

# Oracle Database: Programming with PL/SQL

หลักสูตร Introduction to PL/SQL นี้เป็นหลักสูตรพื้นฐานหลักสูตรแรกสำหรับผู้ที่ต้องการเริ่มต้นการเขียนโปรแกรมกับ ORACLE Database โดยหลักสูตรจะสอนการใช้ภาษา PL/SQL เพื่อนำไปประยุกต์ใช้กับภาษา SQL ในการจัดการกับ ORACLE Database เบื้องต้น

ภาษา PL/SQL เป็นภาษา Programming ของ ORACLE โดยจะกล่าวถึง โครงสร้างและการเขียนโปรแกรมโดยใช้ภาษา PL/SQL เบื้องต้น เช่น การใช้ Exceptions, Cursors, การสร้าง Stored Procedures, Function, Trigger และสามารถที่จะนำพัฒนา Application หลากๆ อย่างโดยใช้ tools ของ ORACLE เอง เช่น Oracle Form และ Oracle Report etc. นอกจากนี้ ผู้เรียนสามารถนำความรู้จากคอร์สนี้ไปพัฒนา ปรับปรุง แก้ไข ระบบ ERP ของ ORACLE ได้เช่นกัน

นอกจากทฤษฎีแล้ว ผู้เข้าอบรมจะได้ฝึกทำแบบฝึกหัดแต่ละบทตามโจทย์ที่มาจากตัวอย่างการใช้งานจริง ทำให้ผู้เข้าอบรมสามารถนำความรู้และประสบการณ์ที่ได้ไปประยุกต์ใช้กับงานจริงได้

## หลักสูตรนี้เหมาะสำหรับ

- PL/SQL Developer
- Forms Developer
- Application Developer
- Database Administrator
- Technical Consultant
- Portal Developer
- System Analyst

หลักสูตร Oracle Database: Programming with PL/SQL มีระยะเวลาอบรม 4 วัน หัวข้อหลักสูตรโดยละเอียด มีดังนี้

### **Introduction and Declare PL/SQL Identifiers**

- Overview of PL/SQL
- Identify the benefits of PL/SQL Subprograms
- Overview of the types of PL/SQL blocks
- Create a Simple Anonymous Block
- How to generate output from a PL/SQL Block?
- List the different Types of Identifiers in a PL/SQL subprogram
- Usage of the Declarative Section to Define Identifiers
- Use variables to store data
- Identify Scalar Data Types
- The %TYPE Attribute
- What are Bind Variables?
- Sequences in PL/SQL Expressions

### **Write Executable Statements**

- Describe Basic PL/SQL Block Syntax Guidelines
- Learn to Comment the Code
- Deployment of SQL Functions in PL/SQL
- How to convert Data Types?
- Describe Nested Blocks
- Identify the Operators in PL/SQL

### **Interaction with the Oracle Server**

- Invoke SELECT Statements in PL/SQL
- Retrieve Data in PL/SQL
- SQL Cursor concept
- Avoid Errors by using Naming Conventions when using Retrieval and DML Statements
- Data Manipulation in the Server using PL/SQL
- Understand the SQL Cursor concept
- Use SQL Cursor Attributes to Obtain Feedback on DML
- Save and Discard Transactions

### **Control Structures**

- Conditional processing using IF Statements
- Conditional processing using CASE Statements
- Describe simple Loop Statement
- Describe While Loop Statement
- Describe For Loop Statement
- Use the Continue Statement

### **Composite Data Types**

- Use PL/SQL Records
- The %ROWTYPE Attribute
- Insert and Update with PL/SQL Records
- INDEX BY Tables
- Examine INDEX BY Table Methods
- Use INDEX BY Table of Records

### **Explicit Cursors**

- What are Explicit Cursors?
- Declare the Cursor
- Open the Cursor
- Fetch data from the Cursor
- Close the Cursor
- Cursor FOR loop
- The %NOTFOUND and %ROWCOUNT Attributes
- Describe the FOR UPDATE Clause and WHERE CURRENT Clause

### **Exception Handling**

- Understand Exceptions
- Handle Exceptions with PL/SQL
- Trap Predefined Oracle Server Errors
- Trap Non-Predefined Oracle Server Errors
- Trap User-Defined Exceptions
- Propagate Exceptions
- RAISE\_APPLICATION\_ERROR Procedure

### **Creating Stored Procedures**

- Describe and create a procedure
- Create procedures with parameters
- Differentiate between formal and actual parameters
- Use different parameter-passing modes
- Invoke a procedure
- Handle exceptions in procedures
- Remove a procedure

### **Creating Stored Functions**

- Describe the uses of functions
- Create stored functions
- Invoke a function
- Remove a function
- Differentiate between a procedure and a function

### **Creating Packages**

- Describe packages and list their components
- Create a package to group together related variables, cursors, constants, exceptions, procedures, and functions
- Designate a package construct as either public or private
- Invoke a package construct
- Describe the use of a bodiless package
- Overload package procedures and functions
- Use forward declarations
- Create an initialization block in a package body
- Manage persistent package data states for the life of a session
- Use PL/SQL tables and records in packages
- Wrap source code stored in the data dictionary so that it is not readable

### **Using Oracle-Supplied Packages in Application Development**

- Describe how the DBMS\_OUTPUT package works
- Use UTL\_FILE to direct output to operating system files
- Use the HTP package to generate a simple Web page
- Describe the main features of UTL\_MAIL
- Call the DBMS\_SCHEDULER package to schedule PL/SQL code for execution

### **Creating Triggers**

- Describe the different types of triggers
- Describe database triggers and their uses
- Create database triggers
- Describe database trigger-firing rules
- Remove database triggers

### **Applications for Triggers**

- Create additional database triggers
- Explain the rules governing triggers
- Implement triggers

### **Dynamic SQL and Metadata**

- Describe the execution flow of SQL statements
- Build and execute SQL statements dynamically using Native Dynamic SQL (that is, with EXECUTE IMMEDIATE statements)
- Compare Native Dynamic SQL with the DBMS\_SQL package approach
- Use the DBMS\_METADATA package to obtain metadata from the data dictionary as XML or creation DDL that can be used to re-create the objects