

Oracle® PL/SQL: programmering - 5 dagar

kurser 493

- Du får lära dig att**
- Utveckla effektiva PL/SQL-program för arbete med Oracle-databaser
 - Skapa lagrade procedurer och funktioner för maximal återanvändning och minimalt kodunderhåll
 - Konstruera moduluppbyggda applikationer med paket
 - Hämta data med front-end-applikationer
 - Bulkbinda kollektioner för att öka hastigheten vid dataflyttning
 - Anropa native dynamisk SQL för att utveckla abstrakt högnivåkod

Sammanfattning Språket Oracle PL/SQL är ett flexibelt procedurtillägg till SQL som ger ökad produktivitet, högre prestanda, bättre skalbarhet, bärbarhet och säkerhet. Kursen ger dig praktisk erfarenhet av att skriva PL/SQL-program. Du lär dig att bygga lagrade procedurer, designa och köra modulära applikationer, samt öka datarörelsens effektivitet.

Vem bör delta Programmerare och andra som arbetar med PL/SQL. Kunskaper om SQL och PL/SQL, i nivå med kurs 926, "Oracle Database 11g: omfattande introduktion", eller kurs 925, "SQL programmeringsspråk: omfattande introduktion", förutsätts.

Praktiska övningar På kursen får du erfarenhet av att skriva modulär PL/SQL-kod. Övningarna omfattar:

- Kapsla in datamanipuleringsuttryck i lagrade procedurer och paket
- Komplexa datamanipulationer med cursors
- Utnyttja EXCEPTIONs för att hantera körningsfel
- Skapa triggerfunktioner för att hantera dataintegritet och datahärledning
- Använda svaga och starka cursor-variabler för dynamisk SQL
- Denormalisera data med funktioner skapade av användaren
- Skapa globala variabler i paket utan kropp för sessionsbearbetning

Oracle® PL/SQL: programmering - 5 dagar

kurser 493

Introduktion och översikt

Grunderna i pl/SQL

- Deklarera variabler
- Förankra variabler till databasdefinitioner
- Skapa flödesstyrning

PL/SQL-funktioner i Oracle10i och 11g

- PL/Scope i Oracle 11g
- Processflöde med CASE-uttryck
- Referenser till PL/SQL-data i DML
- Förbättra prestandan med intern sammanställning
- Hantera regelbundna uttryck med Oracle 10g-funktioner
- Sammansatta triggers i Oracle 11g
- Multiset-operatörer för kollektioner

Teknik för datamanipulering

Underhålla data med dml-satser

- Använda klausulen RETURNING INTO
- Lösa problemet med hämtning från flera håll

Hämta data med hjälp av cursors

- Implikationer av explicita och implicita cursors
- En cursors attribut
- Förenkla cursor-bearbetning med FOR-slingor
- Inbäddade cursor-uttryck i SELECT-satser

Cursor-variabler

- Starka kontra svaga cursor-variabler
- Överföra cursor-variabler till andra program
- Definiera REF CURSORS i paket

Utveckla välstrukturerad och felfri kod

Felhantering med hjälp av exceptions

- Fördefinierade respektive användardefinierade EXCEPTIONS
- Spridning och räckvidd
- "Förnyade försök" med problematiska transaktioner under användning av EXCEPTION

Avlusa pl/SQL-block

- Förenkla testning och avlusning med villkorad sammanställning
- Tolka meddelanden från kompilatorn
- Tillämpa strukturerade testningstekniker

Uppnå maximal återanvändbarhet

Skriva lagrade procedurer och funktioner

- Anropa logik i servern
- Överföra indata- och utdataparametrar

- Utföra en självständig transaktion
- "Definer"-rättigheter kontra "Invoker"-rättigheter

Koda funktioner skrivna av användaren

- Anropa PL/SQL-funktioner från SQL
- Bygga tabellvärderade funktioner

Utveckla säkra triggerfunktioner

- Utnyttja :OLD och :NEW bindevariabler i triggerfunktioner på radnivå
- Implementera komplexa funktionsregler
- Undvika opålitliga triggerkonstruktioner
- Använda schema- och databastrigger

Utnyttja komplexa datatyper

Samlingstyper

- PL/SQL-tabeller, nästlade tabeller och VARRAYs
- Stegning genom täta och icke följdriktiga samlingar

Bulkbinding för högre prestanda

- Flytta data till och från PL/SQL-block
- BULK COLLECT INTO
- FORALL
- BULK cursor-attribut
- Hantering av BULK EXCEPTION

Anropa native dynamic SQL

Finess-hantera kompilatorn

- Satsen EXECUTE IMMEDIATE
- Satsen RETURNING INTO

Typer av dynamisk SQL

- Skapa SQL-satser vid körning
- Självgenererande standardkod

Tips och tekniker för paketering

- Paketstruktur: SPEC och BODY
- Undvika problem med beroenden
- Överbelasta för polymorfiska effekter
- Bedöma applikationsramarna
- Paket utan kropp för alla applikationsdefinitioner
- Deklarera och använd beständiga globala variabler