

## Experiment-5

**Student Name:** Gauri Prabhakar

**UID:** 18BCS6201

**Branch:** 18AITAIML-2

**Section/Group:** B

**Semester:** 7

**Date of Performance:** 6<sup>th</sup> October, 2021

**Subject Name:** Advanced Database Management Lab

**Subject Code:** CSP - 434

### 1. Aim/Overview of the practical:

To implement PL/SQL programming using Control Structures.

### 2. Task to be done:

To implement PL/SQL programming using Control Structures.

### 3. Steps to be followed:

**Implementing IF/ELSE control statement by comparing two integers and printing the greater of the two:**

#### 1. SET SERVEROUTPUT ON;

DECLARE

NUM1 number:=&N1;

NUM2 number:=&N2;

BEGIN

IF NUM1>NUM2 THEN

DBMS\_OUTPUT.PUT\_LINE('THE GREATER NUMBER IS = '|| NUM1);

ELSE

DBMS\_OUTPUT.PUT\_LINE('THE GREATER NUMBER IS = '|| NUM2);

END IF;

END;

/

```
Run SQL Command Line

PL/SQL procedure successfully completed.

SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  NUM1 number:=&N1;
  3  NUM2 number:=&N2;
  4  BEGIN
  5  IF NUM1>NUM2 THEN
  6  DBMS_OUTPUT.PUT_LINE('THE GREATER NUMBER IS = ' || NUM1);
  7  ELSE
  8  DBMS_OUTPUT.PUT_LINE('THE GREATER NUMBER IS = ' || NUM2);
  9  END IF;
 10  END;
 11  /
Enter value for n1: 8
old  2: NUM1 number:=&N1;
new  2: NUM1 number:=8;
Enter value for n2: 9
old  3: NUM2 number:=&N2;
new  3: NUM2 number:=9;
THE GREATER NUMBER IS = 9

PL/SQL procedure successfully completed.

new  3: NUM2 number:=9;
```

## Creating a table MARKS:

2. CREATE TABLE MARKS(Name varchar(50), Marks int);  
INSERT INTO MARKS VALUES('Gauri',100);  
INSERT INTO MARKS VALUES('Lilly',80);  
INSERT INTO MARKS VALUES('Rose',70);  
INSERT INTO MARKS VALUES('Chelsea',50);  
INSERT INTO MARKS VALUES('Amy',90);

```
Run SQL Command Line

SQL> CREATE TABLE MARKS(Name varchar(50), Marks int);

Table created.

SQL> INSERT INTO MARKS VALUES('Gauri',100);

1 row created.

SQL> INSERT INTO MARKS VALUES('Lilly',80);

1 row created.

SQL> INSERT INTO MARKS VALUES('Rose',70);

1 row created.

SQL> INSERT INTO MARKS VALUES('Chelsea',50);

1 row created.

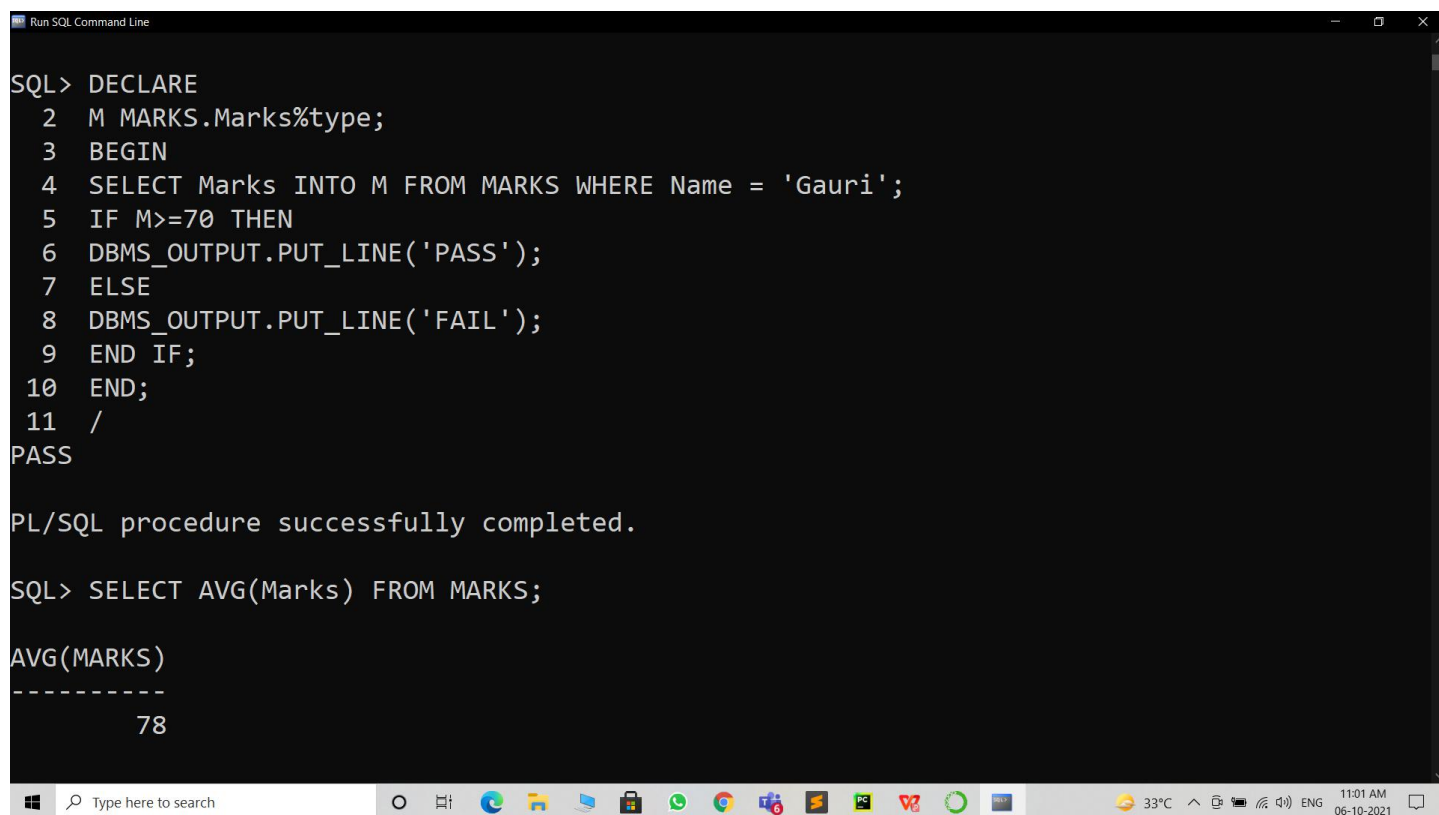
SQL> INSERT INTO MARKS VALUES('Amy',90);

1 row created.
```

**Implementing IF/ELSE control statement on the table MARKS and printing the average marks from the table:**

**3. DECLARE**

```
M MARKS.Marks%type;
BEGIN
SELECT Marks INTO M FROM MARKS WHERE Name = 'Gauri';
IF M>=70 THEN
DBMS_OUTPUT.PUT_LINE('PASS');
ELSE
DBMS_OUTPUT.PUT_LINE('FAIL');
END IF;
END;
/
SELECT AVG(Marks) FROM MARKS;
```



```
Run SQL Command Line
SQL> DECLARE
 2  M MARKS.Marks%type;
 3  BEGIN
 4  SELECT Marks INTO M FROM MARKS WHERE Name = 'Gauri';
 5  IF M>=70 THEN
 6  DBMS_OUTPUT.PUT_LINE('PASS');
 7  ELSE
 8  DBMS_OUTPUT.PUT_LINE('FAIL');
 9  END IF;
10  END;
11  /
PASS

PL/SQL procedure successfully completed.

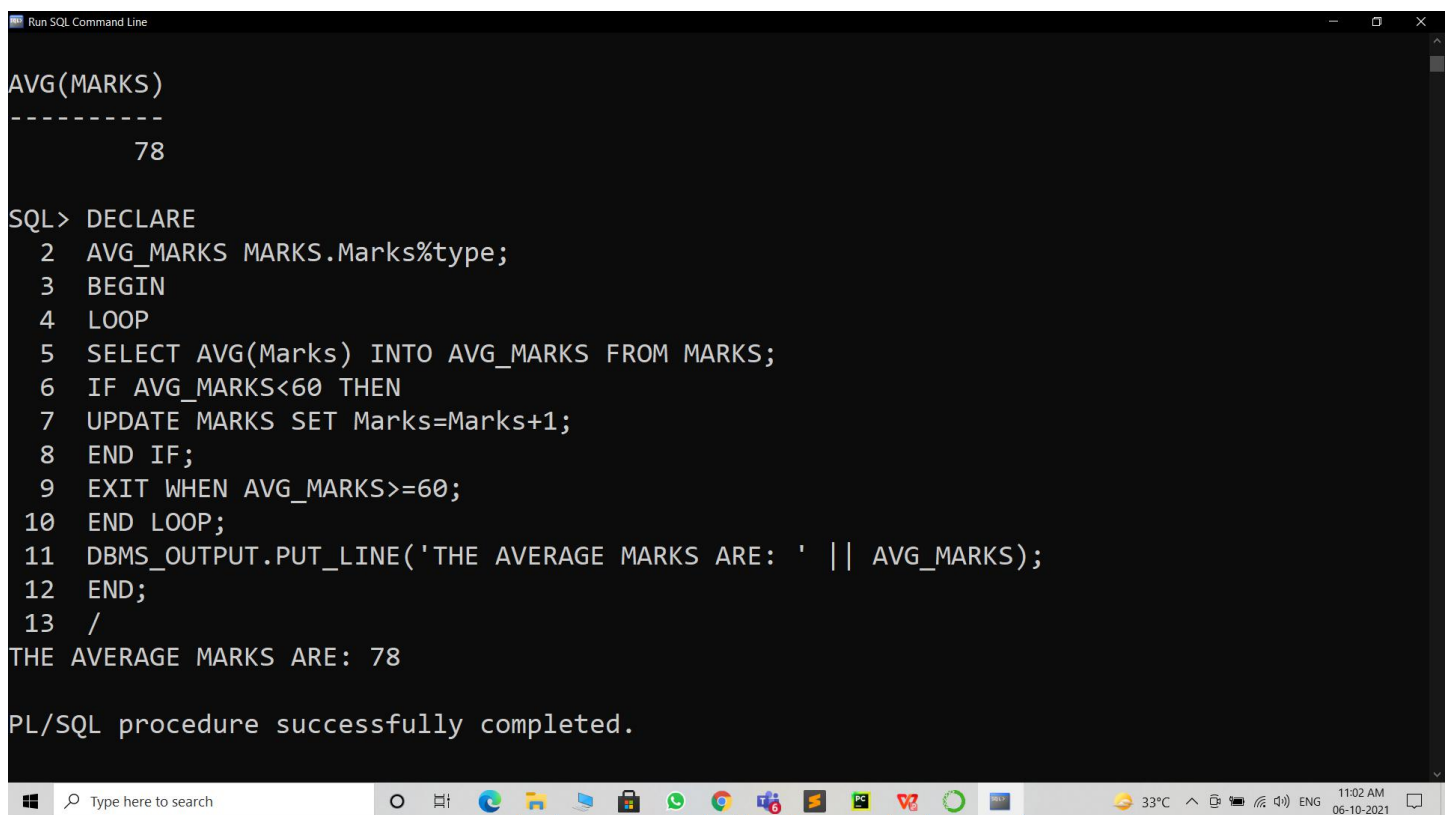
SQL> SELECT AVG(Marks) FROM MARKS;

AVG(MARKS)
-----
          78
```

## Implementing LOOP control structure and printing the average of the marks in the table MARKS:

### 4. DECLARE

```
AVG_MARKS MARKS.Marks%type;
BEGIN
LOOP
SELECT AVG(Marks) INTO AVG_MARKS FROM MARKS;
IF AVG_MARKS<60 THEN
UPDATE MARKS SET Marks=Marks+1;
END IF;
EXIT WHEN AVG_MARKS>=60;
END LOOP;
DBMS_OUTPUT.PUT_LINE('THE AVERAGE MARKS ARE: ' || AVG_MARKS);
END;
/
```



```
Run SQL Command Line

AVG(MARKS)
-----
          78

SQL> DECLARE
 2  AVG_MARKS MARKS.Marks%type;
 3  BEGIN
 4  LOOP
 5  SELECT AVG(Marks) INTO AVG_MARKS FROM MARKS;
 6  IF AVG_MARKS<60 THEN
 7  UPDATE MARKS SET Marks=Marks+1;
 8  END IF;
 9  EXIT WHEN AVG_MARKS>=60;
10  END LOOP;
11  DBMS_OUTPUT.PUT_LINE('THE AVERAGE MARKS ARE: ' || AVG_MARKS);
12  END;
13  /

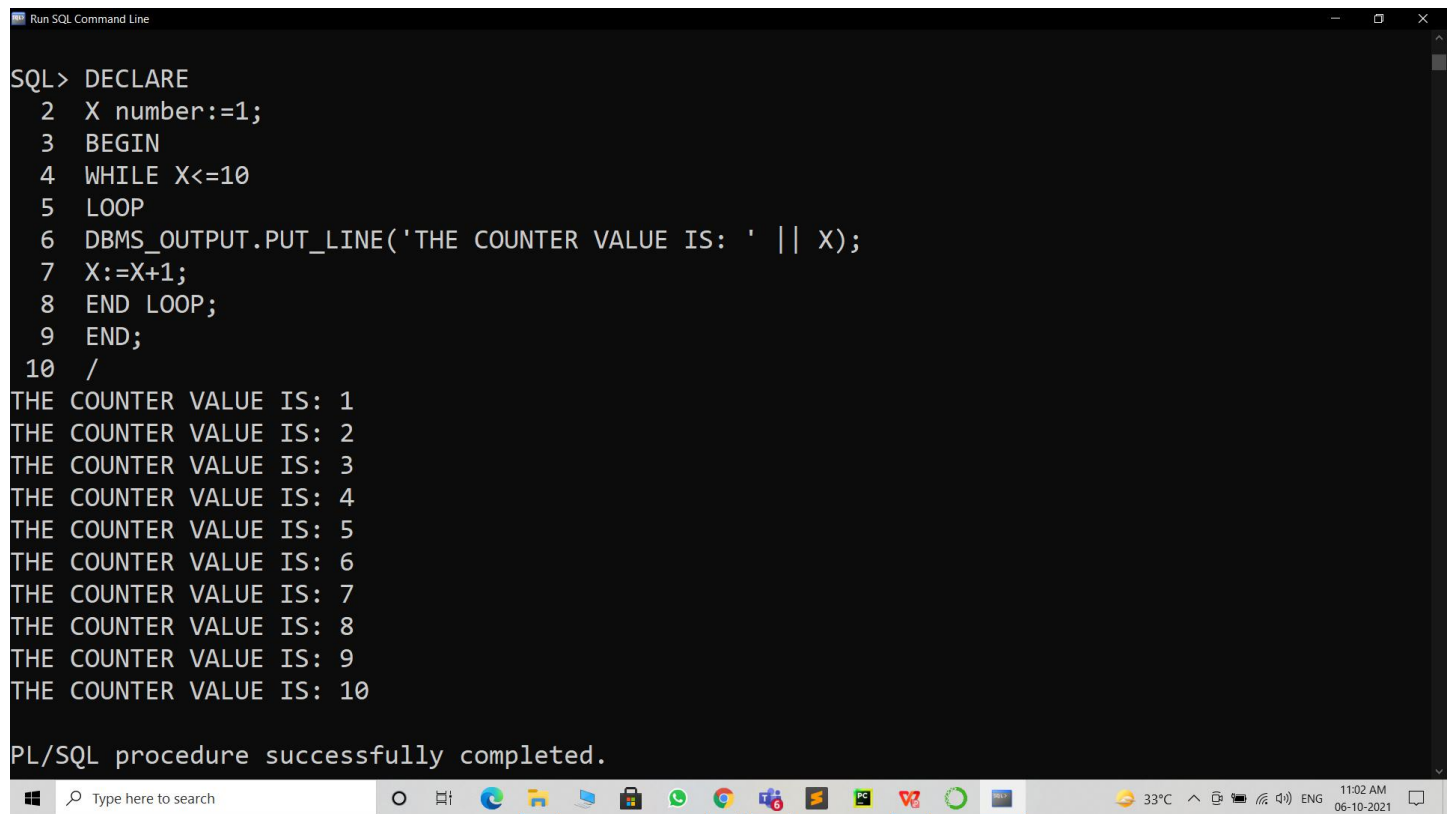
THE AVERAGE MARKS ARE: 78

PL/SQL procedure successfully completed.
```

## Implementing the WHILE LOOP:

### 5. DECLARE

```
X number:=1;  
BEGIN  
WHILE X<=10  
LOOP  
DBMS_OUTPUT.PUT_LINE('THE COUNTER VALUE IS: ' || X);  
X:=X+1;  
END LOOP;  
END;  
/
```

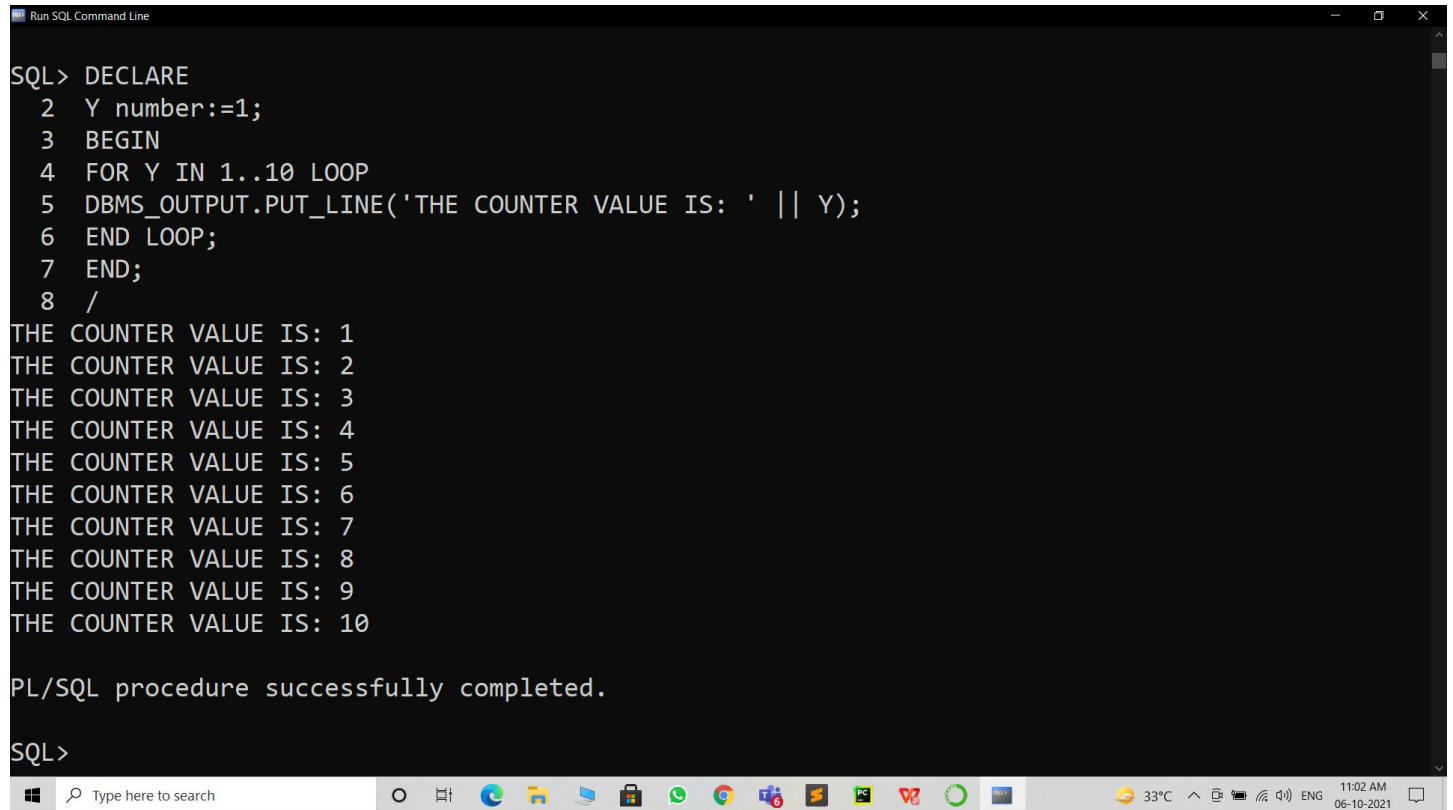


```
Run SQL Command Line  
SQL> DECLARE  
2 X number:=1;  
3 BEGIN  
4 WHILE X<=10  
5 LOOP  
6 DBMS_OUTPUT.PUT_LINE('THE COUNTER VALUE IS: ' || X);  
7 X:=X+1;  
8 END LOOP;  
9 END;  
10 /  
THE COUNTER VALUE IS: 1  
THE COUNTER VALUE IS: 2  
THE COUNTER VALUE IS: 3  
THE COUNTER VALUE IS: 4  
THE COUNTER VALUE IS: 5  
THE COUNTER VALUE IS: 6  
THE COUNTER VALUE IS: 7  
THE COUNTER VALUE IS: 8  
THE COUNTER VALUE IS: 9  
THE COUNTER VALUE IS: 10  
  
PL/SQL procedure successfully completed.
```

## Implementing the FOR LOOP:

### 6. DECLARE

```
Y number:=1;
BEGIN
FOR Y IN 1..10 LOOP
DBMS_OUTPUT.PUT_LINE('THE COUNTER VALUE IS: ' || Y);
END LOOP;
END;
/
```



```
SQL> DECLARE
2 Y number:=1;
3 BEGIN
4 FOR Y IN 1..10 LOOP
5 DBMS_OUTPUT.PUT_LINE('THE COUNTER VALUE IS: ' || Y);
6 END LOOP;
7 END;
8 /
THE COUNTER VALUE IS: 1
THE COUNTER VALUE IS: 2
THE COUNTER VALUE IS: 3
THE COUNTER VALUE IS: 4
THE COUNTER VALUE IS: 5
THE COUNTER VALUE IS: 6
THE COUNTER VALUE IS: 7
THE COUNTER VALUE IS: 8
THE COUNTER VALUE IS: 9
THE COUNTER VALUE IS: 10

PL/SQL procedure successfully completed.

SQL>
```

#### 4. Result/Output/Writing Summary:

- Successfully implemented IF/ELSE control structure.
- Successfully implemented FOR and WHILE control LOOPS.
- Successfully understood the functioning and importance of the above mentioned.
- Successfully implemented CONTROL STRUCTURES on a table.

#### 5. Learning outcomes (What I have learnt):

- How to implement IF/ELSE on SQL Command Line.
- How to implement FOR LOOP on SQL Command Line.
- How to implement WHILE LOOP on SQL Command Line.
- How to implement CONTROL STRUCTURES on a table.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			

