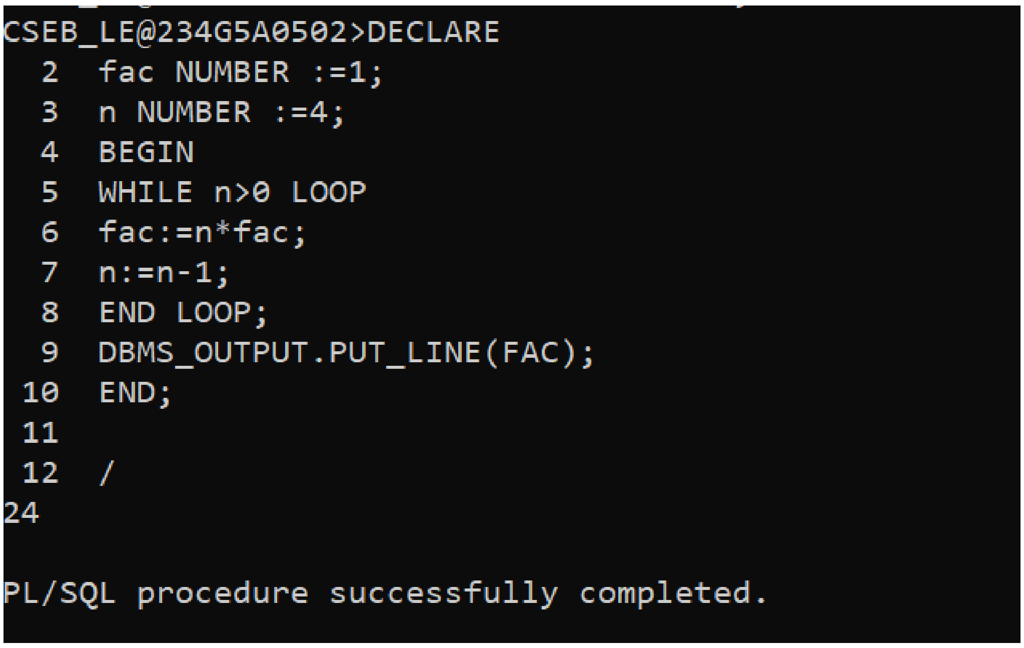
EXP-10

PROGFRAM :-



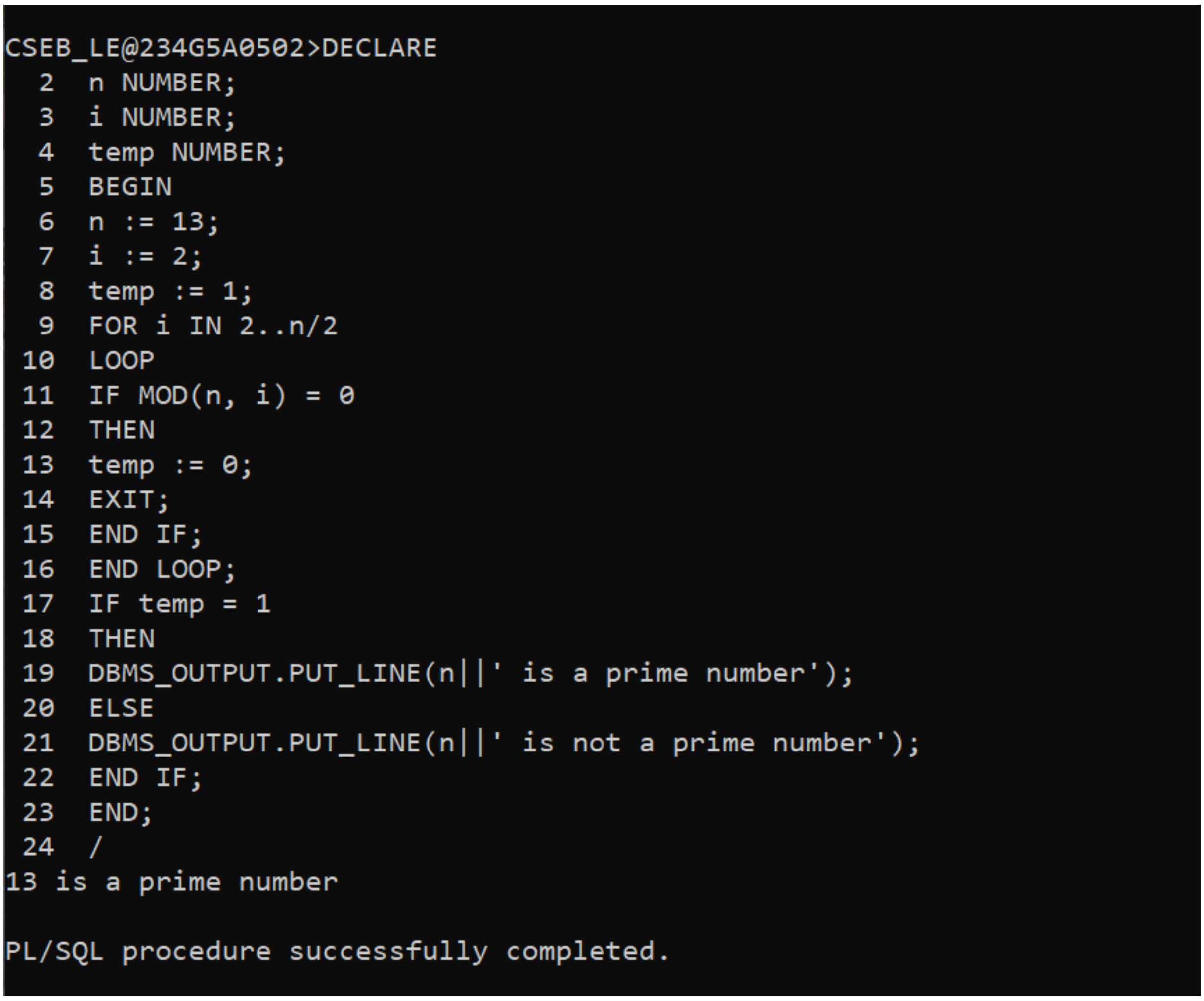
```
CSEB_LE@234G5A0502>DECLARE
fac NUMBER :=1;
n NUMBER :=4;
BEGIN
WHILE n>0 LOOP
fac:=n*fac;
n:=n-1;
END LOOP;
DBMS_OUTPUT.PUT_LINE(FAC);
END;

/
24

PL/SQL procedure successfully completed.
```

EXP-11

PROGRAM :-

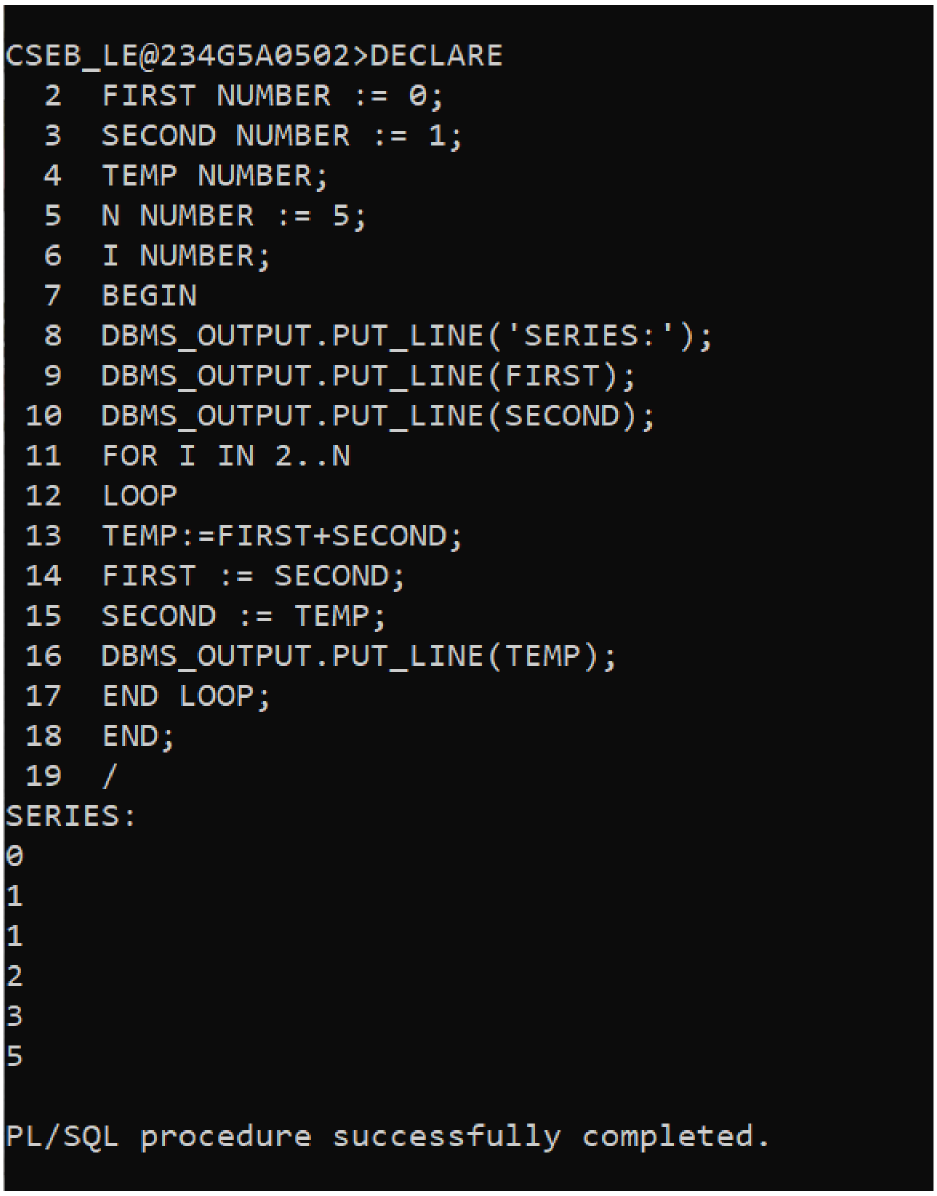


```
CSEB_LE@234G5A0502>DECLARE
n NUMBER;
i NUMBER;
temp NUMBER;
BEGIN
n := 13;
i := 2;
temp := 1;
FOR i IN 2..n/2
LOOP
IF MOD(n, i) = 0
THEN
temp := 0;
EXIT;
END IF;
END LOOP;
IF temp = 1
THEN
DBMS_OUTPUT.PUT_LINE(n||' is a prime number');
ELSE
DBMS_OUTPUT.PUT_LINE(n||' is not a prime number');
END IF;
END;
/
13 is a prime number

PL/SQL procedure successfully completed.
```

EXP-12

PROGRAM :-

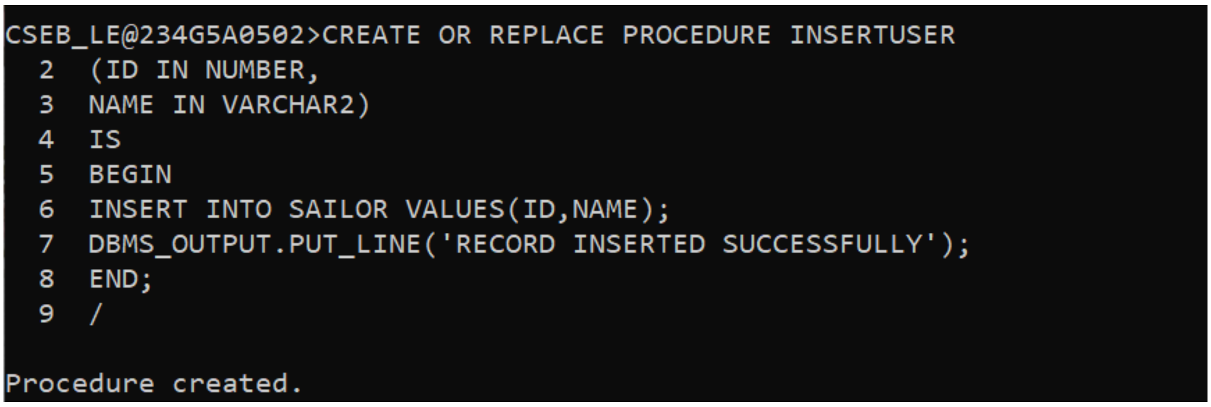


```
CSEB_LE@234G5A0502>DECLARE
FIRST NUMBER := 0;
SECOND NUMBER := 1;
TEMP NUMBER;
N NUMBER := 5;
I NUMBER;
BEGIN
DBMS_OUTPUT.PUT_LINE('SERIES:');
DBMS_OUTPUT.PUT_LINE(FIRST);
DBMS_OUTPUT.PUT_LINE(SECOND);
FOR I IN 2..N
LOOP
TEMP:=FIRST+SECOND;
FIRST := SECOND;
SECOND := TEMP;
DBMS_OUTPUT.PUT_LINE(TEMP);
END LOOP;
END;
/
SERIES:
0
1
1
2
3
5

PL/SQL procedure successfully completed.
```

EXP-13

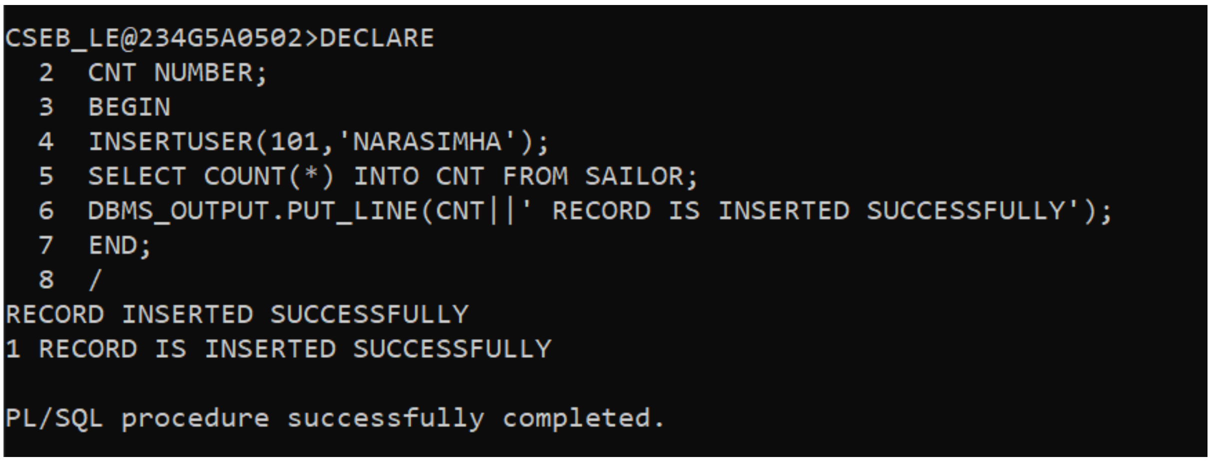
CREATING TABLE :-



```
CSEB_LE@234G5A0502>CREATE OR REPLACE PROCEDURE INSERTUSER
(ID IN NUMBER,
NAME IN VARCHAR2)
IS
BEGIN
INSERT INTO SAILOR VALUES(ID,NAME);
DBMS_OUTPUT.PUT_LINE('RECORD INSERTED SUCCESSFULLY');
END;
/

Procedure created.
```

PROGRAM :-

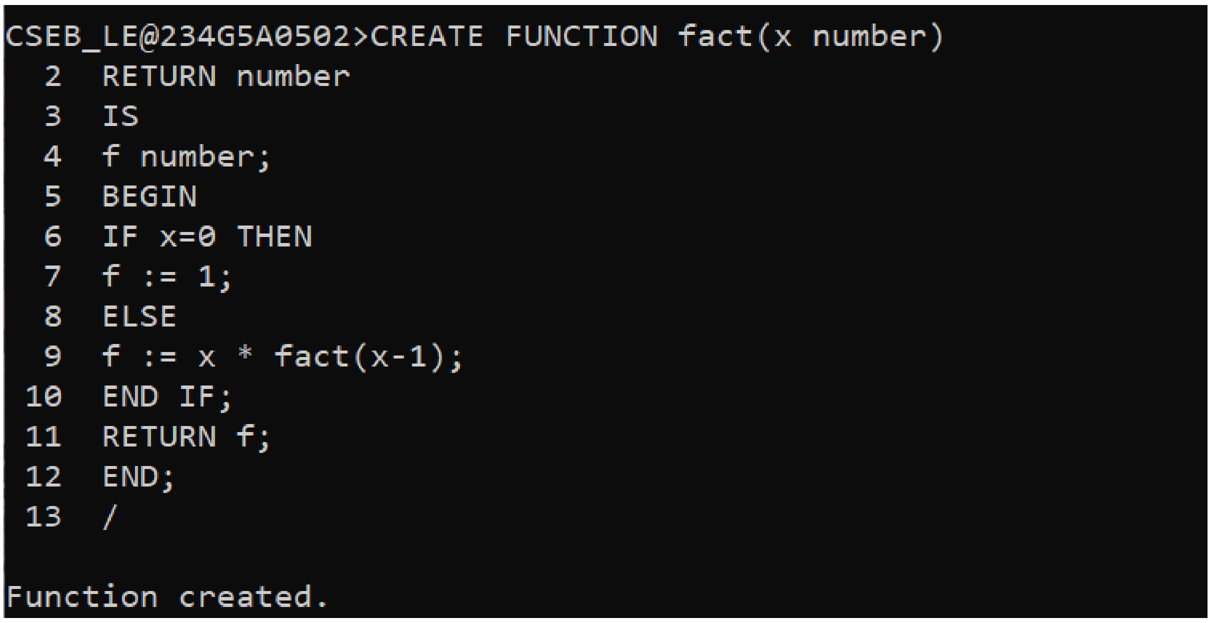


```
CSEB_LE@234G5A0502>DECLARE
CNT NUMBER;
BEGIN
INSERTUSER(101,'NARASIMHA');
SELECT COUNT(*) INTO CNT FROM SAILOR;
DBMS_OUTPUT.PUT_LINE(CNT||' RECORD IS INSERTED SUCCESSFULLY');
END;
/
RECORD INSERTED SUCCESSFULLY
1 RECORD IS INSERTED SUCCESSFULLY

PL/SQL procedure successfully completed.
```

EXP-14

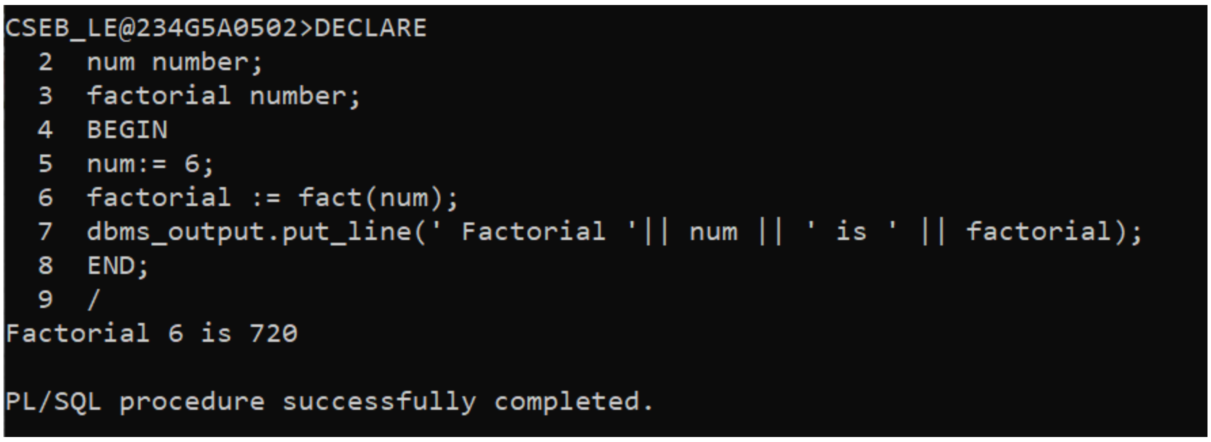
CREATING FUNCTION :-



```
CSEB_LE@234G5A0502>CREATE FUNCTION fact(x number)
RETURN number
IS
f number;
BEGIN
IF x=0 THEN
f := 1;
ELSE
f := x * fact(x-1);
END IF;
RETURN f;
END;
/

Function created.
```

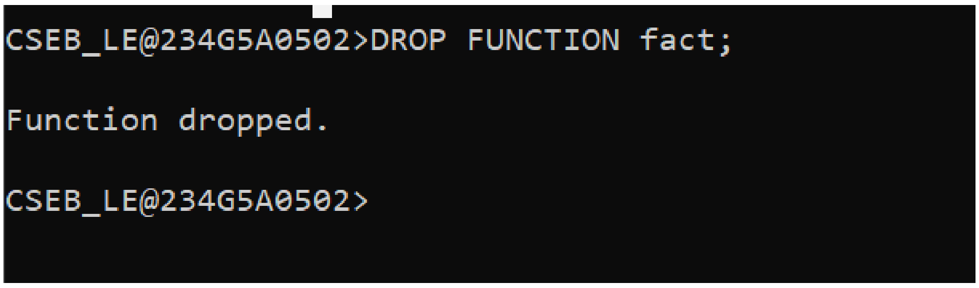
PROGRAM :-



```
CSEB_LE@234G5A0502>DECLARE
num number;
factorial number;
BEGIN
num:= 6;
factorial := fact(num);
dbms_output.put_line(' Factorial '|| num || ' is ' || factorial);
END;
/
Factorial 6 is 720

PL/SQL procedure successfully completed.
```

DROP :-



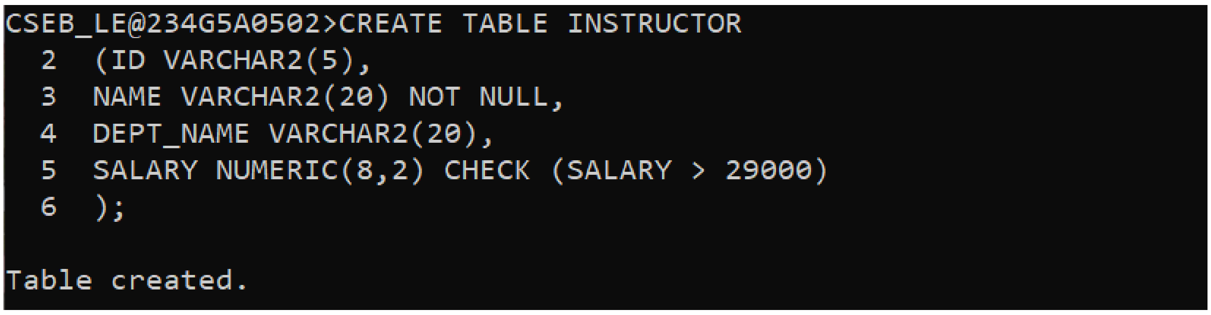
```
CSEB_LE@234G5A0502>DROP FUNCTION fact;

Function dropped.

CSEB_LE@234G5A0502>
```

EXP-15

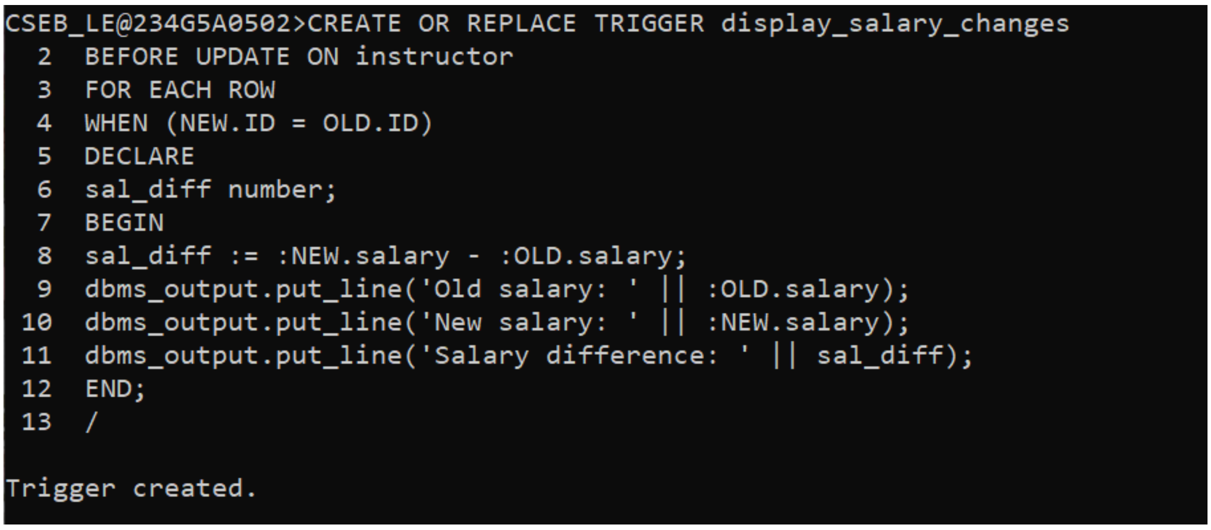
CREATING TABLE :-



```
CSEB_LE@234G5A0502>CREATE TABLE INSTRUCTOR
(ID VARCHAR2(5),
NAME VARCHAR2(20) NOT NULL,
DEPT_NAME VARCHAR2(20),
SALARY NUMERIC(8,2) CHECK (SALARY > 29000)
);

Table created.
```

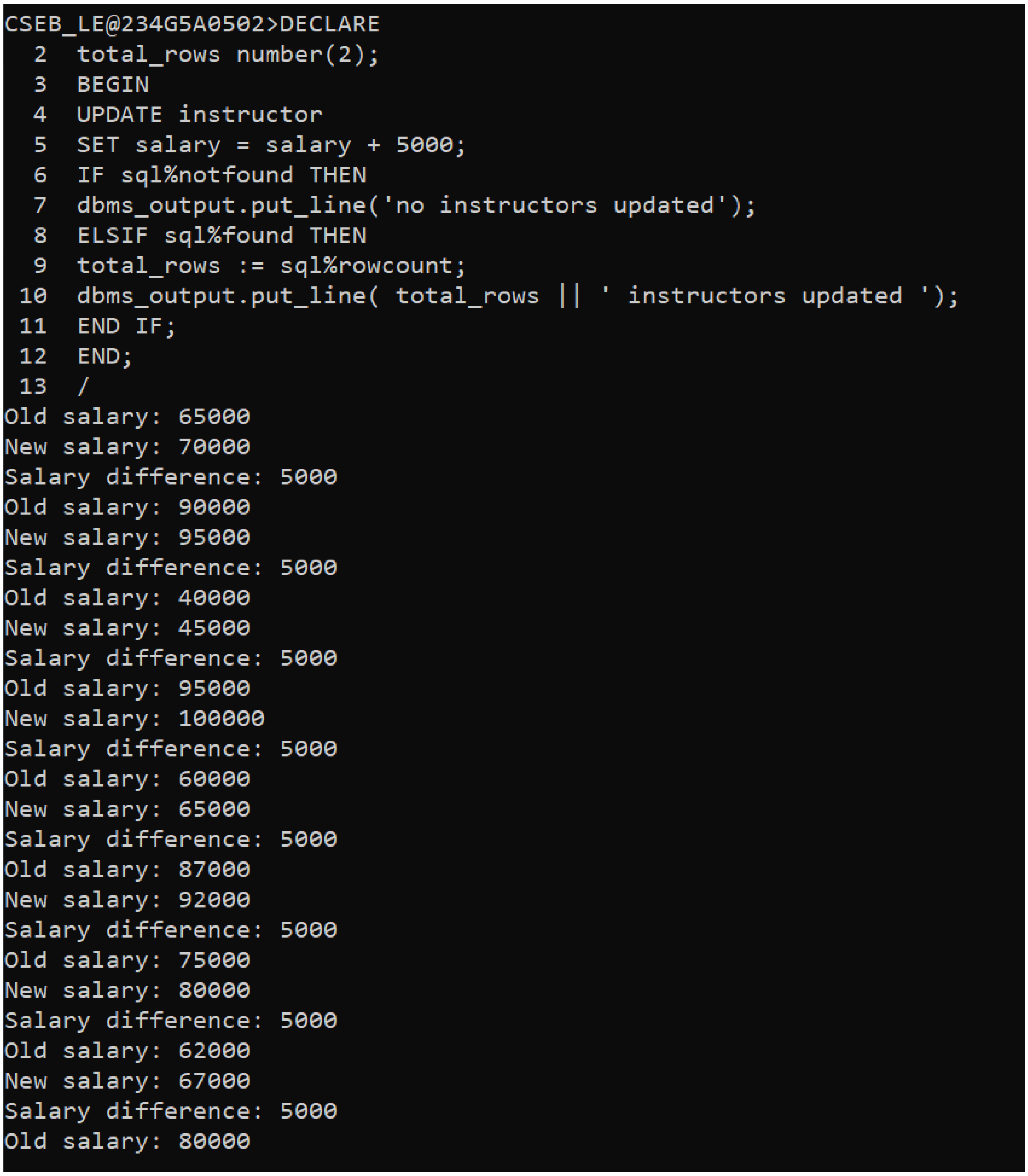
CREATING TRIGGER :-



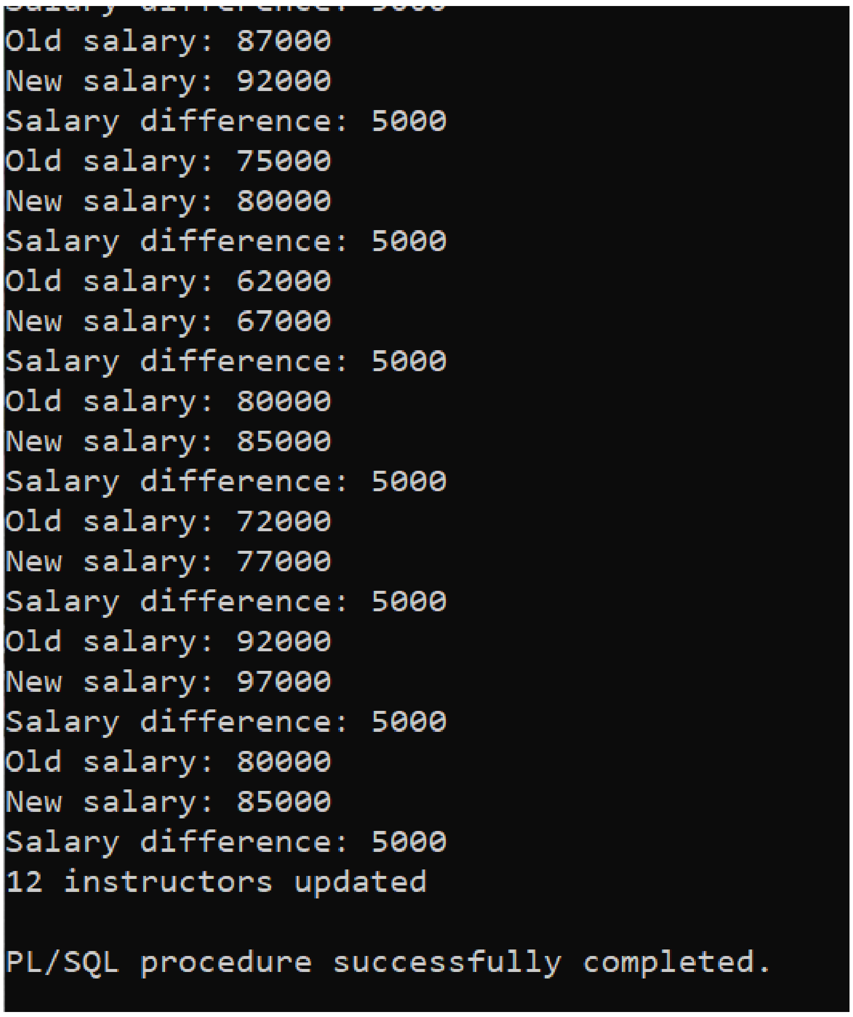
```
CSEB_LE@234G5A0502>CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE UPDATE ON instructor
FOR EACH ROW
WHEN (NEW.ID = OLD.ID)
DECLARE
sal_diff number;
BEGIN
sal_diff := :NEW.salary - :OLD.salary;
dbms_output.put_line('Old salary: ' || :OLD.salary);
dbms_output.put_line('New salary: ' || :NEW.salary);
dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/

Trigger created.
```

PROGRAM :-



```
CSEB_LE@234G5A0502>DECLARE
total_rows number(2);
BEGIN
UPDATE instructor
SET salary = salary + 5000;
IF sql%notfound THEN
dbms_output.put_line('no instructors updated');
ELSIF sql%found THEN
total_rows := sql%rowcount;
dbms_output.put_line( total_rows || ' instructors updated ');
END IF;
END;
/
Old salary: 65000
New salary: 70000
Salary difference: 5000
Old salary: 90000
New salary: 95000
Salary difference: 5000
Old salary: 40000
New salary: 45000
Salary difference: 5000
Old salary: 95000
New salary: 100000
Salary difference: 5000
Old salary: 60000
New salary: 65000
Salary difference: 5000
Old salary: 87000
New salary: 92000
Salary difference: 5000
Old salary: 75000
New salary: 80000
Salary difference: 5000
Old salary: 62000
New salary: 67000
Salary difference: 5000
Old salary: 80000
```

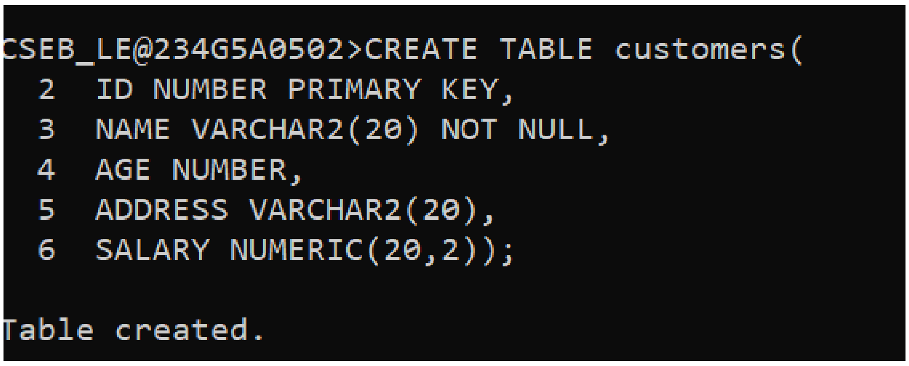


```
Old salary: 87000
New salary: 92000
Salary difference: 5000
Old salary: 75000
New salary: 80000
Salary difference: 5000
Old salary: 62000
New salary: 67000
Salary difference: 5000
Old salary: 80000
New salary: 85000
Salary difference: 5000
Old salary: 72000
New salary: 77000
Salary difference: 5000
Old salary: 92000
New salary: 97000
Salary difference: 5000
Old salary: 80000
New salary: 85000
Salary difference: 5000
12 instructors updated

PL/SQL procedure successfully completed.
```

EXP-16

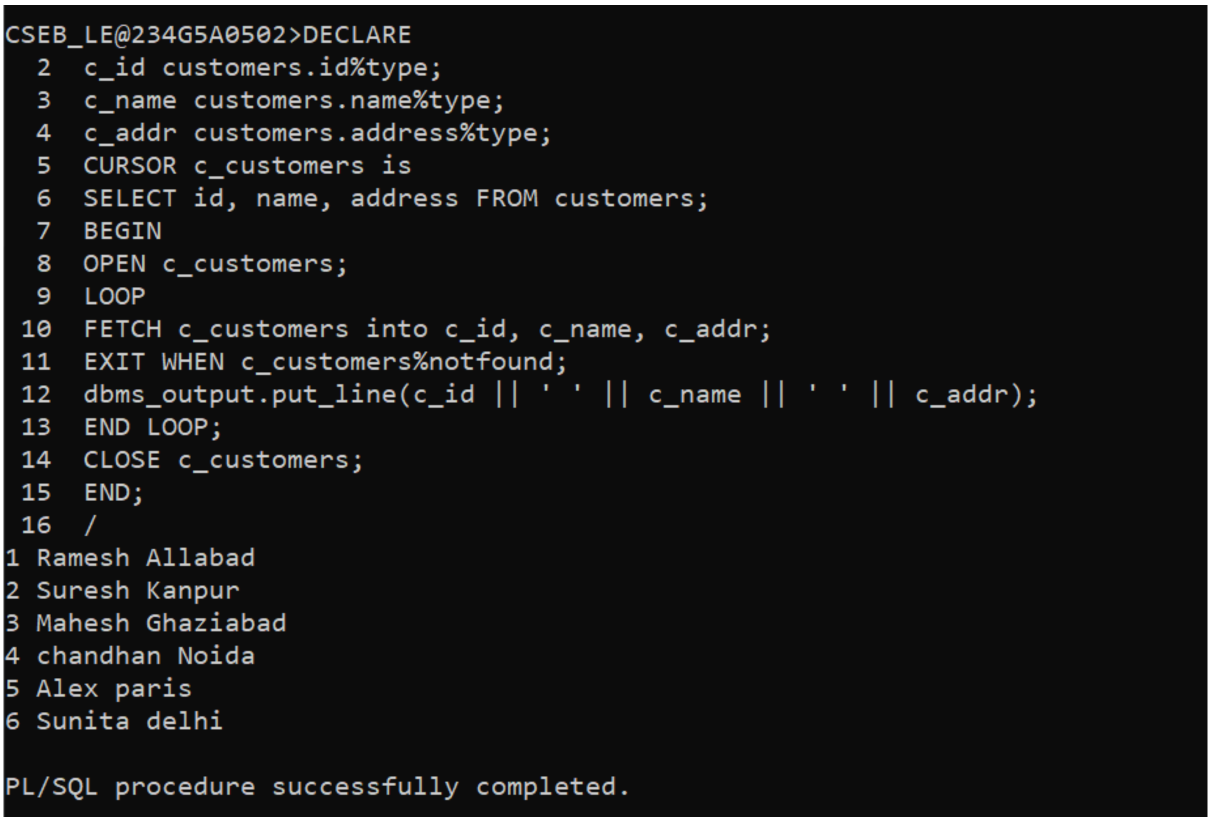
CRDEATING TABLE :-



```
CSEB_LE@234G5A0502>CREATE TABLE customers(
ID NUMBER PRIMARY KEY,
NAME VARCHAR2(20) NOT NULL,
AGE NUMBER,
ADDRESS VARCHAR2(20),
SALARY NUMERIC(20,2));

Table created.
```

PROGRAM :-



```
CSEB_LE@234G5A0502>DECLARE
c_id customers.id%type;
c_name customers.name%type;
c_addr customers.address%type;
CURSOR c_customers is
SELECT id, name, address FROM customers;
BEGIN
OPEN c_customers;
LOOP
FETCH c_customers into c_id, c_name, c_addr;
EXIT WHEN c_customers%notfound;
dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_addr);
END LOOP;
CLOSE c_customers;
END;
/
1 Ramesh Allabad
2 Suresh Kanpur
3 Mahesh Ghaziabad
4 chandhan Noida
5 Alex paris
6 Sunita delhi

PL/SQL procedure successfully completed.
```