Ex.No.9

PL/SQL Conditional and Iterative Statements

Aim:

To manipulate Conditional and Iterative statements in PL/SQL.

PL/SQL

PL/SQL is a combination of SQL along with the procedural features of programming languages.

Block Structure

PL/SQL code is grouped into structures called blocks.

The PL/SQL block divided into three section: declaration section, the executable section and the exception section

The structure of a typical PL/SQL block is shown in the listing:

```
declare
        < declaration section >
begin
        < executable  commands>
exception
        <exception handling>
end;
```

Declaration Section :

Defines and initializes the variables and cursor used in the block

Executable commands :

Uses flow-control commands (such as IF command and loops) to execute the commands and assign values to the declared variables

Exception handling :

Provides handling of error conditions

Creating and Executing PL/SQL Programs

Edit your PL/SQL program in a text editor as text file, and save with '.sql' extension.
Execute the following command once for a session to get displayed the output.

SQL> *set serveroutput on;*

Now execute the program using the following command.

SQL> *start* filename;

(or)

SQL> @filename;

Note : Give absolute path of the filename if you saved the file in some directory.

Ex.

SQL> *start* z:\plsql\ex11; (or) SQL> @ z:\plsql\ex11;

Control Structures

(i) IF Statements

There are three forms of IF statements: IF-THEN, IF-THEN-ELSE, and IF THEN-ELSIF. The third form of IF statement uses the keyword ELSIF (NOT ELSEIF) to introduce additional conditions, as follows:

```
IF condition1 THEN
sequence_of_statements1;
ELSIF condition2 THEN
sequence_of_statements2;
```

```
ELSE
sequence_of_statements3;
END IF;
```

(ii) LOOP and EXIT Statements

There are three forms of LOOP statements. They are LOOP, WHILE-LOOP, and FOR-LOOP.

LOOP

The simplest form of LOOP statement is the basic (or infinite) loop, which encloses a sequence of statements between the keywords LOOP and END LOOP, as follows:

```
LOOP
sequence_of_statements3;
...
END LOOP;
```

With each iteration of the loop, the sequence of statements is executed, then control resumes at the top of the loop. If further processing is undesirable or impossible, you can use the EXIT statement to complete the loop. You can place one or more EXIT statements anywhere inside a loop, but nowhere outside a loop. There are two forms of EXIT statements: EXIT and EXIT-WHEN.

(iii) WHILE-LOOP

The WHILE-LOOP statement associates a condition with a sequence of statements enclosed by the keywords LOOP and END LOOP, as follows:

```
WHILE condition LOOP
sequence_of_statements;
...
END LOOP;
```

(iv) FOR-LOOP

FOR loops iterate over a specified range of integers. The range is part of an iteration scheme, which is enclosed by the keywords FOR and LOOP.

```
FOR counter IN [REVERSE] lower_bound..upper_bound LOOP
sequence_of_statements;
...
END LOOP;
```

The lower bound need not be 1. However, the loop counter increment (or decrement) must be 1. PL/SQL lets you determine the loop range dynamically at run time, as the following example shows:

```
SELECT COUNT(empno) INTO emp_count FROM emp;
FOR i IN 1..emp_count LOOP
...
END LOOP;
```

The loop counter is defined only within the loop

(v) GOTO and NULL statements

The NULL statement can make the meaning and action of conditional statements clear and so improve readability.

```
BEGIN
...
```

```
GOTO insert_row;
...
<<insert_row>>
INSERT INTO emp VALUES ...
END;
```

A GOTO statement cannot branch into an IF statement, LOOP statement, or subblock. A GOTO statement cannot branch from one IF statement clause to another. A GOTO statement cannot branch out of a subprogram. Finally, a GOTO statement cannot branch from an exception handler into the current block.

Q1)Write a PL/SQL program to find the largest of three numbers.

```
Declare
      a number;
      b number;
      c number;
Begin
      dbms_output.put_line('Enter a:');
      a:=&a;
      dbms_output.put_line('Enter b:');
      b:=&b;
      dbms_output.put_line('Enter c:');
      c:=&c;
      dbms_output.put_line('NUMBERS');
      IF a>b AND a>c THEN
            dbms_output.put_line('A is Maximum');
      ELSIF (b>a) AND (b>c) then
            dbms_output.put_line('B is Maximum');
      ELSE
            dbms_output.put_line('C is Maximum');
      END IF;
End;
/
```

Q2) Write a PL/SQL program to swap two numbers
Q3) Write a PL/SQL program to find the factorial of a given number.
Q4) Write a PL / SQL program to check whether the given number is prime or not.
Q5) Write a PL/SQL Block to modify the department name of the department 71 if it is not 'HRD'.

```
Declare
deptname dept.dname%type;
Begin

-- complete the block
End;
/
```