

Oracle Database 11g: PL/SQL und Datenbankprogrammierung Release 2

Duration: 5 Days

What you will learn

Dieser Kurs gibt den Teilnehmern eine Einführung in PL/SQL und unterstützt sie dabei, die Vorteile dieser leistungsstarken Programmiersprache zu verstehen. Darüber hinaus lernen die Kursteilnehmer, wie sie Blöcke von PL/SQL-Anwendungscode erstellen, die von vielen Formular-, Berichts- und Daten-Management-Anwendungen gemeinsam verwendet werden können. Außerdem erfahren sie, wie anonyme PL/SQL-Blöcke sowie Stored Procedures und Stored Functions erstellt werden. Die Teilnehmer lernen, PL/SQL-Stored Program Units zu entwickeln, auszuführen und zu verwalten, beispielsweise Prozeduren, Funktionen, Packages und Datenbank-Trigger. Im Kurs werden die Grundfunktionen vorgestellt, beispielsweise das Debuggen von Funktionen und Prozeduren mithilfe von SQL Developer Debugger. Außerdem lernen die Teilnehmer, wie sie PL/SQL-Unterprogramme und Trigger verwalten, Bezeichner deklarieren und Exceptions abfangen. Den Teilnehmern wird gezeigt, wie sie einige der von Oracle bereitgestellten Packages verwenden. Darüber hinaus lernen die Teilnehmer den Umgang mit dynamischem SQL, Überlegungen zum Design von PL/SQL-Code, die Funktionsweise des PL/SQL-Compilers und die Verwaltung von Abhängigkeiten kennen. Kurse Oracle Database 11g: PL/SQL-Grundlagen und Oracle Database 11g:

PL/SQL-Datenbankprogrammierung.

Die Teilnehmer entwickeln die Programmeinheiten mit Oracle SQL Developer. SQL*Plus und JDeveloper werden als optionale Tools eingeführt.

Dieser Kurs ist auch für Anwender von 10g geeignet. Zwischen den Features von 10g und 11g gibt es nur geringfügige Änderungen.

Codeabläufe (Schleifen, Kontrollstrukturen) mit Bedingungsanweisungen steuern

Mit einigen der von Oracle bereitgestellten PL/SQL-Packages Ausgabe auf dem Bildschirm und in Dateien generieren
PL/SQL-Packages entwerfen und verwenden, die zusammengehörige Konstrukte gruppieren und enthalten

Anonyme PL/SQL-Codeblöcke erstellen

Mit Triggern geschäftliche Herausforderungen bewältigen

PL/SQL-Variablen deklarieren

Related Training

Required Prerequisites

Oracle Database 11g: Einführung in SQL (Kombination aus Oracle Database 11g: SQL Grundlagen I Basiswissen und Oracle Database 11g: SQL Grundlagen II F

Oracle Database 11g: SQL Fundamentals I Release 2

Suggested Prerequisites

Programmiererfahrung

Course Objectives

Überladene Packageunterprogramme für größere Flexibilität erstellen

Trigger erstellen, um Geschäftsanforderungen gerecht zu werden

Stored Procedures und Stored Functions erstellen, verwenden, debuggen und beschreiben

PL/SQL-Features und -Syntax beschreiben

Anonyme PL/SQL-Blöcke entwerfen, die effizient ausgeführt werden

PL/SQL-Code für vordefinierte Datentypen, lokale Unterprogramme, zusätzliche PRAGMAs und standardisierte Konstanten und Exceptions konzipieren

PL/SQL-Packages entwerfen und verwenden, die zusammengehörige Konstrukte gruppieren und enthalten

Laufzeitfehler behandeln

Abhängigkeiten zwischen PL/SQL-Unterprogrammen verwalten

PL/SQL-Programmkonstrukte verwenden sowie Codeabläufe (Schleifen, Kontrollstrukturen und explizite Cursor) mit Bedingungsanweisungen steuern

Mithilfe von Conditional Compilation die Funktionalität in einer PL/SQL-Anwendung ohne Entfernung von Quellcode anpassen

Mit den von Oracle bereitgestellten PL/SQL-Packages Ausgabe auf dem Bildschirm, in Dateien und in E-Mails generieren

Dynamisches SQL für größere Flexibilität bei der Codierung erstellen

Course Topics

Einführung

Lernziele

Kursagenda

Schema Human Resources (HR) beschreiben

In diesem Kurs verfügbare PL/SQL-Entwicklungsumgebungen

Einführung in SQL Developer

Einführung in PL/SQL

PL/SQL – Überblick

PL/SQL-Unterprogramme – Vorteile

Typen von PL/SQL-Blöcken – Überblick

Einfache anonyme Blöcke erstellen und ausführen

Ausgaben von PL/SQL-Blöcken generieren

PL/SQL-Bezeichner deklarieren

Verschiedene Bezeichnertypen in PL/SQL-Unterprogrammen

- Bezeichner im deklarativen Bereich definieren
- Daten in Variablen speichern
- Skalare Datentypen
- Attribut %TYPE
- Bind-Variablen
- Sequences in PL/SQL-Ausdrücken

Ausführbare Anweisungen erstellen

- Grundlegende Richtlinien für die Syntax von PL/SQL-Blöcken beschreiben
- Code kommentieren
- SQL-Funktionen in PL/SQL
- Datentypen konvertieren
- Verschachtelte Blöcke
- Operatoren in PL/SQL

Mit dem Oracle-Server interagieren

- SELECT-Anweisungen in PL/SQL aufnehmen, um Daten abzurufen
- Daten in PL/SQL mit der Anweisung SELECT abrufen
- SQL-Cursorkonzept
- Fehler bei Abfrage- und DML-Anweisungen mithilfe von Benennungskonventionen vermeiden
- Daten im Server mit PL/SQL bearbeiten
- SQL-Cursorkonzept verstehen
- Durch SQL-Cursorattribute Feedback zu DML erhalten
- Transaktionen speichern und verwerfen

Kontrollstrukturen erstellen

- Bedingte Verarbeitung mit IF-Anweisungen
- Bedingte Verarbeitung mit CASE-Anweisungen
- Einfache LOOP-Anweisungen
- WHILE-Schleifen
- FOR-Schleifen
- CONTINUE-Anweisungen

Mit zusammengesetzten Datentypen arbeiten

- Mehrere Werte verschiedenen Typs mit PL/SQL-Records speichern
- Attribut %ROWTYPE
- Mit PL/SQL-Records einfügen und aktualisieren
- Mehrere Werte desselben Datentyps mit INDEX BY-Tabellen speichern
- INDEX BY-Tabellenmethoden
- INDEX BY-Tabelle mit Records

Explizite Cursor

- Explizite Cursor verstehen
- Cursor deklarieren
- Cursor öffnen
- Daten aus Cursors lesen
- Cursor schließen
- Cursor FOR-Schleifen
- Attribute %NOTFOUND und %ROWCOUNT
- Klauseln FOR UPDATE und WHERE CURRENT

Exceptions behandeln

- Exceptions verstehen
- Exceptions mit PL/SQL behandeln
- Vordefinierte Oracle-Serverfehler abfangen
- Nicht vordefinierte Oracle-Serverfehler abfangen
- Benutzerdefinierte Exceptions abfangen
- Exceptions propagieren
- Prozeduren vom Typ RAISE_APPLICATION_ERROR

Stored Procedures erstellen

- Modularisierte Unterprogramme und Unterprogramme mit Schichten erstellen
- Entwicklung mit PL/SQL-Blöcken modularisieren
- PL/SQL-Ausführungsumgebung
- PL/SQL-Unterprogramme – Vorteile
- Anonyme Blöcke und Unterprogramme – Unterschiede
- Stored Procedures mit dem Befehl CREATE und SQL Developer erstellen, aufrufen und entfernen
- Prozedurparameter und Parametermodi
- Prozedurinformationen mit Data Dictionary Views und SQL Developer anzeigen

Stored Functions und Debugging-Unterprogramme erstellen

- Stored Functions mit dem Befehl CREATE und SQL Developer erstellen, aufrufen und entfernen
- Vorteile von Stored Functions in SQL-Anweisungen angeben
- Schritte zum Erstellen von Stored Functions angeben
- Benutzerdefinierte Funktionen in SQL-Anweisungen
- Einschränkungen beim Aufrufen von Funktionen aus SQL-Anweisungen
- Seiteneffekte beim Aufruf von Funktionen aus SQL-Ausdrücken ausschalten
- Funktionsinformationen anzeigen
- Funktionen und Prozeduren debuggen

Packages erstellen

- Vorteile von Packages auflisten
- Packages beschreiben
- Komponenten eines Packages
- Packages entwickeln
- Sichtbarkeit von Packagekomponenten
- Packagespezifikation und Package-Body mit der SQL-Anweisung CREATE und SQL Developer erstellen
- Packagekonstrukte aufrufen
- PL/SQL-Quellcode mit dem Data Dictionary anzeigen

Mit Packages arbeiten

- Unterprogramme in PL/SQL überladen
- Package STANDARD
- Mit Vorwärtsdeklarationen Probleme mit ungültigen Prozedurreferenzen lösen
- Packagefunktionen in SQL – Verwendung und Einschränkungen
- Persistenter Status von Packages
- Persistenter Status eines Packageursors
- Seiteneffekte von PL/SQL-Unterprogrammen ausschalten
- PL/SQL-Record-Tabellen in Packages

Von Oracle bereitgestellte Packages zur Anwendungsentwicklung

- Von Oracle bereitgestellte Packages
- Von Oracle bereitgestellte Packages – Beispiele
- Funktionsweise des Packages DBMS_OUTPUT

Mit dem Package UTL_FILE mit Betriebssystemdateien interagieren

Package UTL_MAIL

Unterprogramme vom Typ UTL_MAIL

Dynamisches SQL

Ausführungsablauf von SQL-Anweisungen

Was ist dynamisches SQL?

Cursorvariablen deklarieren

PL/SQL-Blöcke dynamisch ausführen

PL/SQL-Code mit nativem dynamischem SQL kompilieren

Package DBMS_SQL

DBMS_SQL mit parametrisierten DML-Anweisungen

Umfassende Funktionalität von dynamischem SQL

Überlegungen zum Design von PL/SQL-Code

Konstanten und Exceptions standardisieren

Lokale Unterprogramme

Autonome Transaktionen

Compiler-Hint NOCOPY

Hint PARALLEL_ENABLE

Sessionübergreifender Ergebniscache für PL/SQL-Funktionen

Klausel DETERMINISTIC und Funktionen

Mit Bulk Binding die Performance steigern

Trigger erstellen

Mit Triggern arbeiten

Trigger-Ereignistypen und Body identifizieren

Trigger implementieren – Szenarios für Geschäftsanwendungen

DML-Trigger mit der Anweisung CREATE TRIGGER und SQL Developer erstellen

Trigger-Ereignistypen, Body und Ausführungszeitpunkt identifizieren

Trigger auf Anweisungs- bzw. Zeilenebene – Vergleich

Trigger INSTEAD OF und deaktivierte Trigger erstellen

Trigger verwalten, testen und entfernen

Komplexe, DDL- und Datenbankereignis-Trigger erstellen

Mit komplexen Triggern arbeiten

Ausführungszeitpunkt-Bereiche von komplexen Triggern für Tabellen identifizieren

Struktur komplexer Trigger für Tabellen und Views

Fehler in sich verändernden Tabellen mit komplexen Triggern beheben

Datenbank-Trigger und Stored Procedures – Vergleich

Trigger für DDL-Anweisungen erstellen

Trigger für Datenbankereignisse und Systemereignisse erstellen

Erforderliche Systemberechtigungen zur Trigger-Verwaltung

PL/SQL-Compiler

PL/SQL-Compiler

Initialisierungsparameter für PL/SQL-Kompilierung

Neue PL/SQL-Kompilierungszeitwarnungen

PL/SQL-Kompilierungszeitwarnungen für Unterprogramme – Überblick

Vorteile von Compilerwarnungen

PL/SQL-Kompilierungszeit-Warnmeldungen – Kategorien

Warnmeldungsstufen einstellen – mit SQL Developer, dem Initialisierungsparameter PLSQL_WARNINGS und den DBM:

PL/SQL-Code verwalten

Was ist Conditional Compilation, und wie funktioniert das Feature?

Auswahanweisungen

Vordefinierte und benutzerdefinierte Abfrageanweisungen

Parameter PLSQL_CCFLAGS und Abfrageanweisungen

Benutzerdefinierte Fehler mit Conditional Compilation-Fehleranweisungen auslösen

Package DBMS_DB_VERSION

Quelltext mit DBMS_PREPROCESSOR-Prozeduren ausgeben oder abrufen

Obfuscation und Wrapping von PL/SQL-Code

Abhängigkeiten verwalten

Abhängigkeiten von Schemaobjekten – Überblick

Direkte Objektabhängigkeiten abfragen – View USER_DEPENDENCIES

Objektstatus abfragen

Invalidierung abhängiger Objekte

Direkte und indirekte Abhängigkeiten anzeigen

Fein granulierte Abhängigkeitsverwaltung in Oracle Database 11g

Fernabhängigkeiten

PL/SQL-Programmeinheiten rekompilieren