UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

WORKSHEET 2.2

Student Name: GAURAV KUMAR UID: 21MCA3201

Branch: MCA Section/Group: 9A

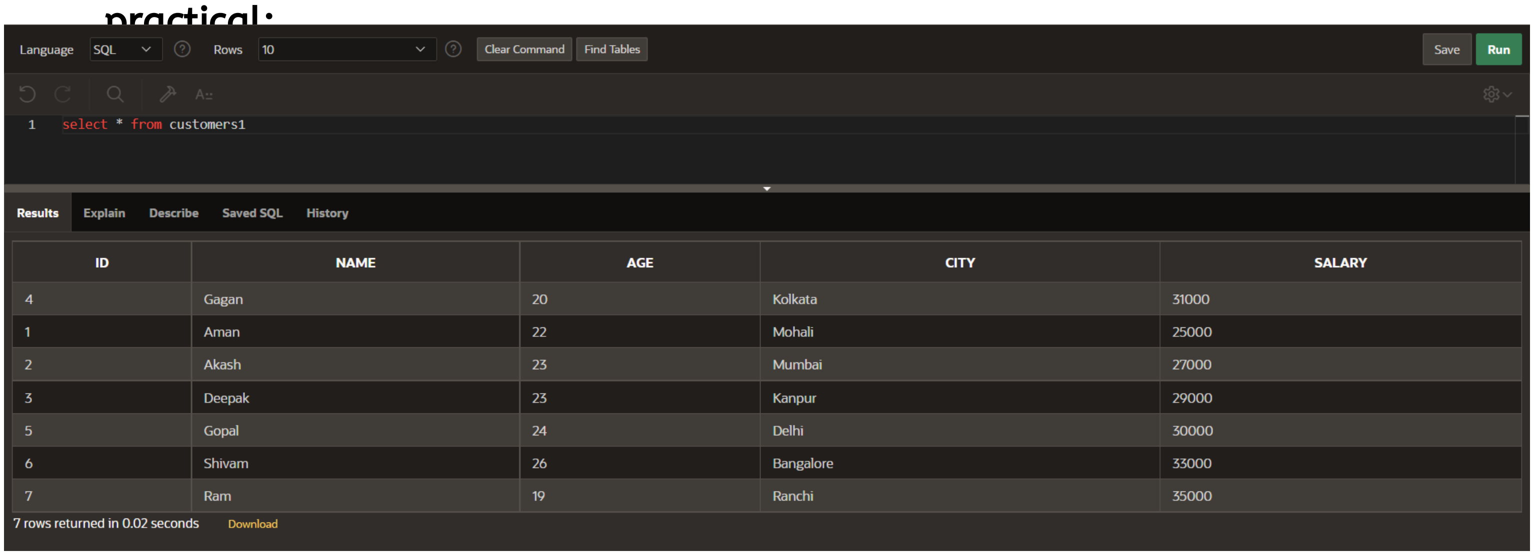
Semester: 1st Semester Date of Performance: 05/01/2022

Subject Name: PL/SQL LAB Subject Code: 21CAP-607

1) Task to be done:

WAP that creates a row-level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table.

2) Steps for experiment/ practical:


Language SQL Rows 10 Clear Command Find Tables Save Run

```
select * from customers1
```

Results Explain Describe Saved SQL History

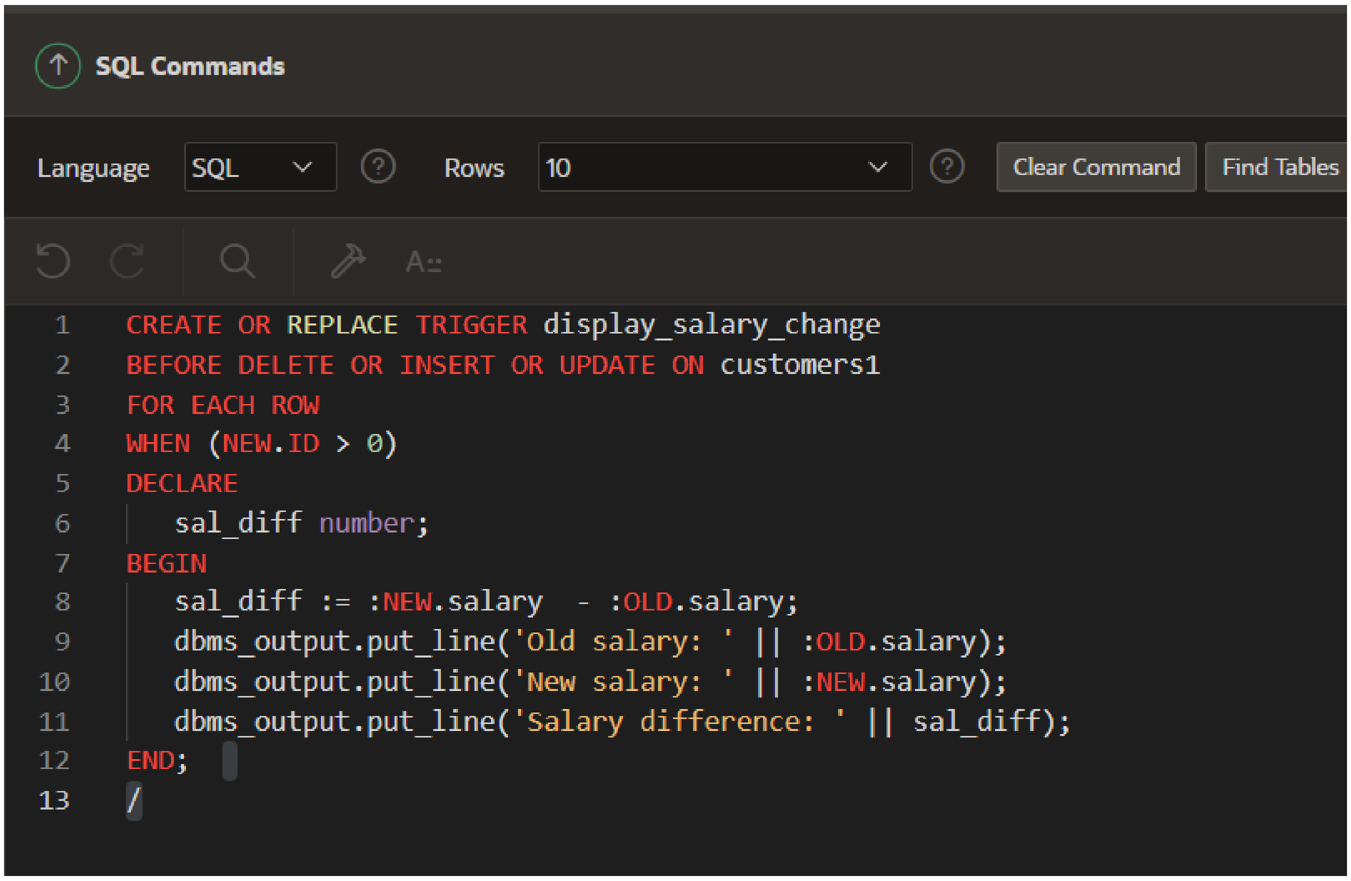
ID	NAME	AGE	CITY	SALARY
4	Gagan	20	Kolkata	31000
1	Aman	22	Mohali	25000
2	Akash	23	Mumbai	27000
3	Deepak	23	Kanpur	29000
5	Gopal	24	Delhi	30000
6	Shivam	26	Bangalore	33000
7	Ram	19	Ranchi	35000

7 rows returned in 0.02 seconds Download

CREATING TRIGGER

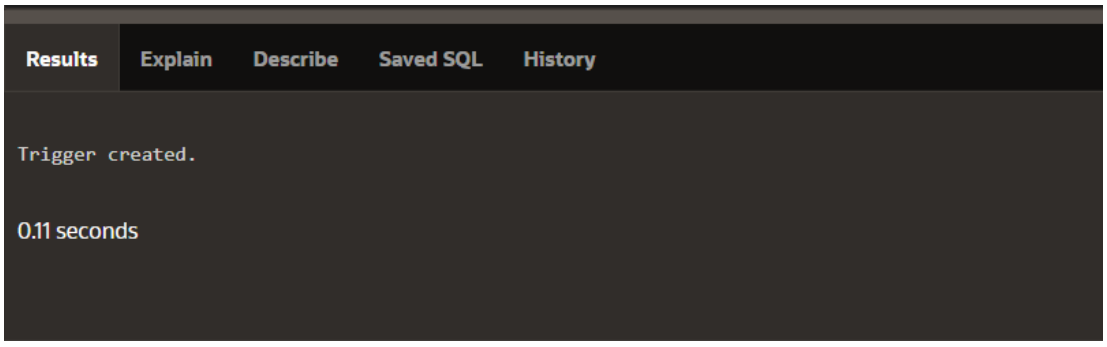
```
CREATEORREPLACETRIGGER display_salary_change
BEFORE DELETEORINSERTORUPDATEON customers1
FOR EACH ROW
WHEN (NEW.ID > 0)
DECLARE
```

```
  sal_diff number;
BEGIN
  sal_diff:= :NEW.salary - :OLD.salary;
  dbms_output.put_line('Old salary: ' || :OLD.salary);
  dbms_output.put_line('New salary: ' || :NEW.salary);
  dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
```



```
CREATE OR REPLACE TRIGGER display_salary_change
BEFORE DELETE OR INSERT OR UPDATE ON customers1
FOR EACH ROW
WHEN (NEW.ID > 0)
DECLARE
   sal_diff number;
BEGIN
   sal_diff := :NEW.salary  - :OLD.salary;
   dbms_output.put_line('Old salary: ' || :OLD.salary);
   dbms_output.put_line('New salary: ' || :NEW.salary);
   dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
```

OUTPUT

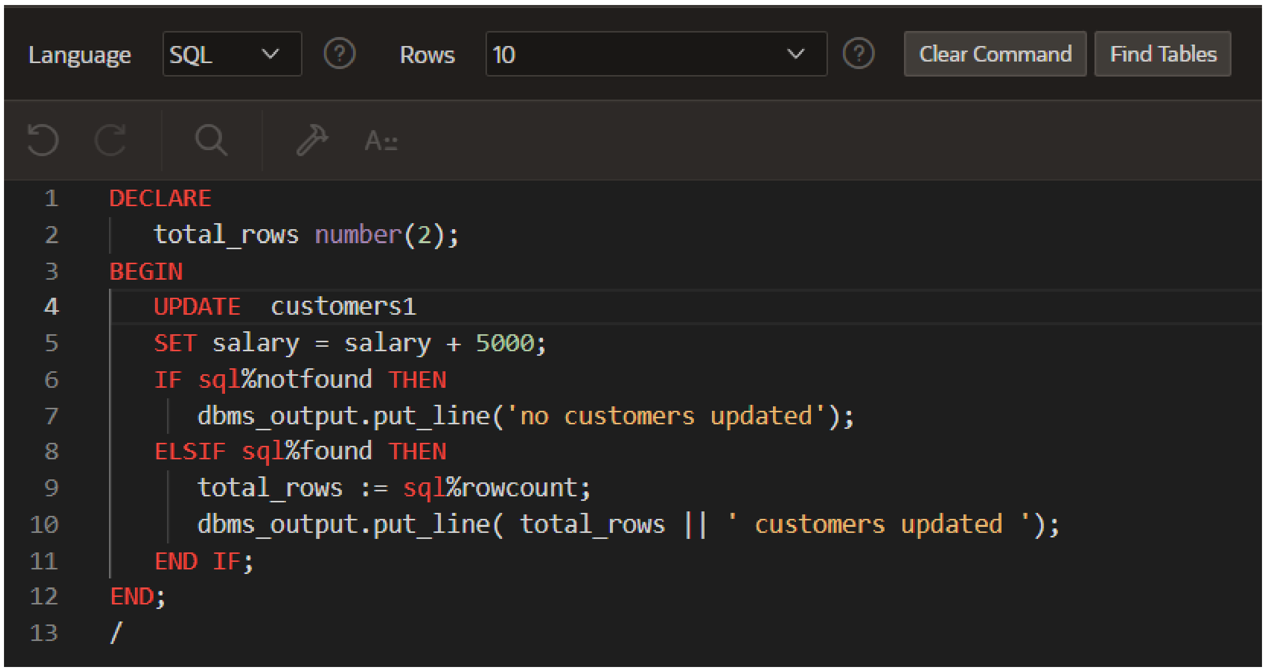


```
Trigger created.

0.11 seconds
```

1) Triggering a trigger:-

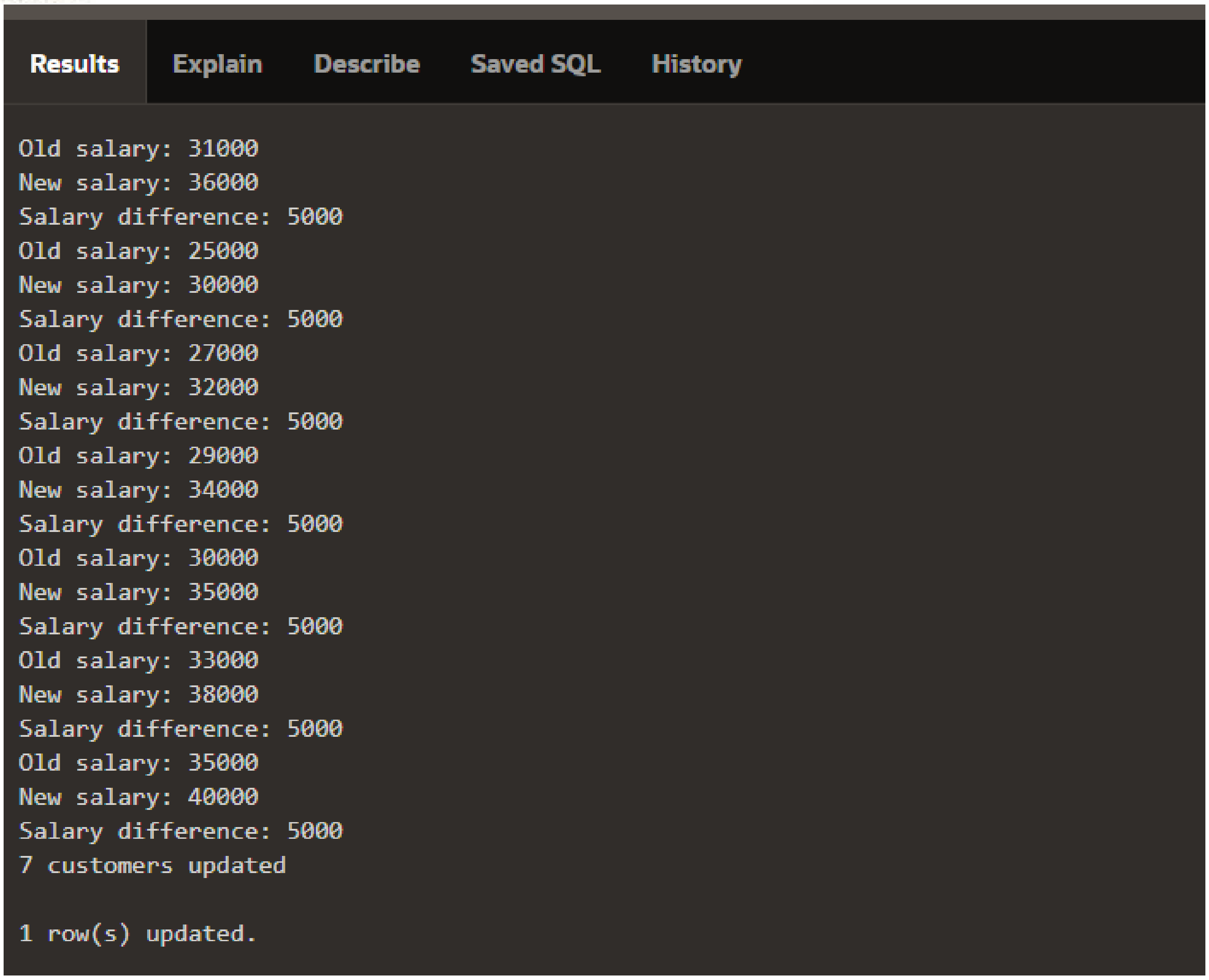
```
DECLARE
  total_rows number(
2);
BEGIN
  UPDATEcustomers1
  SET salary = salary +
5000;
  IF sql%notfound THEN
    dbms_output.put_line('no customers
updated');
  ELSIF sql%found THEN
    total_rows := sql%
rowcount;
    dbms_output.put_line( total_rows || ' customers
updated ');
  END IF;
END;
```



Language: SQL
Rows: 10

```
DECLARE
   total_rows number(2);
BEGIN
   UPDATE  customers1
   SET salary = salary + 5000;
   IF sql%notfound THEN
      dbms_output.put_line('no customers updated');
   ELSIF sql%found THEN
      total_rows := sql%rowcount;
      dbms_output.put_line( total_rows || ' customers updated ');
   END IF;
END;
/
```

2) Output:



```
Old salary: 31000
New salary: 36000
Salary difference: 5000
Old salary: 25000
New salary: 30000
Salary difference: 5000
Old salary: 27000
New salary: 32000
Salary difference: 5000
Old salary: 29000
New salary: 34000
Salary difference: 5000
Old salary: 30000
New salary: 35000
Salary difference: 5000
Old salary: 33000
New salary: 38000
Salary difference: 5000
Old salary: 35000
New salary: 40000
Salary difference: 5000
7 customers updated

1 row(s) updated.
```

4) Learning outcomes (What I have learnt):

1. Implementation of Trigger
2. Triggering a Trigger.

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5