Capgemini

PLSQL

Lesson 02: Loops and Conditional constructs

Lesson Objectives

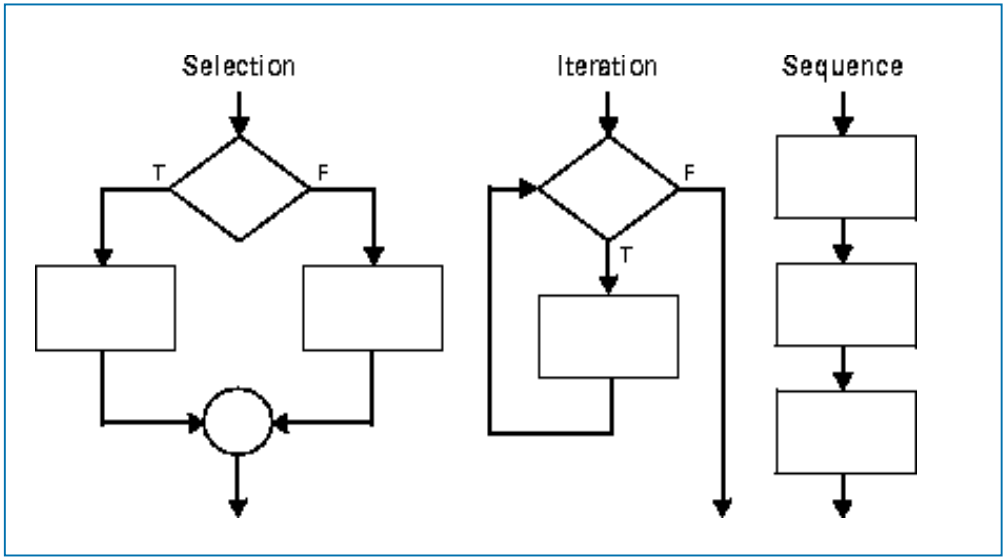
To understand the following topics:

- Loop and conditional constructs
  - If construct
  - Simple Loop
  - For
  - While

2.1: Loops and conditional constructs
Types of Programmatic Constructs

Programmatic Constructs are of the following types:

- Selection structure
- Iteration structure
- Sequence structure



2.2: If Construct
IF - Syntax

Given below is a list of Programmatic Constructs which are used in PL/SQL:

- Conditional Execution:
  - This construct is used to execute a set of statements only if a particular condition is TRUE or FALSE.
  - Syntax:

```
IF Condition_Expr
THEN
      PL/SQL_Statements
END IF;
```

2.2: If Construct
IF Construct - Example

For example:

```
IF v_staffno = 100003
THEN
    UPDATE staff_master
    SET staff_sal = staff_sal + 100
    WHERE staff_code = 100003 ;
END IF;
```

2.2: If Construct
IF Construct - Example (Contd…)

To take alternate action if condition is FALSE, use the following syntax:

```
IF Condition_Expr THEN
     PL/SQL_Statements_1 ;
ELSE
     PL/SQL_Statements_2 ;
END IF;
```

2.2: If Construct
IF Construct - Example (Contd…)

To check for multiple conditions, use the following syntax.

```
IF Condition_Expr_1
    THEN
            PL/SQL_Statements_1 ;
    ELSIF Condition_Expr_2
    THEN
            PL/SQL_Statements_2 ;
    ELSIF Condition_Expr_3
    THEN
            PL/SQL_Statements_3 ;
            ELSE
            PL/SQL_Statements_n ;
    END IF;
```

Note: Conditions for NULL are checked through IS NULL and IS NOT NULL predicates.

2.2: Loop
Simple Loop - Syntax

Looping

- A LOOP is used to execute a set of statements more than once.
- Syntax:

```
LOOP
    PL/SQL_Statements;
END LOOP ;
```

2.2: Loop
Simple Loop (Contd…)

For example:

```
DECLARE
     v_counter  number := 50 ;
BEGIN
LOOP
     INSERT INTO department_master
                         VALUES(v_counter,'new dept');
               v_counter := v_counter + 10 ;
END LOOP;
     COMMIT ;
END ;
/
```

2.2: Loop
Simple Loop – EXIT statement

EXIT

- Exit path is provided by using EXIT or EXIT WHEN commands.
- EXIT is an unconditional exit. Control is transferred to the statement following END LOOP, when the execution flow reaches the EXIT statement.

2.2: Loop
Simple Loop – EXIT statement (Contd…)

Syntax:

```
BEGIN
      .....
      .....
      LOOP                                        IF <Condition> THEN

                  . . . . .
                  EXIT ;          -- Exits loop immediately
            END IF ;
      END LOOP;
      LOOP
            …….
            …….
            EXIT WHEN <condition>
      END LOOP;

      . . . . .                   -- Control resumes here
      COMMIT ;
      END ;
```

2.2: Loop
Simple Loop – EXIT statement (Contd…)

For example:

```
DECLARE
      v_counter number := 50 ;
BEGIN
      LOOP
          INSERT INTO department_master
           VALUES(v_counter,'NEWDEPT');
      DELETE  FROM emp WHERE deptno = v_counter;
                  v_counter := v_counter + 10 ;
                                                  EXIT WHEN
v_counter >100 ;
      END LOOP;
      COMMIT ;
END ;
```

Note: As long as v_counter has a value less than or equal to 100, the loop continues.

2.3: For Loop
For - Syntax

FOR Loop:

Syntax:

```
FOR Variable IN [REVERSE] Lower_Bound..Upper_Bound
LOOP
      PL/SQL_Statements
END LOOP ;
```

2.3: While Loop
While Loop - Syntax

WHILE Loop

The WHILE loop is used as shown below.

- Syntax:

```
WHILE Condition
LOOP
      PL/SQL  Statements;
END LOOP;
```

EXIT OR EXIT WHEN can be used inside the WHILE loop to prematurely exit the loop.

2.4: Labeling Loops
Labeling of Loops

Labeling of Loops:

- The label can be used with the EXIT statement to exit out of a particular loop.

```
BEGIN
    <<Outer_Loop>>
    LOOP
            PL/SQL
            << Inner_Loop>>
            LOOP
                    PL/SQL  Statements ;
                    EXIT Outer_Loop WHEN <Condition
Met>
            END LOOP  Inner_Loop
    END LOOP  Outer_Loop
END ;
```

SUMMARY

In this lesson, you have learnt:

- Different programmatic constructs in PL/SQL are
- Selection structure,
- Iteration structure,
- Sequence structure

Review Question

Question 1: While using FOR loop, Upper Bound and Lower Bound must be integers.

- True / False

Question 2: _____ is used to exit out of loop.