ODAT-3 - ORACLE AI DATABASE: PL/SQL FUNDAMENTALS

Categoria: **Database**

INFORMAZIONI SUL CORSO

Durata:	Categoria:	Qualifica Istruttore:	Dedicato a:	Produttore:
5 Giorni	Database	Oracle Certified Professional	Sviluppatore	Oracle

OBIETTIVI

PREREQUISITI

Aver seguito il corso Oracle SQL Workshop o avere conoscenze equivalenti

CONTENUTI

1: Introduction

- -Objectives 1-2
- -Lesson Agenda 1-3
- -Course Objectives 1-4
- -Course Road Map 1-5
- -Lesson Agenda 1-11
- -Human Resources (HR) Schema for This Course 1-12
- -Agenda 1-13
- -Class Account Information 1-16
- -Appendixes and Practices Used in This Course 1-17
- -Lesson Agenda 1-18
- -Oracle Database 23ai: Focus Areas 1-19
- -Oracle Database 23ai 1-20
- -Lesson Agenda 1-22
- -PL/SQL Development Environments 1-23
- -SQL Developer for VSCode 1-24
- -Oracle SQL Developer 1-25
- -Specifications of SQL Developer 1-26
- -SQL Developer Interface 1-27
- -Coding PL/SQL in SQL*Plus 1-28
- -Lesson Agenda 1-34
- -Oracle SQL and PL/SQL Documentation 1-35
- -Additional Resources 1-36
- -Summary 1-37

-Introduction Practice Overview: Getting Started 1-38

2: Introduction to PL/SQL
-Course Road Map 2-2
-Objectives 2-3
-Agenda 2-4
-Limitations of SQL 2-5
-Why PL/SQL? 2-6
-About PL/SQL 2-8
-Benefits of PL/SQL 2-9
-PL/SQL Runtime Architecture 2-11
-iv
-PL/SQL Block Structure 2-12
-Agenda 2-13
-Block Types 2-14
-Examining an Anonymous Block 2-16
-Executing an Anonymous Block 2-17
-Agenda 2-18
-Enabling Output of a PL/SQL Block 2-19
-Viewing the Output of a PL/SQL Block 2-20
-Summary 2-22

3: Declaring PL/SQL Variables
-Course Road Map 3-2
-Objectives 3-3
-Agenda 3-4
-Variables 3-5
-Variables in PL/SQL 3-6
-Requirements for Variable Names 3-8
-Using Variables in PL/SQL 3-9
-Declaring and Initializing PL/SQL Variables 3-10
-Agenda 3-11
-Declaring and Initializing PL/SQL Variables 3-12
-Initializing Variables Through a SELECT Statement 3-13
-Types of Variables 3-15
-Declaring Variables 3-16
-Guidelines for Declaring and Initializing PL/SQL Variables 3-17
-Guidelines for Declaring PL/SQL Variables 3-18
-Naming Conventions of the PL/SQL Structures Used in This Course 3-19
-Data Types for Strings 3-20
-Delimiters in String Literals 3-21
-Data Types for Numeric values 3-22
-Data Types for Date and Time values 3-23
-Data Type Conversion 3-25
-Agenda 3-27
-The %TYPE Attribute 3-28
-Declaring Variables with the %TYPE Attribute 3-29
-Declaring Boolean Variables 3-30

-LOB Data Type Variables 3-31
-Composite Data Types: Records and Collections 3-32
-Agenda 3-33
-Bind Variables 3-34
-Bind Variables: Examples 3-35
-Using AUTOPRINT with Bind Variables 3-36
-Summary 3-37

4: Writing Executable Statements
-Course Road Map 4-2
-Objectives 4-3
-Agenda 4-4
-Lexical Units in a PL/SQL Block 4-5
-PL/SQL Block Syntax and Guidelines 4-7
-Commenting Code 4-8
-SQL Functions in PL/SQL 4-9
-SQL Functions in PL/SQL: Examples 4-10
-Using Sequences in PL/SQL Blocks 4-11
-Agenda 4-15
-Nested Blocks 4-16
-Nested Blocks: Example 4-17
-Variable Scope and Visibility 4-18
-Using a Qualifier with Nested Blocks 4-20
-Challenge: Determining the Variable Scope 4-21
-Agenda 4-23
-Operators in PL/SQL 4-24
-Operators in PL/SQL: Examples 4-25
-Programming Guidelines 4-26
-Indenting Code 4-27
-Summary 4-28

5: Using SQL Statements Within a PL/SQL Block
-Course Road Map 5-2
-Objectives 5-3
-Agenda 5-4
-SQL Statements in PL/SQL 5-5
-SELECT Statements in PL/SQL 5-7
-Retrieving Data in PL/SQL: Example 5-9
-Retrieving Data in PL/SQL 5-10
-Naming Ambiguities 5-11
-Avoiding Naming Ambiguities 5-12
-Agenda 5-13
-Using PL/SQL to Manipulate Data 5-14
-Insert Data: Example 5-15
-Update Data: Example 5-16
-Delete Data: Example 5-17
-Merging Rows 5-18
-Agenda 5-20

-SQL Cursor 5-21
-SQL Cursor Attributes for Implicit Cursors 5-23
-Summary 5-25

6: Writing Control Structures
-Course Road Map 6-2
-Objectives 6-3
-PL/SQL Control Structures 6-4
-Agenda 6-5
-IF Statement 6-6
-IF-ELSIF Statements 6-7
-Simple IF Statement 6-8
-IF THEN ELSE Statement 6-9
-IF ELSIF ELSE Clause 6-10
-NULL Value in an IF Statement 6-11
-Agenda 6-12
-CASE Expressions 6-13
-Searched CASE Expressions 6-14
-CASE Statement 6-16
-Handling Nulls 6-17
-Logic Tables 6-18
-Boolean Expression or Logical Expression? 6-19
-Agenda 6-20
-Iterative Control: LOOP Statements 6-21
-Basic Loops 6-22
-Basic Loop: Example 6-23
-WHILE Loops 6-24
-WHILE Loops: Example 6-25
-FOR Loops 6-26
-FOR Loops: Example 6-28
-FOR Loop Rules 6-29
-Suggested Use of Loops 6-30
-Nested Loops and Labels 6-31
-Nested Loops and Labels: Example 6-32
-PL/SQL CONTINUE Statement 6-33
-PL/SQL CONTINUE Statement: Example 1 6-34
-PL/SQL CONTINUE Statement: Example 2 6-35
-Summary 6-36

7: Working with Composite Data Types
-Course Road Map 7-2
-Objectives 7-3
-Agenda 7-4
-Composite Data Types 7-5
-PL/SQL Records Versus Collections 7-7
-Agenda 7-8
-PL/SQL Records 7-9
-Creating a PL/SQL Record 7-10

-Creating a PL/SQL Record: Example 7-11
-PL/SQL Record Structure 7-12
-%ROWTYPE Attribute 7-14
-Creating a PL/SQL Record: Example 7-16
-Advantages of Using the %ROWTYPE Attribute 7-17
-Another %ROWTYPE Attribute: Example 7-18
-Inserting a Record by Using %ROWTYPE 7-19
-Updating a Row in a Table by Using a Record 7-20
-Agenda 7-21
-Associative Arrays (INDEX BY Tables) 7-22
-Associative Array Structure 7-24
-Steps to Create an Associative Array 7-25
-Creating and Accessing Associative Arrays 7-26
-Associative Arrays with Record Values 7-27
-Using Collection Methods 7-28
-Using Collection Methods with Associative Arrays 7-29
-Nested Tables 7-30
-Nested Tables: Syntax and Usage 7-31
-Variable-Sized Arrays (Varrays) 7-32
-VARRAYs: Syntax and Usage 7-33
-Summarizing Collections 7-34
-Summary 7-35

8: Using Explicit Cursors
-Course Road Map 8-2
-Objectives 8-3
-Agenda 8-4
-Cursors 8-5
-Implicit Cursors 8-6
-Explicit Cursor 8-7
-Controlling Explicit Cursors 8-8
-Agenda 8-9
-Declaring the Cursor 8-10
-Opening the Cursor 8-11
-Fetching Data from the Cursor 8-12
-Closing the Cursor 8-14
-Cursors and Records 8-15
-Cursor FOR Loops 8-16
-Explicit Cursor Attributes 8-18
-%ISOPEN Attribute 8-19
-%ROWCOUNT and %NOTFOUND: Example 8-20
-Cursor FOR Loops Using Subqueries 8-21
-Agenda 8-22
-Cursors with Parameters 8-23
-Agenda 8-25
-FOR UPDATE Clause 8-26
-WHERE CURRENT OF Clause 8-28
-WHERE CURRENT OF Clause: Example 8-29

-Summary 8-31

9: Handling Exceptions
-Course Road Map 9-2
-Objectives 9-3
-Agenda 9-4
-What is an exception? 9-5
-Handling an Exception: Example 9-6
-Understanding Exceptions with PL/SQL 9-7
-Handling Exceptions 9-8
-Exception Types 9-9
-Agenda 9-10
-Syntax to Trap Exceptions 9-11
-Guidelines for Trapping Exceptions 9-13
-Trapping Internally Predefined Exceptions 9-14
-Internally Defined Exception Trapping: Example 9-15
-Trapping Predefined Exceptions 9-16
-Functions for Trapping Exceptions 9-19
-Trapping User-Defined Exceptions 9-21
-RAISE Statement 9-22
-Trapping User-Defined Exceptions 9-23
-Propagating Exceptions in a Sub-Block 9-24
-The RAISE_APPLICATION_ERROR Procedure 9-25
-Summary 9-28

10: Introducing Stored Procedures and Functions
-Course Road Map 10-2
-Objectives 10-3
-Agenda 10-4
-What are PL/SQL Subprograms? 10-5
-Differences Between Anonymous Blocks and Subprograms 10-7
-Agenda 10-8
-Procedure: Syntax 10-9
-Creating a Procedure 10-10
-Invoking a Procedure 10-12
-Agenda 10-13
-Function: Syntax 10-14
-Creating a Function 10-15
-Invoking a Function 10-16
-Passing a Parameter to the Function 10-17
-Invoking the Function with a Parameter 10-18
-Summary 10-19

11: Creating Procedures
-Course Road Map 11-2
-Objectives 11-3
-Agenda 11-4
-Modularized Program Design 11-5

-Modularizing Code with PL/SQL 11-6
-Benefits of Modularization 11-7
-What are PL/SQL Subprograms? 11-8
-Agenda 11-9
-Procedures 11-10
-What are Procedures? 11-11
-Creating Procedures: Overview 11-12
-Creating Procedures 11-13
-Compiling Procedures 11-16
-Calling Procedures 11-17
-Calling Procedures Using SQL Developer for VS Code 11-19
-Procedures 11-20
-What are Parameters and Parameter Modes? 11-21
-Formal and Actual Parameters 11-22
-Procedural Parameter Modes 11-23
-Comparing the Parameter Modes 11-24
-Using the IN Parameter Mode: Example 11-25
-Using the OUT Parameter Mode: Example 11-26
-Using the IN OUT Parameter Mode: Example 11-27
-Passing Parameters to Procedures 11-28
-Passing Actual Parameters: Creating the raise_sal Procedure 11-29
-Passing Actual Parameters: Examples 11-30
-Using the DEFAULT Option for the Parameters 11-31
-Agenda 11-32
-Handled Exceptions 11-33
-Handled Exceptions: Example 11-34
-Exceptions Not Handled 11-35
-Exceptions Not Handled: Example 11-36
-Removing Procedures: Using the DROP SQL Statement or SQL Developer 11-37
-Viewing Procedure Information Using the Data Dictionary Views 11-38
-Viewing Procedure Information Using SQL Developer 11-39
-Summary 11-40

12: Creating Functions
-Course Road Map 12-2
-Objectives 12-3
-Agenda 12-4
-Functions 12-5
-Creating Functions syntax 12-6
-Tax Calculation 12-8
-The Difference Between Procedures and Functions 12-9
-Creating Functions: Overview 12-10
-Invoking a Stored Function: Example 12-11
-Using Different Methods for Executing Functions 12-12
-Agenda 12-15
-Using a Function in a SQL Expression: Example 12-16
-Calling User-Defined Functions in SQL Statements 12-17
-Restrictions When Calling Functions from SQL Expressions 12-18

-Side Effects of Function Execution 12-19
-Controlling Side Effects 12-20
-Guidelines to Control Side Effects 12-21
-Agenda 12-22
-Passing Parameters to Functions 12-23
-Named and Mixed Notation from SQL: Example 12-24
-Viewing Functions Using Data Dictionary Views 12-25
-Viewing Function Information Using SQL Developer for VS Code 12-26
-Agenda 12-27
-Removing Functions: Using the DROP SQL Statement or SQL Developer 12-28
-Summary 12-29

13: Debugging Subprograms
-Course Road Map 13-2
-Objectives 13-3
-Agenda 13-4
-Prerequisites for Debugging PL/SQL Subprograms 13-5
-Agenda 13-9
-Debugging a Subprogram: Overview 13-10
-Agenda 13-12
-Debugging: Log Tab Toolbar 13-13
-Tracking Data and Execution 13-15
-Agenda 13-16
-Debugging a Procedure: Example 13-17
-Setting Breakpoints and Compiling emp_list for Debug Mode 13-19
-Compiling the get_location Function for Debug Mode 13-20
-Debugging emp_list and Entering Values for the PMAXROWS Parameter 13-21
-Debugging emp_list: Step Into the Code (F7) 13-22
-Viewing the Data 13-23
-Modifying the Variables While Debugging the Code 13-24
-Debugging emp_list: Step Over Versus Step Into 13-25
-Debugging emp_list: Step Out of the Code (Shift + F7) 13-26
-Debugging emp_list: Step to End of Method 13-27
-Debugging a Subprogram Remotely 13-28
-Summary 13-29
-Practice 13 Overview: Introduction to the SQL Developer Debugger 13-30

14: Creating Packages
-Course Road Map 14-2
-Objectives 14-3
-Agenda 14-4
-PUT_LINE 14-5
-What Is a Package? 14-6
-Advantages of Packages 14-7
-How Do You Create PL/SQL Packages? 14-8
-Components of a PL/SQL Package 14-9
-Application Program Interface 14-10
-Agenda 14-11

-Creating the Package Specification: Using the CREATE PACKAGE Statement 14-12
-Example of a Package Specification: comm_pkg 14-15
-Creating the Package Body 14-16
-Example of a Package Body: comm_pkg 14-17
-Invoking the Package Subprograms: Examples 14-18
-Invoking Package Subprograms: Using SQL Developer for VS Code 14-19
-Creating and Using Bodiless Packages 14-20
-Viewing Packages by Using the Data Dictionary 14-21
-Viewing Packages by Using SQL Developer for VS Code 14-22
-Removing Packages 14-23
-Removing Package Bodies 14-24
-Guidelines for Writing Packages 14-25
-Summary 14-26

15: Working with Packages
-Course Road Map 15-2
-Objectives 15-3
-Agenda 15-4
-Why Overload Subprograms? 15-5
-Overloading Subprograms in PL/SQL 15-6
-Overloading Procedures Example: Creating the Package Specification 15-8
-Overloading Procedures Example: Creating the Package Body 15-9
-Restrictions on Overloading 15-10
-STANDARD Package 15-11
-Overloading and the STANDARD Package 15-12
-Agenda 15-13
-Package Instantiation and Initialization 15-14
-Initializing Packages in the Package Body 15-15
-Using User-Defined Package Functions in SQL 15-16
-User-Defined Package Function in SQL: Example 15-17
-Agenda 15-18
-Package State 15-19
-Serially Reusable Packages 15-20
-Memory Architecture 15-21
-Serially Reusable Packages 15-23
-Persistent State of Packages 15-24
-Persistent State of Package Variables: Example 15-25
-Persistent State of a Package Cursor: Example 15-26
-Executing the CURS_PKG Package 15-28
-Summary 15-29

16: Using Oracle-Supplied Packages in Application Development
-Course Road Map 16-2
-Objectives 16-3
-Agenda 16-4
-Using Oracle-Supplied Packages 16-5
-Oracle-Supplied Packages: Examples 16-6
-Agenda 16-7

-How the DBMS_OUTPUT Package Works 16-8
-Using the UTL_FILE Package 16-9
-UTL_FILE Procedures and Functions 16-11
-File Processing Using the UTL_FILE Package 16-12
-Using the Available Declared Exceptions in the UTL_FILE Package 16-13
-FOPEN and IS_OPEN Functions: Example 16-14
-Using UTL_FILE: Example 16-17
-What Is the UTL_MAIL Package? 16-19
-Setting Up and Using UTL_MAIL 16-20
-UTL_MAIL Subprograms: Summary 16-21
-Installing and Using UTL_MAIL 16-22
-SEND Procedure Syntax 16-23
-SEND_ATTACH_RAW Procedure 16-24
-Sending Email with a Binary Attachment: Example 16-25
-SEND_ATTACH_VARCHAR2 Procedure 16-27
-Sending Email with a Text Attachment: Example 16-28
-Summary 16-33

17: Using Dynamic SQL
-Course Road Map 17-2
-Objectives 17-3
-Agenda 17-4
-What is Dynamic SQL? 17-5
-When do you use Dynamic SQL? 17-6
-Using Dynamic SQL 17-7
-Execution Flow of SQL Statements 17-8
-Dynamic SQL Implementation 17-9
-Agenda 17-10
-Native Dynamic SQL (NDS) 17-11
-Using the EXECUTE IMMEDIATE Statement 17-12
-Dynamic SQL with a DDL Statement: Examples 17-13
-Dynamic SQL with DML Statements 17-14
-Dynamic SQL with a Single-Row Query: Example 17-15
-Executing a PL/SQL Anonymous Block Dynamically 17-16
-BULK COLLECT INTO Clause 17-17
-OPEN FOR Clause 17-18
-Using BULK COLLECT and OPEN FOR Clause 17-19
-Summarizing Methods for Using Dynamic SQL 17-20
-Agenda 17-22
-DBMS_SQL Package: Overview 17-23
-Using the DBMS_SQL Package 17-25
-Using the DBMS_SQL Package Subprograms 17-26
-Using DBMS_SQL with a DML Statement: Deleting Rows 17-27
-Using DBMS_SQL with a Parameterized DML Statement 17-28
-Summary 17-29

18: Creating Triggers
-Course Road Map 18-2

-Objectives 18-3
-Agenda 18-4
-What are triggers? 18-6
-Defining Triggers 18-7
-Why use triggers? 18-8
-Trigger Event Types 18-9
-Available Trigger Types 18-10
-Trigger Event Types and Body 18-11
-Agenda 18-12
-Creating DML Triggers by Using the CREATE TRIGGER Statement 18-13
-Specifying the Trigger Execution Time 18-15
-Creating a DML Statement Trigger: Example – SECURE_EMP 18-16
-Testing the SECURE_EMP Trigger 18-17
-Using Conditional Predicates 18-18
-Multiple Triggers of the Same Type 18-19
-CALL Statements in Triggers 18-20
-Agenda 18-21
-Statement-Level Triggers Versus Row-Level Triggers 18-22
-Creating a DML Row Trigger 18-24
-Correlation Names and Pseudorecords 18-25
-Using OLD and NEW Qualifiers 18-27
-Using OLD and NEW Qualifiers: Example 18-28
-Using the WHEN Clause to Fire a Row Trigger Based on a Condition 18-30
-Trigger-Firing Sequence: Single-Row Manipulation 18-31
-Trigger-Firing Sequence: Multirow Manipulation 18-32
-Trigger Execution Model: Summary 18-33
-Agenda 18-34
-INSTEAD OF Triggers 18-35
-Creating an INSTEAD OF Trigger: Example 18-36
-Creating an INSTEAD OF Trigger to Perform DML on Complex Views 18-37
-Agenda 18-39

19: Creating Compound, DDL, and Event Database Triggers

-Course Road Map 19-2
-Objectives 19-3
-Agenda 19-4
-What is a compound trigger? 19-5
-Working with Compound Triggers 19-6
-Why use compound triggers? 19-7
-Compound Trigger Structure 19-8
-Compound Trigger Structure for Views 19-9
-Compound Trigger Restrictions 19-10
-Agenda 19-11
-Mutating Tables 19-12
-Mutating Table: Example 19-13
-Compound Trigger to Resolve the Mutating Table Error 19-15
-Agenda 19-18
-Creating Triggers on DDL Statements 19-19

-Creating Triggers on DDL Statements: Example 19-20
-Agenda 19-21
-Creating Database Triggers 19-22
-Creating Triggers on System Events 19-23
-LOGON and LOGOFF Triggers: Example 19-24
-Agenda 19-25
-Guidelines for Designing Triggers 19-26
-Summary 19-27

20: Design Considerations for the PL/SQL Code
-Course Road Map 20-2
-Objectives 20-3
-Agenda 20-4
-Standardizing Constants and Exceptions 20-5
-Standardizing Exceptions 20-6
-Standardizing Exception Handling 20-7
-Standardizing Constants 20-8
-Local Subprograms 20-9
-Agenda 20-10
-Definer’s and Invoker’s Rights 20-11
-Specifying Invoker’s Rights: Setting AUTHID to CURRENT_USER 20-12
-Granting Privileges to Invoker’s Rights Unit 20-13
-Agenda 20-14
-Autonomous Transactions 20-15
-Autonomous Transactions: Features 20-16
-Using Autonomous Transactions: Example 20-17
-Agenda 20-19
-Using the NOCOPY Hint 20-20
-Effects of the NOCOPY Hint 20-21
-When Does the PL/SQL Compiler Ignore the NOCOPY Hint? 20-22
-Using the PARALLEL_ENABLE Hint 20-23
-Using the Cross-Session PL/SQL Function Result Cache 20-24
-Declaring and Defining a Result-Cached Function: Example 20-25
-Using the DETERMINISTIC Clause with Functions 20-26
-Using the RETURNING Clause 20-27
-Agenda 20-28
-Using Bulk Binding 20-29
-Bulk Binding: Syntax and Keywords 20-30
-Bulk Binding FORALL: Example 20-31
-Using BULK COLLECT INTO with Queries 20-34
-Using BULK COLLECT INTO with Cursors 20-35
-Using BULK COLLECT INTO with a RETURNING Clause 20-36
-Summary 20-37

21: Tuning the PL/SQL Compiler
-Tuning the PL/SQL Compiler
-Practice Course Road Map 21-2
-Objectives 21-3

-Agenda 21-4
-Optimizing PL/SQL Compiler Performance 21-5
-Initialization Parameters for PL/SQL Compilation 21-6
-Using the Initialization Parameters for PL/SQL Compilation 21-8
-Displaying the PL/SQL Initialization Parameters 21-9
-Displaying and Setting PL/SQL Initialization Parameters 21-10
-Changing PL/SQL Initialization Parameters: Example 21-11
-Agenda 21-12
-PL/SQL Compile-Time Warnings 21-13
-Compiler Warnings: Benefits 21-14
-Categories of PL/SQL Compile-Time Warning Messages 21-15
-Enabling Warning Messages 21-16
-Setting Compiler Warning Levels: Using PLSQL_WARNINGS – Examples 21-17
-Viewing the Current Setting of PLSQL_WARNINGS 21-19
-Viewing Compiler Warnings 21-20
-SQL*Plus Warning Messages: Example 21-21
-Defining PLSQL_WARNINGS for Program Units 21-22
-Agenda 21-23
-Using the DBMS_WARNINGS Package 21-24
-Using the DBMS_WARNING Package Subprograms 21-25
-DBMS_WARNING Procedures: Syntax, Parameters, and Allowed Values 21-26
-DBMS_WARNING Procedures: Example 21-27
-DBMS_WARNING Functions: Syntax, Parameters, and Allowed Values 21-28
-DBMS_WARNING Functions: Example 21-29
-Using DBMS_WARNING: Example 21-30
-Summary 21-32

22: Managing Dependencies
-Course Road Map 22-2
-Objectives 22-3
-Agenda 22-4
-What are dependencies in a schema? 22-5
-How Dependencies Work 22-6
-Dependent and Referenced Objects 22-8
-Querying Object Dependencies: Using the USER_DEPENDENCIES View 22-10
-Querying an Object’s Status 22-11
-Categorizing Dependencies 22-12
-Agenda 22-13
-Direct Dependencies 22-14
-Indirect Dependencies 22-15
-Displaying Direct and Indirect Dependencies 22-16
-Agenda 22-18
-Fine-Grained Dependency Management 22-19
-Fine-Grained Dependency Management: Example 1 22-21
-Fine-Grained Dependency Management: Example 2 22-23
-Guidelines for Reducing Invalidation 22-24
-Object Revalidation 22-25
-Agenda 22-26

-Remote Dependencies 22-27
-Managing Remote Procedure Dependencies 22-28
-Setting the REMOTE_DEPENDENCIES_MODE Parameter 22-29
-Timestamp Checking 22-30
-Signature Checking 22-35
-Agenda 22-36
-Revalidating PL/SQL Program Units 22-37
-Unsuccessful Recompilation 22-38
-Successful Recompilation 22-39
-Recompiling Procedures 22-40
-Agenda 22-41
-Packages and Dependencies: Subprogram References the Package 22-42
-Packages and Dependencies: Package Subprogram References Procedure 22-43
-Summary 22-44

INFO

Materiale didattico: Materiale didattico ufficiale Oracle in formato digitale
Costo materiale didattico: incluso nel prezzo del corso a Calendario
Natura del corso: Operativo (previsti lab su PC)