

Oracle SQL



Marco Skulschus
Marcus Wiederstein

Oracle SQL

Marco Skulschus
Marcus Wiederstein



Oracle SQL

Marco Skulschus
Marcus Wiederstein

Webseite zum Buch:

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

© Comelio Medien 2012

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

ISBN 978-3-939701-47-7

Inhaltsverzeichnis

1. Einführung	18
1. 1. Beispieldatenbank	18
1. 1. 1. Entitäten	18
1. 1. 2. Szenario	19
1. 1. 3. Tabellen und Beziehungen	22
1. 2. Einrichtung der Umgebung	25
1. 2. 1. Oracle Database 10g Express Edition	25
1. 2. 2. Oracle Database 11g Express Edition	28
1. 2. 3. Entwicklungsumgebung	32
1. 3. Einrichtung der Beispiel-Datenbank	35
1. 3. 1. Dateien	36
1. 3. 2. Tabellen und Daten aufspielen	37
2. SQL und relationale Datenbanken	40
2. 1. Das relationale Datenbank-System	40
2. 1. 1. Zentrale Begriffe	40
2. 1. 2. Sichten auf ein relationales Datenbanksystem	42
2. 1. 3. Anforderungen an ein DBMS	44
2. 1. 4. Bestandteile einer Tabelle	45
2. 1. 5. Inhalte einer relationalen Datenbank	46
2. 1. 6. Architektur-Muster beim Einsatz relationaler Datenbanken	48
2. 2. Das relationale Modell	51
2. 2. 1. Grundbegriffe des relationalen Modells	52
2. 2. 2. Semantisches Modell	52
2. 2. 3. Eigenschaften von Daten	54
2. 2. 4. Klassifikation von Datentypen	57
2. 2. 5. Beziehungen zwischen Daten	58
2. 2. 6. Entity-Relationship-Modell	62
2. 2. 7. Normalisierung mit Normalformen	65
2. 2. 8. DB-Anomalien	73

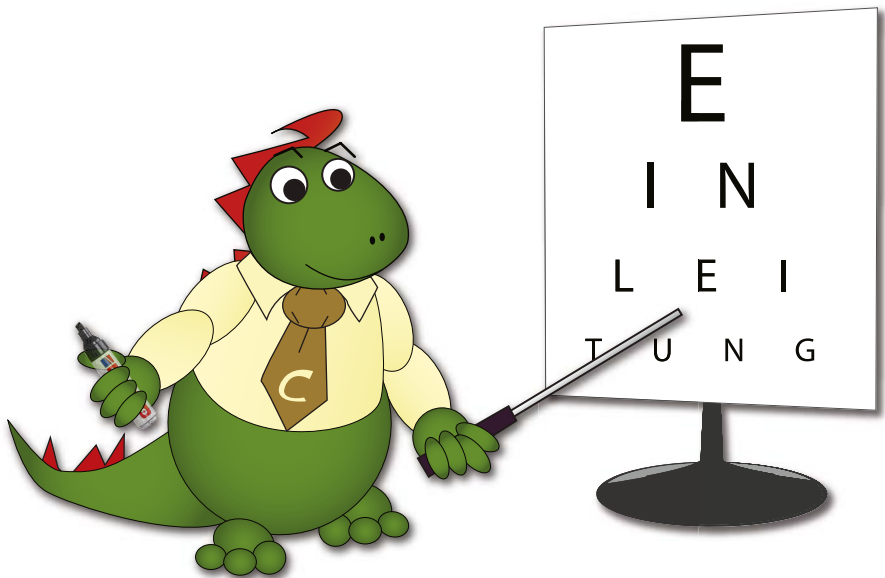
2. 3. Einführung in SQL	75
2. 3. 1. Eigenschaften und Ursprünge der Sprache	75
2. 3. 2. Sprachbereiche und Verwendungsdimensionen	76
2. 3. 3. Anwendungsbereiche und Grundkonzeption	78
2. 3. 4. Erscheinungsformen von SQL	80
2. 3. 5. Mengenkonzepte von SQL bzw. relationale Sprachen	81
3. Einfache Abfragen	89
3. 1. Grundstrukturen von Abfragen	89
3. 1. 1. Basisform einer Abfrage	89
3. 1. 2. Formulierung von Ergebnisrelationen	91
3. 2. Filter mit der WHERE-Klausel	102
3. 2. 1. Funktionsweise eines Filters	103
3. 2. 2. Einfache Operatoren in Bedingungen	106
3. 2. 3. Spezielle Operatoren	111
3. 3. Operatoren für Zahlen, Zeichenketten und Mengen	118
3. 3. 1. Arithmetische Operatoren	119
3. 3. 2. Verkettungsoperator für Zeichenketten	121
3. 3. 3. Mengen-Operatoren	123
3. 4. Sortierung mit ORDER BY	129
3. 4. 1. Einfache Sortierung	130
3. 4. 2. Mehrstufige Sortierung	130
3. 5. Gruppierungen mit GROUP BY	132
3. 5. 1. Standard-Aggregatfunktionen	132
3. 5. 2. Einfache Gruppierung	134
3. 5. 3. Gruppen mit mehreren Spalten	136
3. 5. 4. Gruppierungen mit unterschiedlichen Aggregatfunktionen	137
4. Fortgeschrittene Abfragen	140
4. 1. Abfragen mit mehreren Tabellen	140
4. 1. 1. Prinzip der Abfrage über mehrere Tabellen	140
4. 1. 2. Manuelle Verknüpfungen	142
4. 1. 3. ANSI-SQL-Verknüpfungen	147
4. 2. Unterabfragen	163
4. 2. 1. Ersatz von Wertvorgaben	163
4. 2. 2. Unterabfragen in der Spaltenliste	168
4. 2. 3. Korrelierte Unterabfragen	175
4. 2. 4. Abgeleitete Tabellen	181
4. 2. 5. Prädikate bei Unterabfragen	183

4. 3. Fortgeschrittene Techniken in SQL	187
4. 3. 1. Fallunterscheidungen	188
4. 3. 2. Zugriff auf Pseudospalten	204
4. 3. 3. Hierarchische Untersuchungen	216
5. SQL Funktionen	222
5. 1. Zeichenkettenfunktionen	222
5. 1. 1. Konversionsfunktionen	223
5. 1. 2. Groß- und Kleinschreibung	228
5. 1. 3. Mustervergleiche und Chiffrierung	229
5. 1. 4. Bearbeitung von Leerzeichen	230
5. 1. 5. Andere Zeichenfunktionen	232
5. 2. Mathematische Funktionen	232
5. 2. 1. Winkelfunktionen	233
5. 2. 2. Rundungsfunktionen	233
5. 3. Datums- und Zeitfunktionen	236
5. 3. 1. Zeiterzeugung und Abruf von Zeiten und Daten	236
5. 3. 2. Zeitberechnung	237
5. 3. 3. Zeitformatierung	240
5. 3. 4. Systemfunktionen für Zeit und globale Informationen	243
5. 4. Aggregatfunktionen	247
5. 4. 1. Standardaggregate	247
5. 4. 2. Streuungsmaße	248
5. 4. 3. Quantitativer Zusammenhang: Korrelationsmaße	250
5. 4. 4. Funktionaler Zusammenhang: Lineare Regressionsmaße	251
6. Abfragen und Analysen	256
6. 1. Erweiterte Gruppierungen	256
6. 1. 1. Sinn und Zweck von erweiterten Gruppierungen	257
6. 1. 2. GROUPING SETS	260
6. 1. 3. ROLLUP	269
6. 1. 4. CUBE	271
6. 1. 5. GROUPING-Funktionen	273
6. 2. Rangordnungen erstellen	279
6. 2. 1. Rangfolgen einrichten	279
6. 2. 2. Hitparaden erstellen	282
6. 2. 3. Rangfolgen mit Verteilungen bestimmen	282
6. 2. 4. Quantilsbestimmung	286
6. 2. 5. Histogrammerzeugung	288

6. 2. 6. Individuelle Reihennummern für Datensätze	290
6. 3. Statistische Analysen	291
6. 3. 1. Bereichsaggregation	291
6. 3. 2. Zentrierter Durchschnitt	296
6. 3. 3. Zusammenfassende Gruppenwerte	297
6. 3. 4. Erste und letzte Werte einer Untergruppe	304
6. 3. 5. Versetzte Zeilendaten ohne Selbstverknüpfung	307
6. 3. 6. Zuordnung in Häufigkeitsverteilungen	310
6. 3. 7. Lineare Regression	314
6. 3. 8. Hypothetische Verteilungsrechnung	316
6. 3. 9. Chi-Quadrat-Test	318
6. 4. Erweiterte Abfragetechniken	320
6. 4. 1. Allgemeine Tabellenausdrücke (CTE)	320
6. 4. 2. Pivotierung und Unpivotierung von Ergebnissen	323
6. 5. Einfache Berichte mit SQL*Plus	338
6. 5. 1. Einfache Berichte	338
6. 5. 2. Gruppierungen und Aggregate	349
6. 5. 3. Ausgabevarianten und Berichtformate	359
7. DB-Objekte	379
7. 1. Tabellen	379
7. 1. 1. Allgemeine Syntax	379
7. 1. 2. Schlüssel und Fremdschlüssel	384
7. 1. 3. Tabellen ändern	388
7. 1. 4. Tabellen aus Abfragen	391
7. 2. Tabellenbezogene Objekte	392
7. 2. 1. Indizes	392
7. 2. 2. Zähler/Sequenzen	394
7. 2. 3. Programmierbare Objekte und Trigger	396
7. 3. Sichten	396
7. 3. 1. Grundlagen	397
7. 3. 2. Beispiele für Sichten	399
8. Datenmanipulation	404
8. 1. Daten einfügen mit INSERT	404
8. 1. 1. Standardfall und allgemeine Syntax	404
8. 1. 2. Einfügen aus Abfrage	408
8. 1. 3. Einfügen in mehrere Tabellen	410

8. 1. 4. Fehler	412
8. 1. 5. Rückgabewerte	416
8. 2. Daten aktualisieren mit UPDATE	417
8. 2. 1. Standardfall und allgemeine Syntax	417
8. 2. 2. Aktualisierung auf Basis anderer Tabellendaten	421
8. 2. 3. Verwendung von Unterabfragen	422
8. 2. 4. Fehler	423
8. 2. 5. Rückgabe von Daten	426
8. 3. Daten löschen mit DELETE	427
8. 3. 1. Standardfall und allgemeine Syntax	428
8. 3. 2. Einsatz von Unterabfragen	429
8. 3. 3. Fehler	430
8. 3. 4. Rückgabe von Daten	431
8. 4. Transaktionen in DML-Operationen	432
8. 4. 1. Grundlagen	432
8. 4. 2. Anweisungen für Transaktionssteuerung	434
8. 4. 3. Sicherungspunkte	436
8. 4. 4. Komplexe Transaktionssteuerung	437

Vorwort



Vorwort

Herzlich willkommen zu einem Fachbuch aus dem Verlag Comelio Medien. Dieses Buch aus dem Bereich Datenbanken soll Sie dabei unterstützen, die Oracle SQL zu lernen, um DB-Objekte zu erstellen und zu pflegen, aber auch Daten zu erfassen, zu aktualisieren und zu löschen wie insbesondere auch Abfragen auf diesen Daten durchzuführen. 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 der Abfragesprache SQL und damit auch an Neulinge im Bereich Oracle. Hilfreich kann es sein, wenn Sie bereits mit MS Access oder vielleicht auch intensiv mit MS Excel gearbeitet haben, aber notwendig ist dies nicht. Erfahrungen mit anderen Programmiersprachen sind nicht notwendig. Da das Buch sehr beispielorientiert aufgebaut ist und sich SQL auch sehr gut als erste Syntax („Programmiersprache“ wäre nicht ganz der richtige Ausdruck) eignet, ist ein Einstieg in die allgemeine Softwareentwicklung sehr gut möglich. Darüber hinaus setzt natürlich Oracle SQL auf den SQL-Standard auf, sodass sehr viele Techniken auch in anderen Datenbanken genutzt werden können und Sie so ein ganz allgemeines Wissen erwerben, das Sie sehr lange Zeit und in verschiedenen Situationen nutzen können.

Schreibkonventionen

Das Thema und damit auch dieses 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: <http://www.comelio-medien.com/>

Kontaktmöglichkeiten

- Die Autoren erreichen Sie an ihrem Arbeitsplatz unter info@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 Dokumentation 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/oracle_11g_für_Oracle-Seminare)
 - [http://www.comelio-seminare.com/seminare/titel/xml für XML-Seminare](http://www.comelio-seminare.com/seminare/titel/xml_für_XML-Seminare)
- Weitere Bücher zur Oracle-Technologie im gleichen Verlag:
 - *Oracle PL/SQL* - ISBN 978-3-939701-40-8
 - *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/kurz-referenzen>
 - 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, 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 wesentliche Konzepte von relationalen Datenbanken und des relationalen Datenmodells. Es beschreibt, wie eine Datenbank aufgebaut ist und wie man selbst von einer Beschreibung des abzubildenden Sachverhalts in der semantischen Analyse über die Technik der Normalisierung zu einer geeigneten Tabellenstruktur kommt.
3. Das dritte Kapitel stellt das erste von drei Kapiteln dar, welche den zentralen Teil des Buchs bilden: Abfragen und Analysen. Sie lernen, mithilfe der `SELECT`-Anweisung Daten aus Tabellen abzurufen, indem Sie die interessierenden Spalten sowie Eigenschaften von Filtern, der Sortierreihenfolge oder der Gruppierung vorgeben.
4. Das vierte Kapitel zeigt, wie Sie komplexe Abfragen mit mehreren Tabellen formulieren. Sie sehen zunächst, welche unterschiedlichen Möglichkeiten es gibt, Tabellen zu verknüpfen. Danach lernen Sie die verschiedenen Arten von Unterabfragen kennen, mit denen man komplexe Bedingungen und dynamische Filter formulieren sowie auch übersichtliche Abfragestrukturen erstellen kann.
5. Das fünfte Kapitel listet mit Beispielen viele nützliche Funktionen auf, die man in SQL verwenden kann. Dazu zählen Funktionen für die Manipulation von Zeichenketten, die Durchführung mathematischer und statistischer Operationen, den Umgang mit Zeit und Datum oder auch die Behandlung von besonderen Datentypen.

6. Das sechste Kapitel fokussiert die Formulierung von Analysen, statistischen Auswertungen und die Gestaltung von sehr einfachen Berichten. Zu den vermittelten Kenntnissen gehören die Erstellung von Abfragen mit Kumulationsrechnungen, gleitenden Aggregaten oder auch die Pivot-Darstellung von Ergebnissen. Die Berichte werden über SQL*Plus erstellt und können zu Ergebnissen in CSV-/MS Excel-Dateien oder HTML-Dokumenten führen.
7. Das siebte Kapitel setzt nicht voraus, dass schon Daten in der Datenbank enthalten sind, sondern zeigt die SQL-Anweisungen für das Einfügen (`INSERT`), Aktualisieren (`UPDATE`) und Löschen (`DELETE`) von Daten. Jeweils sieht man die typischen Standardfälle, welche mit den einzelnen Befehlen umgesetzt werden können, und auch die Protokollierung von Fehlern und die Verwendung von Unterabfragen. Darüber hinaus geht es auf die Einrichtung und Verwendung von Transaktionen ein, welche die sichere Verkettung von inhaltlich zusammenhängenden DB-Operationen ermöglichen.
8. Das achte Kapitel zeigt Ihnen, wie Sie mit SQL als Alternative zur grafischen Oberfläche in Programmen wie SQL Developer zur DB-Verwaltung Objekte in der Datenbank anlegen können. Sie lernen also, mit SQL-Befehlen Tabellen anzulegen, Spalteneigenschaften zu definieren, Primärschlüssel und Fremdschlüssel vorzugeben oder Änderungen an diesen Objekten durchzuführen. Dann folgt, wie man Index-/Sequenz-Objekte und auch Sichten erstellen kann.

Autoren

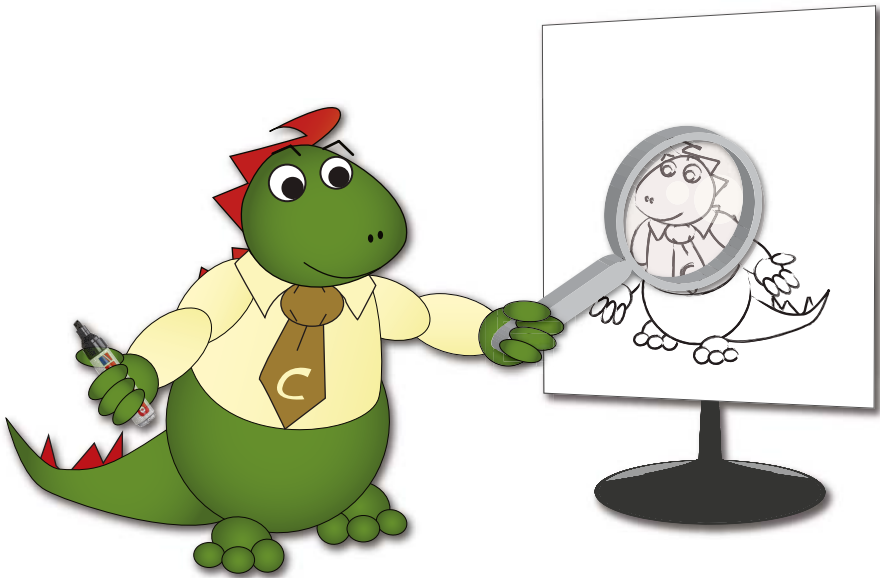
Marco Skulschus

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. Dabei gehen wir davon aus, dass Sie nicht nur in SQL bzw. in Oracle SQL neu sind, sondern auch noch kein Beispiel-System für Tests und Übungen eingerichtet haben.

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 ein-

geschränktem 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 oder Dozenten kennenlernen.

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.

Die Anmeldungen und damit die umsatzrelevanten Tätigkeiten müssen 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.

Es ist für die Durchführung eines gebuchten Seminars 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 Wei-

terentwicklung 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 individuell gebucht. Sie haben vielmehr 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.

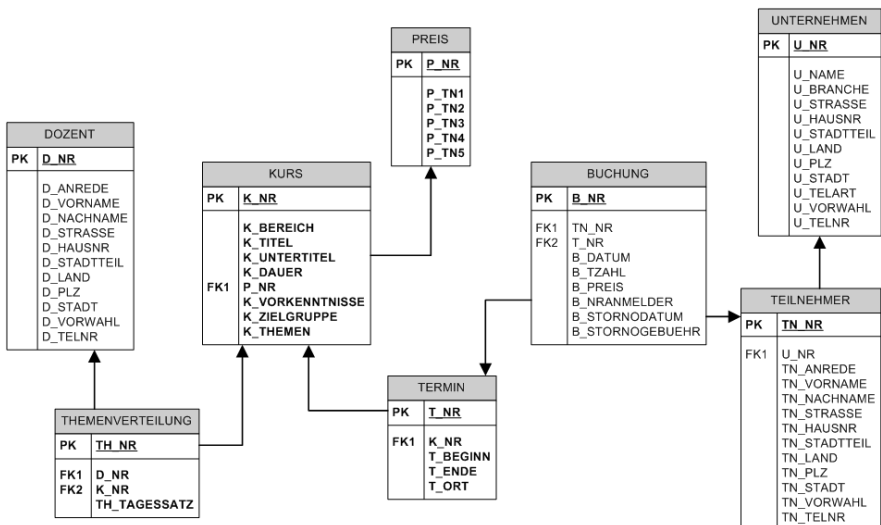


Abbildung 1.1: Das Datenmodell von Kurse NRW

Nach diesen Überlegungen kommt man zu folgendem Modell, das für das gesamte Buch als Standardmodell bzw. als einfachste Ausgangsbasis dienen wird. Es be-

sitzt 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. 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 Unternehmensnummer besitzen und sich selbst oder Kollegen für einen Kurs angemeldet haben. Dies ist notwendig, weil es relativ viele Arbeitnehmer gibt, die ihre Seminare selbst ausuchen 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

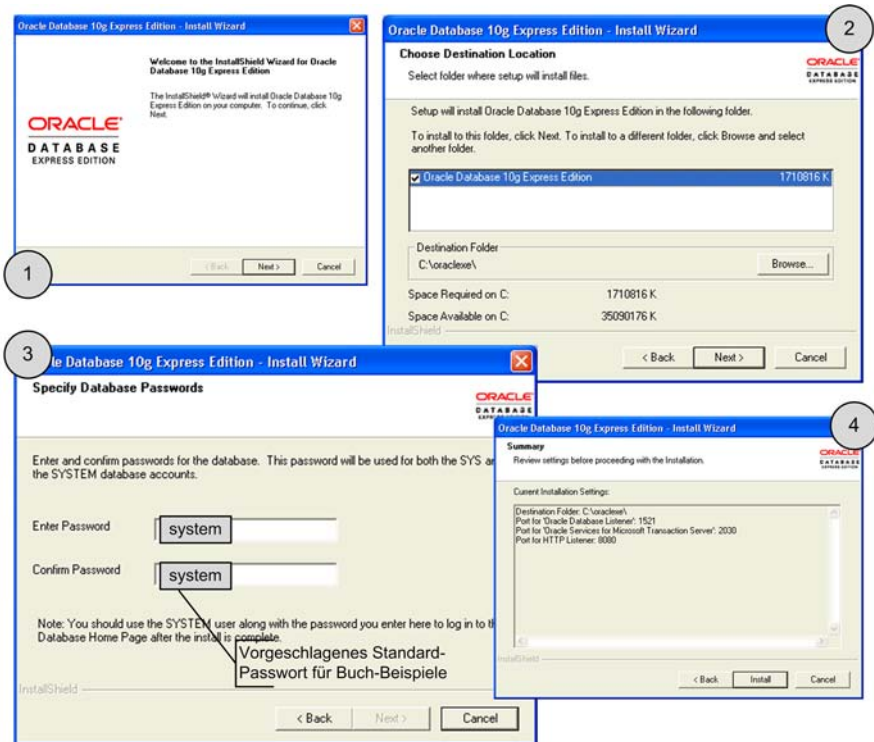
Sie können alle Beispiele aus diesem Buch leicht nachvollziehen, da in den Download-Dateien sowohl die benutzte Beispiel-Datenbank wie auch sämtliche Skripte zur Verfügung stehen. Lediglich die Oracle-Datenbank selbst müssen Sie von der Oracle-Webseite herunterladen. Die hier verwendeten Express-Editionen haben den Vorteil, dass ihre Installation bzw. ihr Betrieb deutlich 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>

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

Viele Teilnehmer werden Oracle 10g nutzen und noch nicht die 11g-Variante und möchten vermutlich auch bei der Express-Variante das jeweils passende System installieren, sodass wir hier beide Installationen zeigen.

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*.

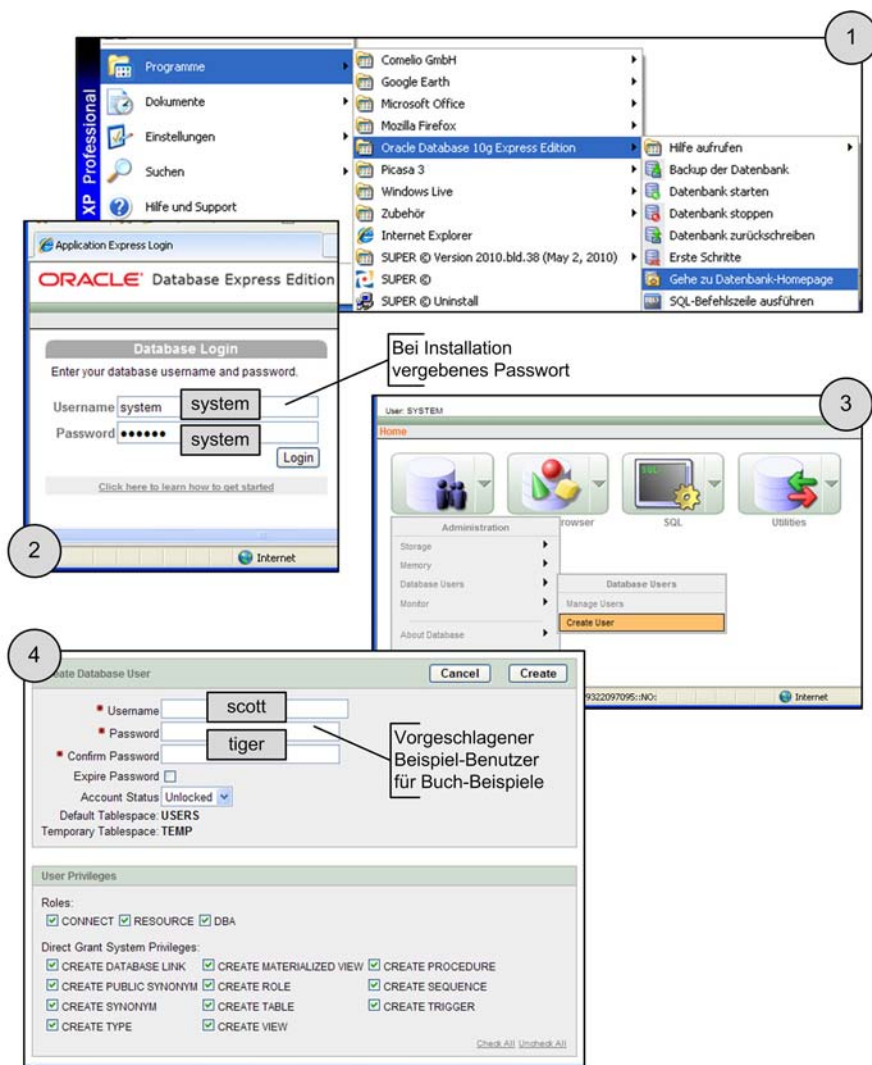


Installation Oracle 10g Express

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-HOME PAGE**.



Einrichten eines Benutzers auf der DB-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.

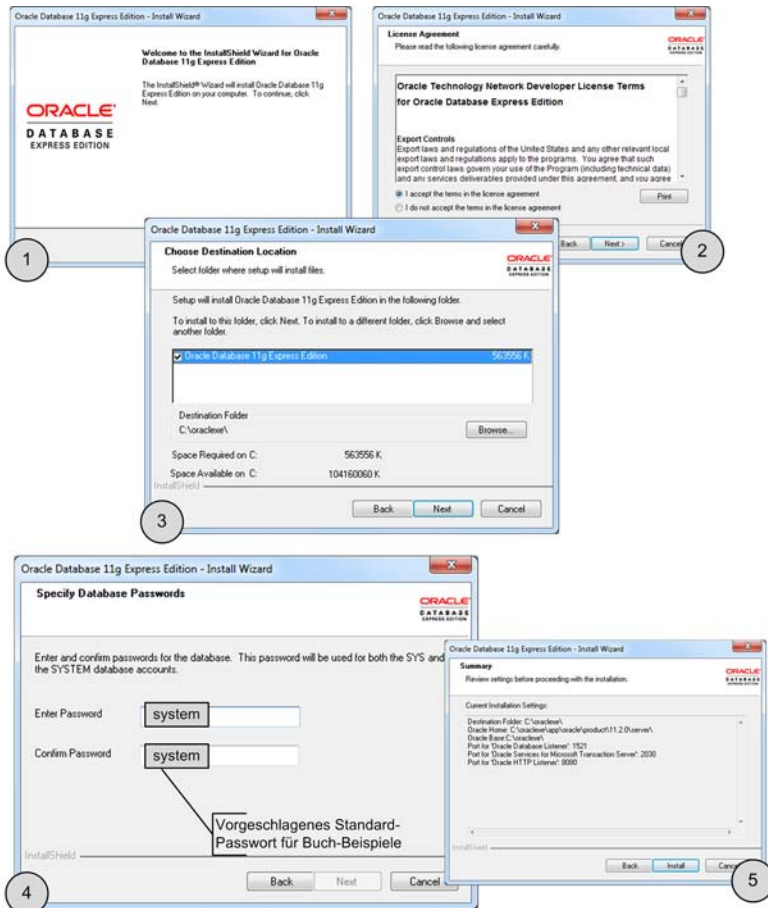
3. Wählen Sie *ADMINISTRATION / DATABASE USERS / CREATE USER*, um einen neuen Benutzer zu erstellen.
4. Legen Sie einen neuen Benutzer an. Die Buch-Beispiele verwenden einen Benutzer namens `SCOTT` mit dem Passwort `TIGER`, der vor langer Zeit in Oracle als Standardbenutzer vorhanden war. Sie können 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. 2. Oracle Database 11g Express Edition

Auch von der der 11g-Ausgabe der Datenbank gibt es eine Express-Version.

Diese ist sicherlich geeigneter, wenn man sich den neuesten Technologien beschäftigen möchte, die über SQL und PL/SQL genutzt werden können.

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*.



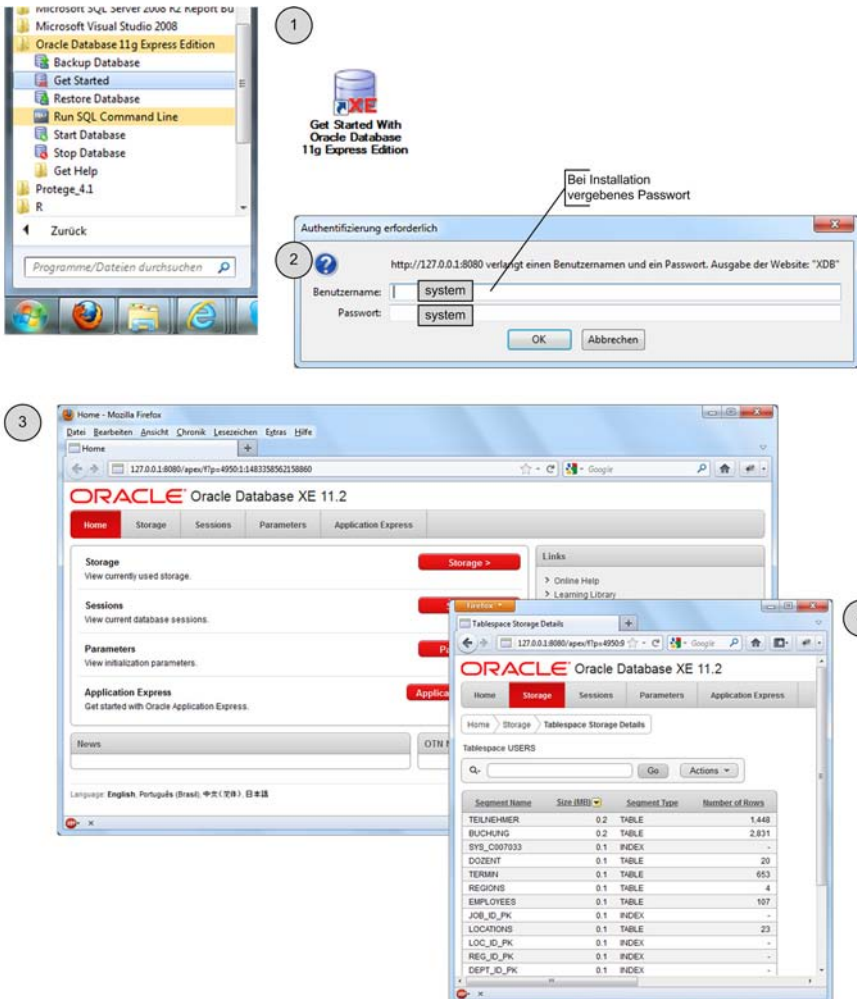
Installation Oracle 11g Express

Über die Datenbank-Homepage kann man bei der Version 11g nicht so umfassend mit der Datenbank umgehen wie bei der Vorgänger-Version. Daher zeigen wir im Folgenden nur die wichtigsten Schritte, um sie zu starten und bspw. vorhandene Tabellen und ihre Befüllung zu sehen.

1. Wählen Sie **START / PROGRAMME / ORACLE 11G DATABASE EXPRESS EDITION / GET STARTED**. Alternativ finden Sie eine Verknüpfung auf dem Desktop.

1. Einführung

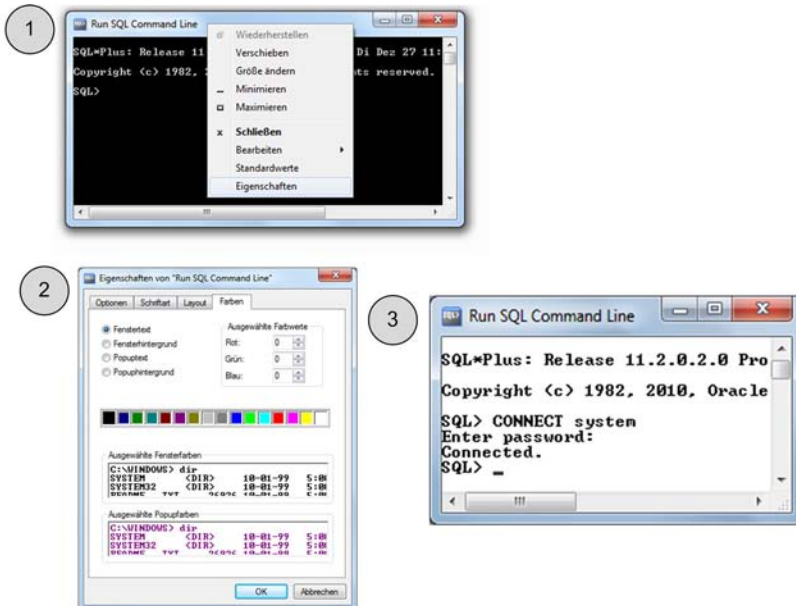
2. Dies öffnet in Ihrem Internet-Browser die Konfigurationswebseite, mit der Sie grafisch die Datenbank administrieren können.
3. Melden Sie sich mit dem `SYSTEM`-Konto und Ihrem bei der Installation vergebenem Passwort an.



Datenbank-Homepage für allgemeine System-Einstellungen

Um einen Benutzer anzulegen, verwenden wir dagegen SQL*Plus, ein Konsolenprogramm, mit dem man – allerdings ohne großen grafischen Komfort – Befehle zur Datenbank senden kann.

1. Wählen Sie *START / PROGRAMME / ORACLE 11G DATABASE EXPRESS EDITION / RUN SQL COMMAND LINE*.
2. Die Hintergrundfarbe dieses Programms ist schwarz, während die Schriftfarbe weiß ist. Sie können dies ändern, indem Sie über die rechte Maustaste das Kontextmenü vom oberen Bildrand öffnen. Hier finden Sie *EINSTELLUNGEN* für die Größe und Farbe des Programms sowie auch Befehlen zum Kopieren, Markieren und Einfügen von SQL-Befehlen.



Verwendung von SQL*Plus

3. Melden Sie sich mit dem `SYSTEM`-Konto und Ihrem bei der Installation vergebenem Passwort an, indem Sie `CONNECT system` eingeben und die Eingabetaste drücken. Danach werden Sie nach dem bei der Installation vergebenem Passwort gefragt.

4. Erstellen Sie mit dem nachfolgenden Skript den Benutzer `SCOTT`, der sich über das Passwort `tiger` anmelden kann und ausreichend Berechtigungen zum Arbeiten erhält.

```
CREATE USER scott IDENTIFIED BY tiger;  
GRANT CONNECT, RESOURCE TO scott;  
GRANT DBA TO scott;
```

121_01.sql: Benutzer anlegen

1. 2. 3. Entwicklungsumgebung

Neben der Client-Software SQL*Plus oder der Web-Ansicht über die Datenbank-Homepage (Version 10g) gibt es auch noch zwei weitere Werkzeuge, die sehr leicht zu installieren und kostenlos verfügbar sind.

➔ 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:

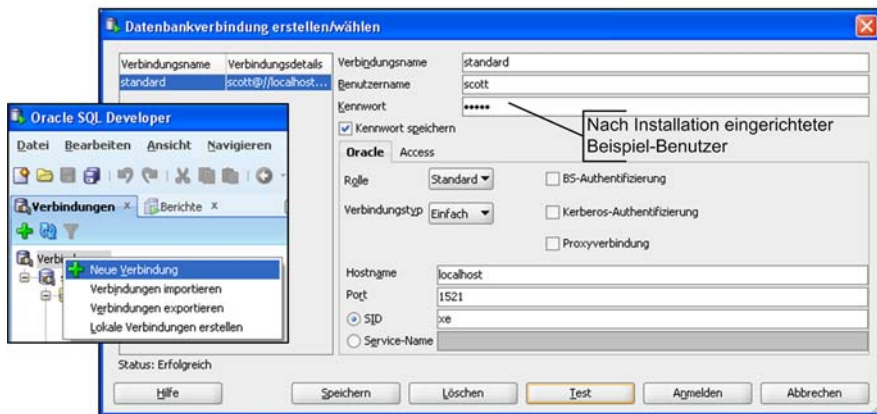
<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

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.

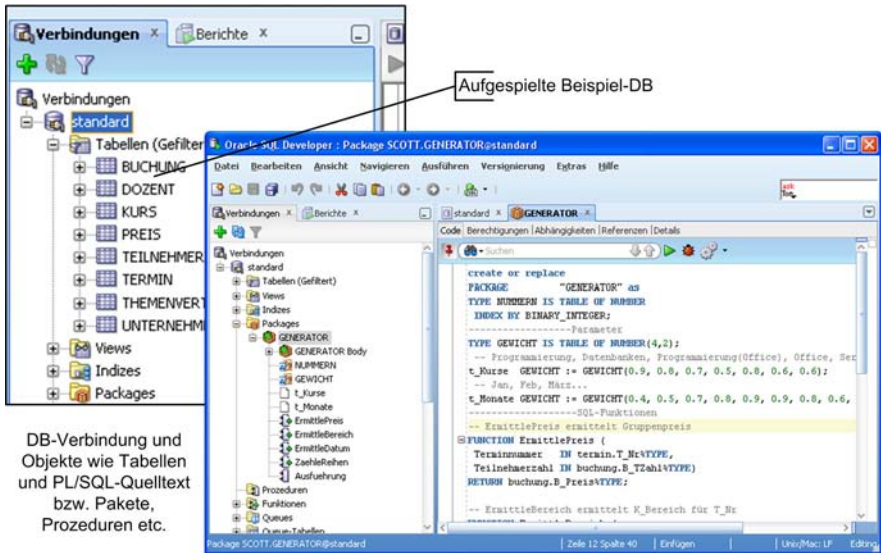
Entpacken Sie das ZIP-Archiv in einem Ordner wie bspw. bei all Ihren anderen Programmen und starten Sie dann *sqldeveloper.exe*.

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-Objekte anzeigen und bearbeiten

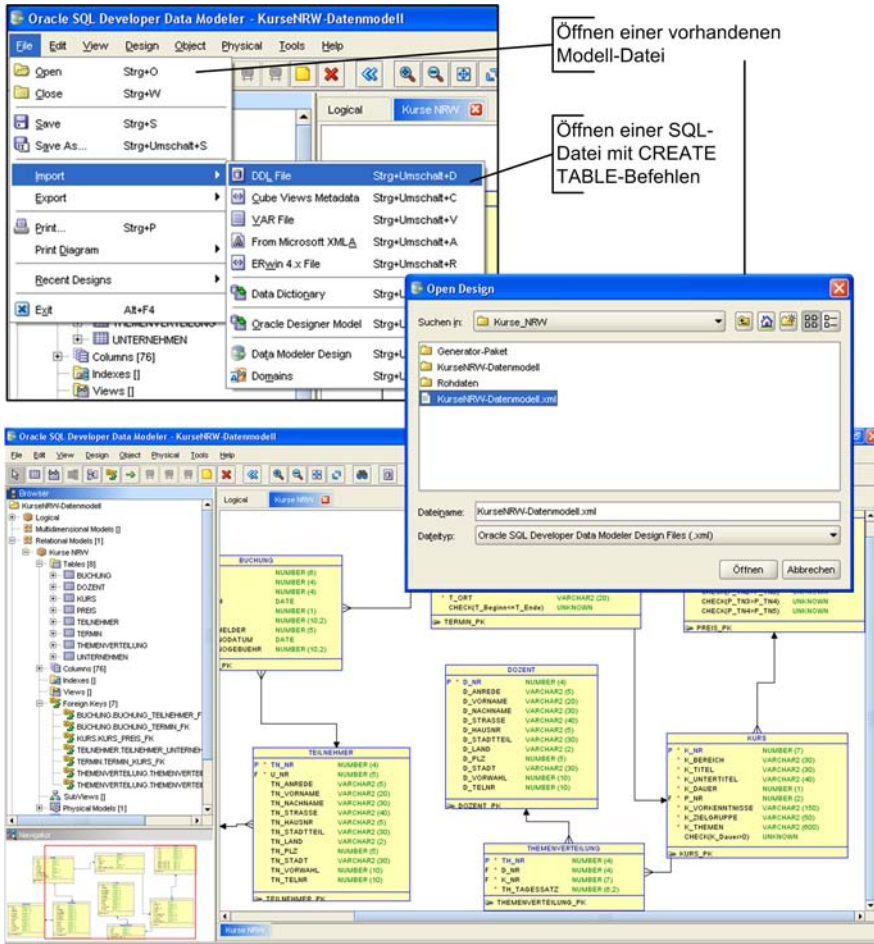
➔ Oracle SQL Developer Data Modeler

Ähnlich verhält es sich mit dem Modellierungswerkzeug. Auch hier enthält die größere der herunterladbaren ZIP-Archive bereits die Java-Laufzeitumgebung, sodass keine eigentliche Installation notwendig ist. Entpacken Sie lediglich das Archiv in einen Ordner Ihrer Wahl wie bspw. in den Ordner für die Programme. Starten Sie dann *datamodeler.exe*.

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.



Oracle SQL Developer Data Modeler

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.

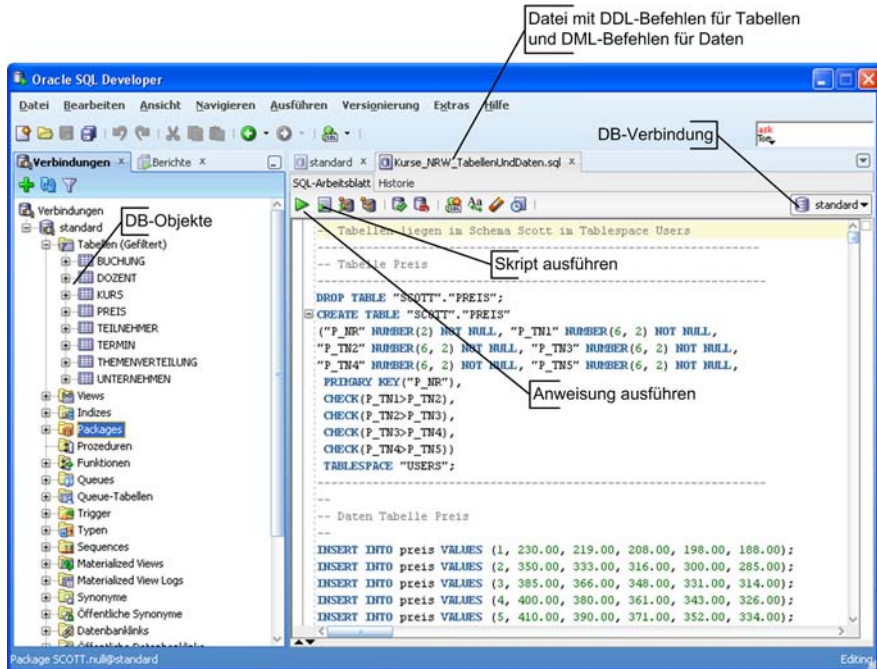
Der Ordner *Rohdaten* enthält die folgenden Dateien:

- *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.

Im Ordner *Generator-Paket* befindet sich ein PL/SQL-Paket, das verschiedene Prozeduren und Funktionen enthält, mit denen man die Tabelle `BUCHUNG` auffüllen kann. Diese Daten wurden mithilfe eines Zufallsgenerators erzeugt. Die Zufallszahlenfolge ist dabei zwar zufällig, wird aber immer wieder gleich generiert. Für die Nutzung des Pakets müssen erst die Spezifikation und dann der Körper des Pakets in die Datenbank eingefügt werden. Dazu liegen zwei Skripte vor, die nur ausgeführt werden müssen.

Die Ausführung selbst ist dann mit einer Prozedur möglich, die in einem anonymen Block in der Datei *Generator-Paket-Ausfuehrung.sql* genutzt wird. Es genügt auch hier, nur dieses Skript auszuführen, um den Generator zu starten. Sie müssen sich mit diesen Paket nicht beschäftigen, wenn Sie die Datenbank über die Datei *Kurse_NRW_TabellenUndDaten.sql* aufspielen. Dies wird im nächsten Abschnitt beschrieben.

1.3.2. Tabellen und Daten aufspielen



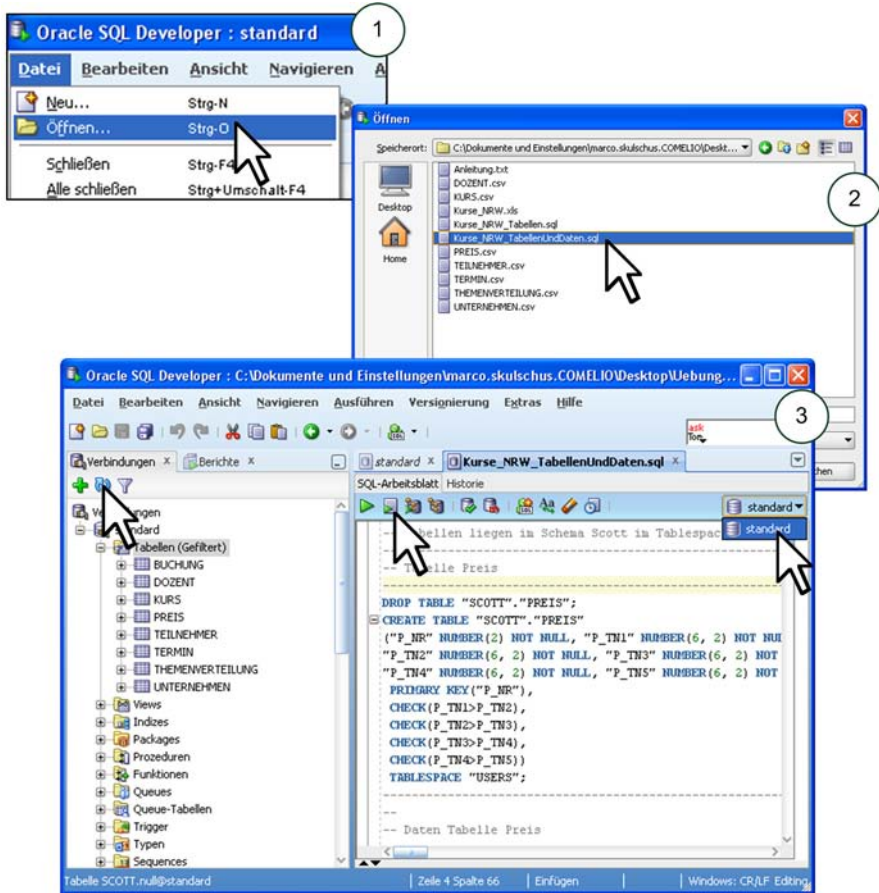
Oracle SQL Developer – Übersicht

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

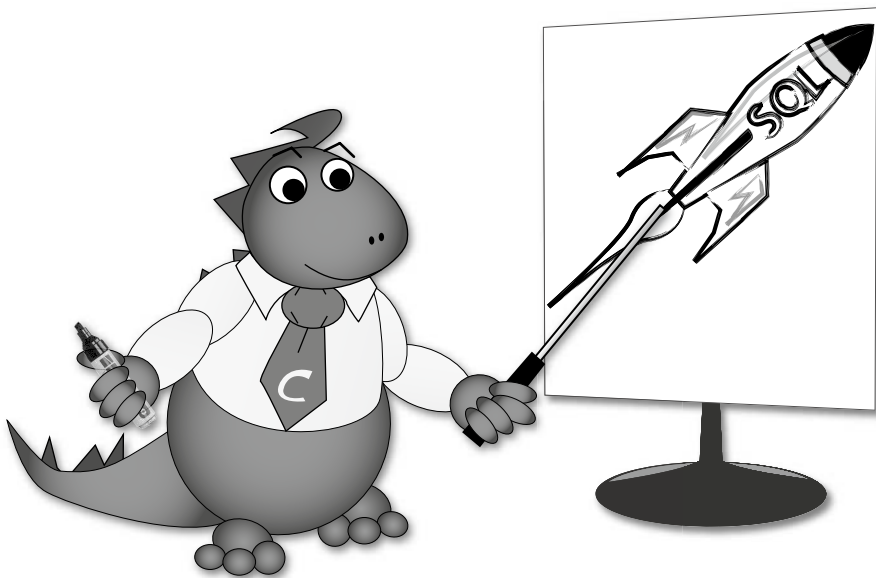
zuvor noch mit *BEARBEITEN* / *ERSETZEN* die Zeichenkette "SCOTT" . aus der gesamten Datei löschen.

5. Aktualisieren Sie dann die Tabellenaufstellung, 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 den Daten.



Installation der Beispiel-DB

SQL und relationale Datenbanken



2. SQL und relationale Datenbanken

Dieses Kapitel gibt einen kurzen Überblick über die Ursprünge von SQL und sein theoretisches Fundament außerhalb der konkreten Syntax. Wir starten mit einer Darstellung über das relationale Datenbank-System und seine Komponenten, gehen dann über zum relationalen Modell und einer Zusammenstellung der wichtigsten Konzepte in diesem Bereich und kommen dann schließlich zu einer Vorstellung von SQL als Sprache, um mit relationalen Datenbanken zu arbeiten.

2. 1. Das relationale Datenbank-System

Dieser Abschnitt stellt die wesentlichen Begriffe rund um den Themenbereich des Relationalen Datenbanksystems (RDBS) dar. Neben einer umfassenden Algorithmen-Lösung für eine effiziente Speicherung und Abfrage von Daten sowie einem Transaktionsmanagement bei konkurrierendem Mehrbenutzerzugriff stellt eine relationale Datenbank eine konkrete Anwendung des relationalen Modells dar. Dieses Modell wird allerdings erst das Thema des nächsten Abschnitts sein. Hier geht es nun zunächst darum, die verschiedenen Komponenten, aus denen eine relationale Datenbank zusammengesetzt ist, vorzustellen.

2. 1. 1. Zentrale Begriffe

Im Alltag verschwimmen oft die Begriffe *Datenbanksystem* und *Datenbank* zu einer Einheit, die sie aber aus technischer Sicht nicht sind. Der trügerische Eindruck entsteht, weil die Daten innerhalb der eigentlichen Datenbank oftmals direkt verbunden mit ihrem Ausgabemedium gesehen werden. Im Internet ist es vielleicht noch am einfachsten, die klare Trennung von Datenbank und Ausgabemedium festzumachen: Ergebnisse einer Abfrage bspw. in einem Webshop gelangen im Browser innerhalb einer HTML-Seite zum Benutzer. Sobald die ausgegebenen Bücher, Preise, ISBN-Nummern und was sonst noch bei einem Internet-Buchhändler abgefragt werden kann, vom Webserver in eine HTML-Datei geschrieben werden,

dient die HTML-Seite bzw. der Internet-Browser als Ausgabemedium, während die Datenbank von dieser Ausgabe losgelöst ist und auch keine direkte Verbindung mehr zu ihr besteht. Diese müsste erst wieder durch eine erneute Abfrage aufgenommen werden.

Das Datenbanksystem besteht aus zwei Komponenten, von denen die eine die Datenbank mit den Daten selbst und die andere das Datenbankmanagementsystem ist, das die Datenbank verwaltet. Verwaltung bedeutet im obigen Beispiel, dass das DBMS die Anfrage nach einem speziellen Buch oder nach beliebig vielen Büchern, die die vom Benutzer vorgegebenen Kriterien erfüllen, an die Datenbank schickt und die zurückgegebenen Daten über den Webserver an den Shopbesucher weiterleitet. Gleichzeitig erfüllt es eine Sicherungsfunktion, da über die Schnittstelle Browser dem Benutzer nur bestimmte Aktionen innerhalb des DBS gestattet werden. Er kann so bspw. nur Daten abfragen und Daten in Form einer Bestellung schreiben. Es wird ihm darüber hinaus vollständig verwehrt, Datensätze zu löschen, welche die angebotenen Bücher betreffen.

- *Datenbanksystem* heißt eine Struktur, die aus Datenbank und Datenbankmanagementsystem besteht.
Der korrekte Oberbegriff für die Datenbank mit ihren Daten und das Datenbankmanagementsystem mit Verwaltungsfunktionen lautet *Datenbanksystem*.
- *Datenbank* heißt der Speicherort strukturierter Daten.
Die *Datenbank* (engl. *Data Base*) als solche stellt den Speicherort sämtlicher Daten auf einem physikalischen Datenträger wie CD-ROM, DVD, Festplatte usw. dar. In Abhängigkeit vom jeweiligen Dateisystem und Datenbankmodell werden diese Daten verwaltet. Zugriffszeit und Speicherplatz wiederum sind hauptsächlich abhängig von der Hardware des Speicherträgers und der Hardware der Zugriffsgeräte (z.B. unterschiedliche Geschwindigkeiten bei Festplatten und CD-ROM-Laufwerken). Eine weitere Einflussquelle stellt auch das Datenmodell selbst dar. Auf dieser Ebene unterscheidet man auch einen Basisbereich, der die tatsächlichen Daten enthält, und einen Systemkatalog (engl. *Data Dictionary*) mit Informationen über die logische Struktur der gesamten Datenbank. Diese Informationen können auch mit dem Begriff der *Metadaten* (Daten über Daten) bezeichnet werden.
- *Datenbankmanagementsystem* heißt die Komponente eines DBS, das Datenverwaltung und Datensicherheit bereitstellt.

Der zweite Bestandteil eines Datenbanksystems ist das *Datenbankmanagementsystem* (engl. *Data Base Management System*), das die Schnittstelle zwischen der Datenbank und der Kommunikationsschnittstelle zum Benutzer bildet. *MySQL* ist beispielsweise eine solches DBMS mit vielen verschiedenen Funktionen für die Einrichtung und Benutzung der Datenbank: Datenzugriff, Gewährleistung von Datensicherheit, Datenmanipulation, Datenverwaltung und Einrichtung des Datenmodells.

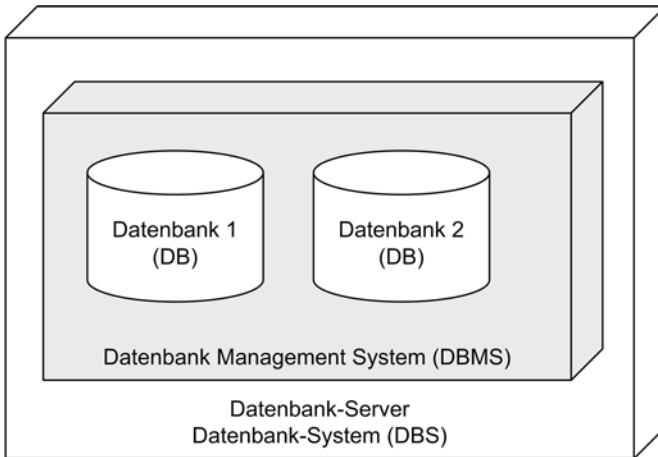


Abbildung 2.1: Zentrale Begriffe und ihre Zusammenhänge

2. 1. 2. Sichten auf ein relationales Datenbanksystem

Mit dem Begriff *Sicht* (engl. *View*) sollen unterschiedliche Betrachtungsweisen einer Datenbankkonzeption bezeichnet werden. Es entspricht einem Drei-Ebenen-Modell, das von *ANSI/SPARC* (engl. *American National Standards Institute/Standards Planning and Requirements Committee*) in den 70er Jahren empfohlen wurde. Diese Konzeption diente der Vereinheitlichung und Vereinfachung aktueller und künftiger Datenbanksysteme. Man hatte bereits die Bedeutung der Fähigkeit eines DBS erkannt, sich weiterzuentwickeln und den Datenbestand auch in neuen Technologien weiterhin benutzbar zu machen. Um kompatibel zu allen Systemen und migrationsfähig für alle kommenden Technologien zu bleiben, bildete die Forderung nach Unabhängigkeit der drei Ebenen untereinander das Kernelement der

Empfehlung. Die Arbeit, die Sie sich heute machen, müssten Sie also zumindest laut dieser Theorie noch in 20-30 Jahren verwenden können.

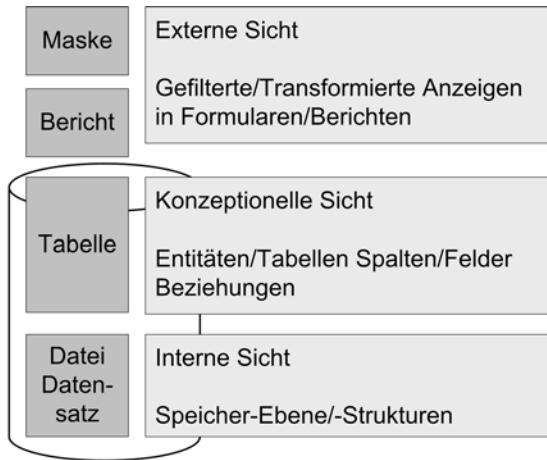


Abbildung 2.2: Sichten auf ein relationales Datenbanksystem

Die einzelnen Begriffe sind dabei durchaus wörtlich zu nehmen, wie die folgenden Erklärungen zeigen:

- **Interne Sicht:**
Direkt auf Datenbankebene beschreibt diese Sichtweise den Blick auf die Datenorganisation und die Art und Weise, wie die Datensätze auf einem geeigneten Speichermedium abgelegt werden. Mögliche Aufgaben aus dieser Perspektive sind die Zugriffszeiten, die vorhandene Datenmenge und Eigenschaften des Datenmodells.
- **Konzeptionelle Sicht:**
Schaut man aus dieser Perspektive auf eine Datenbank, tritt das Datenmodell zu Tage, also die Objekte im Weltausschnitt, der mithilfe der Datenbank modelliert werden sollte. Die in der Realität gefundenen Objekte erhalten in der Konzeption bzw. im Datenmodell bestimmte Eigenschaften, die mit den Eigenschaften in der Wirklichkeit übereinstimmen. Ein Unternehmen würde hier also für das Objekt Mitarbeiter die Eigenschaften Sozialversicherungsnummer, Alter, Adresse und natürlich den Namen oder die Fertigkeiten er-

fassen. Zusätzlich kümmert sich die konzeptionelle Sicht darum, Beziehungen zwischen diesen Objekten einzurichten, die schnelle und einfache Abfragen erlauben.

- *Externe Sicht:*

Benutzer können von außen mithilfe von vorbereiteten oder eigenen Abfragen Datensätze suchen und ändern. Dabei kann man entweder direkt SQL-Befehle benutzen oder verschiedene Hilfsmittel (Stichwörter, Auswahllisten, Berichte oder Profisuche). Diese Möglichkeiten werden Ihnen vom DB-Administrator oder DB-Entwickler zur Verfügung gestellt. Sie regeln gleichzeitig auch die Art und Weise der verschiedenen Möglichkeiten, eine DB von außen zu betrachten. Nicht jeder darf alle Daten sehen, da Informationen wie Umsatzzahlen zwar hochinteressant, aber nicht so sehr für Kunden bestimmt sind, z. B. die Produktinformationen.

2.1.3. Anforderungen an ein DBMS

Um wie in den nächsten Abschnitten ein gutes Datenmodell zu erzeugen, das nicht bereits nach den ersten Einträgen oder ersten umständlichen Abfragen geändert oder sogar gelöscht werden muss, kann man sich in jede einzelne Personengruppe hineinendenken, die eine DB benutzen wird. Ihre verschiedenen Motivationen und Interessen führen zu allgemeinen Wünschen, wie der Weltausschnitt in einer DB abgebildet sein sollte. Diese Wünsche lassen sich folgendermaßen in einer Kriterienliste zusammenfassen:

- *Gute Repräsentation der realen Welt:*

Die Datenbank muss ein getreues Abbild der Realität mit aktuellen Daten liefern.

- *Keine Datenredundanz:*

Die Informationen einer Datenbank sollen in ihrer Beschreibung und Speicherung einzigartig sein.

- *Unabhängigkeit der Daten mit Blick auf Benutzung:*

Die Programme zur Benutzung der Datenbank sollen auf Hard- und Software-Seite (physische Unabhängigkeit) wie auf organisatorischer Seite (logische Unabhängigkeit) unabhängig sein.

- **Leichter Datenzugriff:**
Auch mit Datenbanken nicht vertrauten Benutzern muss ein einfacher Informationszugriff möglich sein.
- **Sicherheit und Vertraulichkeit:**
Sowohl auf physischer (Datenverlust durch Hardwarebeschädigung oder Mutwillen) wie auch auf logischer Seite (inhaltliche Zugriffskontrolle) müssen Sicherheitsstandards bestehen.
- **Leistung der Abfragen und Anwendungsprogramme:**
Auch bei gleichzeitigem Zugriff auf die Daten muss eine akzeptable Antwortzeit gegeben sein.

2. 1. 4. Bestandteile einer Tabelle

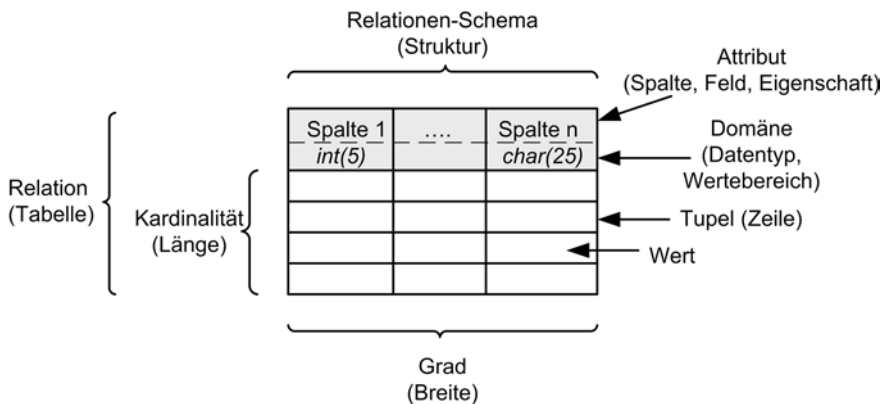


Abbildung 2.3: Übersicht über die Eigenschaften von Relation und Tabelle

Die einzelnen Bestandteile und Eigenschaften einer Tabelle lassen sich mit verschiedenen Begriffen beschreiben, die hier im Zusammenhang gezeigt werden sollen:

- Die gesamte Tabelle heißt *Relation*.
- Ein Datensatz der Relation / Tabelle nennt sich *Tupel*.