CONTENTS

Duration: 35 Hours

1 PL/SQL Concepts.
- 1.1 PL/SQL Client Server Architecture
  - 1.1.1 Use PL/SQL Anonymous Blocks
  - 1.1.2 Understand How PL/SQL gets executed.
- 1.2 PL/SQL in SQL*Plus
  - 1.2.1 Use Substitution Variables
  - 1.2.2 Use DBMS_OUTPUT.PUT_LINE statement

2 General Programming Language Fundamentals
- 2.1 PL/SQL Programming Fundamentals
  - 2.1.1 Make Use of PL/SQL Components
  - 2.1.2 Male Use of PL/SQL Variables
  - 2.1.3 Handle PL/SQL Reserved Words
  - 2.1.4 Make Use of Identifiers in PL/SQL
  - 2.1.5 Make Use of Anchored Data Types
  - 2.1.6 Declare and Initialize Variables
  - 2.1.7 Understand the Scope of a Block, Nested Blocks, and Labels

3 SQL in PL/SQL
- 3.1 Making Use of DML in PL/SQL
  - 3.1.1 Use the SELECT INTO syntax for variable initialization
  - 3.1.2 Use DML in PL/SQL Block
  - 3.1.3 Make use of sequence in a PL/SQL Block

4 Conditional Control: IF Statements
- 4.1 IF statement
  - 4.1.1 Use IF-THEN statements
  - 4.1.2 Use IF-THEN-ELSE statement
- 4.2 ELSIF Statement
- 4.3 Nested IF Statement

5 Conditional Control: CASE Statements
- 5.1 CASE statement
  - 5.1.1 Use the CASE statement
  - 5.1.2 Use Searched CASE statement
- 5.2 CASE Expression

6 Iterative Control: Part 1
- 6.1 Simple Loops
  - 6.1.1 Use Simple Loops with EXIT condition
  - 6.1.2 Use Simple Loops with EXIT WHEN condition
- 6.2 WHILE Loop
- 6.3 Numeric FOR Loops
  - 6.3.1 Use the Numeric FOR loops with IN option
  - 6.3.2 Use the Numeric FOR loops with REVERSE Option

7 Iterative Control: Part 2

- 7.1 The Continue Statement
  - 7.1.1 Use the CONTINUE statement
  - 7.1.2 Use the CONTINUE WHEN condition
  - 7.1.3 Nested Loops

8 Error Handling and Built-In Exceptions
- 8.1 Handling Errors
  - 8.1.1 Understand the importance of Error Handling
- 8.2 Built-in Exception

9 Exceptions
- 9.1 Exception Scope
- 9.2 User defined Exception
- 9.3 Exception Propagation
  - 9.3.1 Understand How exception Propagate
  - 9.3.2 Re-raise Exception.

10 Exceptions: Advanced Concepts
- 10.1 RAISE_APPLICATION_ERROR
- 10.2 EXCEPTION_INIT Pragma
- 10.3 SQLCODE and SQLERRM

11 Introduction to Cursors
- 11.1 Cursor Manipulation
  - 11.1.1 Make Use of Record Types
  - 11.1.2 Process an Explicit Cursor
  - 11.1.3 Make Use of Cursor Attributes
- 11.2 Using Cursor FOR loops and Nested Cursors
  - 11.2.1 Use a Cursor FOR loop
  - 11.2.2 Process Nested Cursors

12 Advanced Cursors
- 12.1 Using Parameters with Cursors.
- 12.2 FOR UPDATE and WHERE CURRENT cursors

13 Triggers
- 13.1 What are Triggers
  - 13.1.1 Understand what a Trigger is
  - 13.1.2 Use BEFORE and AFTER triggers
- 13.2 Type of Triggers
  - 13.2.1 Use Row and Statement Triggers
  - 13.2.2 Use INSTEAD of Triggers

14 Compound Triggers
- 14.1 Mutating Table Issue
- 14.2 Compound Trigger

15 Collections.
- 15.1 PL/SQL Tables
  - 15.1.1 Use Associative Array
  - 15.1.2 Use Nested Tables
- 15.2 Varray

- 16 Records
  - 16.1 Record Types
    - 16.1.1 Use Table-Based and Cursor-Based records
    - 16.1.2 Use User-Defined Records
  - 16.2 Nested Records
  - 16.3 Collection Records
- 17 Native Dynamic SQL
  - 17.1 EXECUTE IMMEDIATE Statement
  - 17.2 OPEN-FOR, FETCH and CLOSE statements

- 18 Procedures
  - 18.1 Creating Procedures
    - 18.1.1 Create Procedures
    - 18.1.2 Query the Data Dictionary for information on Procedures
  - 18.2 Passing parameters into and out of Procedures

- 19 Functions
  - 19.1 Creating and Using Functions
    - 19.1.1 Create Stored Functions
    - 19.1.2 Make use of Functions
    - 19.1.3 Invoke Functions in SQL statements
- 20 Packages
  - 20.1 The Benefits of Using Packages
    - 20.1.1 Create Package Specification
    - 20.1.2 Create Package Bodies
    - 20.1.3 Call Stored Packages
    - 20.1.4 Create Package Variables and Cursors