Database Systems LAB # 13

Control Structures in Oracle RDBMS

In addition to SQL commands,PL/SQL can also process data using flow of statements. The flow of control statements are classified into the following categories.

- Conditional control –Branching
- Iterative control – looping
- Sequential control

```
Enable the output    set serveroutput on size 30000;
```

BRANCHING in PL/SQL:

Sequence of statements can be executed on satisfying certain condition.
If statements are being used and different forms of if are:

1. Simple IF
2. ELSIF
3. ELSE IF

SIMPLE IF:
Syntax:

IF condition THEN
statement1;
statement2;
END IF;

IF-THEN-ELSE STATEMENT:
Syntax:
IF condition THEN
statement1;
ELSE
statement2;
END IF;

ELSIF STATEMENTS:
Syntax:
IF condition1 THEN
statement1;
ELSIF condition2 THEN
statement2;
ELSIF condition3 THEN
statement3;

```
ELSE
Statement4;
END IF;
```

NESTED IF:

Syntax:

```
IF condition THEN
statement1;
ELSE
IF condition THEN
statement2;
ELSE
statement3;
END IF;
END IF;
ELSE
statement3;
END IF;
```

Examples:

```
declare
a number;
b number;
begin
a:=&a;
b:=&b;
if a=b then
dbms_output.put_line('BOTH ARE EQUAL');
elsif a>b then
dbms_output.put_line('A IS GREATER');
else
dbms_output.put_line('B IS GREATER');
end if;
end;
```

SELECTION IN PL/SQL(Sequential Controls) :

SIMPLE CASE

Syntax:

```
CASE SELECTOR
WHEN Expr1 THEN statement1;
WHEN Expr2 THEN statement2;
:
ELSE
Statement n;
END CASE;
```

SEARCHED CASE:
CASE
WHEN searchcondition1 THEN statement1;
WHEN searchcondition2 THEN statement2;
::
ELSE
statementn;
END CASE;

Examples 1:

```
declare
grade varchar(3);
begin
grade:='&grade';
CASE grade
   WHEN 'A' THEN dbms_output.put_line('Excellent');
   WHEN 'B' THEN dbms_output.put_line('Very Good');
   WHEN 'C' THEN dbms_output.put_line('Good');
   WHEN 'D' THEN dbms_output.put_line('Fair');
   WHEN 'F' THEN dbms_output.put_line('Poor');
   ELSE dbms_output.put_line('No such grade');
END CASE;

end;
```

Examples 2:

```
declare
grade varchar(3);
begin
 CASE
   WHEN grade = 'A' THEN dbms_output.put_line('Excellent');
   WHEN grade = 'B' THEN dbms_output.put_line('Very Good');
   WHEN grade = 'C' THEN dbms_output.put_line('Good');
   WHEN grade = 'D' THEN dbms_output.put_line('Fair');
   WHEN grade = 'F' THEN dbms_output.put_line('Poor');
   ELSE dbms_output.put_line('No such grade');
END CASE;

end;
```

ITERATIONS IN PL/SQL:

Sequence of statements can be executed any number of times using loop construct.
It is broadly classified into:

- Simple Loop
- For Loop
- While Loop

SIMPLE LOOP
Syntax:

```
LOOP
statement1;
EXIT [WHEN Condition];
END LOOP;
```

Example:

```
Declare
a number: =0;
Begin
Loop
a := a+25;
exit when a=250;
end loop;
dbms_output.put_line(a);
end;
```

WHILE LOOP
Syntax

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

Example:

```
Declare
i number:=0;
j number:=0;
begin
While i<=100 Loop
j := j+i;
i := i+2;
end loop;
dbms_output.put_line('the value of j is' ||j);
end;
```

FOR LOOP
Syntax:

```
FOR counter LowerBound..UpperBound
LOOP
statement1;
END LOOP;
```

Example:

```
Declare
j number:=0;
Begin
For i in 1..100
Loop
j := j+i;
End loop;
dbms_output.put_line('the value of j is' ||j);
End;
```