Lab 10: Control Structures

Objective(s):

To learn the Control Structures

Control Structures

1. CONDITIONAL CONTROL STRUCTURE

Simple IF Statements:

In PL/SQL, simple conditions can be applied by IF-THEN-ELSE-END IF statements (space between END and IF).

Syntax:

```
If <condition> Then
<action>
Else
<action>
End If;
```

Nested IF-THEN-ELSE Statement:

Either set of actions of the result of the first If statement can include further if statements before specific actions are performed.

Syntax:

```
If <condition1> Then
<Statementt1>;
Else
If <condition2> Then
<Statement2>;
End If;
End If;
```

CASE STATEMENT (no return value):

```
case
when n = 1 then Action1;
when n = 2 then Action2;
when n = 3 then Action3;
when ( n > 3 and n < 8 ) then Action4;
else ActionOther;
end case;
```

CASE EXPRESSION (returns value):

```
text := case
when n = 1 then one
when n = 2 then two
when n = 3 then three
when ( n > 3 and n < 8 ) then four_through_seven
else other
end;
```

2. ITERATIVE CONTROL STRUCTURE

There are three types of loops:

1) Basic Loop.

2) FOR Loop.

3) WHILE Loop.

- Looping constructs a second type of control structures:..

1) Basic Loop to provide repetitive actions without overall condition.
2) FOR Loop to provide iterative control of actions based on a count.
3) WHILE Loop to provide iterative control of actions based on a condition.
4) EXIT statement to terminate loops. Without the EXIT statement loop would be infinite.

BASIC LOOP:

- The simplest form of loop statement is the ***basic (or infinite) loop***, which encloses a sequence of statements b/w the keywords LOOP and END LOOP (space between END and LOOP).
- Basic loop is like a DO-WHILE loop in C.
- A basic loop allows execution of its statement at least once, even if condition is already met upon entering the loop.
- Without the exit statement, loop would be indefinite.

Basic Loop Syntax:

```
Loop -- delimiter
statement1; -- statement
……
Exit (When condition); -- delimiter
End Loop;
```

Example:

```
Declare
        A number:=1;
Begin
        Loop
                Dbms_output.put_line('Bahria University');
                A:=a+1;
                Exit when a>10;
        End loop;
End;
```

WHILE Loop:

- Use the WHILE Loop to repeat statements while a condition is true.
- The condition is evaluated at the start of each iteration.

Syntax:

```
WHILE condition
LOOP
statement1;
statement2;
…..
END LOOP;
```

Example:

```
Declare
A number:=1;
Begin
While a<10
Loop
Dbms_output.put_line('Bahria University');
A:=a+1;
End loop;
End;
```

FOR Loop:

Syntax:

```
FOR counter IN [REVERSE] lower_bound .. Upper_bound
LOOP
statement1;
statement2;
…
END LOOP;
```

Exmaple:

```
Begin
        For a in 1..10
        Loop
                Dbms_output.put_line('Bahria University');
        End loop;
End;
```

Exercise

1. Write a PL/SQL program using CASE statement to assign COMM to employee 7369. COMM will be assigned with respect to the JOB of the employee 7369.

   If JOB = CLERK then COMM = 5% of sal

   If JOB = ANALYST then COMM = 10% of sal

   If JOB = MANAGER then COMM = 15% of sal

   The commission of the employee must be update in the EMP table.

2. Write a PL/SQL program using Basic Loop to print multiplication table from 2 to 10. Each table must be printed from 1 to 10.

3. Write a PL/SQL program using While Loop to display all DEPTNO,DNAME and LOC from DEPT table. Assuming the difference between two deptno is 10.

4. Write a PL/SQL program using FOR LOOP to insert 3 new DEPTNO in the dept table. New DEPTNO should have a difference of 10 between them and must be greater then the existing maximum DEPTNO in DEPT table.