

MES College of Engineering Pune-01
Department of Computer Engineering

Name of Student:	Class:
Semester/Year:	Roll No:
Date of Performance:	Date of Submission:
Examined By:	Experiment No: Part A-04 & A-05

PART: A) ASSIGNMENT NO: 04, 05

AIM: **Unnamed PL/SQLcode block:** Use of Control structure and Exception handling is mandatory.

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To learn the concept of procedural language.
- To learn Control structure and Exception handling.

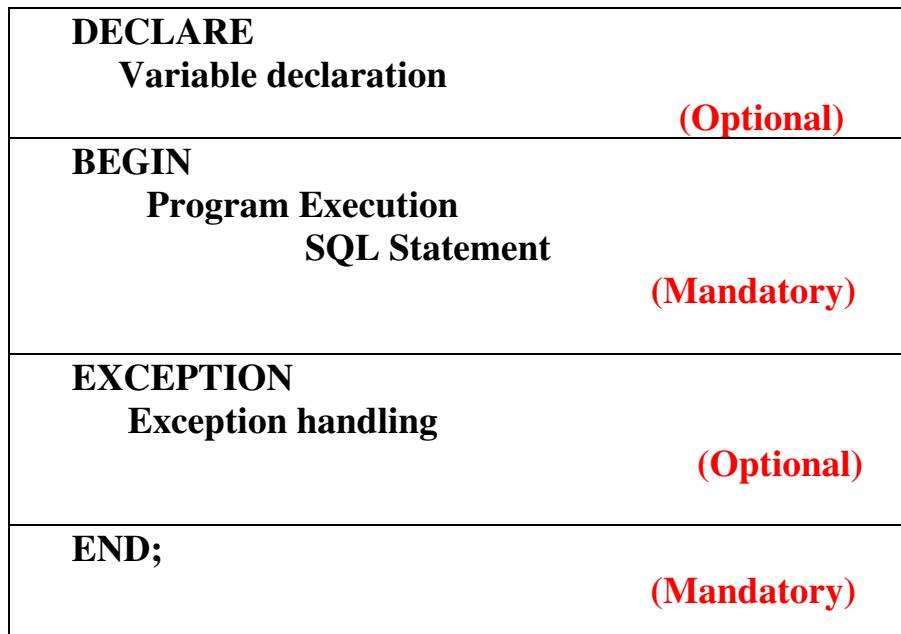
APPTRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Database: MySQL/ Oracle 11g Database.

THEORY:

PL/SQL Introduction:

- PL SQL basically stands for "Procedural Language extensions to SQL".
- It is Extension of Structured Query Language (SQL) that is used in Oracle.
- Unlike SQL, PL/SQL allows the programmer to write code in procedural format.
- It combines the data manipulation power of SQL with the processing power of procedural language to create a super powerful SQL queries.
- It allows the programmers to instruct the compiler 'what to do' through SQL and 'how to do' through its procedural way.
- PL/SQL is a completely portable, high-performance transaction-processing language.
- PL/SQL provides a built-in, interpreted and OS independent programming environment.



A. Control Statements IF:

- PL/SQL supports the programming language features like conditional statements and iterative statements.
- Its programming constructs are similar to how you use in programming languages like Java and C++.
- There are different syntaxes for the IF-THEN-ELSE statement.

Syntax: (IF-THEN statement):

```

IF condition
  THEN
    Statement: It is executed when condition is true}
  END IF;

```

Syntax: (IF-THEN-ELSE statement):

```

IF condition
  THEN
    {statements to execute when condition is TRUE...}
  ELSE
    {statements to execute when condition is FALSE...}
  END IF;

```

Syntax: (IF-THEN-ELSIF statement):

```

IF condition1
  THEN
    {statements to execute when condition1 is TRUE...}
  ELSIF condition2

```

THEN{ statements **to execute when** condition2 **is TRUE...** }**END IF;****Syntax: (IF-THEN-ELSIF-ELSE statement):**

IF condition1

THEN{ statements **to execute when** condition1 **is TRUE...** }

ELSIF condition2

THEN{ statements **to execute when** condition2 **is TRUE...** }**ELSE**{ statements **to execute when** both condition1 and condition2 are **FALSE...** }**END IF;****B. PL/SQL Loop:**

- The PL/SQL loops are used to repeat the execution of one or more statements for specified number of times.
- These are also known as iterative control statements.
- Syntax for a basic loop:**

LOOP

Sequence of statements;

END LOOP;**• Types of PL/SQL Loops****1. Basic Loop / Exit Loop**

LOOP

statements;

EXIT;

{ or EXIT WHEN condition; }

END LOOP;**2. While Loop**

WHILE <condition>

LOOP statements;

END LOOP;**Important steps to follow when executing a while loop:**

- ✓ Initialise a variable before the loop body.
- ✓ Increment the variable in the loop.

- ✓ EXIT WHEN statement and EXIT statements can be used in while loops but it's not done oftenly.

3. For Loop

FOR counter **IN** initial_value .. final_value **LOOP**

 LOOP statements;

END LOOP;

- ✓ initial_value : Start integer value
- ✓ final_value : End integer value

Important steps to follow when executing a while loop:

- ✓ The counter variable is implicitly declared in the declaration section, so it's not necessary to declare it explicitly.
- ✓ The counter variable is incremented by 1 and does not need to be incremented explicitly.
- ✓ EXIT WHEN statement and EXIT statements can be used in FOR loops but it's not done oftenly.

4. GOTO Statement

BEGIN

...

 GOTO insert_row;

...

 <<insert_row>>

 INSERT INTO emp VALUES ...

END;

C. Exception Handling

- An exception is an error condition during a program execution. PL/SQL supports programmers to catch such conditions using EXCEPTION block in the program and an appropriate action is taken against the error condition. There are two types of exceptions.

1. System-defined exceptions

2. User-defined exceptions

DECLARE

 <declarations section>

 BEGIN

 <executable command(s)>

```

    EXCEPTION
        <exception handling goes here >
        WHEN exception1 THEN
            exception1-handling-statements
        WHEN exception2 THEN
            exception2-handling-statements
        WHEN exception3 THEN
            exception3-handling-statements
        WHEN others THEN
            exception3-handling-statements
    END;

```

D. Raising Exceptions

- Exceptions are raised by the database server automatically whenever there is any internal database error, but exceptions can be raised explicitly by the programmer by using the command RAISE. Following is the simple syntax for raising an exception:

```

DECLARE
    exception_name EXCEPTION;
BEGIN
    IF condition THEN
        RAISE exception_name;
    END IF;
EXCEPTION
    WHEN exception_name THEN statement;
END;

```

IMPLEMENTATION:

Consider Tables:

1. Borrower (Roll_no, Name, Date of Issue, Name of Book, Status)
2. Fine (Roll_no, Date, Amt)

- Accept Roll_no and Name of Book from user.
- Check the number of days (from date of issue).
- If days are between 15 to 30 then fine amount will be Rs 5 per day.
- If no.of days > 30, per day fine will be Rs 50 per day & for days less than 30, Rs 5 per day.
- After submitting the book, status will change from I to R.

- If condition of fine is true, then details will be stored into fine table.
- Also handles the exception by named exception handler or user define exception handler.

OR

Write a **PL/SQL code block** to calculate the area of a circle for a value of radius varying from 5 to 9. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns, radius and area.

Note: Instructor will frame the problem statement for writing PL/SQL block in line with above statement.

CONCLUSION:

QUESTIONS:

1. What are the advantages of PLSQL over SQL?
2. List Different Pre-defined Exceptions.
3. Explain User-defined exceptions.