Oracle Database 19c/21c/23ai PL/SQL Program Units

Course information

Days : 3
Total lessons : 12
Suggested Prerequisites :
- Oracle Database 11g: PL/SQL Fundamentals
- Previous programming experience

Training includes :
- Experienced trainer(s)
- Pre-test and Post-test
- Practices and solutions

Public price :
- 15,000 baht(THB) : 1 person

In-house price :
- 45,000 baht(THB) : Economic Class : 1 - 5 people
- 67,000 baht(THB) : Small Class : 6 - 10 people
- 86,000 baht(THB) : Middle Class : 11 - 20 people
- 105,000 baht(THB) : Large Class : 21 - 30 people

All prices exclude VAT 7 %

Course details

Day 1

Introduction
Lesson 1 : Creating Stored Procedures
Lesson 2 : Creating Stored Functions
Lesson 3 : Creating Packages
Lesson 4 : Working with Packages

Day 2

Lesson 5 : Using Oracle-Supplied Packages in Application Development
Lesson 6 : Using Dynamic SQL
Lesson 7 : Design Considerations for PL/SQL Code
Lesson 8 : Creating Triggers

Day 3

Lesson 9 : Creating Compound, DDL, and Event Database Triggers
Lesson 10 : Using the PL/SQL Compiler
Lesson 11 : Managing PL/SQL Code
Lesson 12 : Managing Dependencies

Lesson details
Lesson 1 : Creating Stored Procedures
- Describe and create a stored 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

Lesson 2 : Creating Stored Functions
- Describe the uses of functions
- Create stored functions
- Invoke a function
- Remove a function
- Differentiate between a procedure and a function

Lesson 3 : 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

Lesson 4 : Working with Packages
- 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

Lesson 5 : 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

Lesson 6 : Using Dynamic SQL
- 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

Lesson 7 : Design Considerations for PL/SQL Code
- Use package specifications to create standard constants and exceptions
- Write and call local subprograms
- Set the AUTHID directive to control the run-time privileges of a subprogram
- Execute subprograms to perform autonomous transactions
- Use bulk binding and the RETURNING clause with DML
- Pass parameters by reference using a NOCOPY hint
- Use the PARALLEL ENABLE hint for optimization

Lesson 8 : Creating Triggers

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

Lesson 9 : Creating Compound, DDL, and Event Database Triggers

- Describe compound triggers
- Describe mutating tables
- Create triggers on DDL statements
- Create triggers on system events
- Display information about triggers

Lesson 10 : Using the PL/SQL Compiler

- Use the PL/SQL compiler initialization parameters
- Use the PL/SQL compile-time warnings

Lesson 11 : Managing PL/SQL Code

- Describe and use conditional compilation
- Hide PL/SQL source code using dynamic obfuscation and the Wrap utility

Lesson 12 : Managing Dependencies

- Track procedural dependencies
- Predict the effect of changing a database object on procedures and functions
- Manage procedural dependencies

_____