
Oracle PL/SQL

Marco Skulschus
Marcus Wiederstein

Webseite zum Buch:

www.comelio-medien.com/buch-katalog/oracle/oracle_pl_sql

© Comelio Medien 2010

Alle Rechte vorbehalten. Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jeder Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für die Vervielfältigung, Übersetzung, Mikroverfilmung und die Einspeicherung und Verbreitung in elektronischen Systemen.

© Comelio GmbH

Comelio GmbH
Goethestr. 34
D-13086 Berlin

Fon: +49 (0) 30-8 14 56 22-00

Fax: +49 (0) 30-8 14 56 22-10

www.comelio-medien.com
info@comelio.com

Umschlaggestaltung, Comelio-Grafiken, Layout & Satz: Nadine Kilian
Druck und Bindung: Printdream, ul. Modra 34, 71-220 Szczecin

Printed in Poland

ISBN 978-3-939701-40-8

Inhaltsverzeichnis

1. Einführung	25
1. 1. Beispieldatenbank	25
1. 1. 1. Entitäten	25
1. 1. 2. Szenario	26
1. 1. 3. Tabellen und Beziehungen	30
1. 2. Einrichtung der Umgebung	32
1. 2. 1. Oracle Database 10g Express Edition	32
1. 2. 2. Entwicklungsumgebung	36
1. 3. Einrichtung der Beispiel-Datenbank	39
1. 3. 1. Dateien	39
1. 3. 2. Tabellen und Daten aufspielen	40
2. Einführung in PL/SQL	47
2. 1. Wozu überhaupt PL/SQL?	47
2. 1. 1. SQL- und PL/SQL-Variante eines Datenzugriffs	47
2. 1. 2. Architektur von PL/SQL-Anwendungen	50
2. 2. Die Struktur von PL/SQL	54
2. 2. 1. Schreibweise und Benennungskonventionen für Variablen	54
2. 2. 2. Blockstruktur	56
2. 2. 3. Einsatzbereiche von PL/SQL	58
2. 2. 4. Guter Programmierstil: Kommentare und Formatierungen	60
3. Grundlegende Syntaxvorstellung	69
3. 1. Basis-Syntax	69
3. 1. 1. Gültige Zeichenmenge	69
3. 1. 2. Literale	70
3. 2. Variablen	74
3. 2. 1. Deklaration	75
3. 2. 2. Verwendung von %TYPE	76
3. 2. 3. Vererbung von Datentypen	77
3. 2. 4. Gültigkeit und Sichtbarkeit	77

3. 3. Datentypen und Operatoren	81
3. 3. 1. Oracle- und PL/SQL-Datentypen	82
3. 3. 2. Subtypen – Eigene Datentypen	84
3. 3. 3. Automatische und explizite Typumwandlung	86
3. 3. 4. Operatoren	90
3. 4. Ausgabemöglichkeiten im Puffer und in Dateien	93
3. 4. 1. Puffer-Ausgaben mit DBMS_OUTPUT	93
3. 4. 2. Datei-Ausgaben mit UTL_FILE	98
3. 4. 3. Ausnahmen des UTL_FILE-Pakets	106
3. 4. 4. Textformatierungen und Ersetzungen	107
3. 5. Verzweigungen	110
3. 5. 1. IF-Verzweigung	111
3. 5. 2. IF-ELSE-Verzweigung	112
3. 5. 3. IF-ELSIF-ELSE-Verzweigung	113
3. 5. 4. CASE mit einzelner Selektor	114
3. 5. 5. Ausnahmen bei CASE	115
3. 5. 6. Selektorlose CASE-Anweisung	116
3. 5. 7. Programmuster bei Verzweigungen	117
3. 6. Schleifen	118
3. 6. 1. LOOP – Einfache Schleife	118
3. 6. 2. Schleifensteuerung mit EXIT und CONTINUE	119
3. 6. 3. Verschachtelte Schleifenkonstruktionen mit Labeln	120
3. 6. 4. WHILE-Schleife	122
3. 6. 5. Numerische FOR-Schleife	124
3. 7. Weitere Anweisungen	126
3. 7. 1. GOTO-Strukturen mit Labeln	126
3. 7. 2. NULL im Anweisungsabschnitt	134
3. 8. Datensätze und %RECORD	135
3. 9. Vorteile bei der Verwendung	136
3. 9. 1. Definition eines Datensatztyps	137
3. 9. 2. Verwendung von Datensätzen	139
3. 9. 3. Datensatz-Erzeugung mit %ROWTYPE	142
4. Transaktionssteuerung	147
4. 1. SQL in PL/SQL	147
4. 1. 1. Zulässigkeit von Anweisungen	147
4. 1. 2. Dynamisches SQL	148
4. 2. Kontextwechsel	150
4. 2. 1. Grundprinzip	151

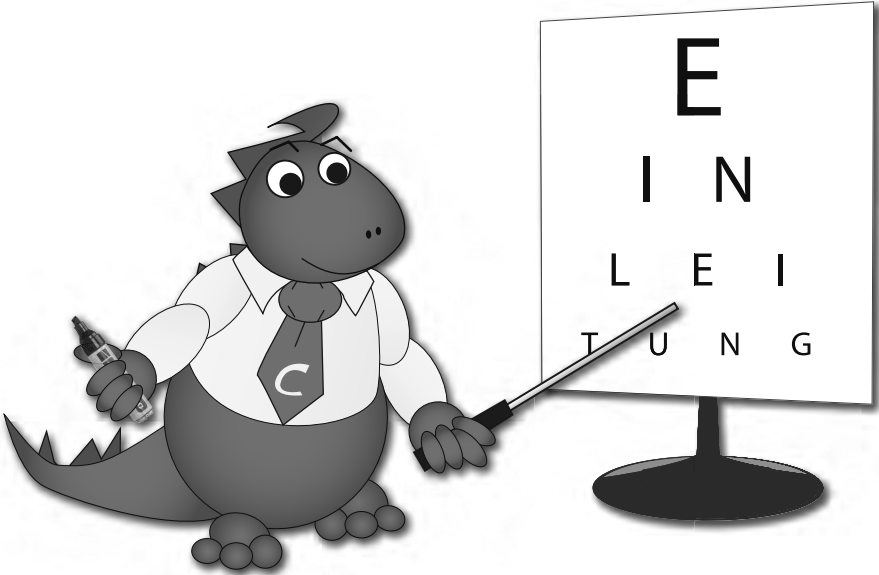
4. 2. 2. Zugriffe über DB-Links	154
4. 2. 3. Einsatz von Synonymen	155
4. 2. 4. Aufruf von Pseudospalten	156
4. 3. Steuerung von Transaktionen	161
4. 3. 1. Grundproblem	161
4. 3. 2. Bestätigen und zurücksetzen von Anweisungen	163
4. 3. 3. Sicherungspunkte	167
4. 3. 4. Transaktionen festlegen	171
4. 3. 5. Tabellen sperren	173
5. Cursor und Mengen	179
5. 1. Prinzip von Cursors	179
5. 1. 1. Erzeugen und verarbeiten	179
5. 1. 2. Cursor-Operationen	181
5. 2. Explizite Cursor	182
5. 2. 1. Deklaration	182
5. 2. 2. Verarbeitung	184
5. 2. 3. Cursor-Schleifen und Cursor-Attribute	188
5. 2. 4. Parametrisierte Cursor und Wertzuweisungen	193
5. 3. Implizite Cursor	195
5. 3. 1. Definition	195
5. 3. 2. Attribute	197
5. 3. 3. Beispiel	198
5. 3. 4. Zeilensperren im Cursor	200
5. 4. Cursor-Variablen	202
5. 4. 1. Deklaration	202
5. 4. 2. Verarbeitung	204
5. 5. Weitere Techniken	209
5. 5. 1. Cursor-Ausdrücke	209
5. 5. 2. Cursor-Ausnahmen	212
6. Fehler und Ausnahmen	217
6. 1. Einführung	217
6. 1. 1. Typologie von Ausnahmen	217
6. 1. 2. Allgemeine Funktionsweise von Ausnahmen	218
6. 1. 3. Vordefinierte Ausnahmen	221
6. 2. Selbst definierte Ausnahmen	223
6. 2. 1. Deklaration und Verwendung	223
6. 2. 2. Fehlermeldungen und Fehlerwerte	225

6. 2. 3. Fehler und Ausnahmen in verschachtelten Blöcken.....	228
6. 2. 4. Verschachtelte Blöcke und besondere Ausnahmen.....	231
6. 2. 5. Verwendung von Ausnahmen bei Schleifen und Blöcken.....	234
7. Collections und ihre Verarbeitung.....	243
7. 1. Collections.....	243
7. 1. 1. Typen.....	243
7. 1. 2. Assoziative Arrays (Index-By-Tabellen).....	244
7. 1. 3. Verschachtelte Tabellen (Nested Tables).....	247
7. 1. 4. Varrays.....	250
7. 1. 5. Multidimensionale Strukturen.....	251
7. 1. 6. Collection-Methoden.....	261
7. 1. 7. Ausnahmen und Collections.....	268
7. 2. Mengenbindung und Mengenverarbeitung.....	268
7. 2. 1. Bindungstypen.....	269
7. 2. 2. Leistungsunterschiede im Vergleich.....	270
7. 2. 3. Mengenübertragung mit FORALL.....	271
7. 2. 4. Verwendung von Cursor-Attributen.....	274
7. 2. 5. Ausnahmebehandlung bei der Mengenbindung.....	277
7. 2. 6. Mengenabfrage mit BULK COLLECT.....	280
7. 2. 7. Einsatz von RETURNING.....	284
8. Dynamisches SQL.....	291
8. 1. Einführung.....	291
8. 1. 1. Grundproblem.....	291
8. 1. 2. Verwendungszweck und Einsatzbereiche.....	293
8. 2. SQL-Befehle mit EXECUTE IMMEDIATE ausführen.....	293
8. 2. 1. Allgemeine Syntax.....	294
8. 2. 2. Beispiel.....	295
8. 3. Dynamisches SQL im Kontext.....	302
8. 3. 1. Verwendung von Cursor-Variablen.....	302
8. 3. 2. Mengenverarbeitung.....	305
8. 3. 3. Beispiele.....	307
9. Prozeduren und Funktionen.....	315
9. 1. Modulare Strukturen in PL/SQL.....	315
9. 1. 1. Typen von Modulen in PL/SQL.....	315
9. 1. 2. Parameter und Parameter-Modi.....	318
9. 1. 3. Fehlersuche und Korrektur.....	325

9. 2. Funktionen und Prozeduren erstellen	332
9. 2. 1. Allgemeine Syntax für Funktionen	332
9. 2. 2. Blockstruktur von Funktionen	334
9. 2. 3. Entscheidungen für Parameter-Modi	339
9. 2. 4. Ausnahmebehandlung	343
9. 2. 5. Übergabe von Werten bei IN und IN OUT	346
9. 2. 6. Rückgaben mit RETURN	348
9. 2. 7. PL/SQL-Funktionen in SQL	354
9. 3. Eigene Prozeduren verwenden	365
9. 3. 1. Die allgemeine Syntax von Prozeduren	365
9. 3. 2. Blockstruktur von Prozeduren	367
9. 3. 3. Speicherort von Prozeduren	369
9. 3. 4. Wahl der Parameter-Modi und Wertübergabe	370
9. 3. 5. Ausnahmebehandlung für Prozeduren	375
9. 3. 6. Übergabe per Referenz und Wert	378
9. 4. Modul-Spezialitäten	379
9. 4. 1. Überladen von Modulen	379
9. 4. 2. Vorwärtsdeklaration und Rekursion	383
9. 4. 3. Einsatz von lokalen Modulen	389
10. Pakete	399
10. 1. Einführung	399
10. 1. 1. Definition und Strukturen	399
10. 1. 2. Vorteile durch die Verwendung von Paketen	400
10. 1. 3. Aufbau von Paketen	401
10. 1. 4. Sichtbarkeit und Gültigkeit	404
10. 2. Eigene Pakete	406
10. 2. 1. Ein Beispieldpaket	406
10. 2. 2. Speicherort von Paketen und ihre Bearbeitung	409
10. 2. 3. Abhängigkeiten zwischen Körpern und Spezifikationen	410
10. 2. 4. Aufruf von Paketen	413
10. 2. 5. Öffentliche und private Elemente	415
10. 2. 6. Berechtigungskonzepte bei gespeicherten Unterprogrammen	417
10. 2. 7. Beispiel-Programm: Testdaten-Erzeugung	422
10. 2. 8. Quelltextschutz und Verschlüsselung	448
10. 3. Eingebaute PL/SQL-Pakete	451
10. 3. 1. Systemüberwachung mit DBMS_UTILITY	452

10. 3. 2. Große Objekte mit DBMS_LOB bearbeiten.....	457
10. 3. 3. Zugriff auf den Shared Pool.....	468
10. 3. 4. Nachrichtenaustausch über Pipes.....	470
10. 3. 5. Datenaustausch via Alarme.....	474
10. 3. 6. Anwendungsoptimierung und -analyse von PL/SQL-Programmen.....	477
11. Trigger.....	501
11. 1. Einführung.....	501
11. 1. 1. Typologie.....	502
11. 1. 2. Einsatzbereiche.....	503
11. 1. 3. Einschränkungen.....	504
11. 2. Trigger programmieren.....	505
11. 2. 1. Allgemeine Trigger-Befehle.....	506
11. 2. 2. DML-Trigger.....	506
11. 2. 3. Syntax-Erweiterungen von Triggern.....	510
11. 2. 4. Instead-of-Trigger.....	518
11. 2. 5. System-Trigger.....	525

Vorwort



Vorwort

Herzlich Willkommen zu einem Fachbuch aus dem Verlag Comelio Medien. Dieses Buch aus dem Bereich Datenbanken soll Sie dabei unterstützen, die umfangreiche SQL-Erweiterung Oracle PL/SQL zu lernen, mit der Sie neben DB-Skripten auch programmierbare DB-Objekte wie Prozeduren, Funktionen und Trigger sowie ganze (Teil-)Anwendungen in Paketen erstellen können. Zusammen mit unseren anderen Büchern zum Thema Oracle lernen Sie das nötige Handwerkzeug zur Programmierung von Oracle über SQL und PL/SQL sowie auch den Einsatz von XML.

Aufbau und Struktur

Ähnlich wie dieses Buch sind auch andere Bücher zum Thema Oracle bei Comelio Medien gestaltet, sodass Sie sehr leicht nebeneinander verwendet werden können, um so dieses interessante Thema aus verschiedenen Blickwinkeln betrachten zu können.

Zielgruppe

Dieses Buch richtet sich an Einsteiger in die PL/SQL-Programmiersprache. Sie sollten schon Erfahrung mit SQL und evtl. auch schon erste Kenntnisse von Oracle haben und nun lernen wollen, wie Sie mit Hilfe von PL/SQL Skripte schreiben und programmierbare DB-Objekte wie Prozeduren erstellen können. Erfahrungen mit anderen Programmiersprachen sind nicht unbedingt notwendig. Da das Buch sehr beispielerorientiert aufgebaut ist und sich PL/SQL auch sehr gut als erste Programmiersprache eignet, ist ein Einstieg in die allgemeine Softwareentwicklung sehr gut möglich. Zusätzlich sehen Umsteiger von anderen Sprachen, mit denen man bspw. Desktop- oder Web-Anwendungen erstellen kann, welche allgemeinen Konzepte bei einer Sprache, die so datenbanknah ist wie PL/SQL, vorhanden sind und wie nützlich eine derartig umfassende Erweiterung zu SQL für die eigene Arbeit ist.

Schreibkonventionen

Das Thema und damit auch diese Buch leben von Beispielen. Quelltext in SQL und PL/SQL wird in diesem Buch in *dicktengleicher Schrift* geschrieben. Sollte es doch einmal wichtig sein, einen besonderen Textbereich hervorzuheben, geschieht dies durch eine **fette, dicktengleiche Schrift**. Wichtige Begriffe, Datei- oder Ordnernamen sind durch *kursive Schrift* gekennzeichnet.

Weitere Buch-Informationen

Wie alle anderen wird auch dieses Buch auf der Comelio Medien-Webseite vorgestellt. Dort können Sie sich über eventuelle Änderungen und Korrekturen sowie Neuauflagen dieses Buches informieren. Dort können Sie auch die Beispiel-Datenbank sowie alle Beispiele dieses Buchs als Quelltext herunterladen.

Die Adresse lautet: www.comelio-medien.com/buch-katalog/oracle/oracle_pl_sql

Kontaktmöglichkeiten

- Die Autoren erreichen Sie an ihrem Arbeitsplatz unter marco.skulschus@comelio.com und marcus.wiederstein@comelio.com. Sie arbeiten beide bei der Comelio GmbH. Webseite: www.comelio.com.
- Den Verlag erreichen Sie unter der Webseite www.comelio-medien.com.

Weitere Informationen

- Offizielle Dokumente finden Sie natürlich auf der sehr umfangreichen Oracle-Webseite, wobei die folgenden Dokumente besonders interessant sind:
 - <http://www.oracle.com/technology/products/database/oracle11g/index.html> (Portalseite der Oracle 11g-R2-Datenbank)
 - <http://www.oracle.com/pls/db112/homepage> (Allgemeine Doku-

mentation zur 11g-R2-Datenbank)

- http://www.oracle.com/pls/db112/portal.portal_db?selected=5&frame=#sql_and_pl_sql_languages (Übersichtsseite zu den Handbüchern von PL/SQL und SQL, wobei insbesondere *PL/SQL Language Reference* und *PL/SQL Packages and Types Reference* für PL/SQL wichtige Informationsquellen sind)
- Seminare und Schulungen zu den vorgestellten Technologien:
 - http://www.comelio-seminare.com/seminare/titel/oracle_11g_für_Oracle-Seminare
 - http://www.comelio-seminare.com/seminare/titel/xml_für_XML-Seminare
- Weitere Bücher zur Oracle-Technologie im gleichen Verlag:
 - *Oracle SQL* - ISBN 978-3-939701-41-5
 - *Oracle, PL/SQL und XML* – ISBN 978-3-939701-10-1
 - *Oracle PL/SQL - Objekte und Objektrelationale Techniken* - ISBN 978-3-939701-42-2
- Weitere kostenlose Lernmaterialien zu Datenbanken und zur Oracle-Technologie im gleichen Verlag:
 - Kostenlose Kurzreferenzen mit Syntax-Darstellungen zu vielen Technologien: <http://www.comelio-medien.com/leserservice/kurzreferenzen>
 - Kostenloses E-Book als PDF *SQL und relationale Datenbanken* (<http://www.comelio-medien.com/buch-katalog/e-books/sql>)

Inhalt nach Kapiteln

Die Themen der einzelnen Kapitel werden im Folgenden kurz aufgelistet:

1. Im ersten Kapitel lernen Sie die Beispieldatenbank kennen und sehen Sie, wie Sie Ihre Arbeitsumgebung einrichten, damit Sie unsere Beispiele nachvollziehen können. Sie wird bei all unseren Oracle-Büchern verwendet. Als Arbeitsumgebung wäre eine Installation der Oracle 11g-Datenbank auf Ihrem lokalen System optimal. Jedoch ist die Verwendung der kostenlosen *Oracle Express Edition* für die meisten Themen ebenfalls völlig ausreichend und lässt sich sehr einfach auf einem privaten Computer durchführen, da diese Ausgabe nicht so viele Systemressourcen benötigt.
2. Das zweite Kapitel gibt einen Überblick über PL/SQL und seinen Einsatzmöglichkeiten bei der Datenbank- und Software-Entwicklung. Dazu zählen erste Beispiele in Form von Skripten, die aus einer Entwicklungsumgebung wie bspw. des kostenlosen *Oracle SQL Developer* direkt zur Datenbank geschickt und dort dann unmittelbar ausgeführt werden.
3. Das dritte Kapitel stellt die grundlegende Syntax von PL/SQL vor. Dazu zählen die Deklaration von Variablen und die Verwendung geeigneter Datentypen oder die verschiedenen Ausgabemöglichkeiten von Ergebnissen in der Standardausgabe oder in Dateien. Für die Programmlogik lernen Sie die verschiedenen Kontrollanweisungen für Fallunterscheidungen und Schleifen kennen. Es schließt mit der Darstellung von Datensatztypen zur Gestaltung von komplexen Datenstrukturen.
4. Das vierte Kapitel zeigt, wie Sie in SQL und PL/SQL Transaktionen beginnen und entweder beenden oder abbrechen und welche verschiedenen Erweiterungen es in PL/SQL gibt, um mehrere Befehle zu einer gemeinsamen Transaktionen zusammenzufassen.
5. Das fünfte Kapitel beschäftigt sich damit, wie man mit Hilfe von Cursors mehrere Zeilen aus Abfragen abrufen kann, um diese dann zu manipulieren oder für sonstige Verarbeitungen zu nutzen. Es stellt dabei die verschiedenen Cursor-Typen mit einzelnen Beispielen vor.
6. Das sechste Kapitel fokussiert die Behandlung von Ausnahmen und Fehler im Programm und zeigt die verschiedenen Möglichkeiten, Fehlernummern

- und ihre Meldungen von der Datenbank abzurufen, eigene Fehler auszugeben oder kritische Programmbereiche mit Hilfe von Ausnahmen robuster zu gestalten.
7. Das siebte Kapitel greift noch einmal das Thema der Behandlung von mehreren Datenreihen auf und führt die Collections ein. Sie erlauben die Speicherung von mehreren einzelnen Werten einfacher Datentypen oder Datensatztypen sowie die Verschachtelung mit weiteren Collections zum Aufbau von komplexen Datengebilden und ihrer gemeinsamen Behandlung.
 8. Das achte Kapitel bietet eine Zusammenfassung verschiedener Themen vorheriger Kapitel, zeigt sie allerdings unter dem Aspekt, wie man die SQL-Anweisungen nicht statisch, sondern dynamisch zusammensetzen und dann ausführen kann.
 9. Das neunte Kapitel verlässt den Bereich der so genannten anonymen Blöcke, d.h. der einfachen PL/SQL-Skripte, die direkt zur Datenbank gesendet und dort ausgeführt werden. Es erklärt, wie Sie Prozeduren und Funktionen erstellen und diese dann über ihren Namen parametrisiert aufrufen können. Es enthält auch weitere Techniken wie lokale Module oder Überladung.
 10. Das zehnte Kapitel führt die Pakete ein. Zum einen zeigt es, wie Sie selbst Pakete mit öffentlichen und privaten Elemente erstellen und dort Prozeduren, Funktionen und Typen wie Datensatztypen oder Cursor unter einem gemeinsamen Namen sammeln. Zum anderen stellt es auch einige schon vorhandene nützliche PL/SQL-Pakete vor.
 11. Das elfte Kapitel präsentiert das Konzept der Trigger-Programmierung und enthält Beispiele für die verschiedenen Arten von Triggern.

Autoren

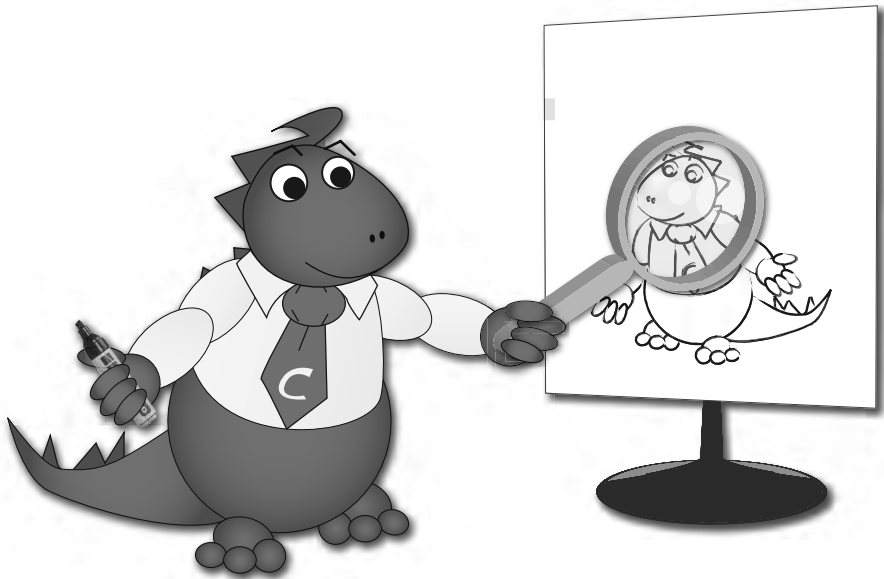
Marco Skulschus

Marco Skulschus studierte Ökonomie in Wuppertal und Paris und schrieb seine Diplomarbeit über die Modellierung von semantischen Informationen in XML-Strukturen. Er ist Autor von vielen Büchern zu Softwaretechnik, darunter insbesondere das Thema XML und Datenbanken. Thematisch arbeitet er in Projekten als Berater und Projektleiter und unterstützt Marktforschungs- und Beratungsunternehmen sowie auch mittelständische Unternehmen dabei, Business Intelligence-Systeme aufzubauen sowie individuelle kaufmännische Software einzuführen.

Marcus Wiederstein

Marcus Wiederstein studierte Elektrotechnik in Bochum und Dortmund. Als Autor konzentriert er sich auf Datenbanken wie Oracle und MS SQL Server sowie die Softwareentwicklung mit Microsoft-Technologien. In Projekten arbeitet er als Projektleiter und Berater. Dort ist er dann für Server-Architekturen und die Themen wie Sicherheit und Hochverfügbarkeit verantwortlich. Thematisch betreffen sie die Bereiche Business Intelligence und klassische kaufmännische Software-Lösungen.

Einführung



1. Einführung

Dieses Kapitel erläutert kurz die Beispiel-Datenbank und erklärt, wie man sie einrichtet.

1. 1. Beispieldatenbank

Wir haben uns für dieses Buch – wie auch für unsere anderen Bücher – eine eigene Beispieldatenbank ausgedacht: Die Firma *Kurse NRW GmbH* bietet EDV- und IT-Seminare in Bereichen wie Programmierung, Datenbanken oder Webdesign an. Informationen, die den Bereich der Schulungs- und Anmeldeverwaltung betreffen, sollen automatisch verwaltet werden.

1. 1. 1. Entitäten

Folgende Entitäten sind vorhanden:

- *Kurse* mit Informationen zu Titel, Bereichszugehörigkeit, Dauer, Vorkenntnissen, Themen und Preisen,
- *Kunden bzw. Teilnehmer der Seminare*, die sowohl Einzelteilnehmer für offene Seminare (Standardkurse zu einem festen Termin) als auch Unternehmenskunden für Firmenseminare (spezielle Seminare am Standort bzw. in den Räumlichkeiten eines Unternehmens) sein können,
- *Dozenten*, die Firmen- und Standardseminare durchgeführt haben, für die Durchführung eingeplant werden können, also mit ihrer Themenverteilung bereitstehen, und einen bestimmten gewünschten Tagespreis sowie einen Mindestpreis haben, der z. B. bei einem Einzelteilnehmer mit dadurch eingeschränktem Seminargewinn für das Schulungsunternehmen auch für den Dozenten noch akzeptabel wäre, und schließlich

- *Mitarbeiter und Geschäftsführer der Kurse NRW GmbH*, die Seminaranmeldungen buchen bzw. in das System einpflegen oder aktualisieren, Marketing betreiben, Dozenten kennen lernen und natürlich umfangreiche Mengen an Kaffee, Gebäck und sonstigen Leckereien vertilgen, die im Teilnehmerschrank aufbewahrt werden und einer ständigen Qualitätskontrolle bedürfen.

1. 1. 2. Szenario

Neben den unmittelbar erkennbaren Strukturen der Schulungsfirma und den daraus abzuleitenden Anforderungen an das Datenmodell gibt es einige Besonderheiten, die die Kurse NRW GmbH auch über die Grenzen von Nordrhein-Westfalen hinaus bekannt gemacht haben.

So hat z. B. der mehr als findige Geschäftsführer Anton Ebenhof viel Zeit und Tüftelei darauf verwandt, ein raffiniertes Rabattsystem einzuführen, das natürlich ebenfalls bei Buchungen und im Rahmen der eigentlichen Kursverwaltung berücksichtigt werden muss. Für jeden Kurs gibt es unterschiedliche Preise. Zum einen – ganz klassisch – einen Preis für Teilnehmer, die sich für einen Standardkurs mit festem Termin anmelden, und zum anderen einen Firmenpreis, der nur gilt, wenn ein Unternehmen einen Dozenten zu sich einlädt, der dann entweder einen Kurs aus dem Katalog oder einen individuellen Kurs für das Unternehmen mit speziellen Inhalten hält. Daraus ergibt sich die Situation, dass der so genannte „Nasenpreis“ (Preis pro Kopf oder pro Teilnehmer) für Unternehmen in diesen Situationen regelmäßig günstiger ist, als wenn Unternehmensangehörige für einen Standardkurs angemeldet würden. Obwohl dies ein im ganzen Bundesgebiet fast schon standardisiertes Verfahren der Preisberechnung ist, empfindet Herr Ebenhof als alter Pfadfinder mit kaufmännischer Ausbildung dieses System als schreiende Ungerechtigkeit, weil einem Marktteilnehmer durch seinen Rechtscharakter eine Vergünstigung gewährt wird, die Individuen mit anderem Rechtscharakter verwehrt wird. Um also Gerechtigkeitsempfinden und Gewinninteresse – die beiden Herzen, die in Anton Ebenhofs Brust pochen – miteinander zu versöhnen, gewährt sein Unternehmen für Gruppenanmeldungen ebenfalls einen Rabatt, der zu einem günstigeren Preis pro Teilnehmer führt. Dies lockt tendenziell mehr Menschen in einen Kurs, weil sie Kollegen oder Freunde leichter durch menschlich-kapitalistisches Vorteilsdenken überzeugen können, sie zu einem günstigeren Preis zu begleiten und wertvolles Wissen für eine besser bezahlte Arbeitsstelle zu erwerben. So verringert sich also – kurz gesagt – der Preis vom teuersten Preis

für einen Einzelteilnehmer bis hin zu einer Zahl von fünf Teilnehmern immer um einige Prozentschritte. Ab fünf Teilnehmern gibt es dann einen Pauschalpreis pro Person für alle Teilnehmer in der Gruppe. An dieser Stelle muss das Seufzen von Elvira Hülzemann erwähnt werden, die die Komplikation beim Anmeldevorgang in der Buchhaltung sofort bemerkt hatte, worauf wir später im Rahmen der Datenerzeugung zu sprechen kommen.

Überhaupt die Anmeldungen: Selbstverständlich müssen sie wegen unterschiedlicher Erfordernisse (Erfolgskontrolle, Anmeldeverwaltung, Steuerinteresse des örtlichen Finanzamtes) eigens verwaltet werden. Problematisch ist allerdings die Berücksichtigung der unterschiedlichen Möglichkeiten, welche Teilnehmer sich selbst oder andere für einen Kurs anmelden. Da die Kurse NRW GmbH gleichzeitig Firmenschulungen wie auch Seminare zu festen Terminen anbietet, gibt es mindestens diese beiden Gruppen. Zusätzlich können Firmen allerdings auch einen Trainer für ein Inhouse-Seminar im Unternehmen einladen. Daher ergeben sich folgende Anwendungsfälle für den Anmeldevorgang:

- Eine Firma meldet mehrere Teilnehmer für ein Seminar zum entsprechenden Preis an.
- Eine Firma bucht einen Dozenten für ein Firmenseminar mit beliebiger Teilnehmerzahl.
- Ein Einzelteilnehmer meldet sich und/oder seine Freunde an.

Bei der Ableitung eines benutzbaren und übersichtlichen Datenmodells sollte es also ein spezielles Ziel sein, dass die unterschiedlichen Fälle gleichzeitig in einer Anmeldungstabelle verwaltet werden können. Zusätzlich muss die Möglichkeit bestehen, dass Anmeldungen auch wieder rückgängig gemacht (storniert) werden können, wobei natürlich die zusammenhängenden Teilnehmer in einer Gruppe identifizierbar bleiben müssen, damit der dann falsche Preis um eine entsprechende Stufe heraufgesetzt werden kann. Diese Regelung soll auch verhindern, dass besonders findige Teilnehmer (wie man sie unter Programmierern leicht finden kann) auf die glorreiche Idee kommen, zunächst immer den ganzen Fußballverein anzumelden und später alle an „Java für Oracle“ nicht so Interessierten wieder abzumelden, um selbst nur den günstigsten Pauschalpreis zu bezahlen.

Es ist für die Durchführung eines so hinreichend sicher gebuchten Seminars zweifelsohne bedeutsam zu wissen, auf welche Dozenten das Unternehmen zurückgreifen kann. Noch wichtiger hingegen ist die Themenverteilung, um die passenden Dozenten mit einem vorher bekannten Tagespreis einem Kurs zuzuordnen und dann Kontakt mit ihnen aufzunehmen. Dabei kann man natürlich von größtmöglicher Unordnung ausgehen: Die einzelnen Dozenten unterrichten unterschiedliche Themen, sie wechseln die Themen, wenn sie selbst Neues gelernt haben oder ein Thema aufgrund persönlicher Weiterentwicklung nicht mehr schulen möchten, und letztlich passen sie auch ihre Tagessätze an ihren Auslastungsgrad, persönlichen Lebensstil und die Anzahl der Familienangehörigen und Hypotheken an.

Kurse werden nicht nur gebucht. Sie haben auch die Eigenschaft, an einem bestimmten Termin und Ort mit einer ebenso bestimmten Länge stattzufinden. Diese Termine wiederholen sich innerhalb bestimmter Perioden für gewisse Kurse und werden (wie bei Standardkursen) entweder weit vorher im Jahr festgelegt oder (wie bei Firmenschulungen) kurzfristig anberaumt.

Aus diesen Überlegungen ergeben sich nun die folgenden neuen Entitäten:

- Preisstufen für die unterschiedlichen Preise, die einzelnen Kursen zugeordnet werden können,
- Themenzuordnungen in Form einer unternehmensweiten Themenverteilung für Dozenten und ihre (Mindest-)Tagessätze (dies ist keine Entität, sondern eine Relation mit zusätzlichen Attributen),
- Termine für Standardkurse und Firmenseminare,
- Anmeldungen der einzelnen Kurstypen von verschiedenen Anmeldungseinheiten (Kunden) und
- Mitarbeiter, die z. B. Anmeldeinformationen eingeben, sich an die Datenbank anmelden, PL/SQL-Prozeduren für einfachere Arbeit anlegen und den Administrator mit der Erzeugung von Testdaten zur Abfrageoptimierung nervös machen können.

Für die beiden Bereiche der Mitarbeiter- und Anmeldungserfassung müssen wir die grundsätzlichen Einschränkungen des Beispiels kurz umreißen: Natürlich

könnte man sämtliche Unternehmensinformationen aus anderen Geschäftsbereichen mit Produkten, Geschäftspartnern und Kunden erfassen. Gleiches gilt für regelrechte Buchungssätze im kaufmännischen Sinne, wie sie bereits für die Erfassung der Anmeldungen und der Dozenten notwendig werden, ohne noch die anderen Geschäftstätigkeiten im Rahmen des Schulungsgeschäftes (wie Einkäufe von Unterrichtsmaterialien oder Ausgaben für Mieten, Abzahlungen, Versicherungen usw.) zu berücksichtigen.

Jede Erweiterung des vorliegenden Basismodells würde jedoch die ohnehin schon nicht geringe Anzahl an Tabellen sprengen und würde eher auf Probleme der Datenmodellierung und in diesem Sinne auf die Besonderheiten des Rechnungswesens hinauslaufen. Unter dieser größeren Realitätsnähe würden allerdings alle anderen Beispiele sehr leiden, weil die Besonderheiten des entstehenden Datenmodells sich unweigerlich auf Standardprozeduren oder mögliche Abfragen niederschlagen würden. Eine Begrenzung der Beispiele auf jene, die das vorliegende Datenmodell liefert, wäre dann teilweise widersinnig, da man dann auch auf das umfangreichere Datenmodell hätte verzichten können. Aus diesen Gründen beschränken wir uns hier auf das Wesentliche – die Seminarverwaltung.

1. 1. 3. Tabellen und Beziehungen

Die nachfolgende Abbildung enthält die Tabellen der Beispiel-DB mit ihren Spalten und den Beziehungen untereinander.

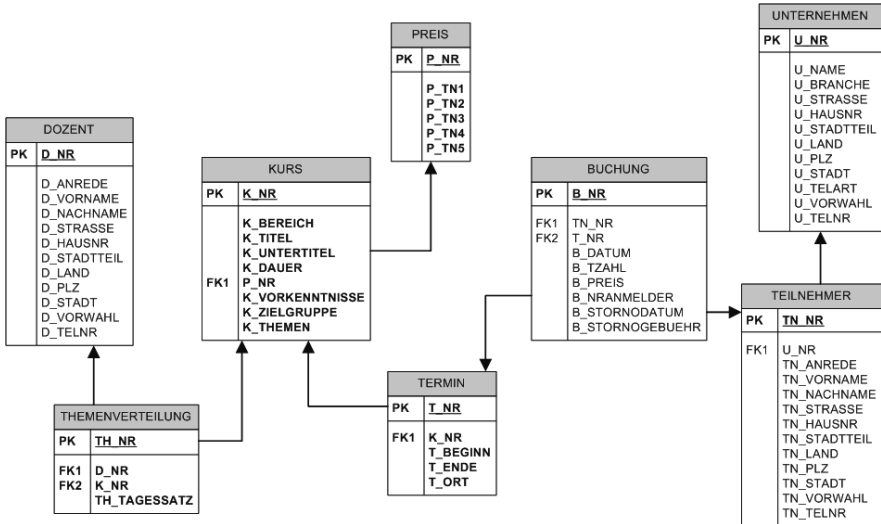


Abbildung 1.1: Das Datenmodell von Kurse NRW

Nach diesen Überlegungen kommt man zu folgendem Modell, das für das gesamte Buch als Standardmodell bzw. als einfachste Ausgangsbasis dienen wird. Es besitzt folgende Entitäten bzw. Tabellen, von denen wir der Übersichtlichkeit halber sowie zum Verständnis einige Besonderheiten kommentieren. Triviale Datenfelder wie z. B. Adressfelder (Vorname, Nachname, PLZ, Stadt usw.) erschließen sich sicherlich von selbst. Ansonsten haben wir einige Vereinfachungen vorgenommen, da eine realistische Seminarverwaltung für das Erlernen der vorgestellten Syntax nicht unbedingt erforderlich ist.

- **KURS**: Die Kursnummer ist ein spezieller Schlüssel, der sich aus zugewiesenen Zahlenwerten für einen Ober- und Unterbereich sowie aus einer fortlaufenden, dreistelligen Nummer am Ende zusammensetzt, so dass Kurse auch über ihre ersten beiden Ziffern grob gegliedert werden können. Der Bereich setzt sich aus Werten wie {Programmierung, Webdesign, Datenbanken...} zusammen.

men. Da jeder Kurs unterschiedliche Preise im Rahmen der oben erwähnten Anwendungsfälle des Anmeldevorgangs besitzt, existiert eine Fremdschlüsselbeziehung zur Tabelle `PREIS`. Weitere Informationen zu Kurspreisen finden sich dann in der Tabelle `THEMENVERTEILUNG` und natürlich als Ergebnis in der Tabelle `BUCHUNG`.

- `PREIS`: Verschiedene Kurse mit ähnlichem Schwierigkeitsgrad, ähnlicher Länge oder ähnlichem Inhalt haben gleiche Preise bzw. die gleiche Preisstufennummer, so dass weniger Preisstufen als Kurse existieren. Die Preise entstehen durch eine gleichmäßige, prozentuale Verringerung des Einzelteilnehmerpreises. Werden mehrere Teilnehmer von einem Unternehmen gleichzeitig zu einem Termin angemeldet, erhalten sie alle jeweils einen günstigeren Preis.
- `DOZENT`: Dozenten stellen wie die Mitarbeiter sehr einfache Entitäten ohne Besonderheiten dar. Sie enthalten Adress- und Namensfelder.
- `THEMENVERTEILUNG`: Diese Tabelle speichert sowohl Informationen zur eigentlichen Themenverteilung (Welcher Dozent führt welchen Kurs durch?) als auch zu seiner Bezahlung in einer Aufteilung von Standard-Tagessatz und Mindest-Tagessatz. Der Mindest-Tagessatz stellt eine Untergrenze dar, bei der ein Kurs mit z. B. nur einem Teilnehmer für ihn noch wirtschaftlich vertretbar wäre. Angestrebt wird ein Lastenausgleich zwischen Schulungsunternehmen und Dozent durch beiderseitig verringerte Einnahmen.
- `TERMIN`: Seminartermine werden im Rahmen einer Jahresplanung festgelegt und können mit vorher bekannt gegebenen Inhalten gebucht werden. Ob sie zustande gekommen sind, lässt sich hier nicht ermitteln, sondern nur durch eine Auswertung der Tabelle `BUCHUNG`. Neben Verknüpfungen zu den Tabellen `KURS` (für die Kursinhalte eines Termins) und `BUCHUNG` (für die erfassten Teilnehmer eines Termins) existieren noch Datumsfelder und eine Ortsangabe.
- `UNTERNEHMEN`: Diese Tabelle enthält wie die Tabellen `DOZENT` oder `TEILNEHMER` keine Besonderheiten. Es existieren Namens- und Adressfelder für die Beschreibung von Unternehmen.
- `TEILNEHMER`: Ganz klassisch werden hier sämtliche Teilnehmer erfasst, die sich für ein Seminar angemeldet haben oder von anderen für eines angemeldet worden sind. Gleichzeitig können sie aber auch eine Unternehmensnum-

mer besitzen und sich selbst oder Kollegen für einen Kurs angemeldet haben. Dies ist notwendig, weil es relativ viele Arbeitnehmer gibt, die ihre Seminare selbst aussuchen und nach entsprechender Rückmeldung und Bestätigung auch selbst buchen können. Sie können daher ebenfalls für Werbemaßnahmen interessant sein, wie es z. B. auch ein Mitarbeiter der Personalabteilung sein kann, der für andere Mitarbeiter ein Seminar oder eine Firmenschulung bucht. Diese Tabelle enthält ausschließlich Daten von Menschen, während die Tabelle `UNTERNEHMEN` ausschließlich Daten von Organisationen (Firmen, Behörden, Körperschaften) speichert.

- **BUCHUNG:** Diese Tabelle stellt das Kernstück der gesamten Datenbank dar, wie es für Informationen üblich ist, die in Unternehmen anfallen und für das Rechnungswesen nötig sind. Erfasst werden allerdings ausschließlich die Anmeldungen und keine richtigen Buchungssätze. Die Tabelle enthält Verknüpfungen zu den Terminen, über die auch die Kursdaten erhältlich sind, und zu den Teilnehmerdaten, über die auch die Unternehmensdaten gefunden werden können. Ein Teilnehmer wird mit seiner Nummer für einen Termin gebucht, der ebenfalls als Nummer vorliegt. Der aus der `PREIS`-Tabelle ermittelte Preis wird in einer speziellen Spalte gespeichert, da er ja abhängig von der Gruppenstärke ist, in der der einzelne Teilnehmer angemeldet wurde. Die Gruppenbeziehung wird über die Erfassung einer Teilnehmernummer in einer speziellen Spalte realisiert, sodass die Teilnehmer für einen Termin über die Nummer eines teilnehmenden Kollegen auffindbar sind. Dies wäre dann der so genannte „Anmelder“.

1. 2. Einrichtung der Umgebung

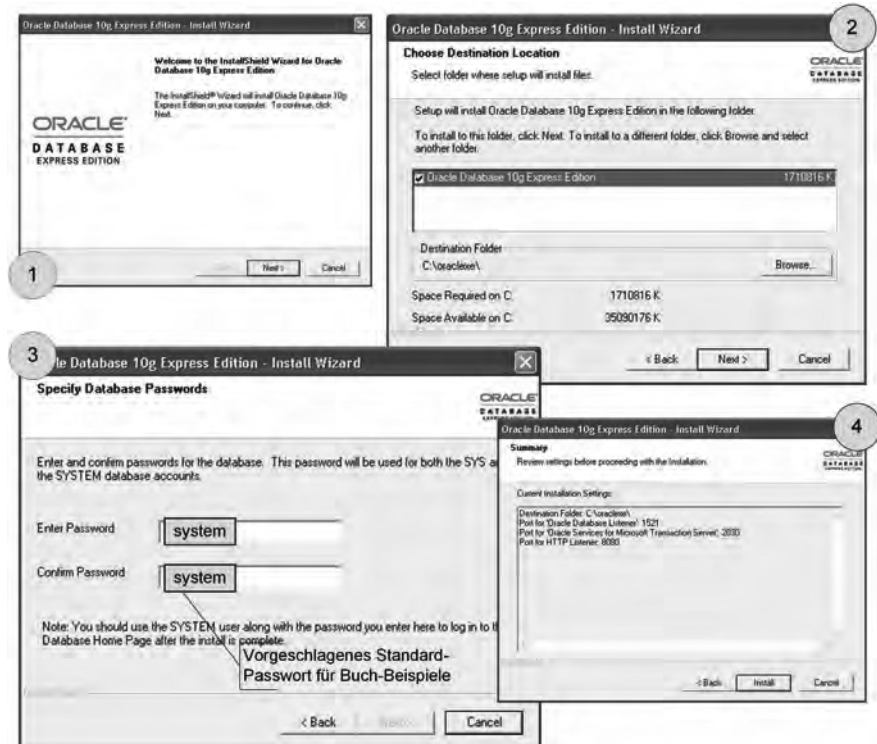
Sie können alle Beispiele aus diesem Buch leicht nachvollziehen, da in den Download-Dateien sowohl die benutzte Beispiel-Datenbank wie auch sämtliche Skripte zur Verfügung stehen.

1. 2. 1. Oracle Database 10g Express Edition

Auch wenn zur Drucklegung nur die Oracle Database 10g Express Edition verfügbar war, und die 11g-Version vermutlich erst im Laufe von 2012 erscheint, kann man die Programmierung PL/SQL mit der Vorgänger-Version sehr gut lernen. Die Express-Editionen haben den Vorteil, dass ihre Installation bzw. ihr Betrieb deut-

lich ressourcenschonender als die „großen“ Server-Versionen sind. Sie finden das notwendige Software-Paket unter der folgenden URL

<http://www.oracle.com/technetwork/database/express-edition/downloads/index.html>



Installation Oracle 10g Express

1. Starten Sie die Installation, nachdem Sie die heruntergeladene Datei entpackt haben, und wählen Sie *NEXT*.
2. Legen Sie das Installationsverzeichnis fest oder akzeptieren Sie die Standardauswahl auf der C-Festplatte und wählen Sie *NEXT*.
3. Geben Sie Passwörter für die beiden Administrator-Konten *SYS* und *SYSTEM*

vor. In der Abbildung schlagen wir Ihnen bspw. `system` vor. Dieses Passwort benötigen Sie später bei einer kleinen Grund-Konfiguration des Systems. Wählen Sie dann wieder `NEXT`.

4. Ihnen wird eine kleine Zusammenfassung angezeigt. Wählen Sie dann `INSTALL`.

Die Express-Edition kann im produktiven Einsatz genutzt werden und ist daher durchaus mehr als nur eine günstige und ressourcenschonende Lernumgebung. In diesem Sinne vereinfachen wir hier die Darstellung sehr und können nur auf andere Quellen wie die Oracle-Dokumentation verweisen, wenn Sie sich intensiver mit der Administration beschäftigen wollen. Wir begrenzen hier die Darstellung auf das absolut Notwendige, um eine Arbeitsumgebung zu haben.

1. Wählen Sie `START / PROGRAMME / ORACLE 10G DATABASE EXPRESS EDITION / GEHE ZU DATENBANK-HOMEPAGE`.
2. Dies öffnet in Ihrem Internet-Browser die Konfigurationswebseite, mit der Sie grafisch die Datenbank administrieren können. Es ist auch möglich, Skripte in SQL und PL/SQL auszuführen, aber natürlich bietet das entsprechende Formular keine Syntax-Unterstützung. Wir benötigen diese Webseite nur, um einen Datenbank-Benutzer anzulegen, mit dem wir die Beispiel-Datenbank erstellen können. Melden Sie sich mit dem `SYSTEM`-Konto und Ihrem bei der Installation vergebenem Passwort an.
3. Wählen Sie `ADMINISTRATION / DATABASE USERS / CREATE USER`, um einen neuen Benutzer zu erstellen.
4. Legen Sie einen neuen Benutzer an. Die Buch-Beispiele verwenden einen Benutzer namens `SCOTT` mit dem Passwort `TIGER`, der vor langer Zeit in Oracle als Standardbenutzer vorhanden war. Sie können auch einen anderen Namen verwenden, müssen dann aber später beim Aufspielen der Tabellen und Daten per Suchen/Ersetzen Ihren Benutzer in das Skript eintragen. Geben Sie diesem Benutzer alle verfügbaren Rechte.

1

2

3

4

Bei Installation vergebenes Passwort

Vorgeschlagener Beispiel-Benutzer für Buch-Beispiele

Einrichten eines Benutzers auf der DB-Homepage

1. 2. 2. Entwicklungsumgebung

Neben der Client-Software oder der Web-Ansicht über die Datenbank-Homepage gibt es auch noch zwei weitere Werkzeuge, die sehr leicht zu installieren und kostenlos verfügbar sind.

1. 2. 2. 1 Oracle SQL Developer

Für die Verwendung von SQL und PL/SQL sowie die Erstellung und Verwaltung von Speicherstrukturen und Datenbank-Objekten können Sie das kostenlose Werkzeug „Oracle SQL Developer“ verwenden. Es dürfte neben der Datenbank selbst das wichtigste Werkzeug sein, das Sie verwenden sollten, um SQL und PL/SQL zu lernen.

Sie erhalten das Programm unter dieser Adresse:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

Eine Installation ist nicht erforderlich. Wenn Sie die vom Speicherplatz größere Version herunterladen, müssen Sie auch nicht unbedingt Java auf Ihrem Rechner installiert zu haben, denn eine Java Runtime Environment ist hier bereits enthalten. Entpacken Sie das ZIP-Archiv in einem Ordner wie bspw. bei all Ihren anderen Programmen und starten Sie dann *sqldeveloper.exe*.

Mit den nächsten Schritten richten Sie eine Verbindung zur Datenbank ein. Wie dann die Tabellen und Daten erzeugt werden, erklärt Ihnen der übernächste Abschnitt.

1. Links befindet sich ein Bereich, in dem die über eine geöffnete DB-Verbindung verfügbaren DB-Objekte in einer Baum-Darstellung aufgelistet werden. Wählen Sie hier aus dem Kontextmenü *NEUE VERBINDUNG*.
2. Tragen Sie in die dafür vorgesehenen Textfelder Benutzernamen und das Passwort für den zuvor eingerichteten Benutzer ein. Die beiden Konten *SYS* und *SYSTEM* können Sie natürlich auch verwenden. Wie in der Abbildung sollte es sich dabei um die Werte *SCOTT* und *TIGER* handeln. Das Passwort können Sie speichern, sodass Sie es nicht jedes Mal neu eingeben müssen.

3. Wählen Sie *TEST*, um die Verbindung zu testen. Wenn links unten eine Bestätigung über die erfolgreiche Verbindung erscheint, können Sie der Verbindung einen Namen im obersten Textfeld geben und *SPEICHERN* wählen. So können Sie bspw. eine Verbindung für verschiedene Benutzer einrichten und dann später einfach auswählen.
4. Wählen Sie *ANMELDEN*, um die Anmeldung durchzuführen.



DB-Verbindung in Oracle SQL Developer einrichten



DB-Verbindung und Objekte wie Tabellen und PL/SQL-Quelltext bzw. Pakete, Prozeduren etc.

DB-Objekte anzeigen und bearbeiten

Sie erhalten das Programm unter dieser Adresse:

<http://www.oracle.com/technetwork/developer-tools/datamodeler/downloads/index.html>

Testen können Sie das Programm, indem Sie die SQL-DDL-Datei für die Beispiel-datenbank (*Kurse_NRW_Tabellen.sql*) importieren oder natürlich direkt die XML-Datei öffnen, welche das Datenmodell enthält.

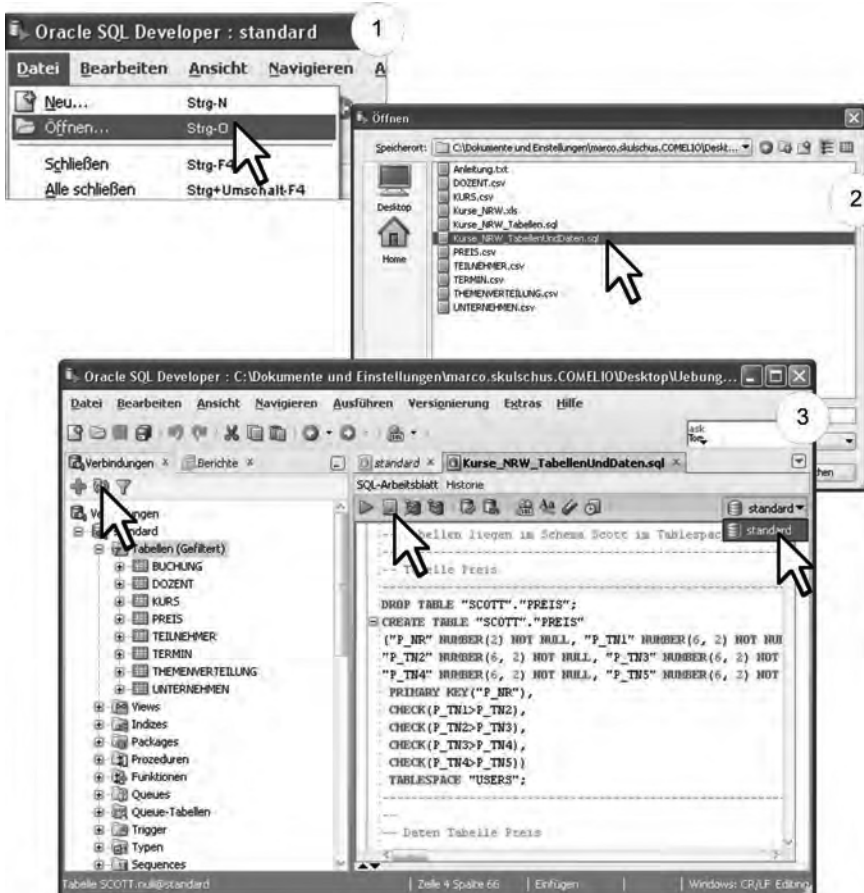
1. 3. Einrichtung der Beispiel-Datenbank

Die Beispiel-Datenbank liegt in verschiedenen Formaten vor, sodass man sie auf verschiedenen Systemen und bei unterschiedlichen Voraussetzungen in der Test-Umgebung nutzen kann.

1. 3. 1. Dateien

In den Beispiel-/Übungsdateien befindet sich neben den Ordnern, welche die jeweiligen Skripte der einzelnen Kapitel enthalten, ein weiterer Ordner namens *Kurse_NRW*. Er enthält wiederum zwei Ordner und eine Abbildung des Datenmodells der Beispiel-Datenbank. Der Ordner *Rohdaten* enthält die folgenden Dateien:

- *Kurse_NRW_TabellenUndDaten.sql*: Diese Datei enthält `CREATE`- und `INSERT`-Anweisungen, welche die gesamte Datenbank mit Tabellen und Daten aufspielen. Dies ist das einfachste Vorgehen.
- *Kurse_NRW_Tabellen.sql*: Diese Datei enthält nur die Tabellen sowie eine Sammlung von `DELETE`-Befehlen, mit denen die Tabellen und Daten in der richtigen Reihenfolge vollständig gelöscht werden können.
- *Kurse_NRW.xls*: In der MS Excel-Datei befinden sich ebenfalls die Datenbank-Daten, welche wiederum einzeln auch in den unterschiedlichen CSV-Dateien zu finden sind. Diese könnte man verwenden, wenn man die Daten über CSV einspielen will.



Installation der Beispiel-DB

Da Sie ohnehin eine Entwicklungsumgebung benötigen, liegt es nahe, diese auch für die Installation der Beispiel-Datenbank zu verwenden.

1. Wählen Sie *DATEI* / *ÖFFNEN*.
2. Wählen Sie die benötigte Datei. Normalerweise sollte dies *Kurse_NRW_TabellenUndDaten.sql* sein, da hier ja sowohl die Tabellen wie auch die Daten enthalten sind.

3. Wählen Sie eine Verbindung zur Datenbank in der Auswahlliste.
4. Führen Sie das Skript mit F5 oder der Schaltfläche *SKRIPT AUSFÜHREN* aus. Wenn Sie keinen Benutzer wie `SCOTT` eingerichtet haben, dann müssen Sie zuvor noch mit *BEARBEITEN / ERSETZEN* die Zeichenkette "`SCOTT`". aus der gesamten Datei löschen.
5. Aktualisieren Sie dann die Tabellenauflistung, um zu kontrollieren, ob die Tabellen vorhanden sind. Wenn Sie eine Tabelle auswählen, öffnet sich ein neues Dokument mit verschiedenen Reitern mit Informationen über die Spalten und auch den Daten.