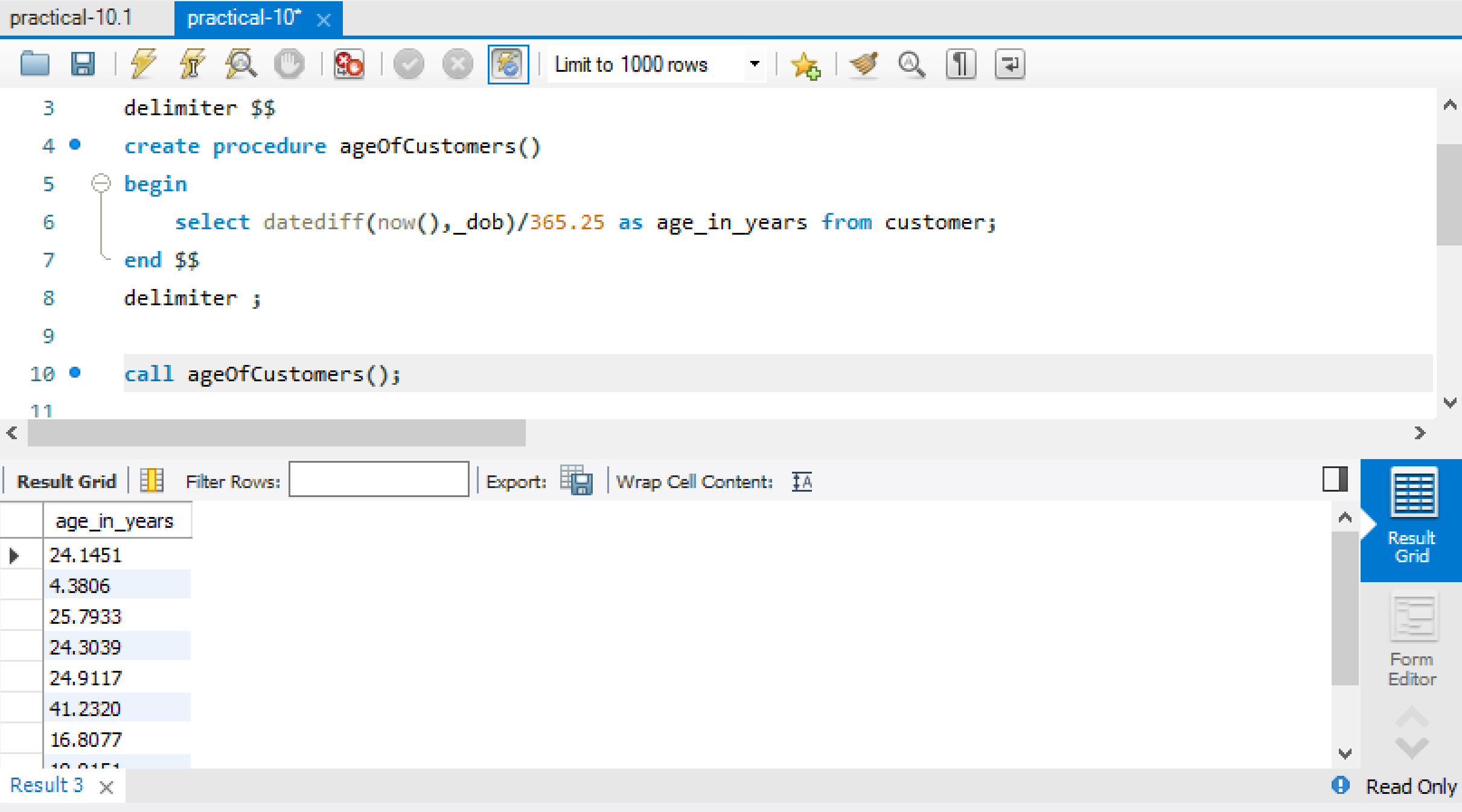
**begin**

**select datediff(now(),\_dob)/365.25 as age\_in\_years from customer;**

**end $$**

**delimiter ;**

**call ageOfCustomers();**



1. Write a PL/SQL stored function to know that, from how many years vehicles are registered with the AP transport department.

**delimiter @**

**create function NoOfyears(v\_id int) returns float**

**begin**

**declare years float;**

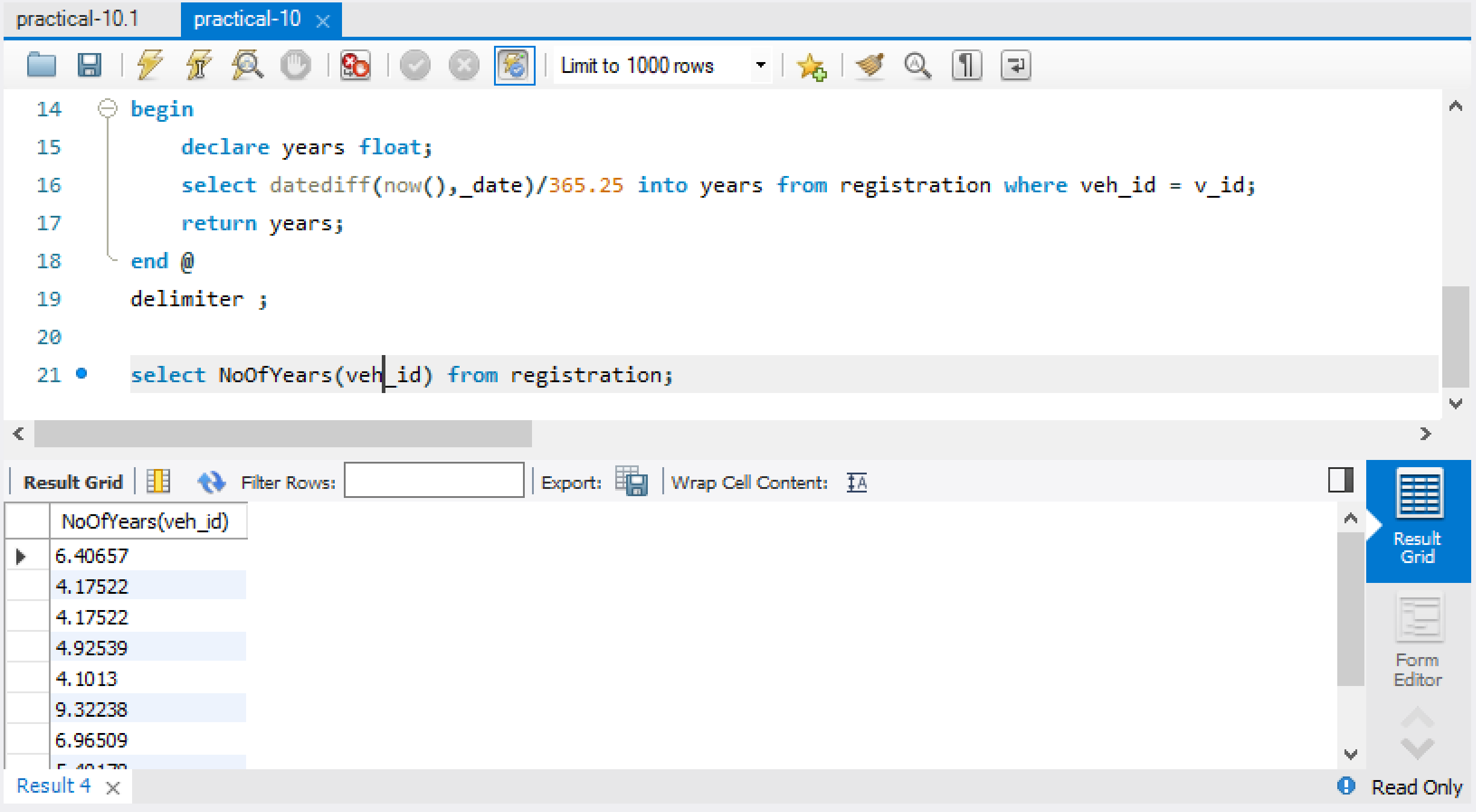
**select datediff(now(),\_date)/365.25 into years from registration where veh\_id = v\_id;**

**return years;**

**end @**

**delimiter ;**

**select NoOfYears(veh\_id) from registration;**



1. Create a trigger before insert to maintain the summary of DealerCenter table into DealerCenterstats. Whenever the capacity of DealerCenters is increased or decreased then the total statistics should be reflected in DealerCenterstats

**create table dealerCenterStats(new\_deal\_id int,new\_deal\_name varchar(25),new\_city varchar(25),new\_street varchar(25), new\_state varchar(25),new\_pincode int,new\_dno int,new\_phno bigint);**

**delimiter $$**

**create trigger new\_dealer before insert on dealer for each row**

**begin**

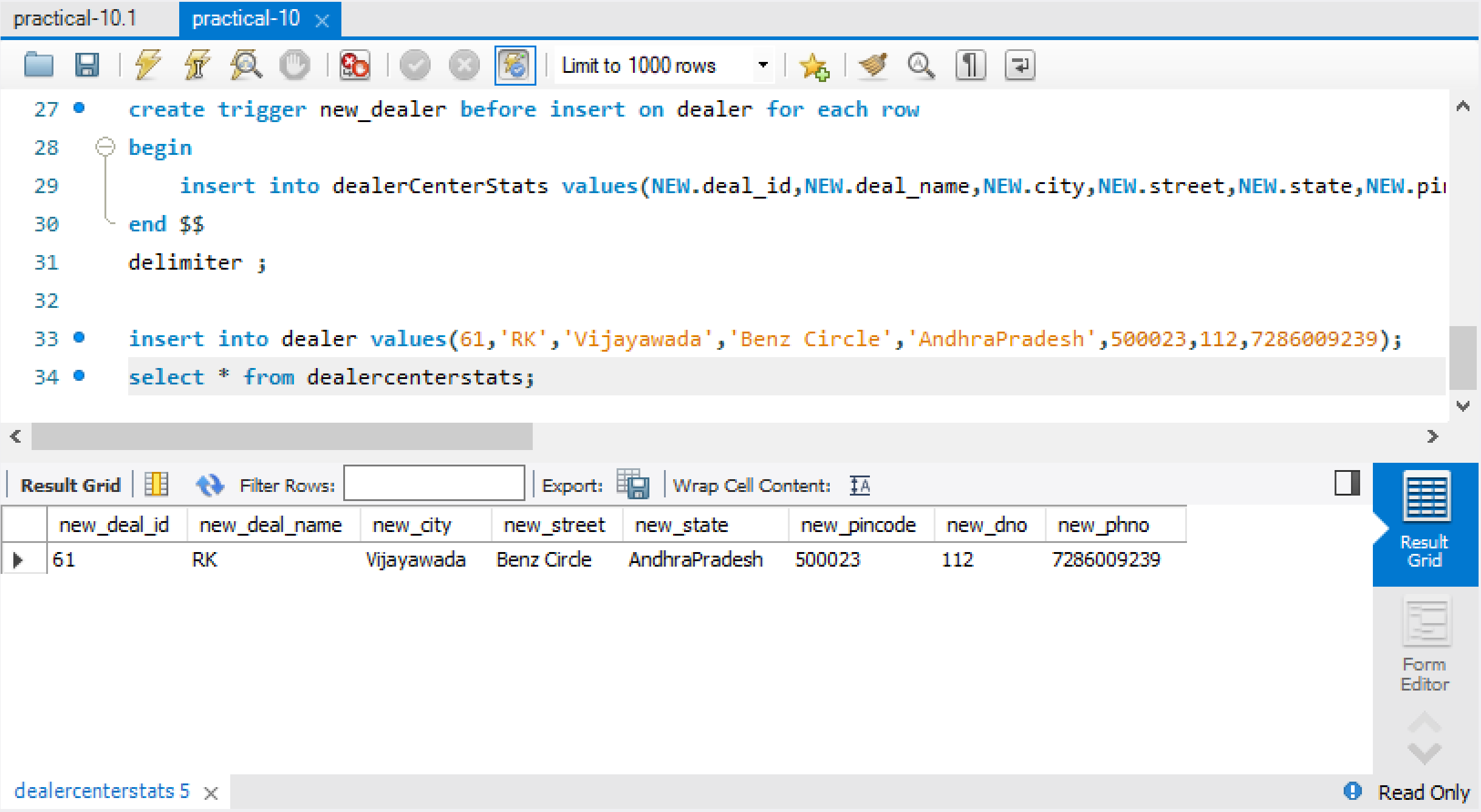
**insert into dealerCenterStats values(NEW.deal\_id, NEW.deal\_name, NEW.city, NEW.street, NEW.state,NEW.pincode,NEW.d\_no,NEW.ph\_no);**

**end $$**

**delimiter ;**

**insert into dealer values(61,'RK','Vijayawada','BenzCircle' ,'AndhraPradesh' ,500023, 112,7286009239);**

**select \* from dealercenterstats;**



1. Create trigger after insert in members table , a trigger should check the value of attribute name and if it is updated then show the message for updating on name in reminder table.

**create table remainder(before\_name varchar(50),after\_name varchar(50));**

**delimiter $$**

**create trigger Customer\_log after update on customer for each row**

**begin**

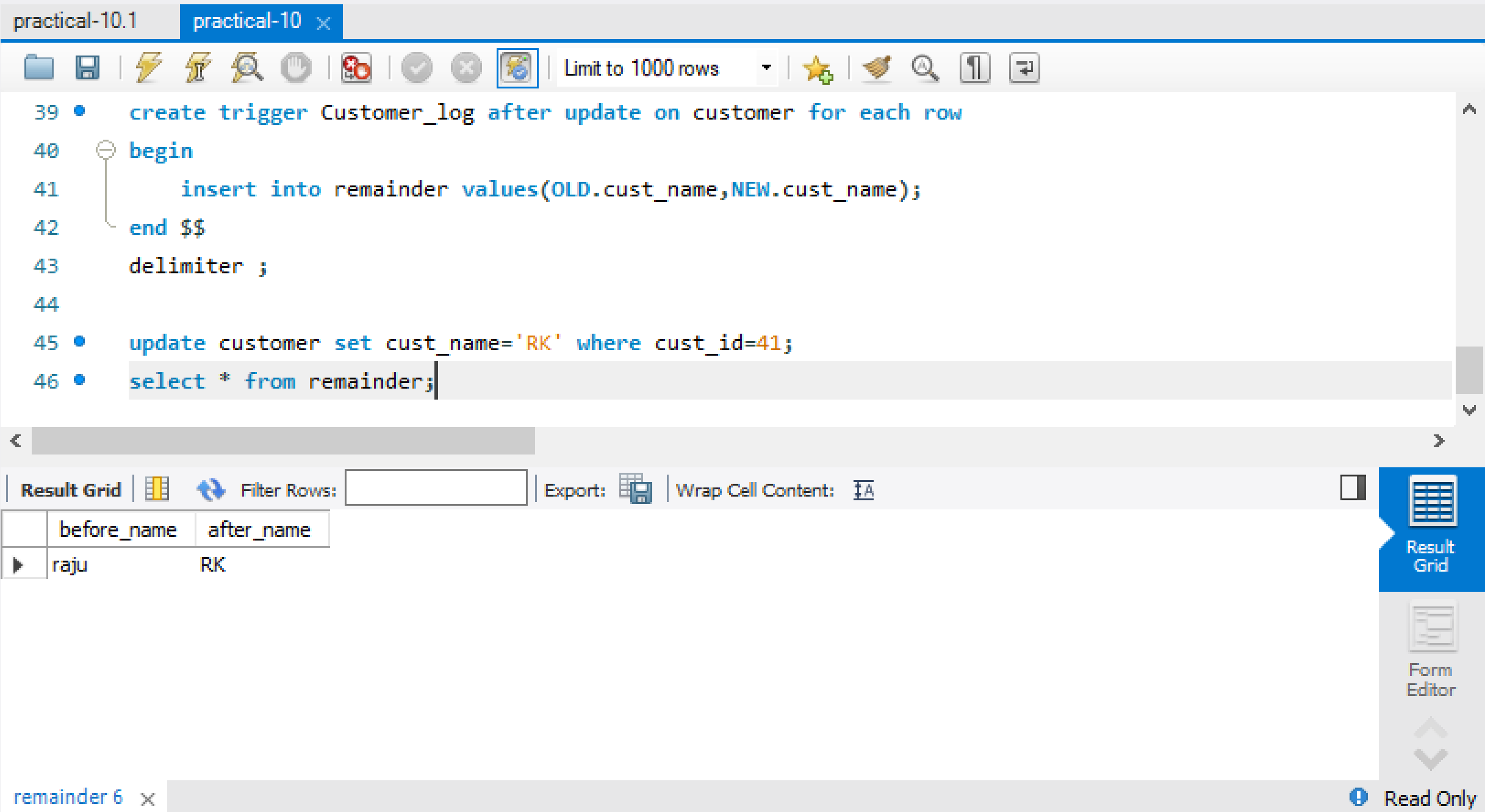
**insert into remainder values(OLD.cust\_name,NEW.cust\_name);**

**end $$**

**delimiter ;**

**update customer set cust\_name='RK' where cust\_id=41;**

**select \* from remainder;**

****

**Case Study 4 : KL UNIVERSITY ERP**

1. Write a Program to create a row level trigger that would fire for INSERT or UPDATE or DELETE operations performed on the Faculty table. The program has to print the salary difference of faculty along with Old salary and New salary

**create table faculty\_Log(operation\_id int primary key auto\_increment,FID int,**

**FNAME varchar(10),Designation varchar(10),Salary int,FMOBILE bigint,**

**FMAIL varchar(20),FADD varchar(10),Branch varchar(10),changed\_at DATETIME NOT NULL,**

**operation varchar(3) NOT NULL,CHECK(operation = 'INS' or operation = 'DEL'));**

**delimiter $$**

**create trigger trig\_faculty\_insert after insert on faculty**

**for each row**

**begin**

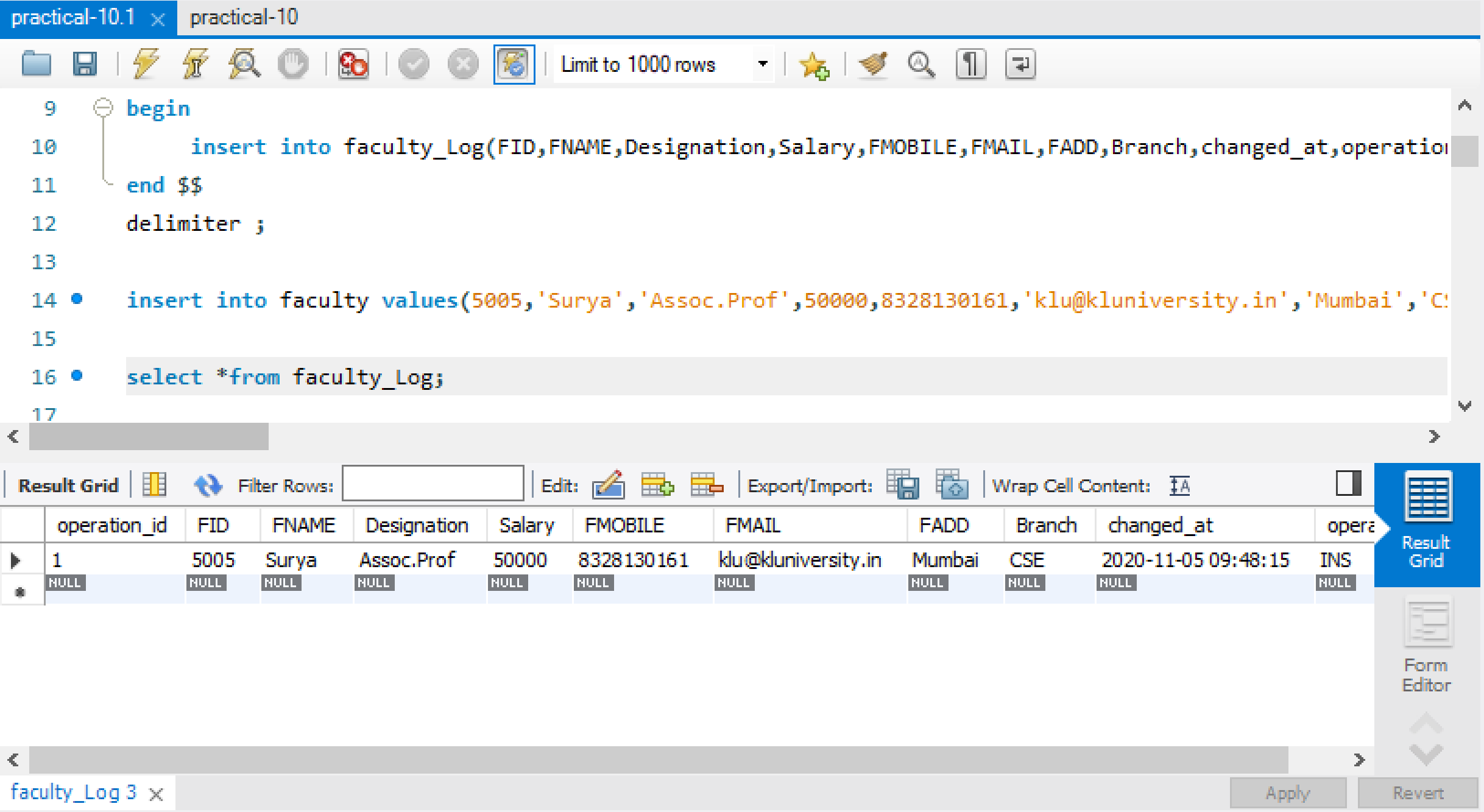
**insert into faculty\_Log(FID,FNAME,Designation,Salary,FMOBILE,FMAIL,FADD,Branch,changed\_at,operation) values(NEW.FID,NEW.FNAME,NEW.Designation,NEW.Salary,NEW.FMOBILE,NEW.FMAIL,NEW.FADD,NEW.BRANCH,current\_timestamp,'INS');**

**end $$**

**delimiter ;**

**insert into faculty values(5005,'Surya','Assoc.Prof',50000,8328130161,'klu@kluniversity.in','Mumbai','CSE');**

**select \*from faculty\_Log;**

****

**delimiter $$**

**create trigger trig\_faulty\_delete after delete on faculty**

**for each row**

**begin**

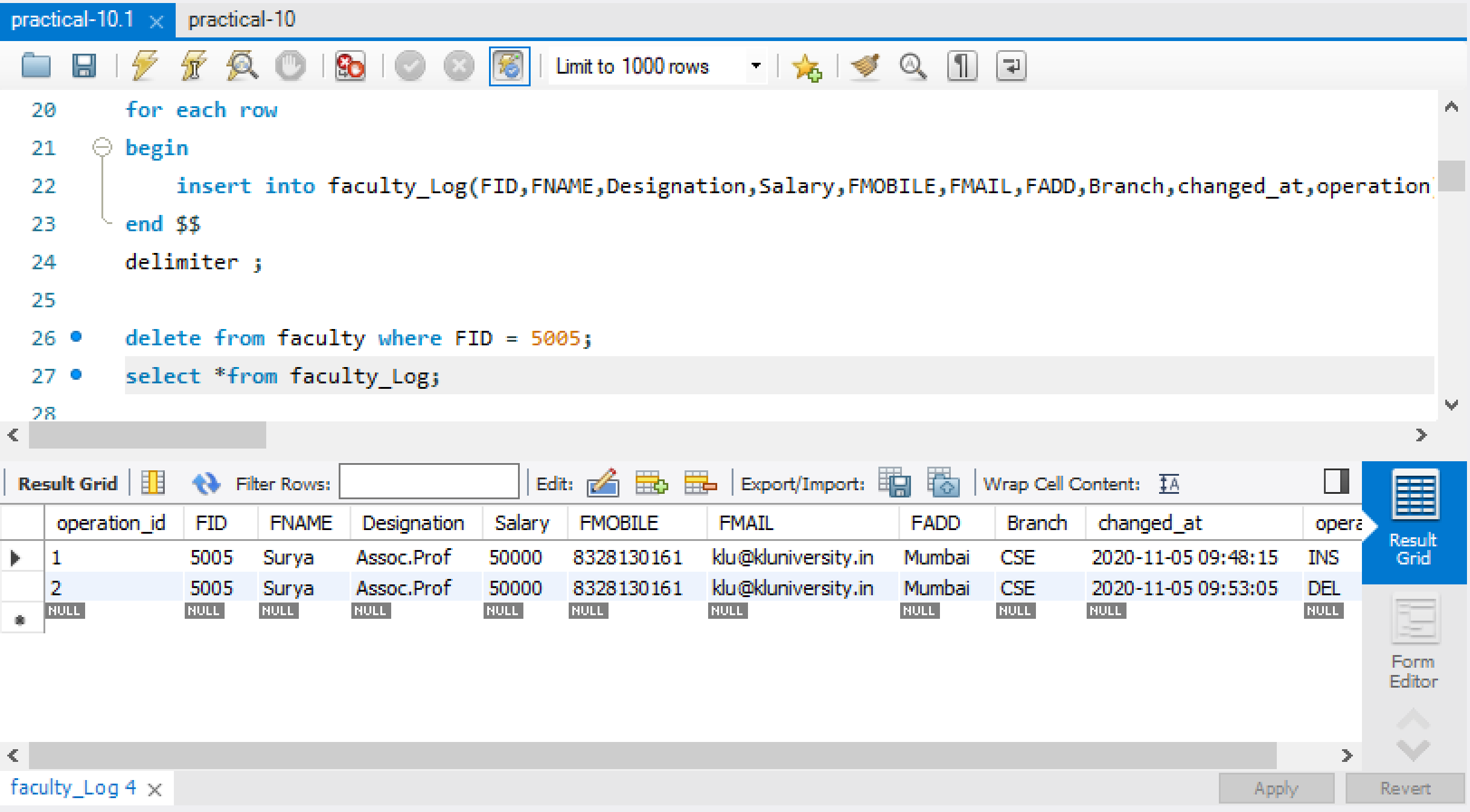
**insert into faculty\_Log(FID,FNAME,Designation,Salary,FMOBILE,FMAIL,FADD,Branch,changed\_at,operation) values(OLD.FID,OLD.FNAME,OLD.Designation,OLD.Salary,OLD.FMOBILE,OLD.FMAIL,OLD.FADD,OLD.BRANCH,current\_timestamp,'DEL');**

**end $$**

**delimiter ;**

**delete from faculty where FID = 5005;**

**select \*from faculty\_Log;**

****

1. Write a Program to create a row level trigger that would fire for INSERT or UPDATE or DELETE operations performed on the LIBRARYBooks table. The program has to print the status of the DML operations(Like Insert, Update and delete) performed

**create table library\_Log(operation\_id int primary key auto\_increment,ACCNO int,**

**updated\_accno int default NULL,BTITLE varchar(30),updated\_btitle varchar(30) default 'No Updation',**

**AUTHOR varchar(30),updated\_author varchar(30) default 'No Updation',PUBLISHER varchar(25),**

**updated\_publisher varchar(25) default 'No updation',EDITION int,updated\_edition int default null,**

**PRICE int,updated\_price int default null,No\_of\_Copies int,updated\_copies int default null,**

**changed\_at DATETIME NOT NULL,operation varchar(20) NOT NULL,**

**CHECK(operation = 'Inserted' or operation = 'Deleted' or operation = 'Updated'));**

**delimiter $$**

**create trigger trig\_library\_insert after insert on library\_Books**

**for each row**

**begin**

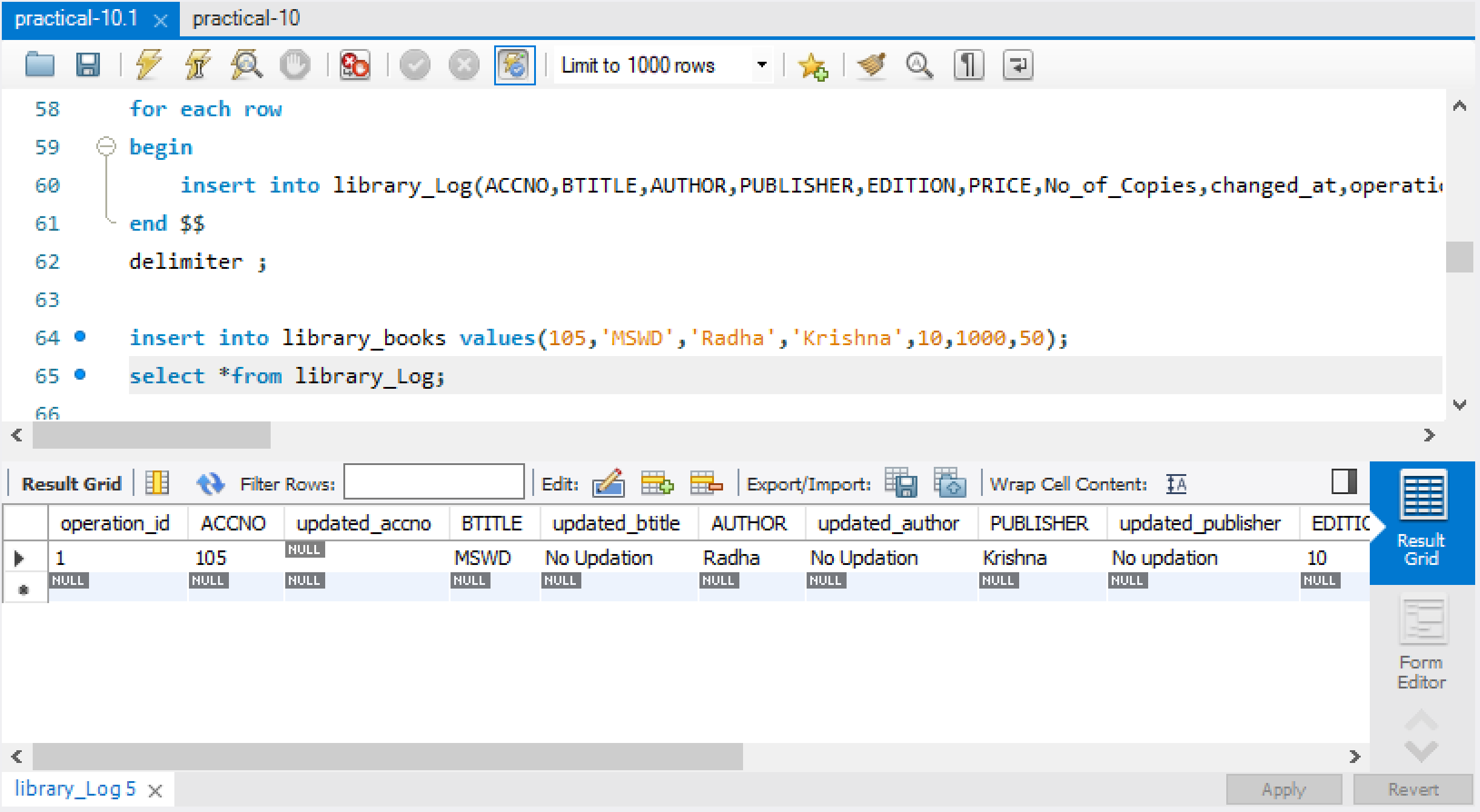
**insert into library\_Log(ACCNO,BTITLE,AUTHOR,PUBLISHER,EDITION,PRICE,No\_of\_Copies,changed\_at,operation) values(new.ACCNO,NEW.BTITLE,NEW.AUTHOR,NEW.PUBLISHER,NEW.EDITION,NEW.PRICE,NEW.No\_of\_Copies,current\_timestamp,'Inserted');**

**end $$**

**delimiter ;**

**insert into library\_Books values(105,'MSWD','Radha','Krishna',10,1000,50);**

**select \*from library\_Log;**

****

**delimiter $$**

**create trigger trig\_library\_update after update on library\_Books**

**for each row**

**begin**

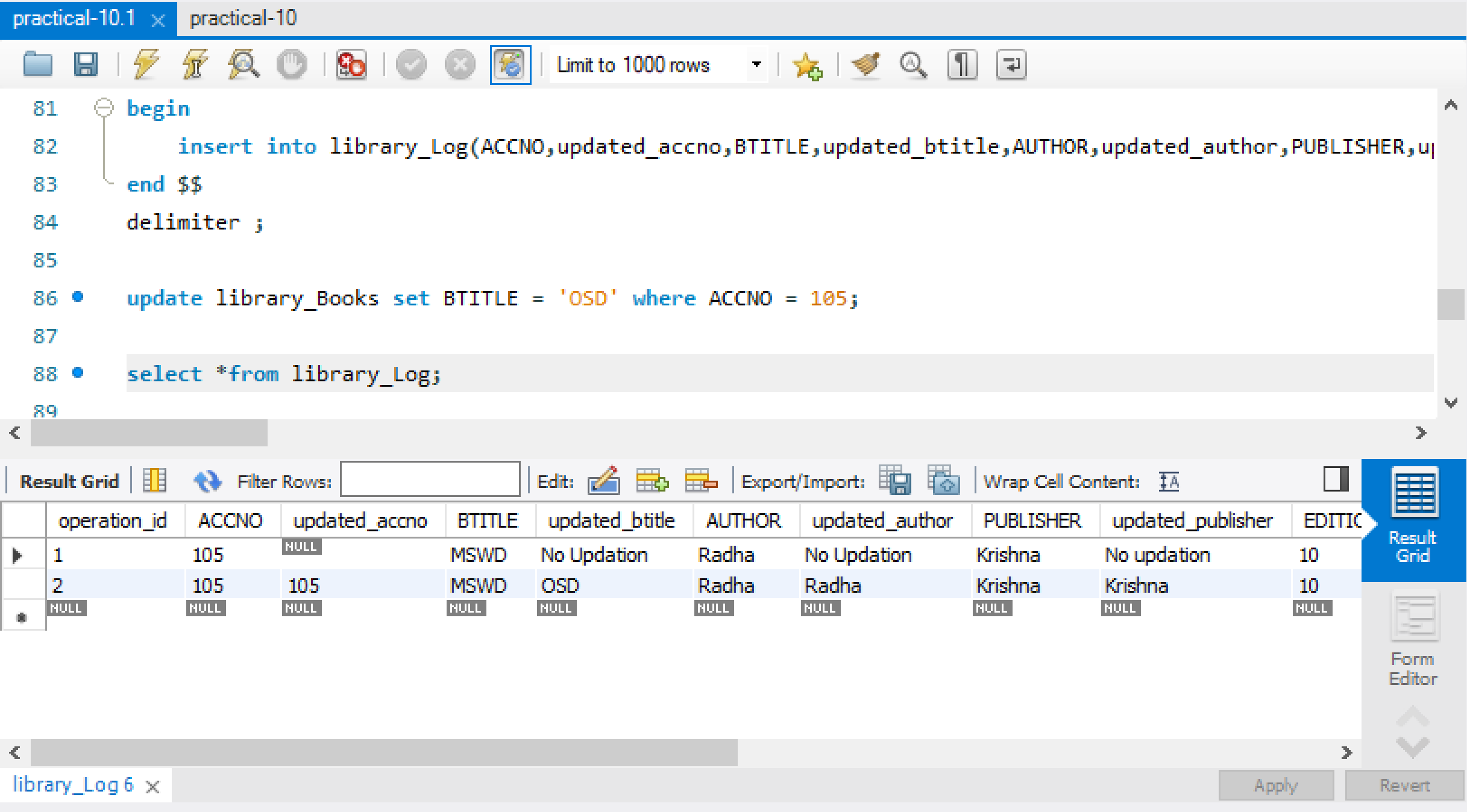
**insert into library\_Log(ACCNO,updated\_accno,BTITLE, updated\_btitle, AUTHOR, updated\_author,PUBLISHER,updated\_publisher,EDITION,updated\_edition,PRICE,updated\_price,No\_of\_Copies,updated\_copies,changed\_at,operation) values (old.ACCNO, new.ACCNO,old.BTITLE,new.BTITLE,old.AUTHOR,new.AUTHOR,old.PUBLISHER,new.PUBLISHER,old.EDITION,new.EDITION,old.PRICE,new.PRICE,old.No\_of\_Copies,new.No\_of\_Copies,current\_timestamp,'Updated');**

**end $$**

**delimiter ;**

**update library\_Books set BTITLE = 'OSD' where ACCNO = 105;**

**select \*from library\_Log;**

****

1. Write a PL/SQL Program to calculate the tax of a faculty based on the below conditions using Functions.

a. If the salary of a faculty is between 0 and 30000 then tax should be 10%

b. If the salary of a faculty is between 30001 and 50000 then tax should be 15%

c. If the salary of a faculty is above 50001 then tax should be 25%

4. Write a PL/SQL Program to create a package that contains the following functions:

a. Function for computing Annual salary of a faculty

**delimiter $@**

**create function annual\_salary(Salary int) returns integer**

**deterministic**

**begin**

**declare anu\_sal int;**

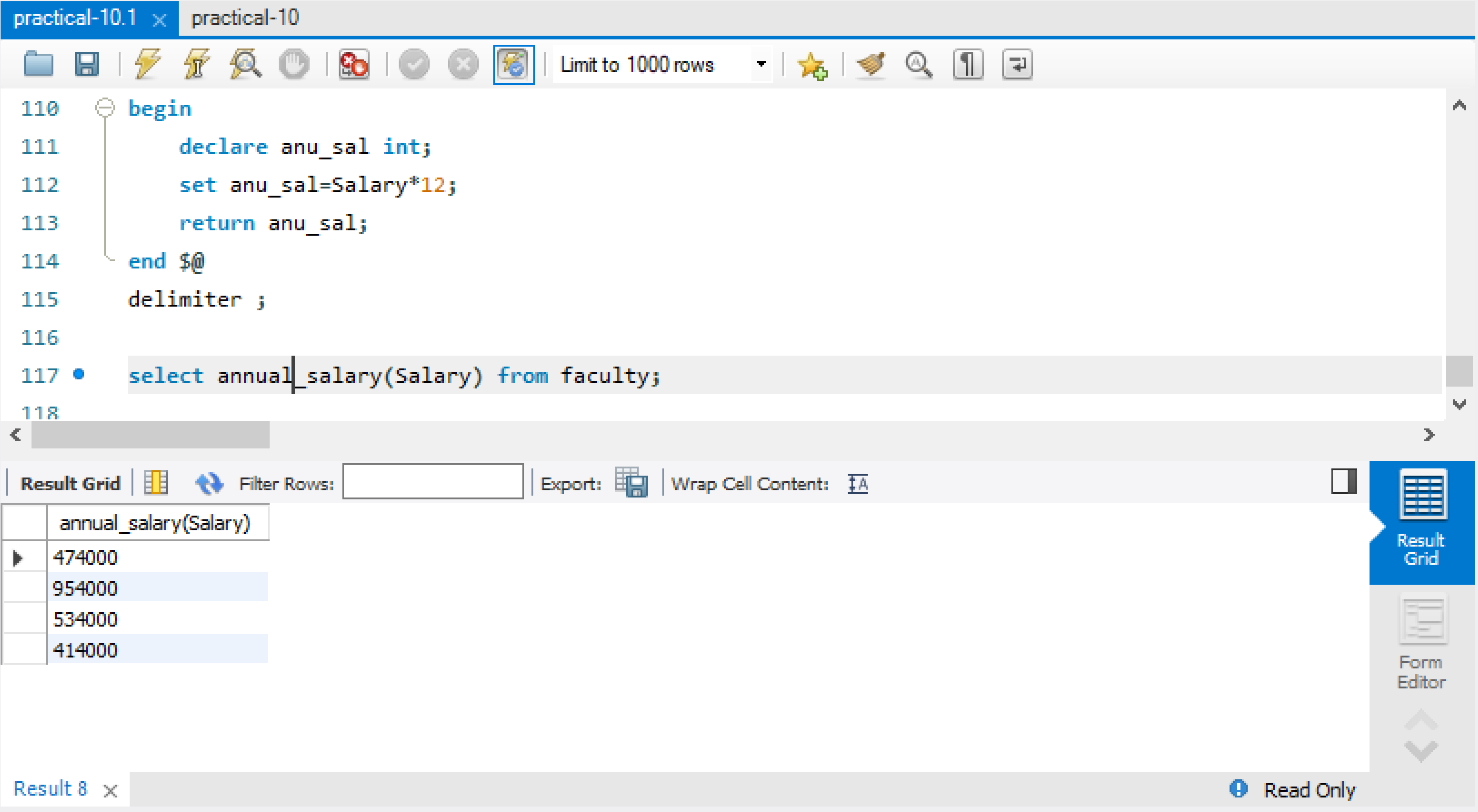
**set anu\_sal=Salary\*12;**

**return anu\_sal;**

**end $@**

**delimiter ;**

**select annual\_salary(Salary) from faculty;**



b. Function to calculate the tax of a faculty based on the conditions in Q3 above.

**delimiter $$**

**create function faculty\_tax(Salary int) returns integer**

**deterministic**

**begin**

**declare tax int;**

**if Salary >= 0 and Salary < 30000 then**

**set tax = 10/100 \* Salary;**

**elseif Salary >= 30001 and Salary < 50000 then**

**set tax = 15/100 \* Salary;**

**elseif Salary >= 50001 then**

**set tax = 25/100 \* Salary;**

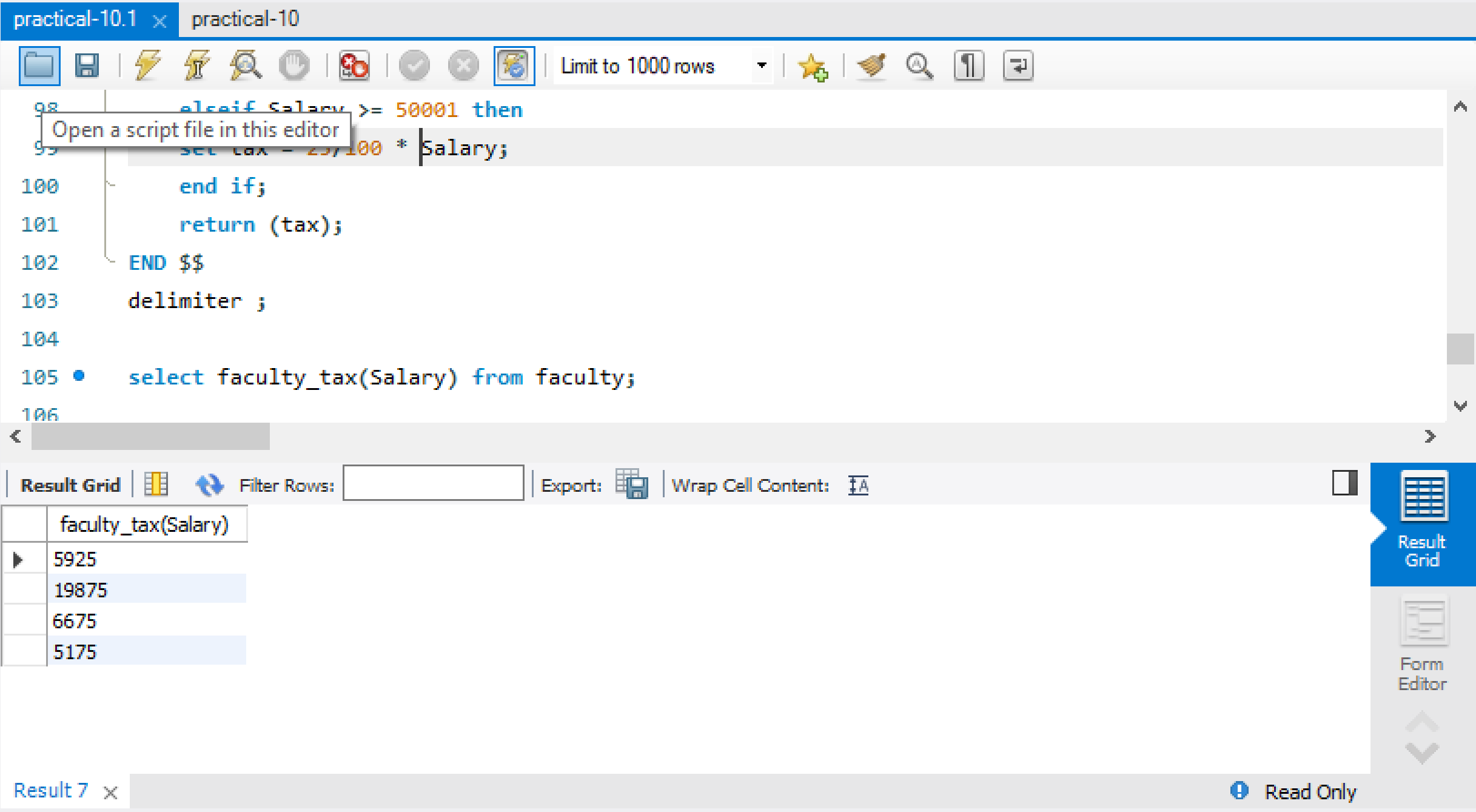
**end if;**

**return (tax);**

**END $$**

**delimiter ;**

**select faculty\_tax(Salary) from faculty;**



**Java Database Connectivity with MySQL**

To connect Java application with the MySQL database, we need to follow 5 following steps.

In this example we are using MySql as the database. So we need to know following informations for the mysql database:

1. **Driver class:**The driver class for the mysql database is **com.mysql.jdbc.Driver**.
2. **Connection URL:**The connection URL for the mysql database is **jdbc:mysql://localhost:3306/sonoo** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name. We may use any database, in such case, we need to replace the sonoo with our database name.
3. **Username:**The default username for the mysql database is **root**.
4. **Password:**It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

Let's first create a table in the mysql database, but before creating table, we need to create database first.

create database transport;

use transport;

create table emp(id **int**(10),name varchar(40),age **int**(3));

In this example, transport is the database name, root is the username and password both.

**import** java.sql.\*;

**class** MysqlCon{

**public** **static** **void** main(String args[]){

**try**{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/sonoo","root","root");

//here sonoo is database name, root is username and password

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from emp");

**while**(rs.next())

System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3));

con.close();

}**catch**(Exception e){ System.out.println(e);}

}

}

The above example will fetch all the records of emp table.

connect java application with the mysql database, **mysqlconnector.jar** file is required to be loaded.

[download the jar file mysql-connector.jar](https://static.javatpoint.com/src/jdbc/mysql-connector.jar)

### Two ways to load the jar file:

1. Paste the mysqlconnector.jar file in jre/lib/ext folder
2. Set classpath

### 1) Paste the mysqlconnector.jar file in JRE/lib/ext folder:

|  |
| --- |
| Download the mysqlconnector.jar file. Go to jre/lib/ext folder and paste the jar file here. |

### 2) Set classpath:

|  |
| --- |
| There are two ways to set the classpath:   * temporary * permanent |

### How to set the temporary classpath

|  |
| --- |
| open command prompt and write: |

C:>set classpath=c:\folder\mysql-connector-java-5.0.8-bin.jar;.;

### How to set the permanent classpath

Go to environment variable then click on new tab. In variable name write **classpath** and in variable value paste the path to the mysqlconnector.jar file by appending mysqlconnector.jar;.; as C:\folder\mysql-connector-java-5.0.8-bin.jar;.;

**POST-LAB**

Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by, Order by, Having.

E\_id E\_name Age Salary

101 AREEB 22 9000

102 DHEERAJ 29 8000

103 RAHUL 34 6000

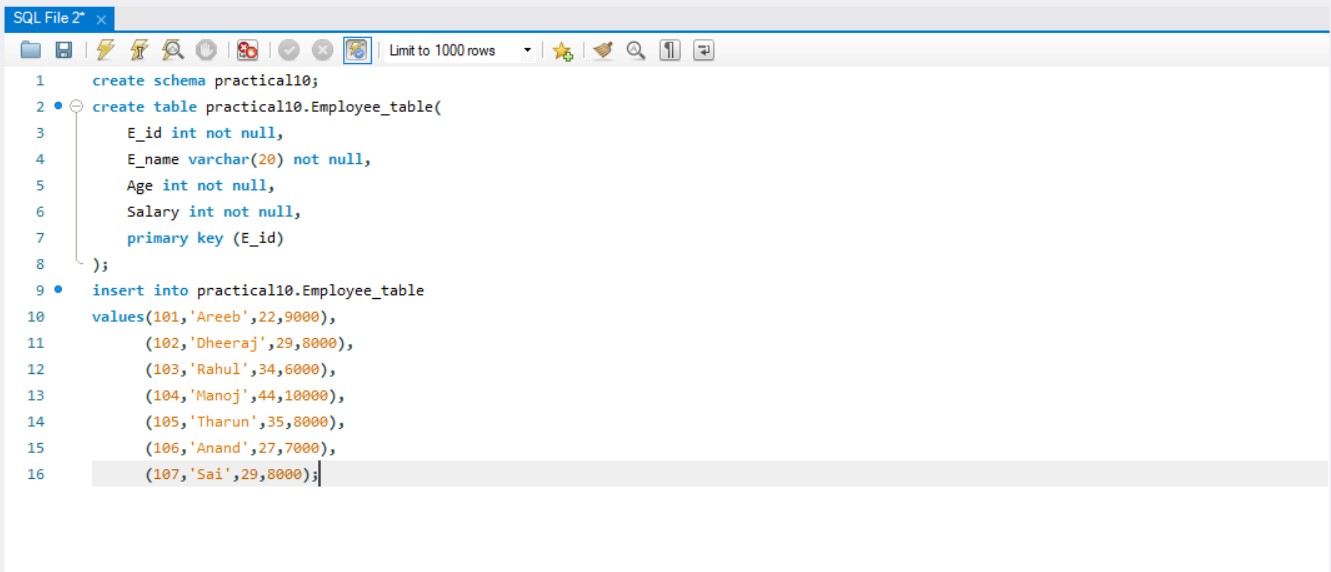
104 MANOJ 44 10000

105 THARUN 35 8000

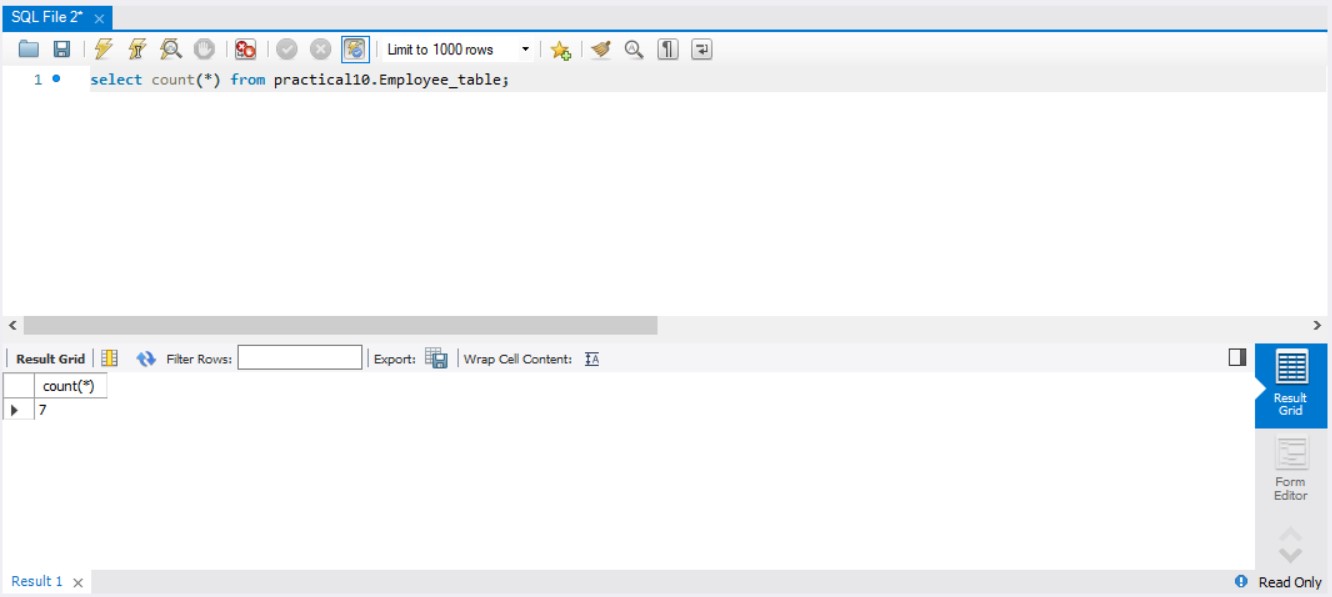
106 ANAND 27 7000

107 SAI 29 8000

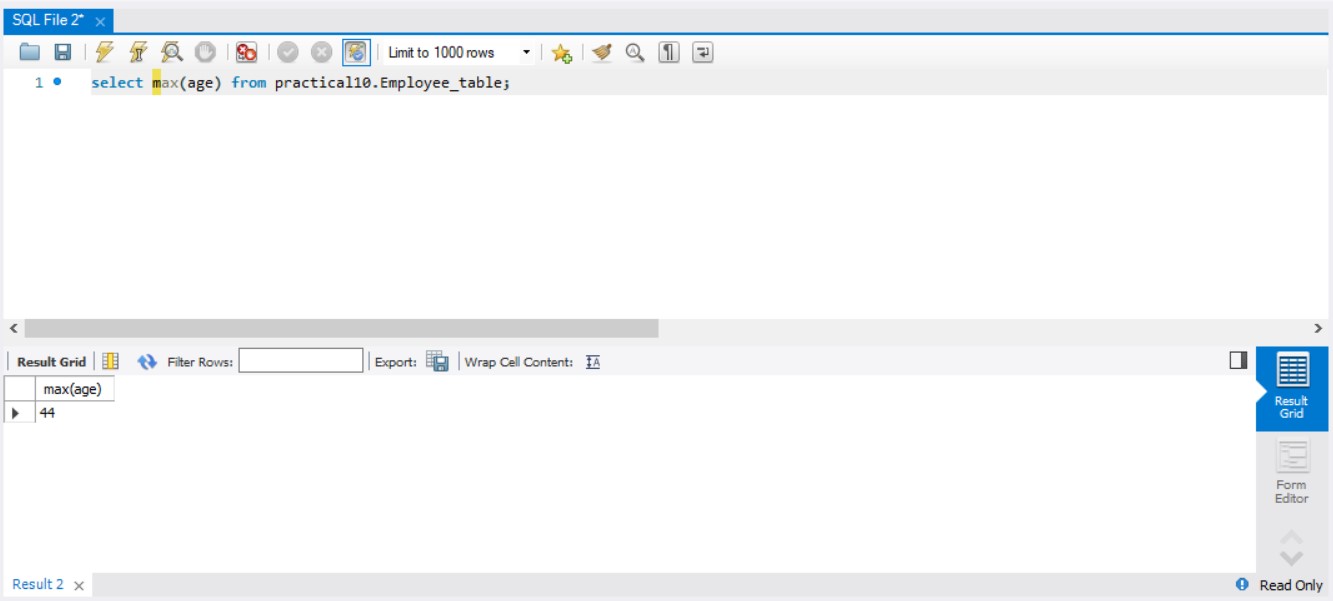
1. Create Employee table containing all Records.



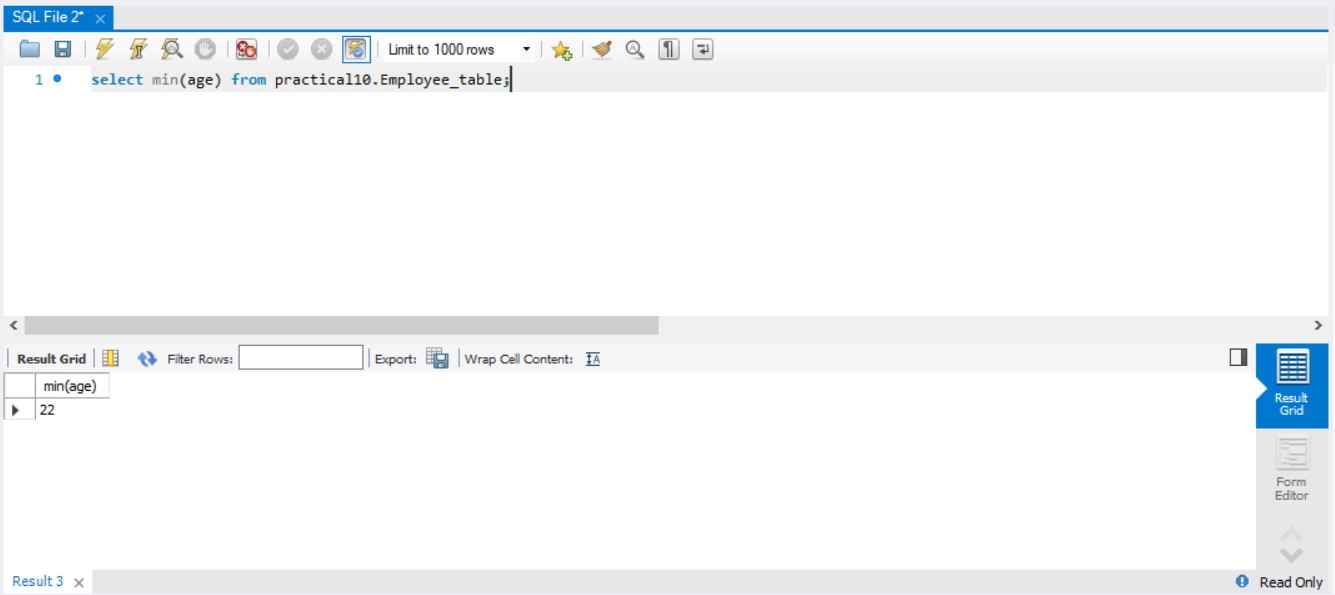
1. Count number of employee names from employee table



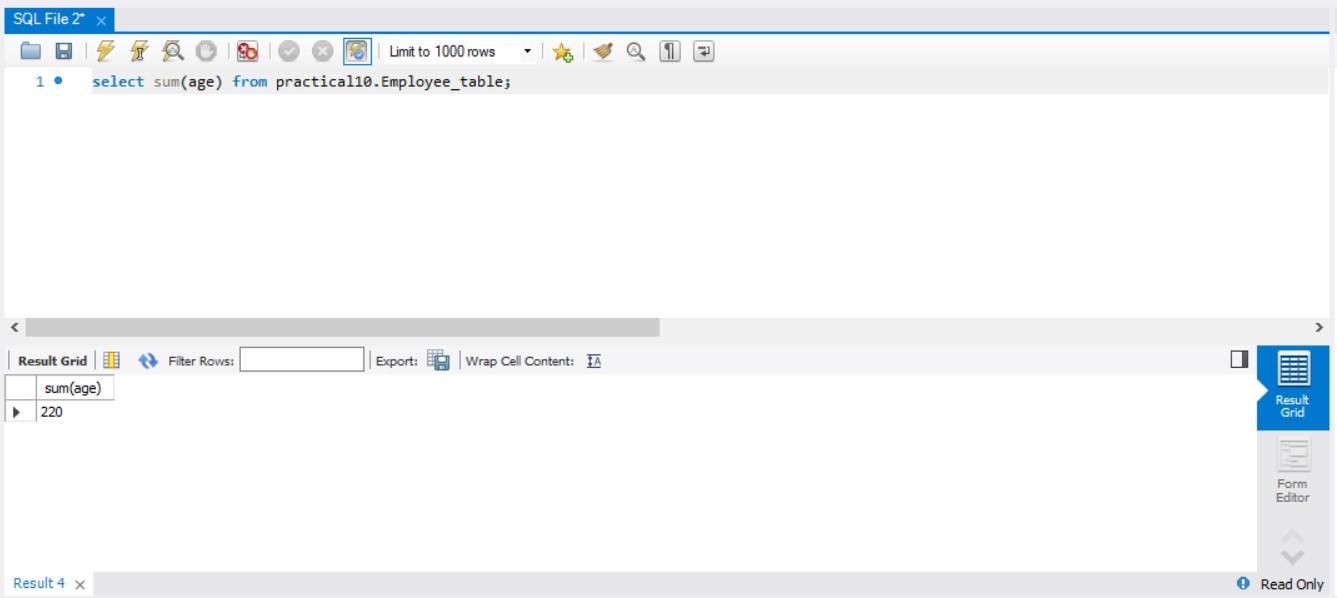
1. Find the Maximum age from employee table.



1. Find the Minimum age from employee table



1. Display the Sum of age employee table



1. Display the Average of age from Employee table.

