

PL/SQL ist eine Programmiersprache für Oracle, mit der Sie die Funktionalität von Oracle Datenbanken und Applikationen optimal nutzen und erweitern können.

Ihr Nutzen

In diesem Kurs lernen die Teilnehmer fortgeschrittene Konzepte, um PL/SQL an der Schnittstelle zwischen der Datenbank und Anwendungen möglichst optimal einzusetzen.

Voraussetzungen

K1000 - Professioneller Einstieg in SQL und ORACLE ~143 oder dem entsprechende Oracle & SQL-Grundkenntnisse, Programmiererfahrung empfohlen

Preis pro Teilnehmer

EUR 1770,- exklusive der gesetzlichen MwSt.

Seminardauer

3 Tag(e)/Day(s)

Hinweise

-

Version: 11g

Seminarinhalte

- 1. Tag
 - * Überblick Entwicklungsumgebungen
 - SQL Developer vs. SQL*Plus
 - * Überlegungen zum Design
 - Vordefinierte Datentypen beschreiben
 - Subtypen erstellen
 - Verschiedene Richtlinien für den Cursor-Entwurf
 - Cursor-Variablen verwenden
 - Cursor-Variablen als Programmparameter übergeben
 - Cursor-Variablen mit statischen Cursors vergleichen
 - * Collections verwenden
 - Collections – Überblick
 - Assoziative Arrays, Nested Tables, Varrays
 - PL/SQL-Programme entwickeln, die Collections verwenden
 - Collections effektiv verwenden
 - * Weitere Schnittstellenmethoden
 - C und Java über PL/SQL aufrufen
- 2. Tag
 - * VDP mit fein granulierter Zugriffskontrolle implementieren
 - Funktionsweise der fein granulierten Zugriffskontrolle
 - Features der fein granulierten Zugriffskontrolle
 - Anwendungskontext beschreiben und erstellen
 - DBMS_RLS-Prozeduren auflisten
 - Policies implementieren
 - Dictionary Views zur fein granulierten Zugriffskontrolle
 - * Large Objects bearbeiten
 - Interne LOBs verwalten, BFILEs beschreiben
 - DIRECTORY-Objekt erstellen und damit auf BFILEs zugreifen
 - Das DBMS_LOB-Package
 - LOBs entfernen
 - Temporäres LOB mit dem DBMS_LOB-Package erstellen
 - * SecureFile-LOBs verwalten
 - Umgebung für SecureFile-LOBs aktivieren
 - Mit SecureFile-LOBs Dokumente speichern
 - BasicFile-LOBs in das SecureFile-LOB-Format konvertieren
 - Performance von SecureFile-LOBs prüfen
 - Deduplication, Komprimierung, Verschlüsselung aktivieren
 - * Performance und Tuning
 - Compiler verstehen und beeinflussen
 - PL/SQL-Code optimieren
 - Inlining innerhalb von Einheiten aktivieren
 - Speicherprobleme identifizieren und optimieren
 - * Performance mit SQL- und PL/SQL-Caching verbessern

- Ergebnis-Cache für SQL-Abfragen und PL/SQL Funktionen
- 3. Tag
 - * PL/SQL-Code analysieren
 - bereitgestellten Packages und Dictionary Views
 - Codierungsinformationen suchen
 - Identifier-Typen mit PL/Scope ermitteln
 - Das DBMS_METADATA Package
 - * PL/SQL-Code – Tracing und Profilerstellung
 - PL/SQL-Programmausführung tracen
 - Profile von PL/SQL-Anwendungen erstellen
 - Code vor SQL-Injection-Angriffen schützen
 - Angriffsflächen verringern
 - DBMS_ASSERT verwenden
 - Immunen Code entwickeln

