

## EXPERIMENT - 10

### (PL/SQL CONDITIONAL STATEMENT AND LOOPING)

1. Write a PL/SQL program to check whether a number is even or odd.

```
SQL> DECLARE
2  N NUMBER := &N;
3  BEGIN
4  IF MOD(N, 2) = 0 THEN
5  DBMS_OUTPUT.PUT_LINE(N || ' IS EVEN');
6  ELSE
7  DBMS_OUTPUT.PUT_LINE(N || ' IS ODD');
8  END IF;
9  END;
10 /
Enter value for n: 5
old 2: N NUMBER := &N;
new 2: N NUMBER := 5;

PL/SQL procedure successfully completed.
```

2. Write a PL/SQL program to check whether a date falls on weekend i.e. SATURDAY or SUNDAY.

```
SQL> DECLARE
2  v_date DATE := TO_DATE('&v_date', 'DD-MM-YYYY');
3  v_day VARCHAR2(10);
4  BEGIN
5  v_day := TO_CHAR(v_date, 'DAY');
6  IF TRIM(v_day) IN ('SATURDAY', 'SUNDAY') THEN
7  DBMS_OUTPUT.PUT_LINE('The date falls on a weekend.');
```

3. Write a PL/SQL program to check whether a given number is positive, negative or zero.

```
SQL> DECLARE
2  num NUMBER := &num;
3  BEGIN
4  IF num > 0 THEN
5  DBMS_OUTPUT.PUT_LINE('Positive Number');
```

4. Write a PL/SQL program to check whether a given character is letter or digit.

```
SQL> DECLARE
2   char_input CHAR(1) := '&char_input';
3   BEGIN
4       IF char_input BETWEEN '0' AND '9' THEN
5           DBMS_OUTPUT.PUT_LINE('It is a Digit.');
```

6 ELIF char\_input BETWEEN 'A' AND 'Z' OR char\_input BETWEEN 'a' AND 'z' THEN  
7 DBMS\_OUTPUT.PUT\_LINE('It is a Letter.');

8 ELSE  
9 DBMS\_OUTPUT.PUT\_LINE('It is neither a Letter nor a Digit.');

10 END IF;  
11 END;  
12 /

Enter value for char\_input: H  
old 2: char\_input CHAR(1) := '&char\_input';  
new 2: char\_input CHAR(1) := 'H';

PL/SQL procedure successfully completed.

5. Write a PL/SQL block to find the maximum of two numbers.

```
SQL> DECLARE
2   num1 NUMBER := &num1;
3   num2 NUMBER := &num2;
4   max_num NUMBER;
5   BEGIN
6       max_num := GREATEST(num1, num2);
7       DBMS_OUTPUT.PUT_LINE('Maximum Number: ' || max_num);
8   END;
9   /
```

Enter value for num1: 10  
old 2: num1 NUMBER := &num1;  
new 2: num1 NUMBER := 10;

Enter value for num2: 15  
old 3: num2 NUMBER := &num2;  
new 3: num2 NUMBER := 15;

PL/SQL procedure successfully completed.

6. Write a PL/SQL block to find the grade of a student based on marks.

```
SQL> DECLARE
2   marks NUMBER := &marks;
3   grade CHAR(1);
4   BEGIN
5       IF marks >= 90 THEN
6           grade := 'A';
7       ELIF marks >= 80 THEN
8           grade := 'B';
9       ELIF marks >= 70 THEN
10          grade := 'C';
11       ELIF marks >= 60 THEN
12          grade := 'D';
13       ELSE
14          grade := 'F';
15       END IF;
16
17       DBMS_OUTPUT.PUT_LINE('Grade: ' || grade);
18   END;
19   /
```

Enter value for marks: 86  
old 2: marks NUMBER := &marks;  
new 2: marks NUMBER := 86;

PL/SQL procedure successfully completed.

7. Write a PL/SQL block to check if a given year is a leap year or not.

```
SQL> DECLARE
2   year NUMBER := &year;
3   BEGIN
4   IF (MOD(year, 4) = 0 AND MOD(year, 100) <> 0) OR (MOD(year, 400) = 0) THEN
5       DBMS_OUTPUT.PUT_LINE(year || ' is a Leap Year');
6   ELSE
7       DBMS_OUTPUT.PUT_LINE(year || ' is not a Leap Year');
8   END IF;
9   END;
10  /
Enter value for year: 2025
old 2:   year NUMBER := &year;
new 2:   year NUMBER := 2025;

PL/SQL procedure successfully completed.
```

8. Write a program in PL/SQL to print the prime numbers between 1 to 50.

```
SQL> DECLARE
2   num NUMBER;
3   is_prime BOOLEAN;
4   BEGIN
5   FOR num IN 2..50 LOOP
6       is_prime := TRUE;
7
8       FOR i IN 2..SQRT(num) LOOP
9           IF MOD(num, i) = 0 THEN
10              is_prime := FALSE;
11              EXIT;
12           END IF;
13       END LOOP;
14
15       IF is_prime THEN
16           DBMS_OUTPUT.PUT_LINE(num);
17       END IF;
18   END LOOP;
19   END;
20  /

PL/SQL procedure successfully completed.
```

9. Write a program in PL/SQL to print 1st n numbers.

```
SQL> DECLARE
2   n NUMBER := &n;
3   BEGIN
4   FOR i IN 1..n LOOP
5       DBMS_OUTPUT.PUT_LINE(i);
6   END LOOP;
7   END;
8   /
Enter value for n: 15
old 2:   n NUMBER := &n;
new 2:   n NUMBER := 15;

PL/SQL procedure successfully completed.
```

10. Write a PL/SQL program to calculate the factorial of a number using a loop.

```
SQL> DECLARE
2   num NUMBER := &num;
3   fact NUMBER := 1;
4 BEGIN
5   FOR i IN 1..num LOOP
6     fact := fact * i;
7   END LOOP;
8   DBMS_OUTPUT.PUT_LINE('Factorial of ' || num || ' is ' || fact);
9 END;
10 /
Enter value for num: 5
old 2:   num NUMBER := &num;
new 2:   num NUMBER := 5;
PL/SQL procedure successfully completed.
```

11. Write a PL/SQL program to display the Fibonacci series up to a specified limit using a loop.

```
SQL> DECLARE
2   n NUMBER := &n;
3   a NUMBER := 0;
4   b NUMBER := 1;
5   temp NUMBER;
6 BEGIN
7   DBMS_OUTPUT.PUT_LINE(a);
8   DBMS_OUTPUT.PUT_LINE(b);
9
10  FOR i IN 3..n LOOP
11    temp := a + b;
12    DBMS_OUTPUT.PUT_LINE(temp);
13    a := b;
14    b := temp;
15  END LOOP;
16 END;
17 /
Enter value for n: 8
old 2:   n NUMBER := &n;
new 2:   n NUMBER := 8;
PL/SQL procedure successfully completed.
```

12. Write a PL/SQL program to find the sum of digits of a number using a loop.

```
SQL> DECLARE
2   num NUMBER := &num;
3   sum_digits NUMBER := 0;
4   digit NUMBER;
5 BEGIN
6   WHILE num > 0 LOOP
7     digit := MOD(num, 10);
8     sum_digits := sum_digits + digit;
9     num := TRUNC(num / 10);
10  END LOOP;
11  DBMS_OUTPUT.PUT_LINE('Sum of Digits: ' || sum_digits);
12 END;
13 /
Enter value for num: 7
old 2:   num NUMBER := &num;
new 2:   num NUMBER := 7;
PL/SQL procedure successfully completed.
```

Submitted By:

Name: ANNESHA DAS

Regd No.: 2301020224

Group: 01 Sem: 4<sup>th</sup>