
Program with PL/SQL

Dauer: 5 Tage Kursnummer: D80186

Überblick:

Dieser Kurs zur Datenbankprogrammierung mit PL/SQL vermittelt zunächst eine Einführung zu PL/SQL und untersucht anschließend die Vorteile dieser leistungsstarken Programmiersprache. Im Rahmen praktischer Übungen machen Sie sich unter Anleitung qualifizierter Oracle-Dozenten mit der Entwicklung von Stored Procedures, Stored Functions, Packages usw. vertraut.

Lernumfang:

Codeabläufe (Schleifen, Kontrollstrukturen) mit Bedingungsanweisungen steuern Zusammengehörige Konstrukte mithilfe von PL/SQL-Packages gruppieren und verbinden Geschäftliche Herausforderungen mithilfe von Triggern bewältigen Ausgabe auf dem Bildschirm und in Dateien mit einigen der von Oracle bereitgestellten PL/SQL-Packages generieren Anonyme PL/SQL-Blöcke, -Funktionen und -Prozeduren erstellen PL/SQL-Variablen deklarieren

Vorteile für Teilnehmer

Durch die Teilnahme an diesem Kurs stellen Sie eine schnelle, zuverlässige, sichere und einfach zu verwaltende Performance sicher. Sie optimieren Datenbank-Workloads, senken die IT-Kosten und erzielen durch die Konsolidierung in Datenbankclouds eine höhere Servicequalität.

Oracle SQL Developer

Die Entwicklung der Programmeinheiten erfolgt mit Oracle SQL Developer. SQL*Plus und JDeveloper werden in diesem Kurs als optionale Tools eingeführt.

Kursumfang

Hinweis: Dieser Kompaktkurs behandelt den Lernstoff aus den beiden Kursen "Oracle Database: PL/SQL Fundamentals" und "Oracle Database: Develop PL/SQL Program Units".

Wer soll teilnehmen?:

PL/SQL-Entwickler Formsentwickler Datenbankadministratoren Technischer Berater Systemanalytiker Entwickler

Kursziel:

- | | |
|---|--|
| ■ Einführung | ■ Control side effects when calling Functions |
| ■ Kursziele | ■ View Functions Information |
| ■ Kursagenda | ■ How to debug Functions and Procedures? |
| ■ Schema Human Resources (HR) beschreiben | ■ Packages |
| ■ In diesem Kurs verfügbare PL/SQL-Entwicklungsumgebungen | ■ Listing the advantages of Packages |
| ■ SQL Developer – Einführung | ■ Describe Packages |
| ■ PL/SQL – Einführung | ■ What are the components of a Package? |
| ■ PL/SQL – Überblick | ■ Develop a Package |
| ■ Vorteile von PL/SQL-Unterprogrammen bestimmen | ■ How to enable visibility of a Package's Components? |
| ■ Typen von PL/SQL-Blöcken – Überblick | ■ Create the Package Specification and Body using the SQL CREATE Statement and SQL Developer |
| ■ Einfache anonyme Böcke erstellen | ■ Invoke the Package Constructs |
| ■ Ausgabe aus PL/SQL-Blöcken generieren | ■ View the PL/SQL Source Code using the Data Dictionary |
-

- PL/SQL-Bezeichner deklarieren
- Verschiedene Typen von Bezeichnern in PL/SQL-Unterprogrammen auflisten
- Bezeichner mithilfe des deklarativen Bereichs definieren
- Daten mithilfe von Variablen speichern
- Skalare Datentypen bestimmen
- Attribut %TYPE
- Bind-Variablen
- Sequences in PL/SQL-Ausdrücken
- Ausführbare Anweisungen erstellen
- Grundlegende Richtlinien zur Syntax von PL/SQL-Blöcken beschreiben
- Code kommentieren
- SQL-Funktionen in PL/SQL bereitstellen
- Datentypen konvertieren
- Verschachtelte Blöcke beschreiben
- Operatoren in PL/SQL bestimmen
- Mit dem Oracle-Server interagieren
- SELECT-Anweisungen in PL/SQL aufrufen
- Daten in PL/SQL abrufen
- SQL-Cursorkonzept
- Fehler bei Abfrage- und DML-Anweisungen mithilfe von Benennungskonventionen vermeiden
- Daten auf dem Server mit PL/SQL bearbeiten
- SQL-Cursorkonzept verstehen
- SQL-Cursorattribute verwenden, um Feedback zu DML zu erhalten
- Transaktionen speichern und verwerfen
- Kontrollstrukturen
- Bedingte Verarbeitung mit IF-Anweisungen
- Bedingte Verarbeitung mit CASE-Anweisungen
- Einfache Schleifenanweisungen beschreiben
- WHILE-Schleifenanweisungen beschreiben
- FOR-Schleifenanweisungen beschreiben
- CONTINUE-Anweisungen
- Zusammengesetzte Datentypen
- PL/SQL-Records
- Attribut %ROWTYPE
- Daten mit PL/SQL-Records einfügen und aktualisieren
- INDEX BY-Tabellen
- Deploying Packages
- Overloading Subprograms in PL/SQL
- Use the STANDARD Package
- Use Forward Declarations to solve Illegal Procedure Reference
- Implement Package Functions in SQL and Restrictions
- Persistent State of Packages
- Persistent State of a Package Cursor
- Control side effects of PL/SQL Subprograms
- Invoke PL/SQL Tables of Records in Packages
- Implement Oracle-Supplied Packages in Application Development
- What are Oracle-Supplied Packages?
- Examples of some of the Oracle-Supplied Packages
- How does the DBMS_OUTPUT Package work?
- Use the UTL_FILE Package to Interact with Operating System Files
- Invoke the UTL_MAIL Package
- Write UTL_MAIL Subprograms
- Dynamic SQL
- The Execution Flow of SQL
- What is Dynamic SQL?
- Declare Cursor Variables
- Dynamically Executing a PL/SQL Block
- Configure Native Dynamic SQL to Compile PL/SQL Code
- How to invoke DBMS_SQL Package?
- Implement DBMS_SQL with a Parameterized DML Statement
- Dynamic SQL Functional Completeness,
- Design Considerations for PL/SQL Code
- Standardize Constants and Exceptions
- Understand Local Subprograms
- Write Autonomous Transactions
- Implement the NOCOPY Compiler Hint
- Invoke the PARALLEL_ENABLE Hint
- The Cross-Session PL/SQL Function Result Cache
- The DETERMINISTIC Clause with Functions
- Usage of Bulk Binding to Improve Performance
- Triggers
- Describe Triggers
- Identify the Trigger Event Types and Body

- INDEX BY-Tabellenmethoden prüfen
- INDEX BY-Record-Tabellen
- Explizite Cursor
- Explizite Cursor – Definition
- Cursor deklarieren
- Cursor öffnen
- Daten aus Cursors lesen
- Cursor schließen
- Cursor-FOR-Schleifen
- Attribute %NOTFOUND und %ROWCOUNT
- Klauseln FOR UPDATE und WHERE CURRENT beschreiben
- 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
- Stored Procedures
- Create a Modularized and Layered Subprogram Design
- Modularize Development With PL/SQL Blocks
- Understand the PL/SQL Execution Environment
- List the benefits of using PL/SQL Subprograms
- List the differences between Anonymous Blocks and Subprograms
- Create, Call, and Remove Stored Procedures
- Implement Procedures Parameters and Parameters Modes
- View Procedure Information
- Stored Functions and Debugging Subprograms
- Create, Call, and Remove a Stored Function
- Identify the advantages of using Stored Functions
- Identify the steps to create a stored function
- Invoke User-Defined Functions in SQL Statements
- Restrictions when calling Functions
- Business Application Scenarios for Implementing Triggers
- Create DML Triggers using the CREATE TRIGGER Statement and SQL Developer
- Identify the Trigger Event Types, Body, and Firing (Timing)
- Differences between Statement Level Triggers and Row Level Triggers
- Create Instead of and Disabled Triggers
- How to Manage, Test and Remove Triggers?
- Creating Compound, DDL, and Event Database Triggers
- What are Compound Triggers?
- Identify the Timing-Point Sections of a Table Compound Trigger
- Understand the Compound Trigger Structure for Tables and Views
- Implement a Compound Trigger to Resolve the Mutating Table Error
- Comparison of Database Triggers to Stored Procedures
- Create Triggers on DDL Statements
- Create Database-Event and System-Events Triggers
- System Privileges Required to Manage Triggers
- PL/SQL Compiler
- What is the PL/SQL Compiler?
- Describe the Initialization Parameters for PL/SQL Compilation
- List the new PL/SQL Compile Time Warnings
- Overview of PL/SQL Compile Time Warnings for Subprograms
- List the benefits of Compiler Warnings
- List the PL/SQL Compile Time Warning Messages Categories
- Setting the Warning Messages Levels: Using SQL Developer, PLSQL_WARNINGS Initialization Parameter, and the DBMS_WARNING Package Subprograms
- View Compiler Warnings: Using SQL Developer, SQL*Plus, or the Data Dictionary Views
- Manage Dependencies
- Overview of Schema Object Dependencies
- Query Direct Object Dependencies using the USER_DEPENDENCIES View
- Query an Object's Status
- Invalidation of Dependent Objects
- Display the Direct and Indirect Dependencies
- Fine-Grained Dependency Management in Oracle Database 12c
- Understand Remote Dependencies
- Recompile a PL/SQL Program Unit

Detaillierte Kursbeschreibung:

- | | | |
|---|---|---|
| <ul style="list-style-type: none">■ Stored Procedures und Stored Functions erstellen und debuggen■ Überladene Packageunterprogramme für größere Flexibilität erstellen■ Geschäftliche Herausforderungen mithilfe von Triggern bewältigen■ Features und Syntax von PL/SQL beschreiben | <ul style="list-style-type: none">■ Anonyme PL/SQL-Blöcke entwerfen, die effizient ausgeführt werden■ PL/SQL-Packages entwerfen, in denen zusammengehörige Konstrukte gruppiert werden■ Laufzeitfehler behandeln■ Abhängigkeiten zwischen PL/SQL-Unterprogrammen verwalten | <ul style="list-style-type: none">■ PL/SQL-Programmkonstrukte verwenden sowie Codeabläufe (Schleifen, Kontrollstrukturen und explizite Cursor) mit Bedingungsanweisungen steuern■ Funktionalität einer PL/SQL-Anwendung mit Conditional Compilation ohne Entfernung von Quellcode anpassen■ Ausgabe auf dem Bildschirm, in Dateien und in E-Mails mit den von Oracle bereitgestellten PL/SQL-Packages generieren■ Dynamisches SQL für größere Flexibilität bei der Codierung erstellen |
|---|---|---|

Weitere Informationen:

Für weitere Informationen oder Buchung kontaktieren Sie uns bitte unter 01/66 55 655 3000

info@globalknowledge.at

www.globalknowledge.at

Global Knowledge Network GmbH, Gutheil-Schoder Gasse 7a, A-1101 Wien