

John-Harry Wieken

MASTER  
CLASS

# SQL

» Einstieg für Anspruchsvolle

inkl. Lösungen und Lerntest,  
Beispiele



ADDISON-WESLEY

[ in Kooperation mit ]

PEARSON  
Studium

# 3

## Die Beispieldatenbanken

Bevor wir uns jetzt den ersten realen SQL-Befehlen im Detail zuwenden, wollen wir noch die beiden Grundlagen betrachten, die Sie für die Beispiele dieses Buches benötigen. Zum einen sind dies die Beispieldatenbanken, auf die sich die meisten SQL-Beispiele dieses Buches beziehen. Zum anderen soll die Installation der Datenbankmanagementsysteme MySQL, MS Access, openBase, Oracle und Firebird beschrieben werden, die Sie für die Bearbeitung der Beispiele nutzen können. Wir werden in diesem Buch Beispiele für alle fünf Systeme zeigen, indem wir auch für alle Übungsaufgaben Lösungen für alle Systeme erstellen, soweit dies sinnvoll möglich ist. Die Lösungen finden Sie auf der beiliegenden CD.

Dieses Kapitel hat zwei Funktionen. Es soll Ihnen zum einen helfen, eine Spielumgebung für die Arbeit mit den Beispielen dieses Buches bereitzustellen. Es soll Ihnen zum anderen aber auch ein wenig Hintergrund zu den Beispielen und der Handhabung der Umgebungen liefern.

Für diejenigen, die sich nur für die Beispiele interessieren und die Datenbankinstallation bereits kennen, reicht es, die Abschnitte 3.1 und 3.2 zu lesen. Für die anderen Leser empfehlen sich auch die weiteren Kapitel, insbesondere diejenigen Abschnitte, die die Sie interessierenden Datenbanken beschreiben.

### 3.1 Die Kursdatenbank

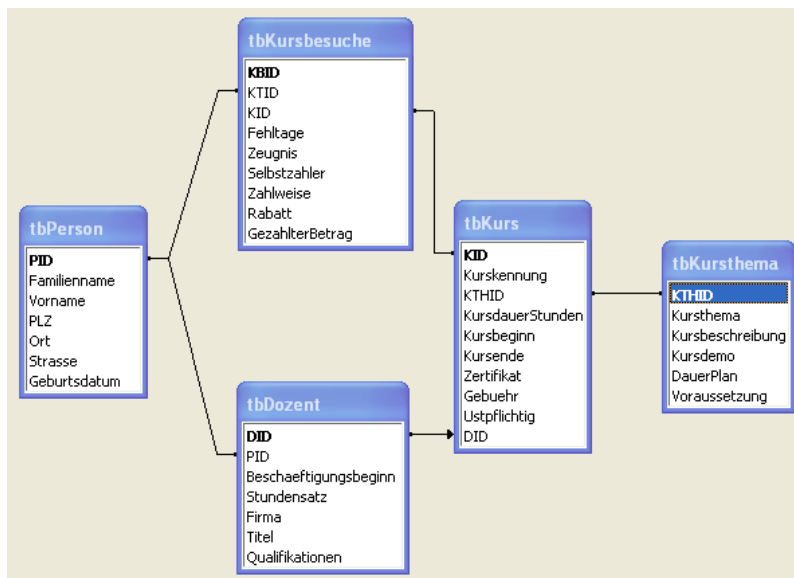
Die wichtigste Beispieldatenbank ist die sogenannte „Kursdatenbank“. Sie beschreibt die Sicht eines Schulungsunternehmens auf seine Kurse. Dabei steht hier der Beispielgedanke im Vordergrund, viele Details sind daher nicht enthalten, die in der Praxis sicherlich eine zentrale Rolle spielen. Das Ziel der Datenbank soll aber auch nicht die reale Abbildung des Schulungsunternehmens sein, sondern ein möglichst verständliches Beispiel für die SQL-Anweisungen.

**Kursdatenbank**

**Datenmodell  
Kursdatenbank**

Zunächst wollen wir uns daher das Modell mit seinen Tabellen und Feldern sowie den möglichen Beziehungen ansehen.

**Abbildung 3.1**  
Datenbankschema  
der Kursdatenbank



Wie Sie in Abbildung 3.1 sehen, besteht die Datenbank aus fünf Tabellen. Kleine Ausschnitte der Daten in den Tabellen `tbPerson` und `tbDozent` haben wir in Kapitel 2 bereits betrachtet. Hier sehen Sie die kompletten Tabellen. Sie erkennen, dass im Datenbankschema die Datenfeldnamen vertikal in der Tabelle eingetragen sind. Werden die realen Daten gezeigt, wird die horizontale Darstellung verwendet, bei der die Datenfeldnamen die oberste Zeile bilden, da zumeist eine ganze Reihe von Datensätzen darunter angegeben werden sollen. Die beiden Tabellen sind mit einer Linie verbunden, die die Beziehung zwischen den Tabellen symbolisiert. Eine Linie verbindet beispielsweise das Primärschlüsselfeld `PID` der Tabelle `tbPerson` mit dem Fremdschlüsselfeld `PID` der Tabelle `tbDozent`. Dies bedeutet, dass inhaltlich die Dozenten mit einer bestimmten `PID` den Personen mit derselben `PID` entsprechen. Auf Details gehen wir später ein. Die Darstellung in Abbildung 3.1 ist MS Access entnommen, ähnliche Darstellungen werden aber von vielen Datenbanksystemen oder entsprechenden Hilfswerkzeugen angeboten.

**tbPerson**

Die Tabelle `tbPerson` beinhaltet die Stammdaten aller Personen, die mit dem Schulungsunternehmen zu tun haben, sei es als Teilnehmer oder Dozent. Die Verwaltungsangestellten und andere Mitarbeiter lassen sich hier natürlich ergänzen, sollen aber im Sinne einer besseren Übersichtlichkeit zunächst nicht betrachtet werden. Für jede Person sind außer ihrem Familiennamen und Vornamen noch die Postleitzahl, der Ort und die Straße sowie ihr Geburtsdatum hinterlegt.

Die Tabelle `tbDozent` beinhaltet zusätzliche Felder für alle Personen, die in dem Unternehmen als Dozenten tätig sind. So wird festgehalten, seit wann sie als Dozent beschäftigt sind, wie hoch ihr Stundensatz ist und welche Titel und Qualifikationen sie mitbringen. Da viele der Dozenten nebenberuflich tätig sind, wird außerdem der Name der Firma gespeichert, in der sie hauptberuflich beschäftigt sind. Über die durch eine Linie symbolisierte Beziehung zur Tabelle `tbPerson` kann ermittelt werden, welche Stammdaten, also Adresse und Geburtsdatum, die einzelnen Dozenten haben.

**tbDozent**

Die Tabelle `tbKursbesuche` beinhaltet alle Teilnehmer an den Kursen. Sie ist wie die Dozententabelle über eine Beziehung mit der Tabelle `tbPerson` verbunden, in der sich die Basisdaten der Kursteilnehmer, insbesondere die Adressdaten befinden. In diesem Sinne bedeutet die Modellierung, dass es sich bei den Kursbesuchern immer um Personen handelt (`tbPerson`), die für einen Kurs die Rolle eines Kursteilnehmers (`tbKursbesuche`) annehmen können.

**tbKursbesuche**

Auf der anderen Seite der Grafik finden Sie die Tabelle `tbKursthema`, die alle angebotenen Kurse thematisch enthält, also jeweils das Thema des Kurses, eine ausführliche Beschreibung und weitere Angaben. Der Primärschlüssel dieser Tabelle ist die `KTHID` (KursThema-Identifikationsnummer). Sie ist in Abbildung 3.1 beispielhaft hervorgehoben. Die Primärschlüssel der anderen Tabellen sind jeweils in Fettschrift markiert und nach demselben Schema benannt. Die Tabelle `tbKursthema` ist aber nicht im Sinne eines konkreten Kurses zu verstehen, sondern als Modul, das inhaltsgleich zu verschiedenen Terminen von verschiedenen Dozenten für verschiedene Teilnehmer angeboten werden kann.

**tbKursthema**

Die konkreten Kurse sind in der Tabelle `tbKurs` aufgelistet. Hier werden die Termine angegeben sowie alle weiteren eventuell für die Durchführung relevanten Angaben gemacht. Über die Beziehung zur Tabelle `tbKursthema` kann auf die genauen Angaben zum Thema eines Kurses zurückgegriffen werden. Diese Angaben sind bei jeder Durchführung identisch, während sich die Termine, die konkrete Dauer, eine eventuelle Umsatzsteuerbefreiung und andere Angaben ändern können. Die Tabelle `tbKurs` beschreibt also die eigentliche Durchführung des Kurses, die darin besteht, dass zu dem geplanten Termin der Dozent und die Kursteilnehmer zusammengebracht werden (auf den Ort wurde hier noch verzichtet). Die Tabelle `tbKurs` ist somit das eigentliche Herzstück des Kursangebotes, obwohl sie vergleichsweise wenig Datenfelder enthält. Über die insgesamt drei Fremdschlüsselattribute werden das Kursthema, die Teilnehmer und der Dozent mit dem Kurs verbunden.

**tbKurs**

## 3.2 Schnelleinstieg und Neustart

Wenn Sie Ihre Datenbank bereits installiert haben, können Sie die Kursdatenbank und wenn Sie wollen auch das zweite Beispiel, die Artikeldatenbank, jetzt schnell installieren. Die Artikeldatenbank wird inhaltlich in den Kapiteln 7 und 8 ausführlich besprochen. Dieser Abschnitt richtet sich also

zunächst ausdrücklich nur an diejenigen Leser, die eine der hier verwendeten Datenbankumgebungen bereits lauffähig installiert haben und sich im Umgang damit bereits etwas auskennen. Für die anderen Leser erfolgt im Anschluss eine ausführliche Beschreibung.

Außerdem wird hier beschrieben, wie Sie bereits installierte Datenbanken wieder in den Ausgangszustand zurücksetzen können. Wenn Sie noch keine Datenbank installiert haben, überspringen Sie diesen Abschnitt und kommen darauf zurück, falls Sie später einmal Ihre Datenbank zurücksetzen möchten.

### 3.2.1 MySQL

**kurse** Alle Daten für die Kursdatenbank stehen auf der CD unter *datenbanken/MySQL/kurse*. Um die Kursdatenbank wieder zurückzusetzen, wird das Skript *LoescheKursdb* aufgerufen und anschließend das Löschen mit „y“ bestätigt.

Um die Datenbank wieder zu laden, wird das Skript *Kursdb* aufgerufen.

Beide Skripte gehen davon aus, dass Sie das Passwort *masterkey* für den Benutzer *root* eingerichtet haben, wie es in der folgenden Installation empfohlen wird. Außerdem gehen beide Skripte davon aus, dass sie aus dem Verzeichnis *datenbanken/MySQL/kurse* heraus ausgeführt werden.

**artikel** Alle Daten für die Artikeldatenbank stehen auf der CD unter *datenbanken/MySQL/artikel*. Um die Kursdatenbank wieder zurückzusetzen, wird das Skript *LoescheArtikeldb* aufgerufen und anschließend das Löschen mit „y“ bestätigt.

Um die Datenbank wieder zu laden, wird das Skript *Artikeldb* aufgerufen.

Beide Skripte gehen davon aus, dass Sie das Passwort *masterkey* für den Benutzer *root* eingerichtet haben, wie es in der folgenden Installation empfohlen wird. Außerdem gehen beide Skripte davon aus, dass sie aus dem Verzeichnis *datenbanken/MySQL/kurse* heraus ausgeführt werden.

### 3.2.2 Oracle

**kurse** Alle Daten für die Kursdatenbank stehen auf der CD unter *datenbanken/Oracle/kurse*. Um die Kursdatenbank wieder zurückzusetzen, wird das Skript *LoescheOraKurse* aufgerufen.

Um die Datenbank wieder zu laden, wird das Skript *OraKurse* aufgerufen.

Beide Skripte gehen davon aus, dass Sie das Passwort *masterkey* für den Benutzer *system* eingerichtet haben, wie es in der folgenden Installation empfohlen wird. Die Existenz des Schemas können Sie in Oracle unter `ADMINISTRATION/DATABASE USERS/MANAGE USERS` kontrollieren. Existiert der Benutzer *kurse*, ist das Schema verfügbar. Sie können sich mit dem Benutzer *kurse* und dem Passwort *pwkurse* anmelden und unter `OBJECT BROWSER/BROWSE/TABLES` den Inhalt überprüfen. Im Tabellenblatt `TABLE` sehen Sie die Struktur, unter `DATA` den Inhalt der Tabellen.

Alle Daten für die Artikeldatenbank stehen auf der CD unter *datenbanken/Oracle/artikel*. Um die Kursdatenbank wieder zurückzusetzen, wird das Skript *LoescheOraArtikel* aufgerufen.

artikel

Um die Datenbank wieder zu laden, wird das Skript *OraArtikel* aufgerufen.

Beide Skripte gehen davon aus, dass Sie das Passwort *masterkey* für den Benutzer *system* eingerichtet haben, wie es in der folgenden Installation empfohlen wird. Die Existenz des Schemas können Sie in Oracle unter ADMINISTRATION/DATABASE USERS/MANAGE USERS kontrollieren. Existiert der Benutzer *artikel*, ist das Schema verfügbar. Sie können sich mit dem Benutzer *artikel* und dem Passwort *pwartikel* anmelden und unter OBJECT BROWSER/BROWSE/TABLES den Inhalt überprüfen. Im Tabellenblatt TABLE sehen Sie die Struktur, unter DATA den Inhalt der Tabellen.

### 3.2.3 Firebird

Alle Daten für die Kursdatenbank stehen auf der CD unter *datenbanken/Firebird/kurse*. Um die Kursdatenbank wieder zurückzusetzen, wird das Skript *LoescheFireKurse* aufgerufen.

kurse

Um die Datenbank wieder zu laden, wird das Skript *FireKurse* aufgerufen.

Alle Daten für die Artikeldatenbank stehen auf der CD unter *datenbanken/Firebird/artikel*. Um die Artikeldatenbank wieder zurückzusetzen, wird das Skript *LoescheFireArtikel* aufgerufen.

artikel

Um die Datenbank wieder zu laden, wird das Skript *FireArtikel* aufgerufen.

Beide Skripte (je Datenbank) gehen jeweils davon aus, dass Sie das Passwort *masterkey* für den Benutzer *SYSDBA* eingerichtet haben, wie es in der folgenden Installation empfohlen wird. Die Existenz des Schemas können Sie in Firebird mit der hier verwendeten Konsole unter DATABASES im Baum des linken Fensters prüfen. Mit einem Doppelklick auf die Datenbank öffnet sich eine Liste, die unter anderem den Eintrag TABLES zeigt. Wählen Sie diesen Eintrag aus, können Sie anschließend durch Doppelklick auf die gewünschte Tabelle im rechten Fenster ein weiteres Fenster mit der Struktur der Tabelle öffnen. In der Registerseite PROPERTIES sehen Sie die Struktur, unter DATA nach der zusätzlichen Auswahl von OPEN den Inhalt der Tabelle.

Wenn Sie eine Datenbank wie *kurse* oder *artikel* neu aufbauen wollen, können Sie dies durch den Start der beiden entsprechenden Skripte tun. Sie müssen sie hinterher unter Umständen noch registrieren. Dazu rufen Sie beispielsweise in der Konsolenoberfläche DATABASE/REGISTER auf. Lassen Sie sich auch nicht davon irritieren, dass die Datenbank nach der Ausführung des Löschkriptes noch sichtbar ist. Mit einem Doppelklick auf den Namen der Datenbank im linken Fenster erhalten Sie eine Fehlermeldung. Die Datenbank ist lediglich noch nicht abgemeldet, was auch nicht notwendig ist, da wir sie wieder neu erzeugen wollen.

### 3.2.4 MS Access

Alle Daten sind für MS Access 2003 vorbereitet. Um den Start zu vereinfachen, sind die Daten als komplette MS Access-Datenbank beigefügt und können unmittelbar genutzt werden.

**kurse** Alle Daten für die Kursdatenbank stehen auf der CD unter *datenbanken/MSAccess/kurse*. Die Datei *Kurse2003.mdb* enthält die komplette Kursdatenbank für MS Access 2003. Um die Kursdatenbank wieder zurückzusetzen, kopieren Sie die Datei einfach erneut von der CD. Für Nutzer älterer Versionen sind die Daten noch als Excel-Dateien beigefügt. Importieren Sie die Excel-Dateien in die Datenbank. Das Verfahren für MS Access wird in Anhang A beschrieben.

**artikel** Alle Daten für die Artikeldatenbank stehen auf der CD unter *datenbanken/MSAccess/artikel*. Die Datei *Artikel2003.mdb* enthält die komplette Artikeldatenbank für MS Access 2003. Um die Artikeldatenbank wieder zurückzusetzen, kopieren Sie die Datei einfach erneut von der CD. Für Nutzer älterer Versionen sind die Daten noch als Excel-Dateien beigefügt. Importieren Sie die Excel-Dateien in die Datenbank. Das Verfahren für MS Access wird in Anhang A beschrieben.

Denken Sie bei einer direkten Kopie von CD daran, gegebenenfalls den Schreibschutz zu entfernen. Klicken Sie dazu im Windows-Explorer mit der rechten Maustaste auf die kopierte Datei, wählen Sie den Befehl EIGENSCHAFTEN und entfernen Sie den Haken beim Schreibschutz.

### 3.2.5 openBase

Alle Daten sind für openOffice.org Base, hier immer kurz openBase genannt, vorbereitet. Um den Start zu vereinfachen, sind die Datenbanken jeweils als komplette Datenbanken beigefügt und können unmittelbar genutzt werden.

**kurse** Alle Daten für die Kursdatenbank stehen auf der CD unter *datenbanken/openBase/kurse*. Die Datei *kurse.odt* enthält die komplette Kursdatenbank für openBase. Um die Kursdatenbank wieder zurückzusetzen, kopieren Sie die Datei einfach erneut von der CD. Für Nutzer älterer Versionen sind die Daten noch als Calc-Dateien beigefügt. Importieren Sie die Calc-Dateien in die Datenbank. Das Verfahren für openBase wird in Anhang A beschrieben.

**artikel** Alle Daten für die Artikeldatenbank stehen auf der CD unter *datenbanken/openBase/artikel*. Die Datei *artikel.odt* enthält die komplette Artikeldatenbank für openBase. Um die Artikeldatenbank wieder zurückzusetzen, kopieren Sie die Datei einfach erneut von der CD. Für Nutzer älterer Versionen sind die Daten noch als Calc-Dateien beigefügt. Importieren Sie die Calc-Dateien in die Datenbank. Das Verfahren für openBase wird in Anhang A beschrieben.

Denken Sie bei einer direkten Kopie von CD daran, gegebenenfalls den Schreibschutz zu entfernen. Klicken Sie dazu im Windows-Explorer mit der rechten Maustaste auf die kopierte Datei, wählen Sie die EIGENSCHAFTEN und entfernen Sie den Haken beim Schreibschutz.

## 3.3 MySQL

### 3.3.1 Gründe für die Nutzung

MySQL ist ein Datenbankmanagementsystem, das von der MySQL AB entwickelt wurde und neben einer kommerziellen Variante auch unter der General Public License (GPL) verfügbar ist. Es darf damit für unsere Übungszwecke frei eingesetzt werden. Die breite Verfügbarkeit und die Lauffähigkeit unter Linux wie unter Windows haben es insbesondere für Internetserver, aber auch für viele mittelgroße Betriebe oder Abteilungen populär gemacht. Kurz vor der Fertigstellung dieses Buches wurde MySQL von Sun übernommen. Was dies für die Zukunft der Datenbank bedeutet, bleibt abzuwarten.

Hinzu kommt, dass MySQL sowohl über eine klassische zeilenorientierte SQL-Befehlseingabe als auch über zahlreiche grafische Front-Ends verfügt, mit denen komfortabel SQL-Befehle direkt eingegeben werden können. Man kann also sehr dicht am System sein, aber auch sehr komfortable Oberflächen nutzen. Für MySQL spricht also:

- Große Popularität und weite Verbreitung
- Freie Verfügbarkeit für die Leser
- Klassische SQL-Eingabemöglichkeiten verbunden mit einer grafischen Benutzeroberfläche

Die Beispiele beruhen auf der Version 5.0.41 für den Server und 5.0.11 für den Client. Natürlich gibt es laufend neuere Versionen, auf denen die Beispiele ebenfalls funktionieren sollten. Als Oberfläche wird MySQL Administrator 1.2.12 gewählt, eine ebenfalls weitverbreitete grafische Oberfläche.

### 3.3.2 Den Server installieren und konfigurieren

Sie finden die MySQL-Version auf der beiliegenden CD. Es werden hier Versionen des MySQL Community Servers verwendet. Neuere Versionen können Sie aus dem Internet herunterladen. Beachten Sie die jeweiligen Lizenzbedingungen, da sich diese für die verschiedenen Server ändern können.

Zur Installation von MySQL verwenden Sie die Dateien im Unterordner *MySQL* auf der beiliegenden CD. Sie finden dort die gepackte Datei *mysql-5.0.41-win32.zip*. Für Linux sind unterschiedliche Versionen für die verschiedenen Systeme verfügbar, deren komplette Bereitstellung den Rahmen dieses Buches sprengen würde. Sie können sie für die Beispiele aber in gleicher Weise nutzen. Sie finden die Downloads unter [dev.mysql.com/downloads](http://dev.mysql.com/downloads).

Version

Nach dem Entpacken und dem Start von *Setup.exe* führt Sie dann der Installationsassistent durch die weitere Installation.



Abbildung 3.2  
Start des Setup-Wizards

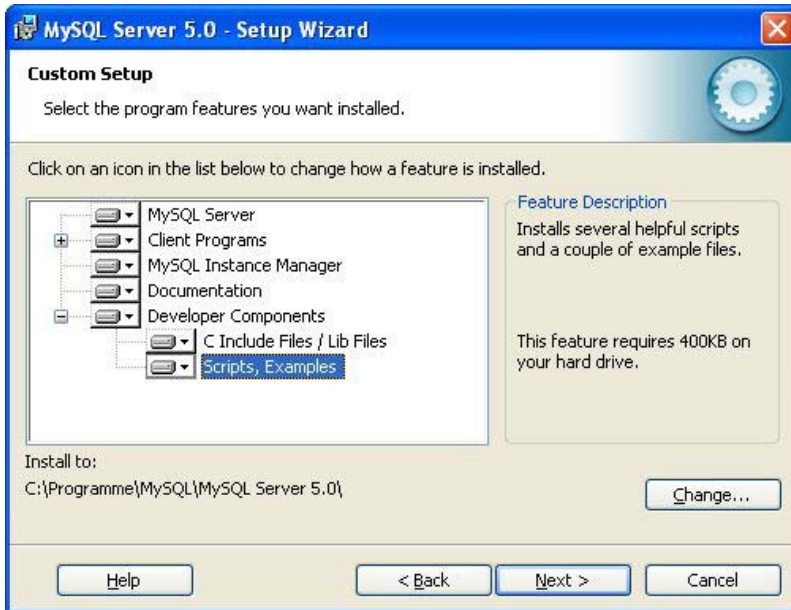


Nach dem Start meldet sich der Installationsassistent, wie in Abbildung 3.2 dargestellt. Die Installation kann in drei verschiedenen Varianten erfolgen (siehe Abbildung 3.3).

Abbildung 3.3  
Verschiedene Installations-  
möglichkeiten für MySQL



Die Unterschiede zwischen den drei Varianten sehen Sie am besten, wenn Sie die individuelle Installation CUSTOM wählen und die komplette Übersicht über die verfügbaren Komponenten betrachten (siehe Abbildung 3.3). Wesentliche Unterschiede liegen insbesondere in der Installation von Entwicklungswerkzeugen, mit deren Hilfe man Anwendungen für MySQL schreiben kann.

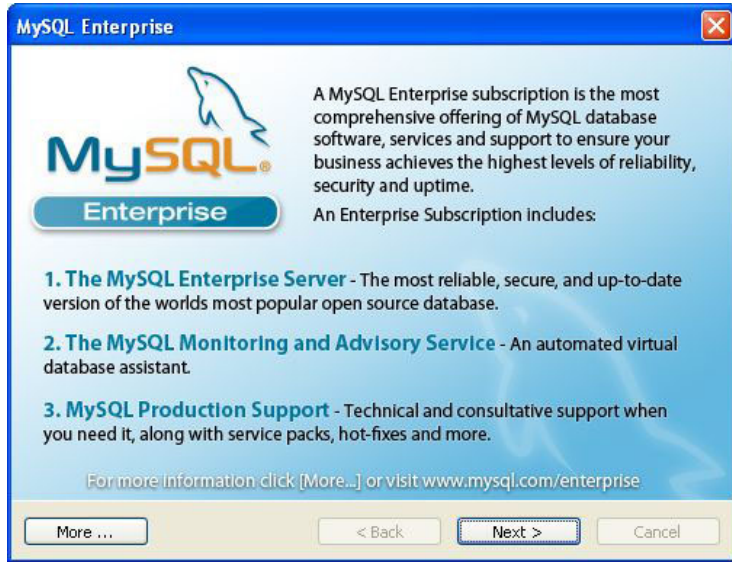


**Abbildung 3.4**  
Benutzerspezifische  
Konfiguration

MySQL wird standardmäßig mit einer C-Bibliothek ausgeliefert, den *C Include Files/Lib Files*, mit denen SQL aus C-Programmen aufgerufen werden kann. Sie können die Entwicklungskomponenten zusätzlich auswählen, wenn Sie planen, mit MySQL auch Anwendungen zu entwickeln. Schnittstellen für andere Programmiersprachen existieren in großem Umfang, müssen aber getrennt installiert werden. Wollen Sie keine eigenen Programme schreiben, können Sie darauf verzichten und später bei Bedarf nachinstallieren.

In jedem Fall, egal, welche Konfigurationskomponente Sie gewählt haben, wird nach der Bestätigung die eigentliche Installation durchgeführt. Nach Abschluss der Installation und vor Beginn der Konfiguration erscheint der Hinweis auf die kommerzielle Version des Enterprise Servers sowie weitere Informationsquellen zu MySQL (siehe Abbildung 3.5).

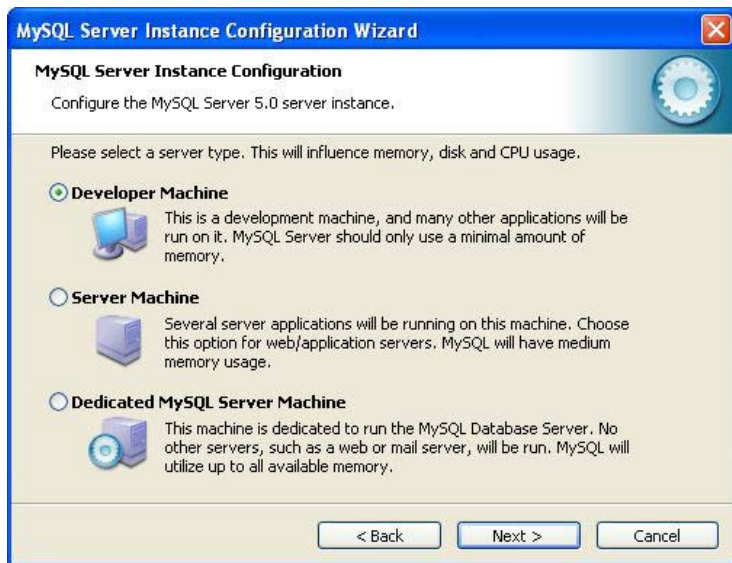
Abbildung 3.5 Bestätigung der Installation



**Konfiguration**

Nach einem weiteren Fenster beginnt die Konfiguration der Datenbank. Bei der Konfiguration haben Sie zunächst die Wahl zwischen der DETAILED CONFIGURATION und der STANDARD CONFIGURATION. Den allergrößten Teil der Beispiele dieses Buches können Sie mit der Standardkonfiguration bearbeiten. Gleichwohl gibt es gerade am Ende des Buches im Rahmen der technischen Beschreibungen einige Beispiele, für die bestimmte Datenhaltungssysteme notwendig sind. Das und das bessere Verständnis der Funktionsweise von MySQL sprechen daher für die Wahl der detaillierten Konfiguration, die dann einige weitere Angaben zu Ihrer Installation erfordert.

Abbildung 3.6 Basiskonfiguration



Sie sehen drei typische Konfigurationen für die verschiedenen Nutzungsmöglichkeiten von MySQL. Die DEVELOPER MACHINE ist die typische Konfiguration für einen Entwicklungsrechner, auf dem die Datenbank einen Teil der Entwicklungsumgebung darstellt. Andere Entwicklungswerkzeuge und eventuell auch andere Datenbankmanagementsysteme werden parallel benötigt. Daher sollte MySQL nicht zu viel Speicher verwenden. Dazu kommt, dass in der Entwicklungsumgebung zumeist nicht allzu große Datenmengen verarbeitet werden müssen. Somit ist diese Konfiguration geeignet, um die Datenbank ressourcenschonend zu betreiben, wenn nicht allzu hohe Anforderungen an die Performance gestellt werden.

#### Developer Machine

Die SERVER MACHINE ist die typische Konfiguration für einen Webserver, bei dem MySQL schon größere Datenmengen und vor allem viele lesende Anfragen bearbeiten muss. Daher werden größere Ressourcen – insbesondere Speicher – auf dem Rechner benötigt. Andererseits müssen noch ein Webserver wie Apache und eine Reihe von PHP-, Python- oder Perlskripten für ein CMS, einen Webshop oder sonstige Anwendungen ausgeführt werden können, sodass nicht alle Ressourcen von MySQL beansprucht werden dürfen. Diese Konfiguration stellt also einen Kompromiss zwischen Performance der Datenbank und anderen Anwendungen dar.

#### Server Machine

Schließlich gibt es noch den dedizierten Datenbankserver, sei es in einer Client/Server-Umgebung oder ebenfalls in einer Webumgebung mit verschiedenen Hardwareservern für die einzelnen Komponenten. In einer solchen Umgebung muss der Datenbankserver auf maximale Performance getrimmt werden und benötigt dafür alle verfügbaren Ressourcen.

#### Dedicated Server Machine

Für unsere Zwecke sollte die Konfiguration als DEVELOPER MACHINE ausreichend sein.



Abbildung 3.7  
Nutzung verschiedener  
Datenablagensysteme

MySQL verwendet im Sinne einer Schichtenarchitektur verschiedene Datenhaltungssysteme, um die in der Datenbank abgelegten Informationen zu speichern. MySQL übernimmt also die Datenorganisation auf der Festplatte nicht selbst, sondern bedient sich dazu wiederum eines Datenhaltungssystems. Die beiden wichtigsten Datenhaltungssysteme sind *MyISAM* und *InnoDB*. Der wesentliche Unterschied ist, dass *MyISAM* zumeist schnellere Datenzugriffe bietet, dafür aber keine Transaktionen, also in sich abgeschlossene Prozessschritte, erlaubt. Solange Sie nur allein in der Datenbank arbeiten oder zumindest nur ein Benutzer ändernd auf die Datenbank zugreift, sind Transaktionen weitgehend überflüssig. Wenn Sie also MySQL nur zu Übungszwecken für dieses Buch einrichten wollen und die Transaktionsbeispiele in Kapitel 13 für Sie nicht relevant sind, reicht die Nutzung von *MyISAM* aus, im anderen Fall ist es besser, die Datenbank von vornherein für Transaktionen vorzubereiten. Wenn Sie `TRANSACTIONAL DATABASE ONLY` auswählen, stellt dies sicher, dass die Daten gleich in *InnoDB* verwaltet werden können. Für unsere relativ kleinen Datenmengen sollten in keinem Fall Probleme mit der Performance entstehen.

Alternativ können Sie die `MULTIFUNCTIONAL DATABASE` verwenden. Dann bleiben alle Optionen offen, Sie müssen dann aber die Verwendung später selbst steuern, was unter Umständen Komplikationen mit sich bringt.

Im Anschluss an die Wahl der „Multifunctional Database“ müssen Sie das Standardsystem für die physische Dateiverwaltung (*InnoDB* oder *MyISAM*) auswählen. Wenn Sie wirklich mit Transaktionen arbeiten wollen, sollten Sie *InnoDB* wählen. Sonst ist nur noch der Pfad für die Speicherung festzulegen.

Das nächste Auswahlfenster bestimmt die Anzahl gleichzeitiger Benutzer der Datenbank. Da wir in unserer Übungsdatenbank eher mit wenigen, maximal drei parallelen Nutzern arbeiten, können Sie entweder die Variante `DECISION SUPPORT (DSS)/OLAP` oder `MANUAL SETTING` mit einer Anzahl von beispielsweise fünf Nutzern wählen.

Im Folgenden geht es um die technische Basis des Servers. Unter Windows wird das System als Dienst eingerichtet. Für die Nutzung zusammen mit anderen Oberflächen ist es sinnvoll, wenn Sie `TCP/IP` aktivieren und den `STRICT MODE` ebenfalls einschalten (siehe Abbildung 3.8).

Danach müssen Sie den Standardzeichensatz festlegen, der in sogenannten „Collations“ gespeichert wird, siehe dazu auch Kapitel 8. Für unsere Zwecke reicht im Normalfall die Standardvariante `Latin-1`.

Danach aktivieren Sie MySQL endgültig als Windows-Service. Sie können den Standardnamen `MySQL` wählen. Starten Sie den Service automatisch mit Windows, wenn Sie ihn regelmäßig zum Testen und Probieren benötigen. Sonst sollten Sie ihn über den Windows-Dienstmanager jeweils aktivieren, wenn Sie ihn benötigen, da die Datenbank beim Start des Dienstes doch einige Zeit benötigt und Ressourcen belegt.

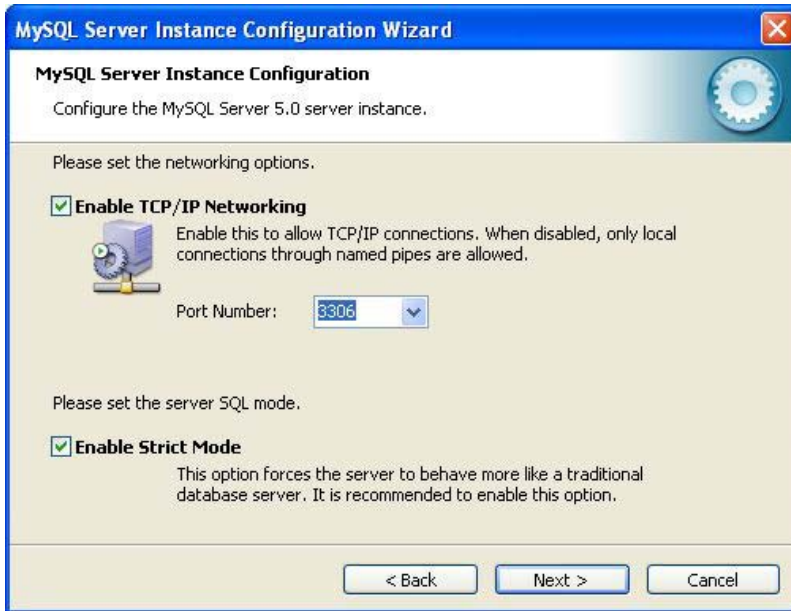


Abbildung 3.8  
Technische Aktivierung

Die Option `INCLUDE BIN DIRECTORY IN WINDOWS PATH` ist dann von Vorteil, wenn Sie die Kommandozeilenwerkzeuge von MySQL nutzen wollen (Fenster EINGABEAUFFORDERUNG). In diesem Fall macht es Sinn, dass das `BIN`-Verzeichnis im Pfad liegt, weil Sie dann den Pfad beim Aufruf der einzelnen Kommandozeilenwerkzeuge nicht immer neu eingeben müssen.

Nach der Definition des Servernamens und des Pfades sind alle wesentlichen Parameter für MySQL festgelegt. Damit Sie sich gleich erstmalig bei MySQL anmelden können, muss in der Datenbank noch mindestens ein Benutzerkonto eingerichtet werden, also ein Benutzer mit dessen Namen Sie sich bei MySQL identifizieren können. Wie jedes Datenbankmanagementsystem hat auch MySQL mindestens einen sogenannten Superuser. Er heißt hier *root*. Sie müssen jetzt das Passwort festlegen, mit dem sich *root* anmelden kann.

Superuser = root

In diesem Buch wird als Passwort für den Superuser immer *masterkey* verwendet. Sie können natürlich ein anderes Passwort verwenden. Ersetzen Sie dann einfach an den entsprechenden Stellen der Listings und Skripte *masterkey* durch das entsprechende Passwort.

Tragen Sie ein Passwort ein und bestätigen Sie es (siehe Abbildung 3.9). Merken Sie sich das Passwort oder schreiben Sie es auf.

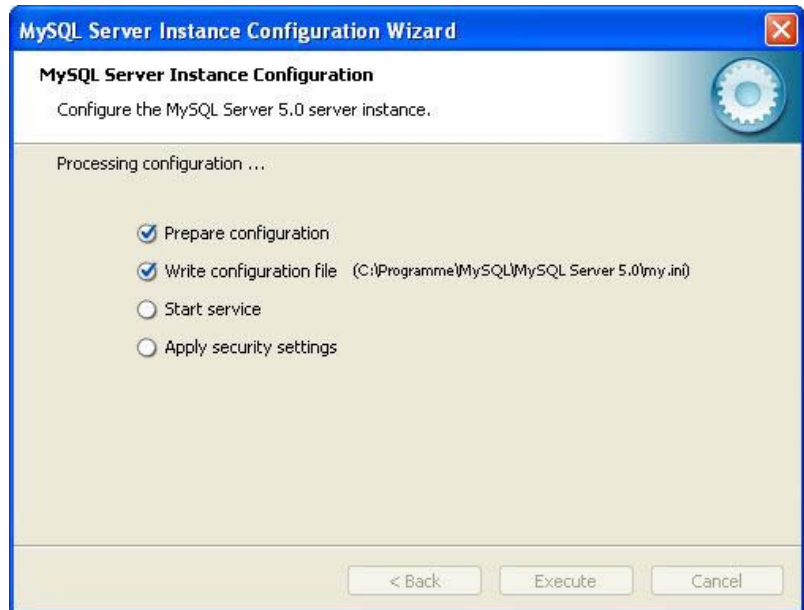
Den zusätzlichen anonymen Account benötigen wir für unsere Beispiele nicht zwingend.

**Abbildung 3.9**  
Einrichten des Superusers in MySQL

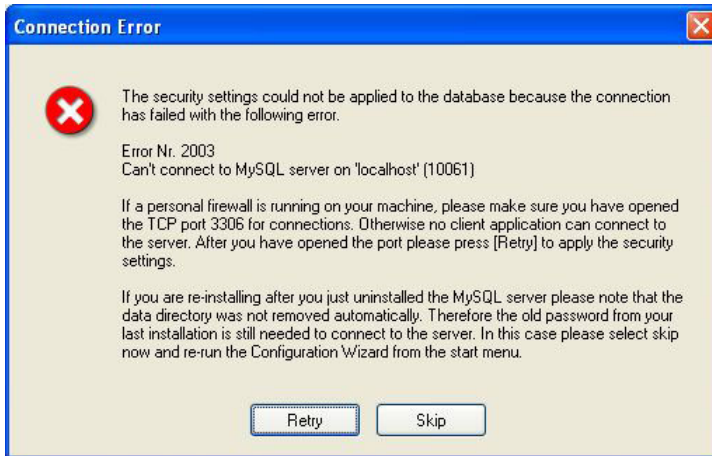


Es beginnt die eigentliche Einrichtung des Servers, die Sie über eine Fortschrittsanzeige verfolgen können (siehe Abbildung 3.10).

**Abbildung 3.10**  
Einrichtung und Start des Servers



Probleme gibt es unter Windows zumeist, wenn die Firewall eingeschaltet ist, weil der Port 3306, über den MySQL kommuniziert, nicht freigegeben ist. Sie erhalten dann eine Fehlermeldung wie in Abbildung 3.11.



**Abbildung 3.11**  
Fehlermeldung bei Konflikten mit der Portfreigabe in der Firewall

In diesem Fall müssen Sie den Port freigeben.



**Abbildung 3.12**  
Firewall unter Windows



Gehen Sie in die Systemsteuerung und öffnen Sie die Windows-Firewall. Unter Vista müssen Sie noch auf SICHERHEITSCENTER klicken, bevor Sie die Windows-Firewall öffnen können. Wenn Sie Ihre Firewall aktiviert haben (siehe Abbildung 3.12), können Sie diese natürlich deaktivieren. Ratsamer ist es aber, gezielt nur den von MySQL verwendeten Port freizugeben.

Dazu müssen zunächst Ausnahmen zugelassen sein. Gehen Sie auf AUSNAHMEN und wählen dort die Option PORT. Sie gelangen in ein kleines Fenster, in dem Sie den Port 3306 freigeben können.

**Abbildung 3.13**  
Freigabe des TCP-Ports  
3306 für MySQL



Damit haben Sie alle Voraussetzungen geschaffen und die Installation von MySQL sollte normal enden (siehe Abbildung 3.14).

Der Server läuft jetzt auf dem System und kann genutzt werden. Sie können dies in der Windows-Systemsteuerung unter den Diensten kontrollieren.

Hier sollten Sie einen Dienst MySQL finden, der gestartet ist. Je nachdem, ob Sie sich für oder gegen einen automatischen Start entschieden haben, finden Sie hier auch den entsprechenden Eintrag. Das Starten und Beenden des Dienstes können Sie ebenfalls manuell vornehmen.



Abbildung 3.14  
Abschluss der  
MySQL-Installation

### 3.3.3 Die Kommandozeile

MySQL bietet nach der Installation nur eine sehr eingeschränkte grafische Oberfläche. Hier macht die Installation einer Ergänzung wie der *MySQL Tools* Sinn.

Sie sollten aber auch die ganz elementare, einfache Kommandozeilenoberfläche von MySQL kennen, falls Ihnen einmal keine andere Oberfläche zur Verfügung steht oder falls Sie einmal Batchprozesse aufbauen und dabei die entsprechenden Komponenten nutzen wollen. Wir beschränken uns hier wieder auf die Windows-Sicht, unter Linux sieht die Umgebung aber prinzipiell sehr ähnlich aus.

Die Kommandooberfläche von MySQL wird direkt bei der Installation bereitgestellt. Installieren Sie MySQL unter Windows, wird das System direkt im Pfad `C:\Programme\MySQL\MySQL Server 5.0` abgelegt. Ein Unterverzeichnis, das dabei angelegt wird, ist das `\bin`-Verzeichnis. Einen Ausschnitt aus dem typischen Inhalt dieses Verzeichnisses sehen Sie in Abbildung 3.15.

Wir wollen zunächst nur mit den beiden Standardprogrammen `mysql.exe` und `mysqladmin.exe` arbeiten. Sie müssen dazu unter Windows über `START/ALLE PROGRAMME/ZUBEHÖR` das Zubehör von Windows wählen. Dort wiederum wählen Sie die `EINGABEAUFFORDERUNG`. Sie gelangen in das Fenster `EINGABEAUFFORDERUNG`. Hier müssen Sie in das MySQL-Verzeichnis navigieren (wenn Sie das Verzeichnis nicht bei der Installation in den Windows-Pfad aufgenommen haben). Falls Sie die alte DOS-Logik nicht mehr griffbereit haben, es reichen im Prinzip wenige Befehle, um dieses Verzeichnis zu erreichen.

**Abbildung 3.15**  
 Inhalt des Verzeichnisses  
 ...\\mysql\\bin (hier unter  
 Windows)

```

Verzeichnis von C:\Programme\MySQL\MySQL Server 5.0\bin
16.03.2008 15:36 <DIR> .
16.03.2008 15:36 <DIR> ..
04.05.2007 11:08 1.490.944 comp-err.exe
04.05.2007 11:08 2.031.616 libmysql.dll
04.05.2007 11:08 1.826.816 myisamchk.exe
04.05.2007 11:08 1.720.320 myisamlog.exe
04.05.2007 11:08 1.744.896 myisampack.exe
04.05.2007 11:08 1.708.032 myisam_ftdump.exe
04.05.2007 11:08 2.039.808 mysql.exe
04.05.2007 11:08 1.990.656 mysqladmin.exe
04.05.2007 11:08 2.052.096 mysqlbinlog.exe
04.05.2007 11:08 1.982.464 mysqlcheck.exe
04.05.2007 11:08 8.421.376 mysqld-debug.exe
04.05.2007 11:08 1.959.056 mysqld-debug.map
04.05.2007 11:08 12.374.016 mysqld-debug.pdb
04.05.2007 11:00 5.701.632 mysqld-nt.exe
04.05.2007 11:00 2.699.520 mysqld-nt.map
04.05.2007 11:00 13.103.104 mysqld-nt.pdb
04.05.2007 11:08 5.701.632 mysqld.exe
04.05.2007 11:08 2.699.520 mysqld.map
04.05.2007 11:08 13.103.104 mysqld.pdb
04.05.2007 11:08 2.035.712 mysqldump.exe
04.05.2007 11:08 1.982.464 mysqlimport.exe
09.01.2007 10:34 3.468.288 MySQLInstanceConfig.exe
04.05.2007 11:08 1.994.752 mysqlmanager.exe
04.05.2007 11:08 556.506 mysqlmanager.map
04.05.2007 11:08 3.944.448 mysqlmanager.pdb
04.05.2007 11:08 1.982.464 mysqlshow.exe
04.05.2007 11:08 2.097.152 mysqltest.exe
04.05.2007 11:08 2.277.376 mysql_client_test.exe
04.05.2007 11:08 1.986.560 mysql_upgrade.exe
04.05.2007 11:08 1.478.656 my_print_defaults.exe
04.05.2007 11:08 1.470.464 perror.exe
04.05.2007 11:08 1.474.560 replace.exe
32 Datei(en) 111.100.010 Bytes
2 Verzeichnis(se), 11.148.677.120 Bytes frei

C:\Programme\MySQL\MySQL Server 5.0\bin>
    
```

**Tabelle 3.1**  
 Basisbefehle zum  
 Erreichen der  
 Kommandozeile  
 von MySQL

Befehl	Bedeutung
cd verzeichnis	Wechsle in das angegebene Unterverzeichnis
cd \	Wechsle auf das Basislaufwerk, aus jedem beliebigen Unterverzeichnis möglich
Laufwerk:	Wechsle in das angegebene Laufwerk
cd ...	Wechsle in das übergeordnete Verzeichnis
dir	Zeige alle Dateien im aktuellen Verzeichnis an

**Beispiel** Nach dem Aufruf des Fensters für die Eingabeaufforderung befinden Sie sich beispielsweise an der unten angegebenen Stelle, während MySQL unter C:\Programme\MySQL\MySQL Server 5.0\bin\ installiert ist.

d:\Dokumente und Einstellungen\Harry>

Sie geben ein:

```

C:
cd Programme
    
```

```
cd MySQL
cd MySQL Server 5.0
cd bin
dir
```

Jetzt müssten Sie in etwa das Bild aus Abbildung 3.15 sehen.

Das Programm *mysqladmin.exe* dient der Administration von MySQL, erfüllt also etwa die Funktion des MySQL Administrators. Hiermit können die grundlegenden Anweisungen für den technischen Betrieb des Datenbankmanagementsystems erteilt werden. Mit dem Aufruf

[mysqladmin.exe](#)

```
C:\Programme\mysql\MySQL Server 5.0\bin>mysqladmin -help
```

können Sie die Optionen aufrufen, die beim Aufruf des Programms angegeben werden können. Der Pfad muss nur eingegeben werden, wenn Sie sich nicht in dem Verzeichnis befinden bzw. der Pfad nicht bei der Installation eingetragen wurde. Im Folgenden werden wir dies voraussetzen und ihn weglassen.

So können Sie sich beispielsweise als Benutzer *root* anmelden und prüfen, ob der MySQL-Server antwortet:

```
mysqladmin -uroot -pmasterkey ping
```

Beachten Sie, dass nach dem Parameter, beispielsweise nach *-p*, der Wert – hier das Passwort – direkt und ohne Leerzeichen angeschlossen wird. Sie können auch mit

```
mysqladmin -uroot -pmasterkey version
```

die aktuelle Version Ihres Servers und einige weitere Statusinformationen erhalten. Wir werden in Abschnitt 3.3.5 *mysqladmin.exe* nutzen, um ein Datenbankschema für unsere Übungsdatenbanken anzulegen.

Das zweite wichtige Programm ist *mysql.exe*. Es dient der Eingabe beliebiger SQL-Anweisungen. Während also *mysqladmin.exe* die grundlegenden technischen Einstellungen steuert, können mit *mysql.exe* die für uns interessanten SQL-Befehle eingegeben werden. Der Aufruf, um die verfügbaren Parameter zu erhalten, lautet:

[mysql.exe](#)

```
mysql -?
```

Denken Sie gegebenenfalls an den Pfadnamen. Sie sehen, dass Sie eine Reihe von Optionen, wiederum beginnend mit einem Bindestrich, angeben können. Die einfachste Form der Anmeldung bei der Datenbank ist:

```
mysql -uroot -pmasterkey
```

Nach diesem Befehl ändert sich der Cursor zu:

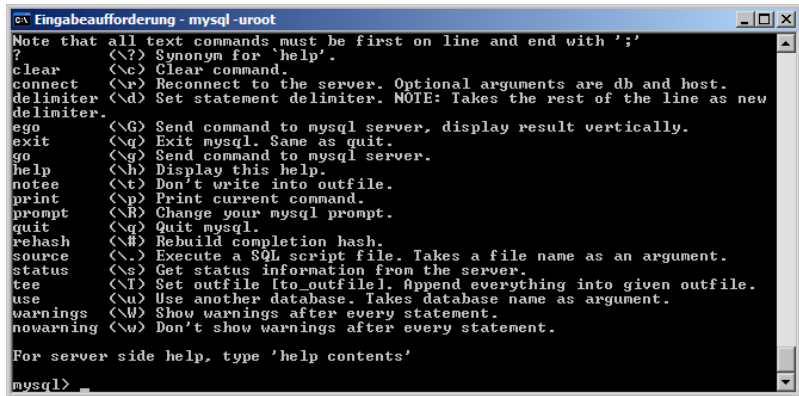
```
mysql>
```

Diese unscheinbare Änderung besagt, dass Sie jetzt nicht mehr auf Betriebssystemebene sind, sondern sich innerhalb eines MySQL-Clients bewegen. Sie sind beim Datenbankmanagementsystem angemeldet und können ab jetzt direkt SQL-Anweisungen sowie einige zusätzliche Kommandos eingeben. Testen Sie dies, indem Sie beispielsweise

```
help
```

eingeben. Sie erhalten eine Liste der möglichen Kommandos, die etwa der Liste in Abbildung 3.16 entsprechen sollte. Die meisten dieser Optionen spielen für uns hier keine große Rolle. Wir wollen aber im nächsten Abschnitt die Funktionsweise am Beispiel der ersten Beispieldatenbank zeigen.

Abbildung 3.16  
Optionen von mysql.exe



Verlassen Sie dazu zunächst *mysql.exe* mit dem Kommando

```
exit (oder quit).
```

Sie gelangen auf die Betriebssystemebene zurück, wie Sie an dem veränderten Cursor sehen.

### 3.3.4 Die grafische Oberfläche MySQL Tools

MySQL meldet sich nach dem Start ausschließlich als Dienst. Dieser besitzt auf Windows eine kleine Oberfläche, die aber nicht zur Eingabe von SQL-Anweisungen geeignet ist. Die zeichenorientierte Oberfläche, die unmittelbar zur Verfügung steht, haben wir gerade getestet. Sie ist letztlich wenig komfortabel. Daher sollten Sie zusätzlich eine grafische Oberfläche installieren. Eine der verfügbaren Oberflächen ist ebenfalls im Unterverzeichnis *MySQL* der CD vorhanden, die *MySQL Tools*. Sie können sie auch direkt von MySQL und verschiedenen Websites laden.

Zur Installation der MySQL Tools starten Sie *mysql-gui-tools 5.0-r12-win32.msi* aus demselben Verzeichnis.



**Abbildung 3.17**  
Start der Installation  
von MySQL Tools

Nach dem Start können Sie die Installation neu erstellen beziehungsweise modifizieren. Eine benutzerdefinierte Installation (CUSTOM SETUP) zeigt Ihnen die einzelnen Komponenten:

- Der SQL-Administrator dient der Verwaltung der Datenbank, also des Service, der Benutzer, der Verbindungen, der Datensicherung und anderer Aktivitäten, die nichts mit dem eigentlichen Inhalt in Form der Tabellen zu tun haben.
- Der Query Browser ist das eigentliche Werkzeug für die Eingabe und Ausführung der SQL-Anweisungen. Die meisten Anweisungen unserer Beispiele sind hier einzugeben.
- Das Migration Toolkit dient der Datenmigration und wird uns nicht weiter beschäftigen.



**Abbildung 3.18**  
Die verschiedenen  
Komponenten der  
grafischen Tools

Nach der Bestätigung erfolgt die Installation. Sie können nach der Installation die Programme über das normale Start-Menü von Windows aufrufen. Starten Sie den MySQL-Administrator für Ihre erste Testanmeldung. Sie werden aufgefordert, sich zu identifizieren (siehe Abbildung 3.19).

Abbildung 3.19  
Anmeldung als root

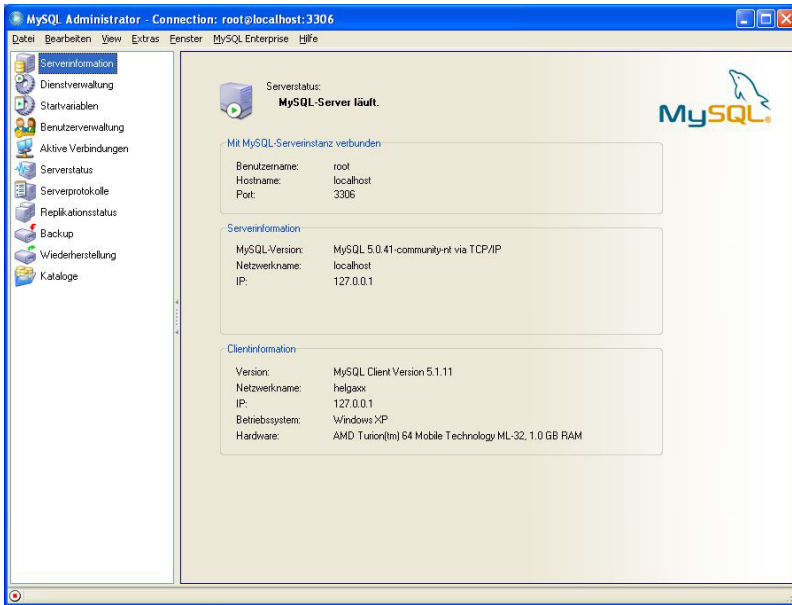


Sie müssen die Adresse des Rechners mit dem Datenbankserver angeben. Dies kann eine TCP/IP-Adresse oder der Name des Rechners im Netzwerk sein. Arbeiten Sie lokal auf einer Maschine, können Sie entsprechend *127.0.0.1* oder einfach *localhost* angeben. Sie erinnern sich, dass MySQL den Benutzer *root* bei der Installation eingerichtet hat? Geben Sie daher hier als Benutzername *root* ein und das Passwort, das Sie bei der Installation gewählt haben, normalerweise *masterkey*. Mehr Informationen haben wir noch nicht und benötigen sie im Moment auch noch nicht.

Nach der erfolgreichen Anmeldung gelangen Sie in das Übersichtsfenster mit allen relevanten Informationen. Die verschiedenen Optionen auf der linken Seite werden uns noch in dem einen oder anderen Fall beschäftigen.

Über den Menüpunkt EXTRAS gelangen Sie zu den anderen Werkzeugen, die auch direkt aufgerufen werden können.

Sie können beispielsweise auch Ihre Verbindung zur Datenbank speichern, wenn Sie sie für künftige Anmeldungen verwenden wollen (siehe Abbildung 3.21).



**Abbildung 3.20**  
Übersichtsfenster des  
MySQL Administrators



**Abbildung 3.21**  
Menüpunkt Extras

Damit ist die Installation der Arbeitsumgebung für MySQL abgeschlossen.

Nach der Installation können Sie die Oberfläche direkt als Windows-Programm starten. Es gibt im Wesentlichen vier Programme unter START/ALLE PROGRAMME/MYSQL:

- MySQL Administrator
- MySQL Migration Toolkit
- MySQL Query Browser
- MySQL System Tray Monitor

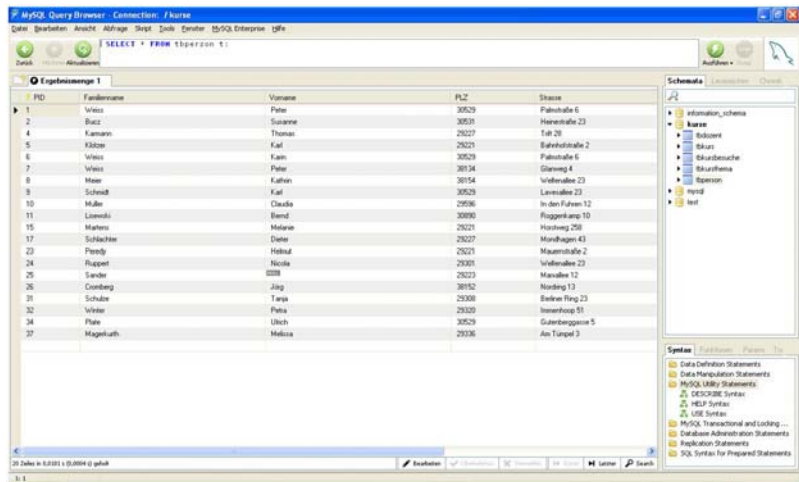
Der Monitor dient der Überwachung des Systems. Mit ihm kann MySQL auch gestartet werden. Sollten Sie beim Aufruf des Administrators oder des Query Browsers eine Fehlermeldung erhalten, so starten Sie mit dem Monitor mit START INSTANCE eine neue Instanz. Sie finden den Monitor unten in der Taskleiste. Sie öffnen ihn mit der rechten Maustaste, die linke startet und beendet den Server ebenfalls.



Der Administrator dient der Administration des Systems, also der Verwaltung von Benutzern und Speicherplatz, der Datensicherung und Wiederherstellung von Daten. Wir wollen darauf in Kapitel 12 zurückkommen.

Der MySQL Query Browser dient der Ausführung der SQL-Kommandos. Er entspricht der *mysql.exe*. Nach dem Start und der Anmeldung erhalten Sie eine Ansicht wie in Abbildung 3.22.

Abbildung 3.22  
Oberfläche MySQL  
Query Browser



Die Grundstruktur erlaubt im oberen Fenster die Eingabe von SQL-Anweisungen. Jede SQL-Anweisung, die Sie im oberen Fenster eingeben, muss durch Drücken der Schaltfläche AUSFÜHREN im rechten Bereich ausgeführt werden. Das Ergebnis wird im unteren Hauptfenster – dem Ergebnisbereich – angezeigt. Dieser Bereich verfügt über Tabs, die immer einschließlich der SQL-Anweisung gelten, sodass Sie immer mit einer Vielzahl von Anweisungen parallel arbeiten können. Rechts haben Sie die verfügbaren Datenbankschemata, die sich erweitern lassen, sodass die Tabellen und auch deren Felder sichtbar werden. Unten finden Sie noch eine sinnvolle Hilfe zur SQL-Syntax.

Weitere Hilfen zum Umgang finden Sie in Anhang A.

### 3.3.5 Eine Beispieldatenbank aufbauen

Um mit MySQL eine Datenbank aufzubauen, sind wie bei fast allen Datenbankmanagementsystemen drei Schritte notwendig. Das entspricht auch der Logik von SQL.

1. Erstellen des Datenbankschemas (manchmal spricht man auch vereinfacht von der Datenbank)
2. Erstellen der Tabellen (auf die weiteren Strukturen verzichten wir hier noch)
3. Eingabe der eigentlichen Daten

## Schritt 1: Datenbankschema erstellen

Das Datenbankschema ist der „Container“, in dem eine Gruppe zusammengehöriger Tabellen zusammengefasst wird. Dies ist die gängige Gruppierung für Projekte, Anwendungen und andere in sich mehr oder weniger geschlossene Informationsbereiche.

Wir wollen jetzt die Datenbankschemata für die Übungsdatenbanken anlegen. Dazu verwenden wir den Benutzer *root*, der bei der Installation eingerichtet wird.

Geben Sie zunächst im Fenster mit der Eingabeaufforderung folgenden Befehl ein, um ein Datenbankschema *kurse* einzurichten:

```
mysqladmin.exe -uroot -pmasterkey CREATE kurse
```

Die wesentliche Syntax ist dabei wie folgt:

```
<pfadname>mysqladmin.exe  
    -uBenutzer [-pPasswort] [CREATE|DROP] Datenbankschema
```

[Aufruf mysqladmin.exe](#)

Mit **CREATE** wird das Datenbankschema erzeugt, wobei es noch nicht vorhanden sein darf. In diesem Fall wird die Anweisung zum Schutz der in dem Schema enthaltenen Daten abgewiesen. Mit **DROP** können Sie ein Schema einschließlich aller enthaltenen Daten löschen.

Die Abkürzung *u* steht für User, *p* für Password. Beachten Sie, dass kein Leerzeichen oder irgendein anderes Zeichen zwischen dem *-u* und dem eigentlichen Benutzernamen respektive zwischen dem *-p* und dem Passwort stehen darf.

Alternativ können Sie als Syntax *--user=root* und *--password=Passwort* verwenden:

```
mysqladmin.exe --user=root --password=masterkey CREATE kurse
```

## Schritt 2: Tabellen erstellen

Nachdem das Datenbankschema eingerichtet ist, kann jetzt auf das Tool *mysql.exe* zurückgegriffen werden, das für normale Anwender ausreichend ist. Die Syntax lautet hier:

```
<pfadname>mysql.exe  
    -uBenutzer [-pPasswort] [Datenbankschema] [< Inputskript]
```

[Aufruf mysql.exe](#)

Das Inputskript enthält wie alle hier verwendeten Skripte Toolaufrufe und SQL-Anweisungen, die ausgeführt werden sollen. Sie können die Skripte mit einem normalen Editor wie dem *Notepad* bearbeiten. Seien Sie aber vorsichtig mit Textverarbeitungsprogrammen wie Word, da Skripte keine Formatierungsanweisungen enthalten dürfen. Prüfen Sie die Skripte und passen Sie Pfade oder Passwörter gegebenenfalls an Ihre Umgebung an.

[Bearbeitung der Skripte](#)

Jetzt kann es losgehen. Zunächst werden die Tabellen angelegt. Dazu können Sie das vorbereitete Skript *CreateKurse.txt* nutzen. Alle Dateien, die jetzt angesprochen werden, finden Sie auf der CD im Verzeichnis *datenbanken\MySQL\kurse*.

[Tabellen anlegen](#)

Die Datei *CreateKurse.txt* beginnt mit folgendem Inhalt:

**Listing 3.1**  
 CREATE TABLE tbPerson  
 DROP TABLE IF EXISTS tbPerson;  
 CREATE TABLE IF NOT EXISTS tbPerson (  
 PID int NOT NULL PRIMARY KEY,  
 Familienname varchar(50) NOT NULL,  
 Vorname varchar(50) NULL,  
 PLZ char(5) NULL,  
 Strasse varchar(50) NULL,  
 Ort varchar(50) NULL DEFAULT 'Celle',  
 Geburtsdatum date);

Im Prinzip handelt es sich dabei um zwei SQL-Anweisungen. Zunächst wird geprüft, ob die Tabelle schon existiert, und diese gegebenenfalls gelöscht. Dann wird eine neue Tabelle *tbPerson* erzeugt. Für diese Befehle benötigen wir den Administrator nicht, da es hier „nur“ um Inhalte geht. Diese SQL-Befehle können wir mit *mysql.exe* ausführen lassen. Wechseln Sie in der Eingabeaufforderung in das Kurs-Verzeichnis */datenbanken/MySQL/kurse* und schicken Sie dann folgenden Befehl ab:

```
mysql.exe -uroot -pmasterkey kurse < CreateKurse.txt
```

Diese Zeile bedeutet im Prinzip:

1. Melde dich als Benutzer *root* bei MySQL an. (Gedanklich haben wir jetzt den Cursor *mysql>* vor uns.)
2. Wähle die Datenbank *kurse* (entspricht einem *mysql>use Kurse*).
3. Führe alle SQL-Befehle aus, die du in der Datei *CreateKurse.txt* findest.
4. Führe die Abmeldung aus (entspricht *mysql>exit*).

Zusätzlich zum Benutzer und zum Passwort muss jetzt das Datenbankschema angegeben werden, das wir im ersten Schritt bereits erzeugt haben. Damit weiß MySQL, in welchem Datenbankschema die Tabellen angelegt werden sollen. Schließlich wird das Skript *CreateKurse.txt* als Input-Datenstrom an das Tool übergeben. Die Verzeichnisstruktur entspricht der Struktur auf Ihrer CD. Bei einer abweichenden Installation müssen hier eventuell Änderungen vorgenommen werden.

**Ergebniskontrolle I** Wir wollen das Ergebnis unserer „Batch“-SQL-Anweisungen zunächst mit *mysql.exe* überprüfen. Dazu wollen wir mit direkten Befehlen und SQL-Anweisungen in *mysql.exe* arbeiten. Zunächst müssen wir uns also wieder anmelden:

```
mysql -uroot -pmasterkey
```

Nach diesem Befehl ändert sich der Cursor wie gehabt zu

```
mysql>
```

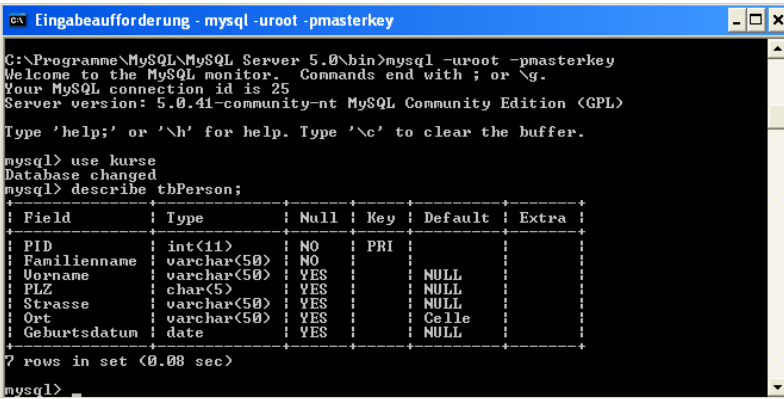
Jetzt müssen wir das richtige Datenbankschema wählen, dazu gibt es die *use*-Option (siehe Abbildung 3.16).

```
mysql>use kurse;
```

Gedanklich befinden wir uns jetzt innerhalb von MySQL und hier wiederum innerhalb der Datenbank `kurse`. Damit haben wir auf alle Tabellen dieser Datenbank Zugriff. Wir wollen dies beispielhaft überprüfen, indem wir uns den Aufbau der Tabelle `tbPerson` anzeigen lassen:

```
mysql>describe tbPerson;
```

Das Ergebnis sollte bei Ihnen etwa wie in Abbildung 3.23 aussehen. Vergleichen Sie das Ergebnis einmal in Ruhe mit der Anweisung `CREATE TABLE tbPerson`, die wir oben als Auszug der Datei `CreateKurse.txt` in Listing 3.1 gesehen haben. Sie sehen das unmittelbare Ergebnis dieser SQL-Anweisung.



```

C:\Programme\MySQL\MySQL Server 5.0\bin>mysql -uroot -pmasterkey
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 5.0.41-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use kurse
Database changed
mysql> describe tbPerson;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| PID | int(11) | NO | PRI | | |
| Familienname | varchar(50) | NO | | | |
| Vorname | varchar(50) | YES | | NULL | |
| PLZ | char(5) | YES | | NULL | |
| Strasse | varchar(50) | YES | | NULL | |
| Ort | varchar(50) | YES | | NULL | |
| Geburtsdatum | date | YES | | NULL | |
+----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>

```

**Abbildung 3.23**  
Tabelle `tbPerson` im  
Kommandozeilenmodus

Wenn Sie das kleine Beispiel mit der zeilenorientierten Oberfläche durchgeführt haben, finden Sie jetzt das Datenbankschema `kurse`, das bereits fünf Tabellen enthält, auch in der Oberfläche der MySQL-Tools. Rufen Sie das Werkzeug `MySQL Query Browser` auf (direkt oder über `EXTRAS` in `MySQL Administrator`). Sie finden im rechten Teil des Bildschirms die Schemata. Eventuell müssen Sie diese aktualisieren (rechte Maustaste), siehe Abbildung 3.24. Mit der rechten Maustaste können Sie außerdem ein Schema zum aktuellen Schema bestimmen. Dieses wird dann in Fettschrift dargestellt. Nur die Tabellen des aktuellen Schemas können direkt verwendet werden, alle anderen Tabellen müssen mit dem Schemanamen qualifiziert werden, also `kurse.tbPerson`, statt einfach `tbperson`. Sie sollten daher das Schema `kurse` als aktuelles Schema auswählen.

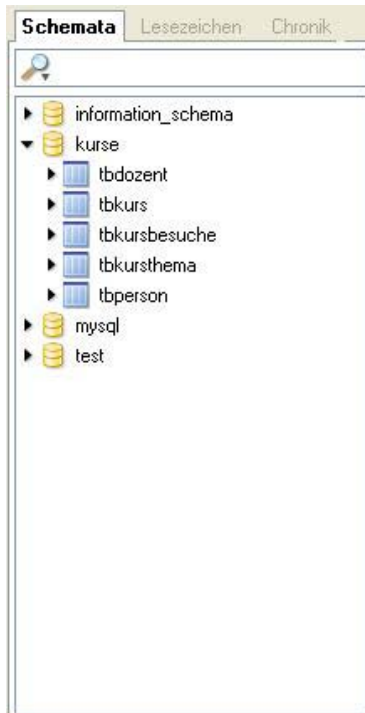
### Ergebniskontrolle II

Die SQL-Anweisung

```
DESCRIBE tbPerson;
```

können Sie jetzt direkt im oberen Fenster eingeben und ausführen (AUSFÜHREN). Sie sollten dann den Inhalt aus Abbildung 3.25 erhalten. Dies ist eine komplette Beschreibung der Tabellenstruktur. Sie sehen in jeder Zeile die Beschreibung eines Datenfeldes mit Namen, Datentyp, der Angabe, ob das Feld leer bleiben darf (`NULL`), ob es Primärschlüssel ist und ob ein Standardwert vorhanden ist. Die Details werden wir in Kapitel 8 betrachten.

**Abbildung 3.24**  
Das Fenster Schemata in MySQL Query Browser mit dem Schema kurse



**Abbildung 3.25**  
Struktur der Tabelle tbPerson

Field	Type	Null	Key	Default	Extra
KID	char(10)	☒☒	NO	PRI	☒☒
Kurskennung	char(50)	☒☒	NO		☒☒
KTHID	int(11)	☒☒	NO		☒☒
KursdauerStunden	int(11)	☒☒	YES	NULL	
Kursbeginn	date	☒☒	YES	NULL	
Kursende	date	☒☒	YES	NULL	
Zertifikat	char(50)	☒☒	YES	NULL	
Gebuehr	float	☒☒	YES	NULL	
Ustpflichtig	char(1)	☒☒	YES	NULL	
DID	int(11)	☒☒	YES	NULL	

Sie haben vielleicht im unteren rechten Teil bereits ein kleines Fenster entdeckt. Dort erhalten Sie Hilfe zu verschiedenen SQL-Aspekten. Diese Hilfe ist sinnvoll, wenn Sie die grundsätzlichen Befehle und die Struktur von SQL verstanden haben. Sie können dann leicht die diversen spezifischen Parameter und Optionen einer SQL-Anweisung nachschlagen. Öffnen Sie beispielsweise `MYSQL UTILITY STATEMENTS`, so finden Sie hier neben dem bereits bekannten `USE` zur Auswahl eines Datenbankschemas auch die SQL-Anweisung `DESCRIBE`. In der Hilfe zu `DESCRIBE` erfahren Sie, dass Sie auch die Abkürzung `DESC` verwenden sowie gezielt einzelne Spalten (columns) auswählen können, beispielsweise mit:

```
DESC tbPerson Ort;
```

### Schritt 3: Daten importieren

Nach der Erzeugung der Tabellenstruktur werden jetzt in einem dritten Schritt die eigentlichen Daten importiert. Dafür wird wieder *mysql.exe* mit identischer Syntax verwendet.

Um gleich eine weitere Tabelle kennenzulernen, sollen zunächst Daten in die Tabelle *tbKurs* geladen werden. Prüfen Sie zunächst die Struktur der Tabelle mit:

```
DESC tbKurs;
```

Wenn Sie auf Ihrer CD in die Datei *Kurse.txt* schauen (*\datenbanken\MySQL\kurse\*), sehen Sie folgende Zeilen:

```
INSERT INTO tbKurs (KID,Kurskennung,KTHID,KursdauerStunden,Kursbeginn,Kursende,Zertifikat,Gebuehr,Ustpflichtig,DID)
```

```
VALUES('CE17','Ce11e17-Word',4,40,'2007-04-23','2007-04-27','eigen',280.00,'J',833);
```

```
INSERT INTO tbKurs (KID,Kurskennung,KTHID,KursdauerStunden,Kursbeginn,Kursende,Zertifikat,Gebuehr,Ustpflichtig,DID)
```

```
VALUES('CE20','Ce11e20-Word',4,40,'2007-07-09','2007-07-13','eigen',280.00,'J',812);
```

```
INSERT INTO tbKurs (KID,Kurskennung,KTHID,KursdauerStunden,Kursbeginn,Kursende,Zertifikat,Gebuehr,Ustpflichtig,DID)
```

```
VALUES('CE23','Ce11e23-Access',5,36,'2007-08-06','2007-08-10','eigen',350.00,'N',812);
```

```
...
```

Es folgen noch eine Reihe weiterer Zeilen. Bei der Datei handelt es sich wieder um SQL-Anweisungen, mit denen die Daten in die Tabelle *tbKurs* eingefügt werden können. Dies erreichen wir mit der Zeile:

```
mysql.exe -uroot -pmasterkey kurse < Kurse.txt
```

Die angegebenen Daten werden in die Tabelle *tbKurs* geladen.

Wir wollen das Laden der Daten in die Tabelle *tbKurs* überprüfen. Dazu genügt uns eine andere einfache SQL-Anweisung, vielleicht die SQL-Anweisung schlechthin:

```
mysql>SELECT * FROM tbKurs;
```

Mit dieser Anweisung werden alle Daten angezeigt, die in der Tabelle *tbKurs* gespeichert sind. Das Ergebnis können Sie der Abbildung 3.26 entnehmen. Achten Sie dabei auf den Zeilenumbruch. Jeder Datensatz und auch alle Erläuterungszeilen sind aus Platzgründen in zwei Zeilen umgebrochen. Vergleichen Sie die ersten drei Zeilen mit den *INSERT INTO*...-Anweisungen, die wir in Listing 3.2 als Ausschnitt der Datei *Kurse.txt* gesehen haben. Sie sehen jetzt das unmittelbare Ergebnis dieser SQL-Anweisungen in der Tabelle *tbKurs*.

**Listing 3.2**  
Einfügen von Daten in die Tabelle *tbKurs*

**Ergebniskontrolle III**

**Abbildung 3.26**  
Inhalt der Tabelle  
tbKurs in mysql.exe

RID	Kurskennung	KTHID	KursdauerStunden	Kursbeginn	Kursende	Zertifikat	Gebuehr	Ustpflichtig	DID
CE17	Celle17-Word	4	40	2007-04-23	2007-04-27	eigen	280	J	833
CE20	Celle20-Word	4	40	2007-07-09	2007-07-13	eigen	280	J	812
CE23	Celle23-Access	5	36	2007-08-06	2007-08-10	eigen	350	N	812
CE24	Celle24-Access	7	40	2007-08-13	2007-08-17	eigen	400	N	812
H89	Hannover89-Access	5	32	2007-07-09	2007-07-13	eigen	350	N	812
H90	Hannover90-Windows	1	18	2007-07-16	2007-07-18	kein	120	J	834
H91	Hannover91-Windows	2	18	2007-07-23	2007-07-25	kein	120	J	834

**Abbildung 3.27**  
Inhalt der Tabelle tbKurs  
in MySQL Query Browser

K...	Kurskennung	KTHID	KursdauerStunden	Kursbeginn	Kursende	Zertifikat	Gebuehr	Ustpflichtig	DID
CE17	Celle17-Word	4	40	2007-04-23	2007-04-27	eigen	280	J	833
CE20	Celle20-Word	4	40	2007-07-09	2007-07-13	eigen	280	J	812
CE23	Celle23-Access	5	36	2007-08-06	2007-08-10	eigen	350	N	812
CE24	Celle24-Access	7	40	2007-08-13	2007-08-17	eigen	400	N	812
H89	Hannover89-Access	5	32	2007-07-09	2007-07-13	eigen	350	N	812
H90	Hannover90-Windows	1	18	2007-07-16	2007-07-18	kein	120	J	834
H91	Hannover91-Windows	2	18	2007-07-23	2007-07-25	kein	120	J	834

Sie können die Eingabe einer SQL-Anweisung jetzt noch mit der ebenfalls bereits aus der zeilenorientierten Oberfläche bekannten Anweisung

```
SELECT * FROM tbKurs;
```

üben, was hoffentlich in MySQL Query Browser im Ergebnis zur Anzeige aller Datensätze der Tabelle tbKurs führt (siehe Abbildung 3.27). Denken Sie an das Ausführen der Anweisung und daran, dass die auszuführende Anweisung markiert sein muss, sonst erhalten Sie die Fehlermeldung:

Sie haben versucht, einen leeren String auszuführen. Bitte geben Sie eine SQL-Anweisung in das Bearbeitungsfeld ein und führen Sie sie dann aus.

Damit wird die Beschreibung der grundsätzlichen Logik von MySQL hier zunächst abgeschlossen. Sie sollten jetzt den grundsätzlichen Umgang beherrschen und mit Details wollen wir uns im Zusammenhang mit den SQL-Anweisungen beschäftigen. Wir wollen jetzt die restlichen Übungstabellen aufbauen, soweit nicht schon geschehen.

**Datenimport** Nachdem Sie die Daten für die erste Tabelle erfolgreich importiert haben, können Sie mit allen weiteren Tabellen genauso verfahren:

```
mysql.exe -uroot -pmasterkey kurse < person.txt
mysql.exe -uroot -pmasterkey kurse < kursbesuche.txt
mysql.exe -uroot -pmasterkey kurse < dozent.txt
mysql.exe -uroot -pmasterkey kurse < kursthema.txt
```

Damit haben Sie die komplette Übungsdatenbank *kurse* in Ihrem MySQL angelegt. Die Funktionsfähigkeit prüfen Sie am besten mithilfe der grafischen Benutzeroberfläche. Geben Sie beispielhaft die SQL-Anweisung

```
SELECT * FROM tabellename;
```

ein und führen Sie sie aus.

Wenn Sie im Query Browser rechts doppelt auf die Tabelle klicken, wird automatisch die passende SQL-Anweisung erzeugt. Sie brauchen sie nur noch auszuführen. Nach der Ausführung können Sie ganz unten die Anzahl ermittelter Datensätze ablesen, beispielsweise „10 Zeilen in 0,0178 s (0,0004 s) geholt“.

### Tip

Sie sollten in *tbDozent* 5 Datensätze, in *tbKurs* 7 Datensätze, in *tbKursbesuche* 18 Datensätze, in *tbKursthema* 11 Datensätze und in *tbPerson* 20 Datensätze finden. Wenn Sie den Inhalt genau überprüfen wollen, sollten Sie Ihre Datenbankinhalte mit den im Anhang abgedruckten Tabellen vergleichen.

Für größere Importe von Daten bietet MySQL noch das zusätzliche Utility *mysqlimport.exe*. Damit können Sie größere Datenmengen komfortabel in die Datenbank einfügen. Insbesondere können tabellarisch aufbereitete Daten sehr schnell eingelesen werden. Diese Daten müssen auch nicht zwingend auf dem Client liegen, sondern können auch direkt auf dem Server liegen. Im Prinzip führt *mysqlimport.exe* die SQL-Anweisung `LOAD DATA INFILE` aus. Wir wollen darauf im Rahmen der Administration und der Utilities zurückkommen. Für unsere kleinen Datenmengen der Übungsdatenbanken und zum Verständnis der SQL-Logik reicht das `INSERT INTO` aus.

### Import

## 3.4 MS Access

### 3.4.1 Gründe für die Nutzung

MS Access ist eines der Datenbanksysteme, die in diesem Buch schwerpunktmäßig als Grundlage der Beispiele und Übungen verwendet werden. Im Fall von MS Access sind die Beispiele für MS Access 2003 erstellt – die zurzeit meistverbreitete Version.

Für die Nutzer älterer Versionen von MS Access (97, 2000) ist hier keine komplette Datenbank beigefügt. Sie können aber, wie die Nutzer aller Versionen von MS Access, eine eigene Datenbank anlegen, indem Sie die beigefügten Excel-Dateien importieren. Das Verfahren ist in Anhang A unter MS Access ausführlich beschrieben.

### Ältere MS Access-Versionen

Für die Nutzer von MS Access 2007 ist die beigefügte Datenbank nutzbar. Sie können sie bei Bedarf auch konvertieren. Bei der Oberfläche gibt es natürlich ein paar Unterschiede, die Sie aber beim Umgang mit SQL nicht besonders behindern sollten.

### Access 2007



**Gründe für MS Access**

MS Access wird hier als eines der Datenbankmanagementsysteme für die Übungen verwendet, obwohl es in vielen Bereichen etwas untypisch ist und von erfahrenen Datenbankanwendern als „kleine Münze“ eingestuft wird. Man sieht da oft so ein virtuelles Naserümpfen. Gleichwohl ist es die Datenbank, mit der viele Anwender als Erstes in Berührung kommen, einfach weil sie als Teil der MS Office-Umgebung von Microsoft den Weg zu vielen Anwendern findet. Es darf auch nicht übersehen werden, dass viele Anwender in Fachabteilungen und auch private Nutzer zunächst MS Access zur Verfügung haben. Die grafische Windows-Oberfläche erleichtert außerdem den Zugang zur Datenbankwelt. Drei wesentliche Gründe haben also dafür gesprochen, trotz aller Probleme und Besonderheiten gerade auch MS Access als eines der Datenbanksysteme zu verwenden, die die Grundlage für die Beispiele in diesem Buch bilden:

- Viele Leser haben MS Access als Übungsdatenbanksystem zur Verfügung. Es ist eines der gängigsten Datenbanksysteme.
- MS Access bietet durch seine Windows-Oberfläche eine vertraute Umgebung, in der viele Arbeiten in gewohnter Umgebung ablaufen können.
- Viele Anwendungen in den Unternehmen entstehen in Fachabteilungen und hier spielt MS Access wegen des breiten Office-Einsatzes eine wichtige Rolle.

Daher werden wir an vielen Stellen gezielt auf die Besonderheiten der grafischen Oberfläche von MS Access eingehen, um den interessierten Anwendern auch deren Nutzung nahezubringen und insbesondere den Zusammenhang zwischen der grafischen Oberfläche und dem darunterliegenden SQL zu erläutern. Dieser Zusammenhang kann auch für diejenigen interessant sein, die nicht MS Access nutzen, da die Struktur der Anweisungen manchmal auch in ihrer Umsetzung in der grafischen Windows-Oberfläche transparenter wird.

### 3.4.2 Eine Beispieldatenbank aufbauen

**MDB-Datei**

Auf die Installation soll hier nicht eingegangen werden. Sie erfolgt im Rahmen der Installation des Office-Paketes. Eine Datenbank für das Datenbankmanagementsystem MS Access besteht aus einer einfachen Datei im Windows-Dateisystem. Ähnlich einer Word- oder Excel-Datei wird für die komplette Datenbank eine Datei angelegt, die die Endung *.mdb* (Microsoft Data Base) erhält.

**Tip**

Sie können MS Access-Datenbanken als Ganzes kopieren, verschieben oder löschen, indem Sie einfach die MDB-Datei kopieren, verschieben oder löschen. Achten Sie aber vor einer solchen Aktion stets darauf, dass die Datenbank nicht geöffnet ist, also kein Benutzer damit arbeitet. Dies können Sie leicht daran erkennen, dass keine Datei mit Sperrern (Locks) existiert. Diese Datei hat denselben Namen wie die MDB-Datei, besitzt aber die Namensendung *.lck*.

Nach dem Start von MS Access können Sie zunächst eine bestehende Datenbank öffnen oder eine neue Datenbank anlegen, wenn Sie DATEN/ÖFFNEN wählen.

Wenn Sie von der mitgelieferten CD aus dem Verzeichnis *datenbanken\MSAccess\kurse* die Datenbankdatei *Kurse2003.mdb* auf Ihre Festplatte kopieren, können Sie schnell starten. Wenn Sie die Datei direkt mit dem Windows-Explorer kopieren, müssen Sie darauf achten, den Schreibschutz zu entfernen, bevor Sie auf die Kopie zugreifen. Dies geschieht im Windows-Explorer durch Anklicken mit der rechten Maustaste, Auswahl der EIGENSCHAFTEN und Entfernen des Hakens für den Schreibschutz.

Öffnen Sie jetzt die Datenbank, indem Sie in MS Access entweder beim Start eine bestehende Datenbank auswählen oder nach dem Start mit DATEI/ÖFFNEN eine Datenbank auswählen. In beiden Fällen müssen Sie zu der Stelle Ihrer kopierten Datei *Kurse2003.mdb* navigieren und diese dann auswählen.

Alternativ können Sie natürlich auch im Windows-Explorer auf die Datei navigieren und sie mit einem Doppelklick starten. Bei der Installation von MS Access wurde die Erweiterung *.mdb* mit MS Access verbunden und führt zum Start des Datenbanksystems.

Sollten Sie beim Start eine Sicherheitswarnung erhalten, können Sie diese ignorieren und die Datenbank öffnen.

Sie können dann links TABELLEN wählen und sollten die fünf Tabellen der Kursdatenbank sehen. Mit einem Doppelklick können Sie deren Inhalt jeweils direkt betrachten.

In der Datenbank stehen die Tabellen nicht nur für sich allein, sondern sind durch Beziehungen miteinander verknüpft. Das ist nicht zwingend erforderlich, erleichtert aber oft die spätere Arbeit. Wählen Sie zur Ansicht der Beziehungen entweder im Hauptmenü EXTRAS/BEZIEHUNGEN oder das entsprechende Symbol in der Titelleiste.



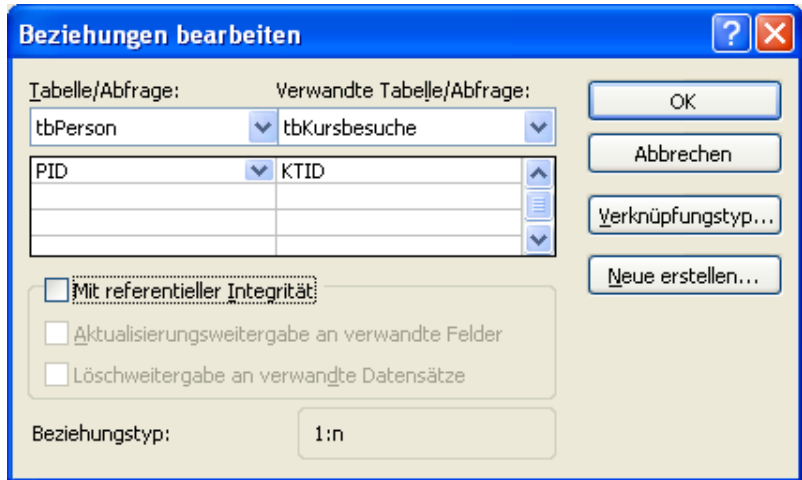
### Beziehungen

Sie bekommen alle Tabellen angezeigt. Verschieben Sie sie und vergrößern Sie sie bei Bedarf so, dass Sie beispielsweise das Ergebnis aus Abbildung 3.1 erreichen. Sie sehen in der Abbildung, dass jeweils zwei Tabellen durch eine Beziehung verbunden sind. Diese Beziehungen enden jeweils auf Höhe eines bestimmten Datenfeldes, beispielsweise ist das Datenfeld PID in der Tabelle *tbPerson* mit dem Datenfeld KTID in der Tabelle *tbKursbesuche* verbunden. Derartige Felder bilden den Primärschlüssel der einen Tabelle und den Fremdschlüssel der anderen Tabelle.

Sie können eine solche Beziehung im Prinzip auf zwei Arten anlegen. Entweder Sie „greifen“ mit der linken Maustaste eines der beiden Felder und ziehen es mit gedrückter Maustaste auf das andere Feld. Sie sehen an der Veränderung des Cursors, wann eine Verbindung zustande kommen kann. Hier lassen Sie das Feld einfach los. Haben Sie sich vertan, markieren Sie die Verbindungslinie und löschen die Verbindung mit der Taste .

Die zweite Möglichkeit ist die Funktion **BEZIEHUNGEN/BEZIEHUNGEN BEARBEITEN**. Hier erhalten Sie ein Fenster wie in **Abbildung 3.28** dargestellt. Sie müssen die an der Beziehung beteiligten Tabellen sowie das Primärschlüssel- und das Fremdschlüsselfeld auswählen. Beachten Sie, dass die Auswahl nicht symmetrisch ist. Sie müssen links als Tabelle die Tabelle mit dem Primärschlüsselfeld angeben. Rechts wird als Detailtabelle die Tabelle mit dem Fremdschlüsselfeld ausgewählt.

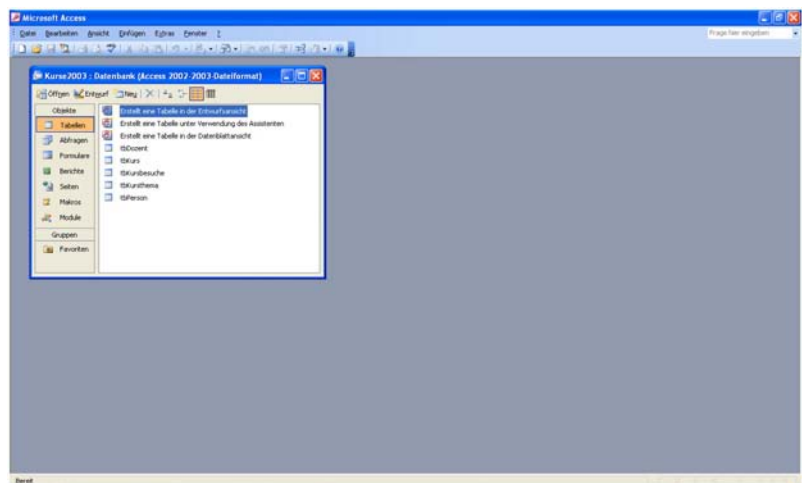
**Abbildung 3.28**  
 Aufbau einer Beziehung  
 zwischen tbPerson und  
 tbKursbesuche in MS  
 Access



In der Datenbank sind die Beziehungen bereits eingetragen. Sie sollten Sie jetzt nicht inhaltlich verändern. Schließen Sie das Beziehungsfenster wieder und speichern Ihre eventuellen grafischen Veränderungen dabei ab.

Damit haben Sie die Kursdatenbank als Musterdatenbank **kurse** vorerst komplett angelegt.

**Abbildung 3.29**  
 Datenbankfenster  
 nach dem Import



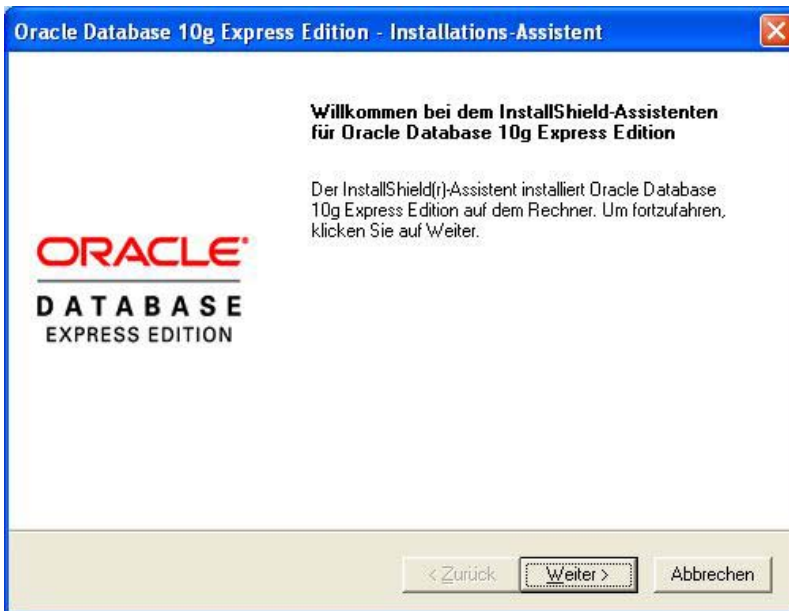
## 3.5 Oracle

### 3.5.1 Gründe für die Nutzung

Oracle ist eines der „großen“ Datenbankmanagementsysteme und neben DB2 das Datenbanksystem, das gerade in Großunternehmen oft das Rückgrat der IT bildet. Oracle ist inzwischen in verschiedenen Versionen in jeder Größenordnung der IT bis hinunter zur Einzelplatzversion für Windows verfügbar und hier auch für nicht kommerzielle Zwecke testweise kostenlos nutzbar. Daher ist es auch hier interessant, es zu nutzen. Oracle war immer wieder Vorreiter bei der Umsetzung komplexer SQL-Erweiterungen, was einerseits für umfangreiche Anwendungen sehr interessant ist, andererseits natürlich umgekehrt aber auch mit dem Nachteil verbunden ist, dass vieles was in Oracle „geht“ in anderen Systemen nicht möglich ist. Wer Oracle bereits kennt oder entscheidet, dass ihn gerade wegen seiner Bedeutung für größere Unternehmen Oracle besonders interessiert, kann die Beispiele auch mit dem beiliegenden *Oracle Express* bearbeiten.

### 3.5.2 Oracle installieren

Zunächst soll hier kurz die Installation der Oracle 10g Express Edition erläutert werden, für die Sie die Hinweise zum Download unter [www.serval.de/SQL](http://www.serval.de/SQL) finden. Starten Sie die Datei auf Ihrem PC. Damit startet der INSTALLATIONS-ASSISTENT (siehe Abbildung 3.30).

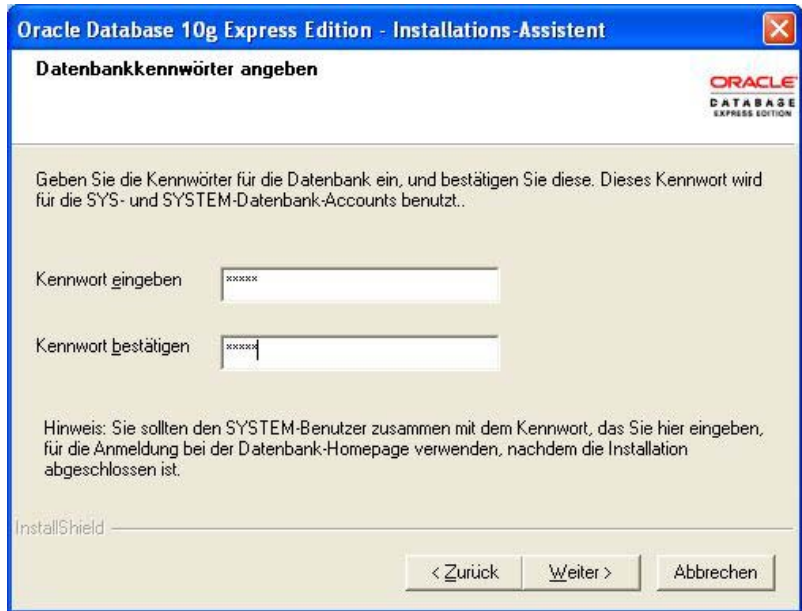


**Abbildung 3.30**  
Startbildschirm des Oracle-Installations-Assistenten

**SYS und SYSTEM**

Nach den gängigen Fragen zur Lizenzierung und zum Zielverzeichnis für die Installation fragt Sie Oracle nach einem Passwort. Dieses Passwort wird gleich für zwei Superuser verwendet, die Oracle *SYS* und *SYSTEM* nennt. Die genauen Unterschiede sind hier nicht relevant, im Wesentlichen wird mit *SYSTEM* gearbeitet, während *SYS* ein „Backup“ für den Administrator darstellt. Verwenden Sie als Passwort am besten wieder *masterkey*, wie in diesem Buch immer empfohlen und worauf auch die Skripte abgestimmt sind.

**Abbildung 3.31**  
Eingabe des Passwortes  
für die Superuser



Danach läuft die Installation des Oracle-Servers als Dienst ab. Sie bekommen die drei wesentlichen Ports angezeigt, die Oracle verwendet. Diese führen zumeist nicht zu größeren Problemen mit der Firewall, wenn doch, finden Sie im Rahmen der Beschreibung der MySQL-Installation eine Beschreibung der Vorgehensweise zur Freigabe einzelner Ports in der Firewall.

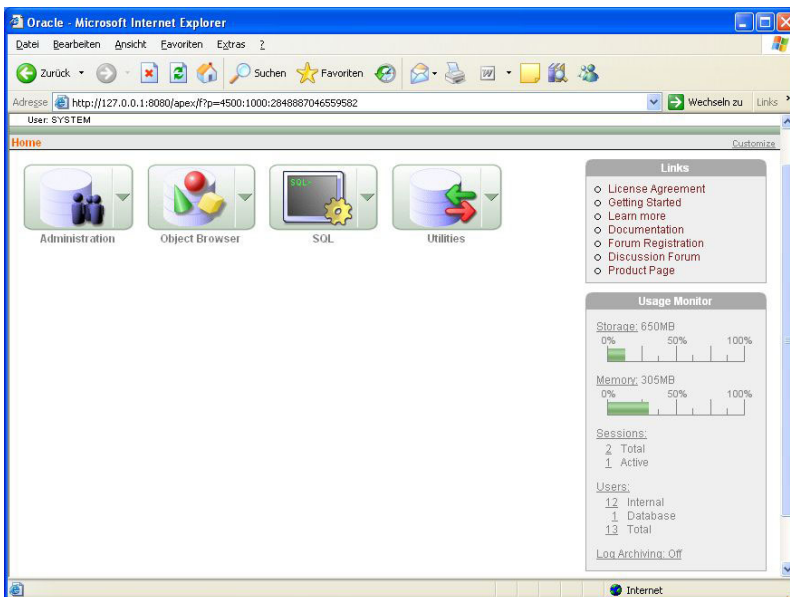
Nach der erfolgten Installation des Servers, die etwas Geduld Ihrerseits erfordert, aber keine weiteren Besonderheiten oder wichtigen Auswahlmöglichkeiten aufweist, können Sie über die normale Windows-Start-Schaltfläche eine Reihe von Funktionen aufrufen (siehe Abbildung 3.32).

Sie sehen, dass Sie den Oracle-Server hier starten und stoppen können, was den Vorteil hat, dass alle Dienste koordiniert werden. Für den Zugriff auf die Datenbank bietet Oracle die SQL-Befehlszeile und die Option *GEHE ZU DATENBANK-HOME PAGE* als grafische Oberfläche an. Auf die Erläuterung der Befehlszeile soll hier verzichtet werden, da dies den Rahmen sprengen würde.



**Abbildung 3.32**  
Oracle-Werkzeuggruppe

Wenn Sie zu der angebotenen Datenbank-Homepage gehen, öffnet sich Ihr Standard-Browser. Die Oberfläche besteht also aus einer Browser-Anwendung. Sie können sich jetzt mit dem Benutzer *SYSTEM* und dem von Ihnen während der Installation angegebenen Passwort anmelden. Danach kommen Sie auf eine Übersichtsseite wie sie die Abbildung 3.33 darstellt.



**Abbildung 3.33**  
Übersichtsfenster der  
Oracle-Umgebung auf  
Browser-Basis

Der Bereich *ADMINISTRATION* dient der kompletten Speicher- und Benutzerverwaltung. Im *OBJECT BROWSER* können Sie sich die Datenbankstruktur und die Inhalte überblickartig anzeigen lassen, während der Bereich *SQL* der Eingabe von SQL-Anweisungen dient und eine zentrale Rolle spielen wird. Die weiteren *UTILITIES* sollen uns hier nicht interessieren. Die einzelnen Bereiche sollen beschrieben werden, wenn wir sie nutzen, was mit der Installation der Beispieldatenbank *kurse* beginnt. Zum Thema „Oracle“ finden Sie im Übrigen in Anhang A eine kurze Beschreibung der Funktionsweise.

### 3.5.3 Die Testdatenbanken importieren

Oracle bietet eine Fülle von Möglichkeiten, um Daten in die Datenbank zu laden. Flexibilität, Performance, Zeichensatzprobleme und Ähnliches bieten hier ein breites Spektrum für Vergleiche. In unserem Rahmen soll nur der einfache Import einer vergleichsweise kleinen Datenmenge im Mittelpunkt stehen, wozu einfache Skripte genutzt werden, die mit dem mitgelieferten *SQLPlus* importiert werden.

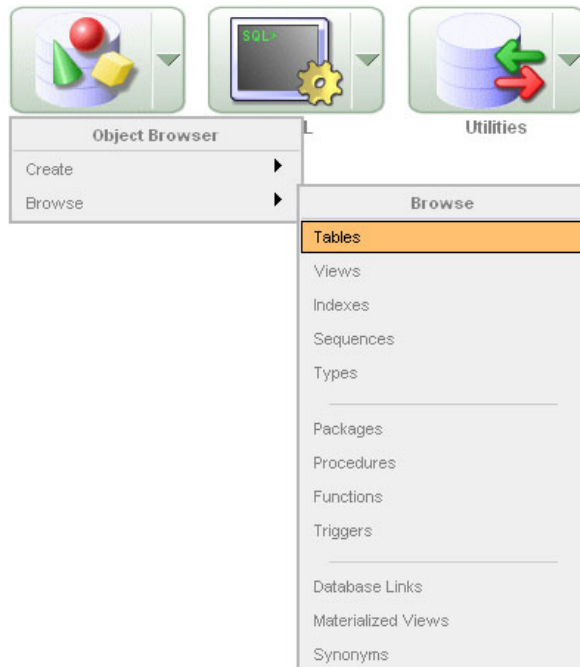
Im ersten Schritt soll das Schema für die Kurs-Beispieldatenbank erzeugt werden. Dazu ist es ausreichend, wenn Sie aus dem CD-Verzeichnis *\Datenbanken\Oracle\kurse* das Skript *OraKurse.bat* durch einen Doppelklick starten. Es wird das ORACLE-Werkzeug *SQLPlus* gestartet. Haben Sie nicht *masterkey* als Passwort verwendet, müssen Sie das Skript kopieren und entsprechend anpassen. Danach läuft das Skript automatisch ab und schließt das Fenster wieder.

Testen Sie den erfolgreichen Ablauf, indem Sie sich über die Browser-Oberfläche anmelden. Verwenden Sie jetzt als Username *kurse* mit dem Passwort *pwkurse*. Hat der Import funktioniert, können Sie über den Objekt-Browser das Ergebnis kontrollieren.

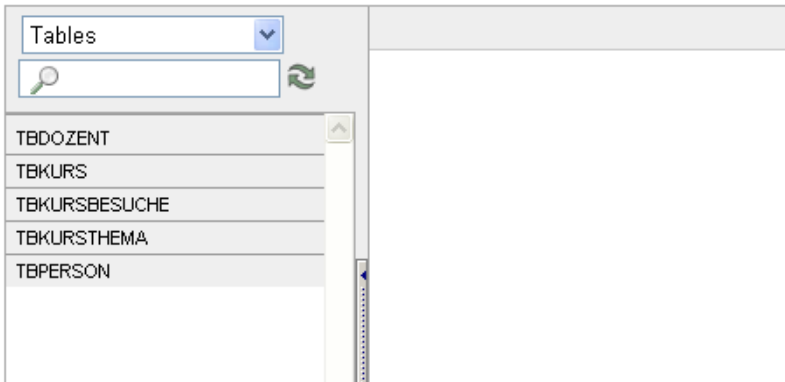


Durch Ausführung des Skripts haben Sie ein Datenbankschema erzeugt, das wie unser Benutzer *kurse* heißt. Sie können dieses Schema jetzt inhaltlich kontrollieren, indem Sie das Auswahlménü des OBJECT BROWSER öffnen, auf BROWSE und dann auf TABLES verzweigen (siehe Abbildung 3.34).

**Abbildung 3.34**  
Der Object Browser erlaubt die Kontrolle des Inhaltes.



Sie kommen in das Übersichtsfenster des Browsers. Hier sind zunächst die fünf Tabellen des Schemas für die Kursdatenbank vorhanden (siehe Abbildung 3.35). Sie können jeweils auf die Tabellennamen klicken, um sich die Struktur der einzelnen Tabellen anzusehen.



**Abbildung 3.35**  
Tabellen des  
Schemas kurse

Sollte die Anmeldung nicht funktionieren oder sollten keine Tabellen vorhanden sein, öffnen Sie die Datei *CreateKurse.sql* im *Oracle*-Ordner unter *Datenbanken* auf Ihrer CD mit einem normalen Editor wie dem *Notepad*. Entfernen Sie die letzte Anweisung `exit` aus der Datei und speichern Sie sie neu auf Ihrer Festplatte ab. Versuchen Sie danach erneut das Skript *OraKurse.bat* zu starten. Sie sollten jetzt nach Eingabe des Passwortes die Fehlermeldungen sehen, die das Problem verursacht haben. Analysieren Sie sie insbesondere im Hinblick auf Benutzer, Passwort und eventuell Verzeichnisse und versuchen Sie es erneut. Wahrscheinlich haben Sie ein falsches Verzeichnis oder ein falsches Passwort verwendet.

#### Fehlerfall

Neben der Erzeugung der Struktur sind durch den Aufruf des Skriptes auch die eigentlichen Daten importiert worden. Dazu dienen die weiteren Zeilen in *OraKurse.bat*, die einen Aufruf und eine Ausführung der SQL-Anweisungen in den Dateien *person.sql*, *kursthema.sql*, *dozent.sql*, *kurse.sql* und *kursbesuche.sql* bewirken.

#### Datenimport

Den Erfolg können Sie wieder im Objekt-Browser überprüfen. Nach Auswahl einer Tabelle im linken Teil des Fensters (siehe Abbildung 3.36) können Sie rechts zwischen verschiedenen Darstellungen wählen. Die Basisdarstellung *TABLE*, die die Struktur der Tabelle zeigt, kennen Sie bereits. Die zweite Darstellung, *DATA*, unmittelbar daneben gibt die in die Tabelle geladenen Daten wieder. Die weiteren Darstellungen geben Strukturen an, die erst später besprochen werden. Interessant ist noch die letzte Darstellung, *SQL*, die die SQL-Anweisung wiedergibt, mit der die Tabelle erzeugt wurde. Sie können dies jeweils mit der Angabe in der Datei *CreateKurse.sql* vergleichen, die wir zum Aufbau der Struktur verwendet haben.



Abbildung 3.36  
Tabelle Person

EDIT	PID	FAMILIENNAME	VORNAME	PLZ	STRASSE	ORT	GEBURTSDATUM
	1	Weiss	Peter	30529	Palmstrasse 6	Hannover	07.11.83
	2	Bucz	Susanne	30531	Heinestrasse 23	Hannover	06.04.76
	4	Karmann	Thomas	29227	Tritt 28	Celle	04.08.54
	5	Kloetzer	Karl	29221	Bahnhofstrasse 2	Celle	13.03.71
	6	Weiss	Karin	30529	Palmstrasse 6	Hannover	05.10.62
	7	Weiss	Peter	38134	Glanweg 4	Braunschweig	02.03.74
	8	Meier	Kathrin	38154	Welfenallee 23	Braunschweig	03.05.81
	9	Schmidt	Karl	30529	Lavesallee 23	Hannover	25.06.49
	10	Mueller	Claudia	29596	In den Führen 12	Breitenhees	-

Die komplette Struktur der Tabellen mit ihren Beziehungen ist der Abbildung 3.1 zu entnehmen.

### Import-Utility

Für den Import umfangreicher Daten kann in Oracle das Original-Import-Utility *imp.exe* verwendet werden. Oracle bietet seit der Version 10 auch die Datenpumpe *impdp.exe* an. Da aber eventuell ältere Oracle-Versionen zum Einsatz kommen, wird diese hier noch nicht verwendet. Für unsere Zwecke reicht aber der Weg über die konventionellen SQL-Anweisungen aus, wie er hier beschrieben wurde.

Weitere Hinweise zur Benutzung der Oracle-Oberfläche finden Sie in Anhang A.

Abfragen können Sie in Oracle ausführen, indem Sie auf der HOME-Seite das Auswahlménú zur SQL-Schaltfläche aufklicken und den Befehl SQL COMMANDS/ENTER COMMAND auswählen. Danach können Sie im oberen Fenster beispielsweise

```
SELECT * FROM tbPerson;
```



eingeben und anschließend die Schaltfläche RUN verwenden. Im unteren Bereich erscheint dann das Ergebnis.

## 3.6 Firebird

Firebird ist der Nachfolger und die Weiterentwicklung des Borland-Datenbankmanagementsystems *Interbase*. Es handelt sich wiederum um eine für Testzwecke frei nutzbare Datenbank, die dank der Vorarbeiten in Interbase eine erstaunliche Stärke und Robustheit für den produktiven Einsatz aufweist. Die geringe Verbreitung ist sicher in erster Linie auf ihre Herkunft aus dem Haus Borland, jetzt CodeGear (Embarcadero), zurückzuführen. Der Ruf von Borland ist der eines hervorragenden Lieferanten von Softwareentwicklungsumgebungen wie Delphi, C++-Builder und inzwischen einer Reihe von

Umgebungen für die Internetprogrammierung, aber eben nicht der eines Datenbankherstellers. Das Schattendasein des Systems ist daher eher auf die fehlende Positionierung im Markt als auf technische Schwächen zurückzuführen.

Das System wurde hier trotz der vergleichsweise geringen Verbreitung auch aufgenommen, da es gerade für Programmierer von Anwendungen mit einer embedded Database eine interessante Alternative bietet. Firebird bietet einen vergleichsweise geringen Funktionsumfang, der sich aber mithilfe sogenannter User Defined Functions (UDF) schnell erweitern lässt. Das System ist sehr schlank und stabil. Es ist aber als eigenständige Datenbank mit grafischer Oberfläche eher untypisch.

Damit Sie dieses System ebenfalls für die Beispiele des Buches nutzen können, wird kurz die Installation und die Bereitstellung der Beispieldatenbanken beschrieben. Die Beispiele sind aber – ähnlich wie bei Oracle – nicht schwerpunktmäßig auf Firebird abgestimmt, sodass einige Beispiele wegen der etwas unterschiedlichen Philosophie von Firebird mangels vorinstallierter Funktionen nicht genutzt werden können.

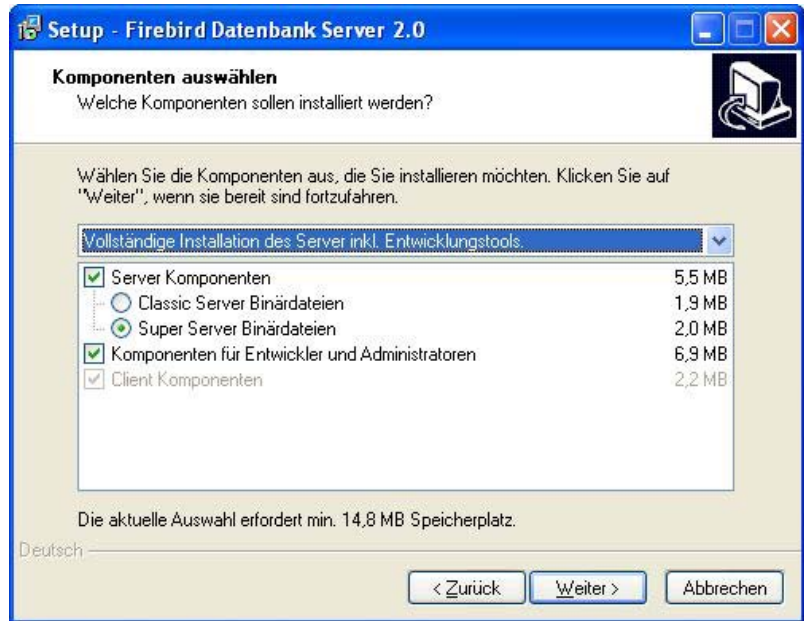
Sie finden alle Hinweise zum Download unter [www.serval.de/SQL](http://www.serval.de/SQL). Starten Sie die Installation der Firebird-Version 2.0.3 mit der entsprechenden Datei *Firebird-2.0.3.12981-1-Win32.exe*. Der Installationsassistent meldet sich mit dem Logo in der Abbildung 3.37.



**Abbildung 3.37**  
Start des Firebird-  
Installationsassistenten

Sie können jetzt wie üblich durch die Installation gehen, indem Sie zunächst die Lizenzvereinbarung lesen. Der folgende Hinweis auf die geänderte Struktur ist nur für die bisherigen Nutzer des Systems relevant. Installieren Sie das System das erste Mal, können Sie den Hinweis ignorieren, alle Dateien sind auf die Version 2 abgestimmt.

**Abbildung 3.38**  
Wahl des Super Servers



Wenn Sie weiter dem Installationsassistenten folgen, gelangen Sie schließlich zu der Abfrage in Abbildung 3.38. Wesentlich ist die Wahl der CLASSIC SERVER- oder der SUPER SERVER-Komponenten. Wegen der besseren Unterstützung ist unter Windows in jedem Fall der SUPER SERVER vorzuziehen.

Die weitere Auswahl ist in Abbildung 3.39 angegeben. Der Guardian ist ein nützliches Hilfsmittel zur Kontrolle des Servers. Die Installation als Server folgt der Logik der anderen Datenbanken. Der automatische Start des Servers bei Systemstart sollte wiederum je nach Häufigkeit der Nutzung entschieden werden.

Nach einigen weiteren Standardeinstellungen ist die Installation abgeschlossen und Firebird bestätigt dies, wie in Abbildung 3.40 zu sehen. Die Installation können Sie in der Windows-Systemsteuerung kontrollieren. Dort müsste bei Ihnen als zusätzliches Symbol der „Guardian“ (Firebird Server Manager) auftauchen. Er dient der Kontrolle des Firebird-Servers.

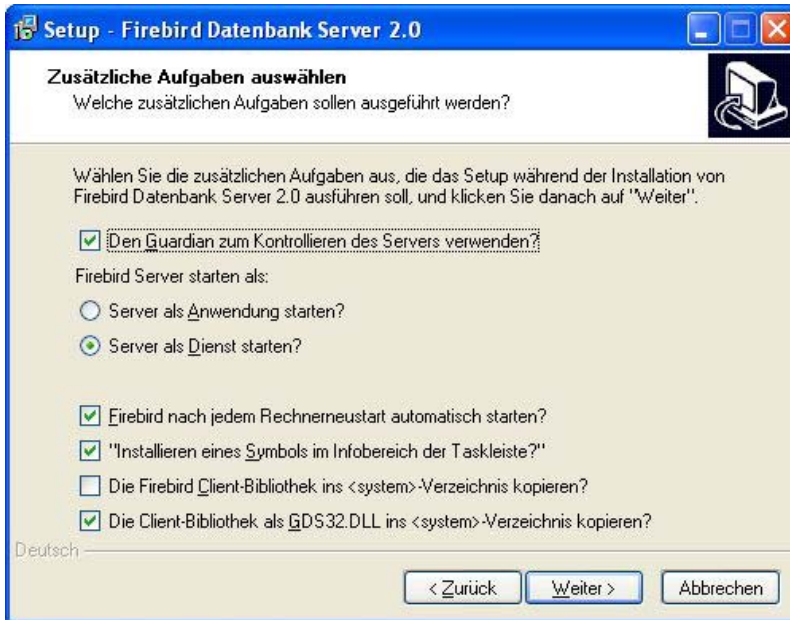


Abbildung 3.39  
Konfiguration der  
Firebird-Installation



Abbildung 3.40  
Abschluss der Installation

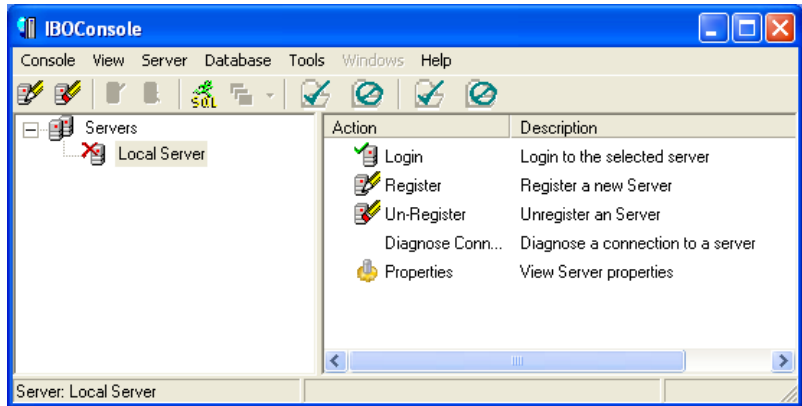
Firebird bietet ähnlich wie MySQL nicht direkt eine grafische Benutzeroberfläche. Allerdings gibt es wieder wie bei MySQL Werkzeuge anderer Anbieter. Eines dieser Werkzeuge der Firma IB, die *IBOConsole*, ist ebenfalls über

einen Verweis auf *www.serval.de/SQL* erreichbar und kann für Testzwecke genutzt werden. Die Installation besteht lediglich aus einem Entpacken des Werkzeugs in ein Verzeichnis Ihres Rechners.

Sie können danach die Datei *IBOConsole.exe* direkt starten. Auf Dauer ist es aber sinnvoller, einen Link auf dem Desktop oder im Programmsystem einzufügen, um den Aufruf zu vereinfachen.

Die Oberfläche der IBOConsole ist in Abbildung 3.41 erkennbar.

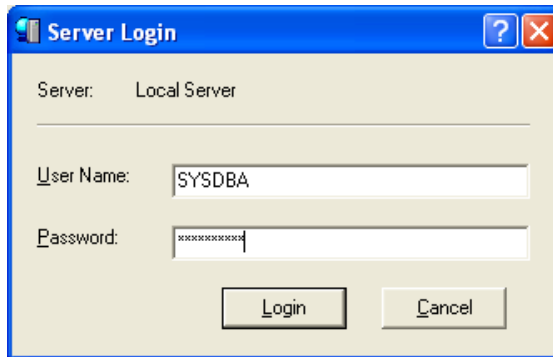
**Abbildung 3.41**  
Anmeldung am  
Firebird-Server



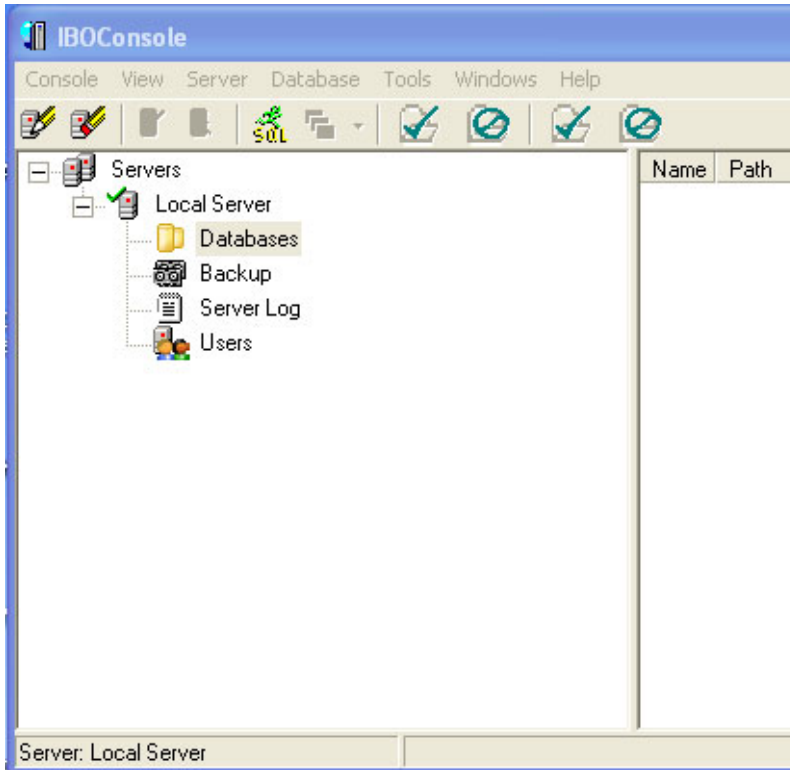
Melden Sie sich beim Server an, entweder über die Funktion im rechten Fensterenteil (LOGIN) oder über die rechte Maustaste (jeweils nach Markierung des Servers).

**SYSDBA** Nutzen Sie dazu den Superuser *SYSDBA* mit dem automatisch erzeugten Passwort *masterkey* (siehe Abbildung 3.42).

**Abbildung 3.42**  
Anmeldung mit dem  
Superuser SYSDBA



Der Server öffnet sich und zeigt die enthaltenen Bestandteile an. Es sind jetzt, wie in Abbildung 3.43 zu sehen, noch keine Datenbanken vorhanden.



**Abbildung 3.43**  
Nach der Anmeldung  
beim Server

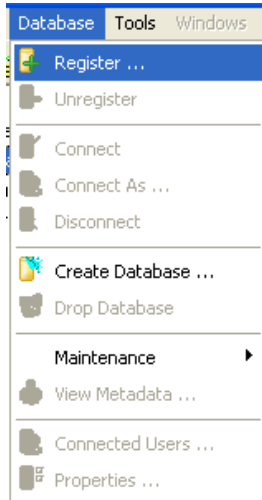
Jetzt können Sie eine Datenbank anlegen. Dazu finden Sie auf der CD unter `\Datenbanken\Firebird\kurse` die Datei `FireKurse.bat`. Diese Datei lädt die in `CreateKurse.sql` niedergeschriebenen SQL-Anweisungen in Firebird, um die Datenbank `kurse` zu erzeugen. Dabei wird eine Datei `Kurse.fdb` (Firebird Database) angelegt.

In der ersten Zeile finden Sie eine Pfadangabe für die Ablage der Datei. Diese entspricht der Standardinstallation. Prüfen Sie, ob der Pfad in der ersten Zeile für Ihr System gültig ist:

```
CREATE DATABASE 'C:\Programme\Firebird\Firebird_2_0\KURSE.fdb';
```

Diese Datei müssen Sie noch beim Server registrieren, damit er sie in seine Dateiverwaltung aufnimmt. Rufen Sie dazu den Menüpunkt `DATABASE/REGISTER` in der Konsole auf.

**Abbildung 3.44**  
 Registrierung der Firebird-  
 Datenbankdatei

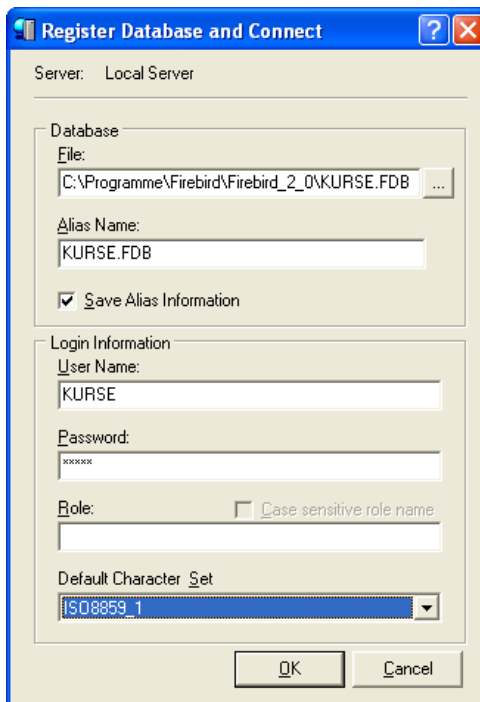


Klicken Sie auf die ...-Schaltfläche und folgen Sie im SELECT DATABASE-Dia-  
 log dem Pfad zu der oben angegebenen Datei, im Normalfall also

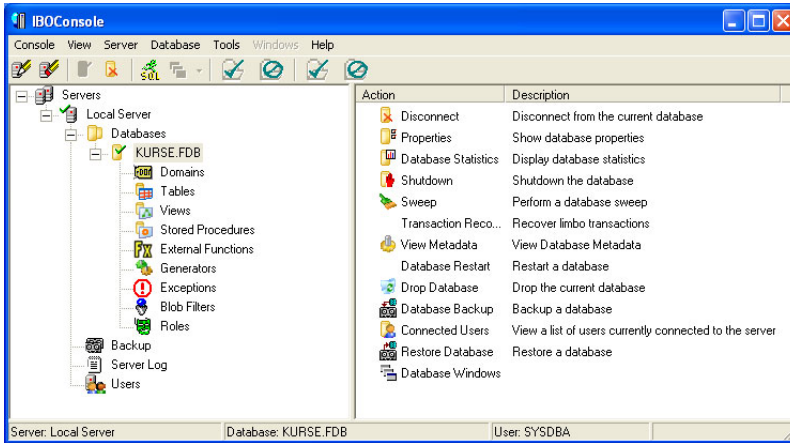
*C:\Programme\Firebird\Firebird\_2\_0*

und wählen Sie die Datei *kurse.fdb* aus (siehe Abbildung 3.45). Sie können  
 noch eine eigene Anmeldung sowie einen Zeichensatz angeben.

**Abbildung 3.45**  
 Registrierung der  
 Datenbank

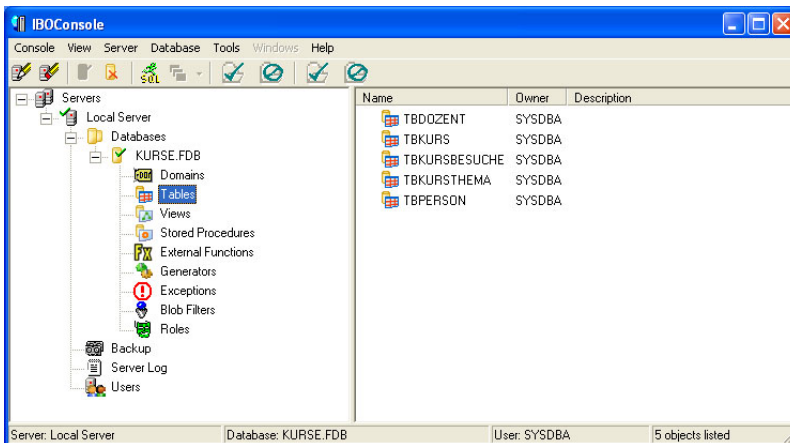


Nach der Registrierung der Datenbank können Sie sich im Hauptfenster mit einem CONNECT anmelden und erhalten die Übersicht in Abbildung 3.46.



**Abbildung 3.46**  
Übersicht über die Datenbank KURSE nach der Anmeldung

Wählen Sie beispielsweise TABLES, dann erhalten Sie die Übersicht in Abbildung 3.47.



**Abbildung 3.47**  
Tabellenübersicht

Hier können Sie sich jetzt über den aktuellen Stand informieren. Ein Hinweis soll an dieser Stelle noch erfolgen. Sollten Sie im Lauf der Arbeit wieder auf den aktuellen Stand kommen wollen, können Sie die Datenbank mit DROP DATABASE entfernen und dann wie beschrieben neu aufbauen. Das Kommando ist auch im Skript *LoescheFireKurse* enthalten. Führen Sie es bei geschlossener Datenbank aus und starten danach wieder *FireKurse*, so haben Sie den Ausgangszustand wiederhergestellt.



**Abbildung 3.48**  
Entfernen der Datenbank



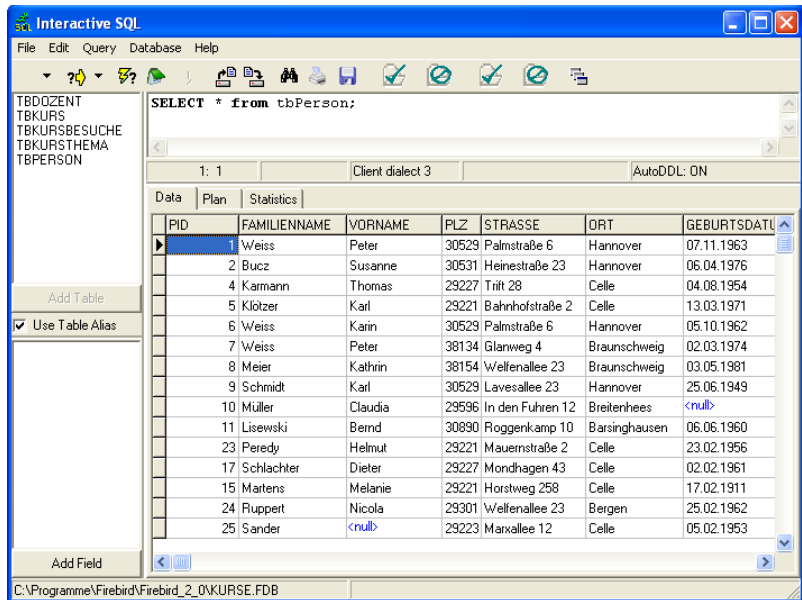
Wenn Sie eine SQL-Abfrage durchführen wollen, wählen Sie TOOLS/INTERACTIVE SQL aus.

Geben Sie im oberen Fenster einfach die SQL-Anweisung ein, beispielsweise `SELECT * from tbPerson;`



und betätigen Sie die Schaltfläche mit dem gelben Blitz oder wählen Sie im Menu QUERY/EXECUTE. Sie sehen das Ergebnis der Abfrage im unteren Fenster. Es sollte etwa wie in Abbildung 3.49 aussehen.

Abbildung 3.49  
Tabelle in Firebird



Weitere Hinweise zum Umgang mit dem „Feuervogel“ finden Sie in Anhang A.

### 3.7 OpenOffice.orgBase

Abschließend sehen wir uns jetzt noch eine Alternative zum Einsatz von MS Access an, die auch viele Ähnlichkeiten mit der MS Access-Umgebung aufweist. So steht mit der Datenbankkomponente *orgBase* von OpenOffice 2.2 eine weitere Datenbank zur freien Nutzung für die hier benötigten Beispiele zur Verfügung. Die Einbindung in eine Office-Umgebung und die für viele Aufgaben unmittelbar nutzbare grafische Oberfläche sind MS Access nicht unähnlich. In der Art der Ausstattung und Nutzung, bei den Datentypen und auch beim SQL-Dialekt gibt es allerdings auch gravierende Unterschiede. So ähnelt MS Access oft mehr einer Entwicklungsumgebung für Datenbankanwendungen unter Windows und speziell unter MS Office, während openOffice.orgbase mehr Ähnlichkeiten mit den „großen Geschwistern“ MySQL oder Firebird aufweist. Auch wenn dies nicht ganz korrekt ist, wollen wir die Datenbank im Folgenden entsprechend der Oberfläche immer kurz als „openBase“ bezeichnen.

openBase wurde hier als „freie“ Alternative zu MS Access aufgenommen. Es ist neben MS Access hier die einzige Datenbank, die direkt mit einer grafischen Oberfläche ausgeliefert wird und somit unmittelbar kontrollierbar ist. Eine Datenbank in openBase ist analog der MS Access-Datei mit der Erweiterung *.mdb* eine einzelne Datei mit der Erweiterung *.odb*. Natürlich werden wir auch hier die direkte Eingabe der SQL-Anweisungen vorstellen, die Oberfläche bietet aber einen schnellen Überblick. Der Funktionsumfang ist sehr gut und stärker am Standard angelehnt als bei MS Access, dafür fehlt weitgehend die Funktionalität eines echten Multiuserbetriebes.

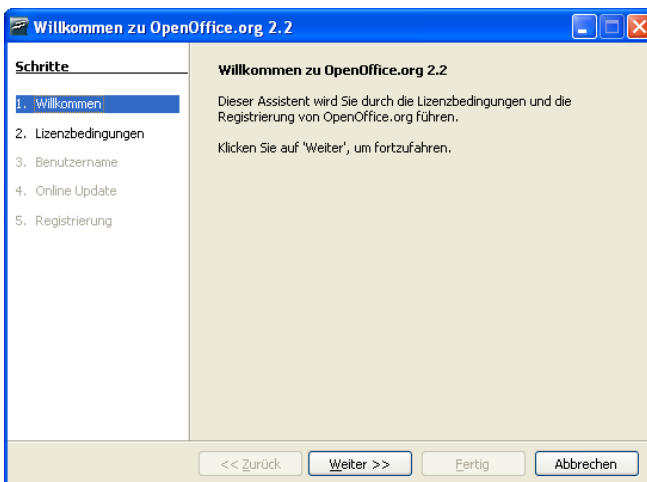
Basis der Datenbank openBase ist die sogenannte HSQL-Engine. Daneben unterstützt openBase auch die Weiterleitung von Anweisungen an andere Datenbanksysteme. Hier wird aber die eigentliche und eigene HSQL-Engine für die Beispiele mit openBase verwendet. Unter dem entsprechenden Schlagwort finden Sie auch die meiste Hilfe im Internet.

HSQL

Wenn Sie openBase als Beispielumgebung für die Übungen dieses Buches nutzen wollen, müssen Sie zunächst OpenOffice installieren. Die Version finden Sie auf der CD im Unterordner *OpenBase*.

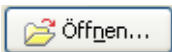
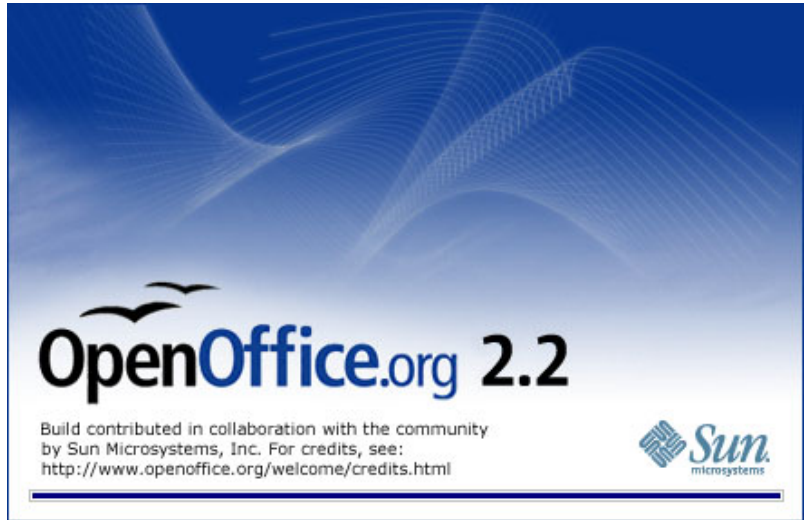
Nach dem Start von *OOosetup.exe* meldet sich der Installationsassistent, der Ihnen ein Installationsverzeichnis vorschlägt. Sie können das komplette Paket oder nur openBase (*OpenOffice.org Base*) installieren. Für unsere Zwecke reicht die Base-Komponente aus. Spätestens, wenn Sie aber Daten importieren oder exportieren wollen, benötigen Sie zumindest noch die Calc-Komponente.

Die weiteren Fragen können Sie entsprechend Ihres Systems beantworten. Beachten Sie die Lizenzbedingungen. OpenOffice ist eine „freie“ Software, die aber den üblichen Restriktionen hinsichtlich einer kommerziellen Nutzung unterliegt. In diesem Buch wird die Version 2.2 verwendet, andere Versionen sind aber ähnlich. Gleiches gilt auch für StarOffice mit Star Base. Natürlich gibt es stets neuere Versionen, die aber zumindest den hier vorausgesetzten Funktionsumfang beinhalten sollten.



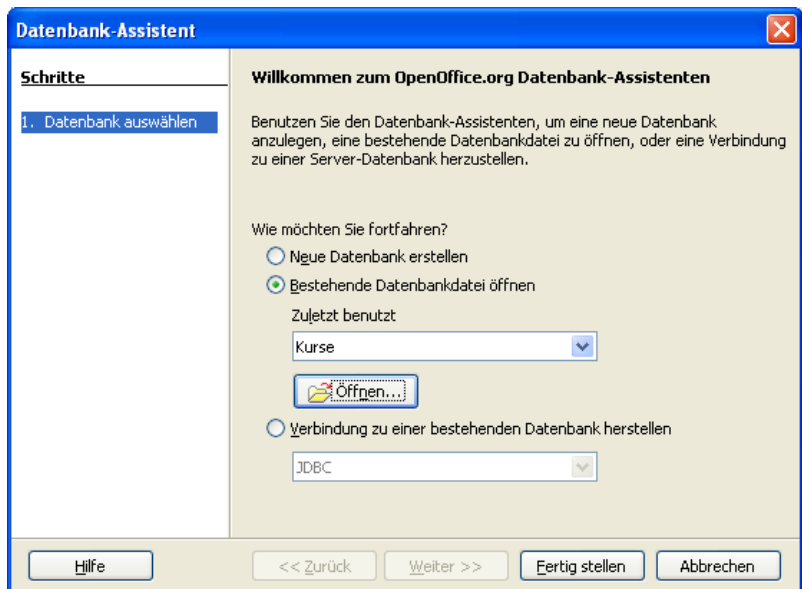
**Abbildung 3.50**  
Start des Installationsassistenten für OpenOffice

Abbildung 3.51  
Installation von OpenOffice



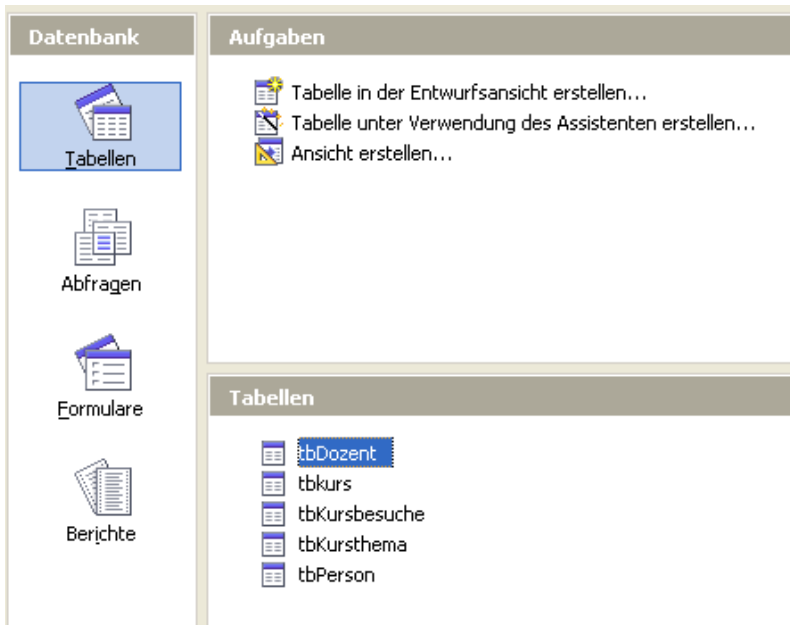
Nach der Installation von OpenOffice können Sie openBase aufrufen. Suchen Sie zuvor auf der CD unter *datenbanken/openBase/kurse* nach der Kursdatenbank *kurse.odt*. Wenn Sie die Datei direkt mit dem Windows-Explorer in ein Verzeichnis Ihres Rechners kopieren, müssen Sie an den Schreibschutz denken. Öffnen Sie die kopierte Datei mit einem rechten Mausklick, wählen Sie EIGENSCHAFTEN und entfernen Sie gegebenenfalls den Haken für den Schreibschutz. Danach können Sie die Datei in openBase mit der Option BESTEHENDE DATENBANKDATEI ÖFFNEN, der Wahl der Datei und der Schaltfläche ÖFFNEN laden.

Abbildung 3.52  
Öffnen der Datenbank



Damit steht die Datenbank zur Bearbeitung zur Verfügung. Wählen Sie **TABELLEN**, um eine Übersicht über alle Tabellen zu bekommen. Dies kann beim ersten Mal einen Augenblick dauern. Sie sehen eine Datenbank mit den Tabellen (siehe Abbildung 3.53). Den Inhalt der Tabellen können Sie durch Auswahl mit einem einfachen Klick rechts als Vorschau sehen. Den kompletten Inhalt einer Tabelle können Sie sich mit einem Doppelklick auf den Tabellennamen oder über die rechte Maustaste mit der Option **ÖFFNEN** ansehen. Sie sehen den kompletten Tabelleninhalt und können ihn mit dem Inhalt im Anhang vergleichen.

Das Fenster schließen Sie mit **DATEI/SCHLIESSEN**.



**Abbildung 3.53**  
Übersicht Tabellen in  
openBase

Die Beziehungen zwischen den Tabellen können Sie über **EXTRAS/BEZIEHUNGEN** sichtbar machen. Sie finden eine ähnliche Darstellung wie in Abbildung 3.1. Durch einen Doppelklick auf eine Beziehung können Sie die beteiligten Felder sowie die beim Ändern oder Löschen von Daten durchzuführenden Aktionen sehen. Die Bedeutung der Beziehungen sehen wir uns in Kapitel 8 an. Sie können dieses Fenster wieder mit **DATEI/SCHLIESSEN** beenden.

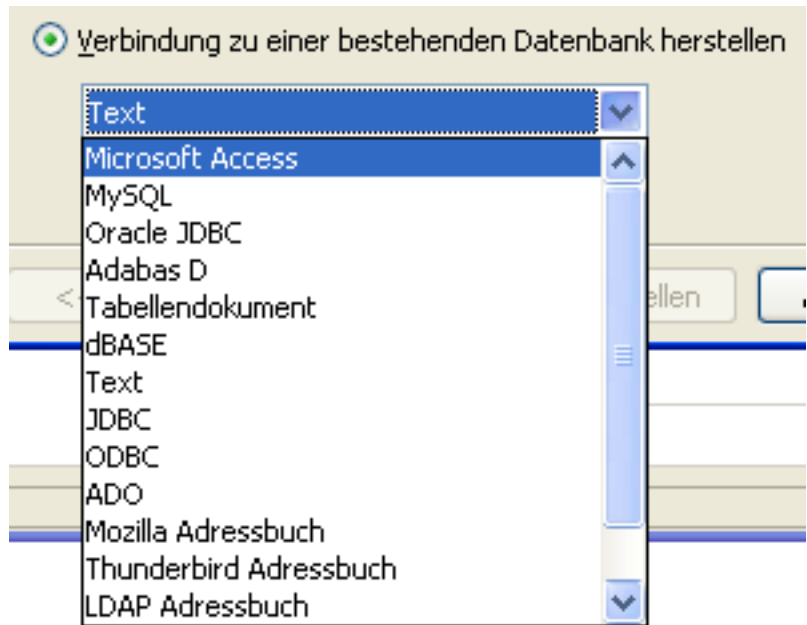
### Beziehungen

Noch eine Anmerkung für die Nutzer von OpenOffice: openBase kann auch als „Oberfläche“ für andere Datenbankmanagementsysteme verwendet werden. So finden Sie in der Liste möglicher Datenbanken in Abbildung 3.54 auch bekannte Systeme wie MS Access, MySQL und Oracle. Auch die Standardverbindungen ODBC, JDBC und ADO stehen zur Verfügung. Die Nutzung dieser Verbindungen kann openBase zu einer attraktiven Oberfläche und zu einer Alternative zu den hier vorgestellten Oberflächen für die ande-

ren Datenbanken machen. Wegen des teilweise untypischen Verhaltens der Oberfläche und Irritationen bei dem zu verwendenden SQL soll dies hier aber nicht geschehen.

Wenn Sie sich mit openBase und dem anderen Datenbanksystem aber schon ein wenig auskennen, können Sie diesen Weg aber probeweise nutzen.

**Abbildung 3.54**  
Verbindung aus  
OpenOffice zu anderen  
Datenbanksystemen



Abschließend soll eine erste Abfrage in openBase erstellt werden, um den Mechanismus zu verstehen. Wählen Sie dazu links ABFRAGEN. Sie haben nun unter AUFGABEN die Möglichkeit, eine Abfrage in der Entwurfsansicht mithilfe eines Assistenten oder in der SQL-Ansicht zu erstellen.

Der Assistent soll uns hier nicht interessieren. Die Entwurfsansicht entspricht der grafischen Oberfläche, die SQL-Ansicht einem reinen Eingabefenster. Wählen Sie die SQL-Ansicht. Sie erhalten ein leeres Fenster. Geben Sie ein:

```
SELECT * FROM tbPerson;
```



Führen Sie die Abfrage anschließend aus, indem Sie entweder das Symbol mit dem grünen Haken oder den Menübefehl BEARBEITEN/ABFRAGE AUSFÜHREN auswählen. Sie sollten eine Tabelle wie in Abbildung 3.55 erhalten.



Sie können dieselbe Abfrage auch in der grafischen Oberfläche von openBase darstellen. Dafür müssen Sie entweder die Schaltfläche mit dem gelben Konstruktionsdreieck oder im Menü ANSICHT/DESIGN-ANSICHT AN-/AUSSCHALTEN auswählen. Das Ergebnis ist die SQL-Anweisung in der Design-Ansicht und sollte etwa wie in Abbildung 3.56 aussehen.

	PID	Familienname	Vorname	PLZ	Ort	Strasse	Geburtsdatum
▶	1	Weiss	Peter	30529	Hannov	Palmstraße	07.11.63
	2	Bucz	Susanne	30531	Hannov	Heinestraße	06.04.76
	4	Karmann	Thomas	29227	Celle	Trift 28	04.08.54
	5	Klötzer	Karl	29221	Celle	Bahnhofstra	13.03.71
	6	Weiss	Karin	30529	Hannov	Palmstraße	05.10.62
	7	Weiss	Peter	38134	Braunsc	Glanweg 4	02.03.74
	8	Meier	Kathrin	38154	Braunsc	Welfenallee	03.05.81
	9	Schmidt	Karl	30529	Hannov	Lavesallee	25.06.49
	10	Müller	Claudia	29596	Breitenl	In den Fuhr	
	11	Lisewski	Bernd	30890	Barsingl	Roggenkam	06.06.60
	15	Martens	Melanie	29221	Celle	Horstswen ?	17.02.61

Datensatz 1 von 14 \*

```
SELECT * FROM "tbPerson"
```

**Abbildung 3.55**  
Ergebnis der Abfrage  
in openBase

	PID	Familienname	Vorname	PLZ	Ort	Strasse	Geburtsdatum
▶	1	Weiss	Peter	30529	Hannov	Palmstraße	07.11.63
	2	Bucz	Susanne	30531	Hannov	Heinestraße	06.04.76
	4	Karmann	Thomas	29227	Celle	Trift 28	04.08.54
	5	Klötzer	Karl	29221	Celle	Bahnhofstra	13.03.71
	6	Weiss	Karin	30529	Hannov	Palmstraße	05.10.62
	7	Weiss	Peter	38134	Braunsc	Glanweg 4	02.03.74
	8	Meier	Kathrin	38154	Braunsc	Welfenallee	03.05.81
	9	Schmidt	Karl	30529	Hannov	Lavesallee	25.06.49
	10	Müller	Claudia	29596	Breitenl	In den Fuhr	
	11	Lisewski	Bernd	30890	Barsingl	Roggenkam	06.06.60
	15	Martens	Melanie	29221	Celle	Horstswen ?	17.02.61

Datensatz 1 von 14 \*

tbPerson

- \*
  - ⚡ PID
  - Familienname
  - Vorname
  - PLZ
  - Ort
  - Strasse

Feld	tbPerson.*			
Alias				
Tabelle	tbPerson			
Sortierung				
Sichtbar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Funktion				
Kriterium				
oder				
oder				
oder				
oder				

**Abbildung 3.56**  
Design-Ansicht in  
openBase

Das Umschalten zwischen der Design-Ansicht und der SQL-Ansicht funktioniert zwar theoretisch in beiden Richtungen, tatsächlich gibt es aber Probleme. Einerseits werden wir später SQL-Anweisungen verwenden, die teilweise für die Design-Ansicht zu komplex sind und dort nicht dargestellt werden können. Andererseits wird bei der Umschaltung von der Design-Ansicht in die SQL-Ansicht die SQL-Anweisung komplett neu generiert. Das kann zu einer anderen Darstellung führen, da der Generator bestimmte Dinge rein schematisch durchführt. Wenn Sie beispielsweise jetzt wieder auf die SQL-Ansicht zurückschalten, indem Sie dieselbe Schaltfläche wieder betätigen, steht als SQL-Anweisung jetzt

```
SELECT * FROM "tbPerson"
```

dort. Die eingefügten Anführungszeichen sind zwar harmlos und insofern sogar sinnvoll, dass sie in jedem Fall zu einer korrekten Schreibweise des Namens ohne Umsetzung durch den SQL-Interpreter führen, zeigen aber andererseits, dass hier die SQL-Anweisung neu generiert wurde. Dies kann später auch zu sehr unübersichtlichen Anweisungen führen, wenn insbesondere Klammern in größerer Zahl entstehen.

Weitere Hinweise zur Nutzung von openBase finden Sie in Anhang A.