

# Oracle PL/SQL Fundamentals (ORPLSQL)

ID ORPLSQL Preis 2.500,- € (exkl. MwSt.) Dauer 5 Tage

## Kursüberblick

This course introduces one to Oracle database programming using the PL/SQL programming language. One will learn the syntax, structure and features of the language. This training will also lay the foundation for the Oracle PL/SQL programming language, allowing one to progress from introductory topics to advanced application design and programming and finally onto writing complex high-performance applications. The focus of the second part of this training will be on the use of database-resident stored program units such as procedures, functions, packages and database triggers. New features introduced with the Oracle 12c release of the database are also explained and demonstrated.

PL/SQL may be considered as one of the basic skill sets required for any Oracle professional, nearly as important as the SQL language itself.

## Zielgruppe

The target audience is all Oracle professionals. Among the specific groups for whom this course will be helpful are:

- Application designers and database developers
- Database administrators
- Web server administrators

## Kursziele

This training begins with an explanation of the intent and usage of the PL/SQL programming language for database applications. Important reasons why one should incorporate PL/SQL modules within the application architecture right from the initial design

and planning phase are presented. Next one learns how to begin building executable PL/SQL program units. One learns about each of the major segments of a working program and how these interact with each other during program execution, including the important error or exception handling capabilities of the language.

In a production environment most PL/SQL program units should be packaged, and these advanced database programming capabilities along with the benefits of using these are discussed. It concludes with extensive demonstrations on how a particular type of database-resident program unit known as a database trigger can be used as part of an advanced database application design.

## Kursinhalt

- Selection & Setup Of The Database Interface
- About BIND & SUBSTITUTION Variables
- Choosing A Database Programming Language
- PL/SQL Language Fundamentals
- DECLARE Section
- BEGIN Section
- Exception Section
- Beyond The Basics: Explicit Cursors
- Beyond The Basics: Nested Blocks
- Beyond The Basics: DECLARED Subprograms
- Introducing Database-Resident Program Units
- Creating Stored Procedures & Functions
- Executing Stored Procedures & Functions
- Maintaining Stored Program Units
- Managing Dependencies
- Creating & Maintaining Packages
- Advanced Package Capabilities
- Advanced Cursor Techniques
- Using System-Supplied Packages
- Database Trigger Concepts

- Creating Database Triggers
- Maintaining Database Triggers
- Implementing System Event Triggers

### **Detaillierter Kursinhalt**

#### **Selection & Setup Of The Database Interface**

- Considering Available Tools
- Selecting The Appropriate Tool
- Oracle NET Database Connections
- Oracle PAAS Database Connections
- Setup SQL Developer
- Setup SQL\*PLUS
- Setup JDeveloper

#### **About BIND & SUBSTITUTION Variables**

- Using SQL Developer
- Using SQL\*PLUS

#### **Choosing A Database Programming Language**

- What Is Database Programming?
- PL/SQL Performance Advantages
- Integration With Other Languages

#### **PL/SQL Language Fundamentals**

- PL/SQL Program Structure
- Language Syntax Rules
- Embedding SQL
- Writing Readable Code
- Generating Database Output
- SQL\*PLUS Input Of A Program Block

#### **DECLARE Section**

- About The DECLARE Section
- DECLARE Primitive Types
- Declaration Options
- NOT NULL
- CONSTANT
- Data Dictionary Integration
- %TYPE
- Declare Simple User-Defined Types
- TYPE ... TABLE
- TYPE ... RECORD
- Extended User-Defined Types

#### **BEGIN Section**

- About The BEGIN Section

- Manipulating Program Data
- Logic Control & Branching
- GOTO
- LOOP
- IF-THEN-ELSE
- CASE

#### **Exception Section**

- About The EXCEPTION Section
- Isolating The Specific EXCEPTION
- PRAGMA EXCEPTION\_INIT
- SQLCODE & SQLERRM Example
- SQL%ROWCOUNT & SELECT...INTO

#### **Beyond The Basics: Explicit Cursors**

- About Explicit Cursors
- Extended Cursor Techniques
- FOR UPDATE OF Clause
- WHERE CURRENT OF Clause
- Using FOR...LOOP Cursors

#### **Beyond The Basics: Nested Blocks**

#### **Beyond The Basics: DECLARED Subprograms**

- Using DECLARED Subprograms
- DECLARED Procedure
- DECLARED Function

#### **Introducing Database-Resident Program Units**

- About Database-Resident Programs
- Physical Storage & Execution
- Types Of Stored Program Units
- Stored Program Unit Advantages
- Modular Design Principles

#### **Creating Stored Procedures & Functions**

- Stored Procedures & Functions
- CREATE Procedure / CREATE Function
- Creating Procedures & Functions
- RAISE\_SALARY() Procedure
- SALARY\_VALID() Function
- The Parameter Specification
- DEFAULT Clause
- SYSTEM & OBJECT Privileges
- Using The Development Tools

#### **Executing Stored Procedures & Functions**

- Calling Procedures & Functions

- Unit Testing With EXECUTE
- ANONYMOUS BLOCK Unit Testing
- Specifying A Parameter Notation
- SQL Worksheet Unit Testing
- Calling Functions From SQL

### **Maintaining Stored Program Units**

- Recompiling Programs
- Mass Recompilation Using UTL\_RECOMP()
- Dropping Procedures & Functions
- DROP Procedure / Function
- Data Dictionary Metadata
- Using USER\_OBJECTS
- Using USER\_SOURCE
- Using USER\_ERRORS
- Using USER\_OBJECT\_SIZE
- Using USER\_DEPENDENCIES

### **Managing Dependencies**

- DEPENDENCY INTERNALS
- TRACKING DEPENDENCIES
- The DEPENDENCY TRACKING Utility
- SQL Developer Dependency Info
- Dependency Strategy Checklists

### **Creating & Maintaining Packages**

- About Packages
- Creating Packages
- Maintaining Packages
- Performance Considerations

### **Advanced Package Capabilities**

- Definer & Invoker Rights
- White Lists & Accessible By
- Persistent Global Objects
- Defining Initialization Logic
- Object Orientation Support

### **Advanced Cursor Techniques**

- USING CURSOR VARIABLES
- Using SYS\_REFCURSOR
- Using CURSOR Expressions

### **Using System-Supplied Packages**

- DBMS\_OUTPUT()
- UTL\_FILE()
- FOPEN() Example

### **Database Trigger Concepts**

- About Database Triggers
- DML EVENT TRIGGER Sub-Types
- DATABASE TRIGGER Scenario
- TRIGGER Execution Mechanisms
- TRIGGERS Within SQL Worksheet

### **Creating Database Triggers**

- STATEMENT-LEVEL TRIGGERS
- Using RAISE\_APPLICATION\_ERROR()
- ROW-LEVEL TRIGGERS
- EXAMPLES OF TRIGGERS
- EMPLOYEE\_SALARY\_CHECK Example
- EMPLOYEE\_JOURNAL Example
- BUDGET\_EVENT Example
- INSTEAD OF TRIGGERS
- Triggers Within An Application

### **Maintaining Database Triggers**

- CALL Syntax
- Trigger Maintenance Tasks
- SHOW ERRORS Trigger
- DROP Trigger
- ALTER Trigger
- Multiple Triggers For A Table
- Handling Mutating Table Issues

### **Implementing System Event Triggers**

- What Are System Event Triggers?
- Defining The Scope
- Available System Events
- System Event Attributes