



Certified Associate

Oracle Database 11g: Program with PL/SQL 1z0-144

Oracle Database 11g: PL/SQL Fundamentals

- Introduction to PL/SQL
- Explain the need for PL/SQL
- Explain the benefits of PL/SQL
- Identify the different types of PL/SQL blocks
- Output messages in PL/SQL
- Declaring PL/SQL Variables
- Recognize valid and invalid identifiers
- List the uses of variables, declare and initialize variables, use bind variables
- List and describe various data types using the %TYPE attribute
- Writing Executable Statements
- Identify lexical units in a PL/SQL block
- Use built-in SQL functions in PL/SQL and sequences in PL/SQL expressions
- Describe when implicit conversions take place and when explicit conversions have to be dealt with
- Write nested blocks and qualify variables with labels
- Write readable code with appropriate indentation
- Interacting with the Oracle Database Server
- Create PL/SQL executable blocks using DML and transaction control statements
- Make use of the INTO clause to hold the values returned by a SQL statement
- Writing Control Structures
- Identify the uses and types of control structures (IF, CASE statements and expressions)
- Construct and identify loop statements
- Apply guidelines when using conditional control structures
- Working with Composite Data Types
- Create user-defined PL/SQL records
- Create a record with the %ROWTYPE attribute
- Create an INDEX BY table and INDEX BY table of records
- Describe the differences among records, tables, and tables of records
- Using Explicit Cursors
- Distinguish between usage of implicit and explicit cursors, use SQL cursor attributes
- Declare and control explicit cursors, use simple loops and cursor FOR loops to fetch data
- Declare and use cursors with parameters
- Lock rows with the FOR UPDATE clause and reference the current row with the WHERE CURRENT OF clause
- Handling Exceptions

- Define PL/SQL exceptions
- Recognize unhandled exceptions
- Handle different types of exceptions (pre-defined exceptions, non-predefined exceptions and user-defined exceptions)
- Propagate exceptions in nested blocks and call applications
- Creating Stored Procedures and Functions
- Differentiate between anonymous blocks and subprograms
- Create a simple procedure and invoke it from an anonymous block
- Create a simple function
- Create a simple function that accepts a parameter
- Differentiate between procedures and functions

Oracle Database 11g: Develop PL/SQL Program Units

- Creating Procedures
- Differentiate between anonymous blocks and subprograms, use a modularized and layered subprogram design, and identify the benefits of subprograms
- Create a simple procedure and invoke it from an anonymous block
- Work with procedures
- Handle exceptions in procedures, remove a procedure, and display a procedure's information
- Creating Functions
- Differentiate between a procedure and a function
- Describe the uses of functions
- Work with functions (create, invoke and remove functions)
- Creating Packages
- Identify the benefits and the components of packages
- Work with packages (create package specification and body, invoke package subprograms, remove a package and display package information)
- Working with Packages
- Overload package subprograms, use forward declarations
- Create an initialization block in a package body
- Manage persistent package data states for the life of a session and use PL/SQL tables and records in packages
- Using Oracle-Supplied Packages in Application Development
- Describe how the DBMS_OUTPUT package works
- Use UTL_FILE to direct output to operating system files
- Describe the main features of UTL_MAIL
- Using Dynamic SQL
- Describe the execution flow of SQL statements
- Use Native Dynamic SQL (NDS)
- Use the DBMS_SQL package
- Design Considerations for PL/SQL Code
- Create standard constants and exceptions
- Write and call local subprograms
- Control the run-time privileges of a subprogram
- Perform autonomous transactions

- Use NOCOPY hint, PARALLEL ENABLE hint and DETERMINISTIC clause
- Use bulk binding and the RETURNING clause with DML
- Creating Triggers
- Describe different types of triggers and their uses
- Create database triggers
- Manage triggers
- Creating Compound, DDL, and Event Database Triggers
- Create triggers on DDL statements
- Create triggers on system events
- Using the PL/SQL Compiler
- Describe the new PL/SQL compiler and features
- Use the new PL/SQL compiler initialization parameters
- Use the new PL/SQL compile time warnings
- Managing PL/SQL Code
- Describe and use conditional compilation
- Hide PL/SQL source code using dynamic obfuscation and the Wrap utility
- Managing Dependencies
- Track and manage procedural dependencies