



### 3 Objekte in ACCESS 2002 und ihre Aufgaben

#### Ziele dieses Kapitels

- ▶ Sie lernen die verschiedenen Objekttypen in ACCESS und deren Aufgaben kennen.
- ▶ Sie lernen, was eine Gruppe ist.
- ▶ Sie lernen etwas über Konzepte zur Erstellung von Datenbanken in Mehrbenutzerumgebungen.
- ▶ Sie lernen etwas über den Datenzugriff über die ODBC-Schnittstelle.



## 3.1 Die Objekttypen im Datenbankfenster

Eine ACCESS-Datenbank besteht aus verschiedenen Objekten. Jedes Objekt können Sie einem bestimmten Objekttyp zuordnen. Aus dem Zusammenspiel dieser Objekte machen Sie aus Ihrer Datenbank eine Anwendung.

Jeder Objekttyp hat dabei ein genau umrissenes Aufgabenfeld, das Sie bei der Konzeption Ihrer Datenbank beziehungsweise Anwendung berücksichtigen sollten.

### 3.1.1 Tabellen

Die Daten in ACCESS-Datenbanken werden in Tabellen gespeichert. Alle anderen Objekte der Datenbank greifen auf diese Tabellen zu, um die dort gespeicherten Daten zu bearbeiten, zu ergänzen, zu löschen oder anzuzeigen.

Alle Änderungen an den Daten, die Sie über die anderen Objekte der Datenbank vornehmen, werden ohne Ausnahme in den Tabellen gespeichert.

### 3.1.2 Abfragen

Bei Abfragen handelt es sich um einmalig festgelegte Regeln, mit denen Benutzer auf die gespeicherten Daten zugreifen.

Sie verwenden Abfragen, um die in den Tabellen gespeicherten Daten anzuzeigen, zu analysieren oder zu ändern.

Sie können in Abfragen zum Beispiel:

- ▶ Berechnungen durchführen, indem Sie Feldwerte auf unterschiedlichste Art und Weise miteinander kombinieren (horizontale Ebene)
- ▶ Daten zusammenfassen
- ▶ Daten filtern und sortieren
- ▶ Daten gruppieren und Berechnungen über die Spalten einer Tabelle ausführen (vertikale Ebene)
- ▶ Änderungen an den Daten durchführen



### 3.1.3 Formulare

Die zentrale Aufgabe von Formularen ist es, eine Oberfläche bereitzustellen, über die Benutzer einer Datenbank gezielt auf die Daten zugreifen können, um die Daten sinnvoll zu bearbeiten, zu erfassen und zu pflegen.

Formulare stellen das Bildschirmobjekt der Datenbank dar. Sie legen in und mit Formularen Bearbeitungsreihenfolgen fest, präsentieren die Daten auf dem Bildschirm und fordern den Benutzer zur Dateneingabe auf.

Die Formulare sind somit das zentrale Bedienelement der Datenbank. In einer guten Datenbank hat der Benutzer lediglich direkten Kontakt mit drei Objekttypen: mit Formularen, Berichten und Datenzugriffsseiten.

### 3.1.4 Berichte

Um mit den gespeicherten Daten sinnvoll arbeiten zu können, müssen Sie diese Daten in einen entsprechenden Zusammenhang bringen, sortieren, berechnen und übersichtlich darstellen.

Berichte dienen dazu, Daten in gedruckter Form wirkungsvoll zu präsentieren. Berichte können Tabellen oder Abfragen als Herkunftsobjekte für die Daten benutzen.

Da Berichte in der Regel nur einen Ausschnitt der gespeicherten Daten wiedergeben sollen, ist die Abfrage die am häufigsten verwendete Datenherkunft für einen Bericht.

### 3.1.5 Datenzugriffsseiten

Datenzugriffsseiten dienen dazu, Daten einer ACCESS-Datenbank oder eines ACCESS-Projektes im Internet oder Intranet eines Unternehmens zur Anzeige oder zur Bearbeitung bereitzustellen.

Datenzugriffsseiten sind in ihrer Zweckbestimmung mit den Formularen und Berichten verwandt. Im Gegensatz zu Formularen und Berichten werden Datenzugriffsseiten aber nicht direkt in der Datenbank gespeichert.

Wenn Sie in ACCESS eine Datenzugriffsseite erstellen, wird die entsprechende Seite außerhalb der Datenbankdatei gespeichert und in der Datenbankdatei eine Verknüpfung zu dieser externen Datei erstellt.



### 3.1.6 Makros

In einer Datenbank werden Sie unter Umständen bestimmte Aktionen immer wieder ausführen müssen. Um diese Vorgänge zu automatisieren, können Sie Makros verwenden.

Die Bandbreite des Einsatzgebietes von Makros reicht von der Ausführung einfacher Aktionen wie den Start des Drucks eines Berichts bis hin zur Ausführung komplexer Operationen.

Ein Makro kann aus einem einzelnen Makro, das mehrere einzelne Makroaktionen beinhaltet, oder einer Ansammlung verschiedener Makros – einer so genannten Makrogruppe – bestehen.

Über Makros und Module entwerfen Sie die Intelligenz der Datenbank. Sie sind das fehlende Glied zwischen den einzelnen Objekten der Datenbank und verbinden diese zu einer leistungsstarken Anwendung.

Makros bieten Ihnen schnelle, vordefinierte Lösungen, um bestimmte Aktionen in Ihrer Datenbank Anwendung zu automatisieren.


Gegenüber Lösungen, die mittels Visual Basic für Applikationen (VBA) erstellt werden, haben Makros den Nachteil, dass

- ▶ sie in ihrer Gestaltung unübersichtlicher sind,
- ▶ sich nicht in dem nötigen Umfang mit Anmerkungen und Dokumentationen für die Entwickler einer Datenbank versehen lassen und
- ▶ weniger flexibel sind, als es VBA-Lösungen sein können.

Letztlich lässt sich jedes Makro auch in VBA-Befehle umsetzen. Wenn Sie ein Makro in ein VBA-Modul konvertieren möchten, gehen Sie folgendermaßen vor.

#### **Makro als Modul speichern**



1. Markieren Sie das Makro im Datenbankfenster.
2. Wählen Sie im Menü **Datei** den Menübefehl **Speichern unter**.
3. In dem Dialog **Speichern unter** ändern Sie die Art des zu speichernden Objektes von **Makro** auf **Modul**.
4. Klicken Sie auf die Schaltfläche , um das neue Modul zu speichern.



### 3.1.7 Module

Die Module in ACCESS-Datenbanken enthalten eine Auflistung von VBA-Deklarationen und -Prozeduren.

Mit VBA steht Ihnen ein wirkungsvolles Instrument zur Verfügung, mit dessen Hilfe sich komplexe Datenbankanwendungen entwickeln lassen.

Generell kann man eine Einteilung der Prozeduren in Sub-Prozeduren und Funktionen vornehmen. Funktionen geben Werte zurück und können in Ausdrücken verwendet werden. Sie werden mittels der Schlüsselwörter **Function** und **End Function** deklariert.

Sub-Prozeduren führen Aktionen aus und liefern keine Werte zurück. Sie werden mittel der Schlüsselwörter **Sub** und **End Sub** deklariert.

Funktionen und Sub-Prozeduren werden in Einheiten zusammengefasst gespeichert. Diese Einheiten werden als Module bezeichnet. Sie können diese Module aber nicht aufrufen oder ausführen. Sie rufen vielmehr die in diesen Modulen gespeicherten Sub-Prozeduren und Funktionen auf.

Sie haben verschiedene Möglichkeiten, VBA-Code in ACCESS auszuführen. Von besonderer Bedeutung sind dabei die Ereignisprozeduren.

Mittels Ereignisprozeduren reagieren Sie auf Aktionen, die durch den Benutzer der Datenbank ausgeführt werden. Aufgrund dieser Aktionen treten bestimmte Ereignisse ein, in denen so genannte Ereignisprozeduren ausgeführt werden können.

Wenn Sie in Formularen, Berichten oder Steuerelementen eine Ereignisprozedur erstellen, stellt Ihnen ACCESS automatisch Vorlagen für die Ereignisprozeduren zur Verfügung, die auf etwaige Besonderheiten der einzelnen Ereignisse eingehen. So können zum Beispiel bestimmte Ereignisse rückgängig gemacht werden, andere Ereignisse wiederum nicht.

Sie brauchen lediglich den Code hinzufügen, den Sie als Reaktion auf ein bestimmtes Ereignis ausführen möchten.

Der wesentliche Vorteil der Nutzung von VBA in Modulen gegenüber der Verwendung von Makros besteht in der größeren Flexibilität und den weitreichenden Möglichkeiten, die Ihnen diese Programmiersprache bietet.

## 3.2 Gruppen

Der Vollständigkeit halber seien hier die Gruppen in der Gruppenleiste erwähnt. Bei Gruppen handelt es sich nicht um Datenbankobjekte im strengen Sinne.




In Gruppen können Sie Verknüpfungen zu Datenbankobjekten anlegen, um einen schnelleren Zugriff auf diese zu bekommen. Sie können oft benutzte Formulare Gruppen hinzufügen und haben sie somit stets schnell verfügbar.

#### Neue Gruppe erstellen



Sie erstellen eine neue Gruppe und fügen dieser Gruppe Objekte hinzu.

1. Wechseln Sie mit der Taste  in das Datenbankfenster.
2. Klicken Sie in der Objektleiste auf den Objekttyp, den Sie der Gruppe hinzufügen möchten.
3. Klicken Sie mit der rechten Maustaste auf das Objekt, das Sie der Gruppe hinzufügen möchten.
4. In dem Kontextmenü wählen Sie **Zur Gruppe hinzufügen**. Klicken Sie dann auf **Neue Gruppe...**
5. In dem Dialogfeld **Neue Gruppe** vergeben Sie einen Namen für die neue Gruppe (zum Beispiel MeineGruppe).

ACCESS hat eine Gruppe in der Gruppenleiste und zusätzlich in der Gruppe eine Verknüpfung zu dem ausgewählten Datenbankobjekt erstellt.

Sie können alle Datenbankobjekte, die Sie der Gruppe hinzugefügt haben, anzeigen, indem Sie in der Gruppenleiste im Datenbankfenster das Symbol der Gruppe anklicken.

Gruppen dienen dazu, Ihnen die Verwaltung der Datenbankobjekte zu erleichtern. Sie können zum Beispiel für ein Formular mit mehreren Unterformularen eine eigene Gruppe erstellen, in die Sie alle miteinander in Beziehung stehenden Objekte aufnehmen: das Formular, die Unterformulare und die als Datenherkunft dienenden Tabellen und Abfragen.



Wenn Sie eine Gruppe oder ein Mitglied der Gruppe löschen, werden lediglich die Verknüpfungen zu den Datenbankobjekten gelöscht, nicht die zugrunde liegenden Datenbankobjekte.

Sie können Gruppen nicht aus anderen Datenbanken importieren oder in andere Datenbanken exportieren.

### 3.3 Importieren und verknüpfen von Objekten in ACCESS-Datenbanken

Bei der Arbeit mit ACCESS-Datenbanken haben Sie die Möglichkeit, die Objekte neu zu erstellen oder aus anderen Datenbanken zu importieren.



Sofern es sich um Tabellenobjekte handelt, haben Sie zusätzlich die Möglichkeit, Tabellen aus anderen Datenbanken beziehungsweise aus anderen Formaten in Ihre ACCESS-Datenbank einzubinden.

Wenn Sie Objekte in eine ACCESS-Datenbank importieren, wird eine Kopie dieses Objektes erstellt und in der Datenbank gespeichert. Eine Ausnahme stellen die Datenzugriffsseiten dar.

Wenn Sie eine Datenzugriffsseite importieren, wird eine Kopie der Datenzugriffsseite erstellt und außerhalb der Datenbankdatei gespeichert. In der Datenbankdatei wird eine neue Verknüpfung zu dieser Datenzugriffsseite angelegt.

### **3.4 Eingebundene Tabellen**

Die Antwort auf die Frage, wann Sie Tabellen importieren und wann Sie diese mit ACCESS-Datenbanken verknüpfen, hängt im Wesentlichen davon ab, was Sie mit den Daten weiterhin vorhaben.

Wenn Sie darauf angewiesen sind, dass Daten aus mehreren Programmen heraus bearbeitet werden, ist es unter Umständen angebracht, diese lediglich in die ACCESS-Datenbank einzubinden.

Wenn Sie Daten ausschließlich in ACCESS bearbeiten wollen, sollten Sie die Daten auf jeden Fall importieren. Denn nur dann können Sie das volle Leistungsspektrum von ACCESS einsetzen.

Sie können für eingebundene Tabellen zum Beispiel die Tabellenstruktur nicht verändern, da sonst andere Programme nicht mehr auf diese Struktur zugreifen beziehungsweise diese erkennen können.

Wenn Sie Tabellen verknüpfen, werden diese nicht physikalisch in die Datenbankdatei aufgenommen. In der ACCESS-Datenbank wird, vereinfacht gesagt, lediglich eine Verbindung zu einer Tabelle gespeichert.

Wenn Sie eine verknüpfte Tabelle löschen, wird nicht die Tabelle, sondern lediglich die Verknüpfung zu dieser Tabelle gelöscht. Die Daten bleiben in der externen Datei erhalten.

Wenn Sie die verknüpfte Datei an einen anderen Speicherort auf der Festplatte bewegen, verweist die gespeicherte Information in der eingebundenen Tabelle in ACCESS auf einen falschen Speicherort. Sie müssen dann diese Informationen aktualisieren oder die Tabelle neu einbinden.



### 3.4.1 Eingebundene Tabellen aus Fremdformaten

Bei der Arbeit mit Fremdformaten besteht ein wesentlicher Unterschied im Umgang mit den Tabellen darin, wie ACCESS mit den jeweiligen Fremdformaten umgehen kann.

Wenn Sie zum Beispiel Textdateien in ACCESS-Datenbanken einbinden, ist ACCESS nur in der Lage, diesen Dateien Datensätze hinzuzufügen. Sie können keine bestehenden Daten in Textdateien verändern.

Neben Einschränkungen dieser Art müssen Sie mit Geschwindigkeitsverlusten bei der Bearbeitung rechnen, da ACCESS für eingebundene Daten aus Fremdformaten die Informationen jeweils in die eine oder andere Richtung unter Verwendung geeigneter Treiber (ISAM oder ODBC) in das entsprechende Format umwandeln muss.

Weitere Informationen zu Treibern finden Sie in der integrierten Hilfe für Microsoft ACCESS 2002 unter der Überschrift: „Unterschiede zwischen ODBC-Treibern und integrierten Treibern für externe Daten.“

- ▶ Der Zugriff auf FoxPro-Tabellen erfolgt nun nicht mehr über ISAM-Treiber sondern über ODBC-Treiber.
- ▶ Um Daten aus dBASE 7.0 oder Paradox 8.0 verwenden zu können, benötigen Sie die Borland Database Engine (BDE) in der Version 4.x oder höher. Es ist weiterhin möglich, Zugriff auf die Daten vorhergehender Versionen von dBase oder Paradox zu erhalten. Allerdings ist auf die Daten, wenn die Tabellen lediglich verknüpft werden, ohne die Borland Database Engine nur ein schreibgeschützter Zugriff möglich.

### 3.4.2 Eingebundene Tabellen aus anderen ACCESS-Datenbanken

Ein gängiges Konzept beim Aufbau von ACCESS-Datenbanken ist es, die Tabellen in Datenbank A (**Back-End** genannt) zu speichern und die anderen Objekte einer Datenbank in der Datenbank B (**Front-End**) zu verwalten. Man spricht hier von geteilten Datenbanken.

Dieses Konzept bietet folgende Vorteile:

- ▶ In Mehrbenutzerumgebungen können die Daten zentral gespeichert und verwaltet werden, während die Programmteile auf den lokalen Computern ausgeführt werden. Dadurch kann die Belastung des Netzwerkes reduziert werden, da nur für den direkten Datenzugriff über das Netzwerk auf die Tabellen zugegriffen werden muss.



- ▶ Um Änderungen an den Objekten der Datenbank vornehmen zu können, müssen Sie die Datenbank exklusiv öffnen. Wenn Sie eine Datenbank exklusiv öffnen, können andere Benutzer einer Datenbank nicht auf diese zugreifen. Durch die Trennung der Datenbestände von den Programmteilen können Sie Veränderungen an den Formularen, Berichten und so weiter vornehmen, ohne dass die anderen Benutzer der Datenbank die Arbeit unterbrechen müssen.
- ▶ Sie können mit unterschiedlichen Versionen von Microsoft ACCESS auf die Tabellen zugreifen. Sie müssen in diesem Fall darauf achten, dass die Datenbankdatei mit den Tabellen der niedrigsten Version von Microsoft ACCESS entspricht, mit der auf diese Tabellen zugegriffen werden soll.

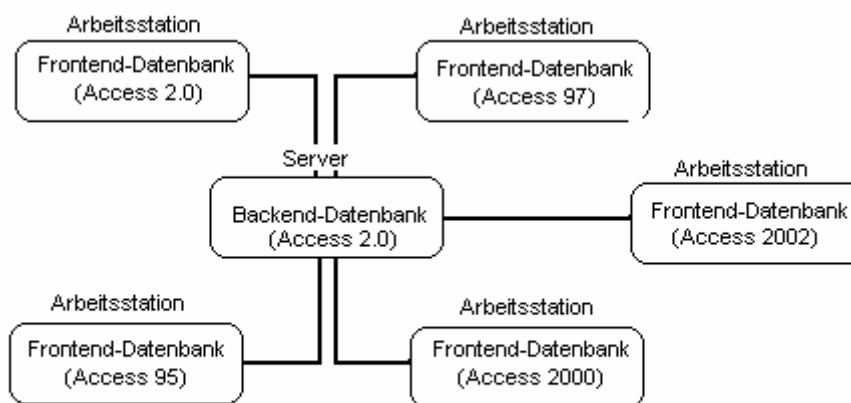


Abb. 3.1: Mögliche Struktur einer geteilten Datenbank unter Verwendung mehrerer Versionen von Microsoft ACCESS

- ▶ Die Sicherung der Daten im Netzwerkbetrieb ist einfacher.

Diese Liste ist nicht vollständig. Sie sollten den Gedanken mitnehmen, dass es abhängig von den Zielsetzungen, die Sie verfolgen, unterschiedliche Konzepte gibt, Datenbanken zu entwerfen.

Bei der Arbeit mit verknüpften Tabellen aus anderen ACCESS-Datenbanken können Sie diese mit anderen Objekten in ACCESS genau so behandeln, als seien sie in der Datenbank direkt gespeichert.

### 3.5 ODBC-Treiber und ODBC-Datenquellen

Zusätzlich zu dem Datenzugriff über die eingebauten Treiber kann der Datenzugriff über die ODBC (Open Database Connectivity)-Schnittstelle erfolgen.



Es handelt sich hierbei um ein Standardprotokoll für den Zugriff auf SQL-Datenbanken. Theoretisch ist es damit möglich, auf jeden Datenbankserver zuzugreifen. Voraussetzung ist jedoch, dass

- ▶ der Hersteller des Datenbankservers einen ODBC-Treiber zur Verfügung stellt, der diesen Zugriff mit ACCESS erlaubt;
- ▶ der Treiber auf dem Computer installiert ist, von dem aus der Zugriff erfolgen soll und
- ▶ eine entsprechende ODBC-Datenquelle auf dem Computer eingerichtet wurde.

Mit Microsoft ACCESS 2002 werden automatisch die ODBC-Treiber für

- ▶ den Microsoft SQL-Server (Sqlsrv32.dll),
- ▶ den Zugriff auf FoxPro Tabellen (vfpodbc.dll) und für
- ▶ den Zugriff auf Oracle Datenbankserver (msorcl32.dll)

installiert. Zusätzliche Treiber und Informationen, ob die Treiber für die Verwendung mit Microsoft ACCESS 2002 getestet oder geeignet sind, erhalten Sie von den jeweiligen Anbietern Ihres Datenbankservers.

### **3.5.1 Was ist eine ODBC-Datenquelle?**

Eine ODBC-Datenquelle, im allgemeinen als DSN (Data Source Name) bezeichnet, setzt sich im wesentlichen zusammen aus der Quelle, die Daten beinhaltet und den benötigten Informationen, um eine Verbindung zu dieser Datenherkunft herzustellen.

Eine solche Datenherkunft kann zum Beispiel der Microsoft SQL-Server sein. Die Verbindungsinformationen beinhalten den Servernamen, den Datenbanknamen, die benötigte Anmeldungs-ID für den Datenbankserver und das dazugehörige Kennwort.

Anhand dieser Informationen kann eine Verbindung zwischen ACCESS und der jeweiligen Datenherkunft hergestellt werden.

### **3.5.2 ODBC-Architektur und Erstellung einer ODBC-Datenquelle**

In der ODBC-Architektur wird durch eine Anwendung wie Microsoft ACCESS eine Verbindung zum ODBC-Treibermanager hergestellt.



Dieser Treibermanager verwendet einen bestimmten, auf dem Computer installierten ODBC-Treiber, um eine Verbindung zu einer bestimmten Datenbank in einer Datenherkunft wie zum Beispiel dem SQL-Server herzustellen.

All diese Informationen sind in einer ODBC-Datenquelle gespeichert, die Sie in Microsoft ACCESS verwenden, um eine Verbindung zu Datenbankservern außerhalb von Microsoft ACCESS herzustellen.

Um eine solche ODBC-Datenquelle (DSN) zu erstellen, führen Sie folgende Schritte aus.

### ODBC-Datenquelle erstellen

1. Öffnen Sie in der Windows-Oberfläche die Systemsteuerung und dann die Verwaltung.
2. Klicken Sie per das Symbol Datenquellen (ODBC).
3. Wählen Sie je nach Typ der zu erstellenden DSN eine der Registerkarten  ,  oder  aus.
4. Zur Erstellung einer neuen DSN klicken Sie auf die Schaltfläche .
5. Wählen Sie den für die Datenquelle in der Liste angeführten Treiber aus. Sofern Sie die MSDE oder den SQL-Server installiert haben, wählen Sie den Treiber für den SQL-Server.
6. Im nächsten Schritt geben Sie
  - ▶ einen Namen an, unter dem die DSN auf Ihrem Computer gespeichert werden soll;
  - ▶ den Standort beziehungsweise Namen des Datenbankservers an, auf dem sich die Datenbank befindet, mit der Sie Ihre ACCESS-Datenbank verbinden wollen.



Die Angabe einer Beschreibung für diese DSN ist optional und dient nur der leichteren Identifizierung der DSN durch den Anwender.



Abb. 3.2: Eingabemaske für den DSN-Namen und des Servers

7. Klicken Sie auf . Geben Sie dann die Informationen an, mit denen Sie auf dem Datenbankserver identifiziert werden können und Zugriff auf den Datenbankserver und die dort gespeicherte Datenbank erhalten.

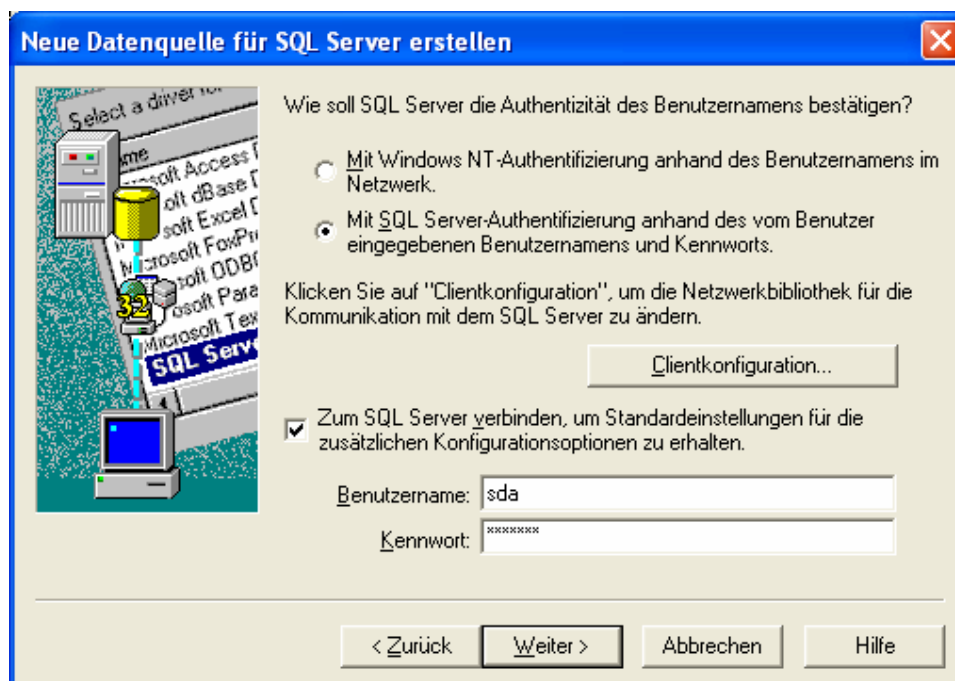


Abb. 3.3: Festlegung der weiteren Verbindungsinformationen

8. Weiterhin müssen Sie angeben, mit welcher Datenbank auf dem Server die Verbindung hergestellt werden soll.



Sie können die Datenbank aus einer Liste der Datenbanken auswählen, die der Assistent auf dem angegebenen Server gefunden hat und Ihnen zur weiteren Bearbeitung anbietet.

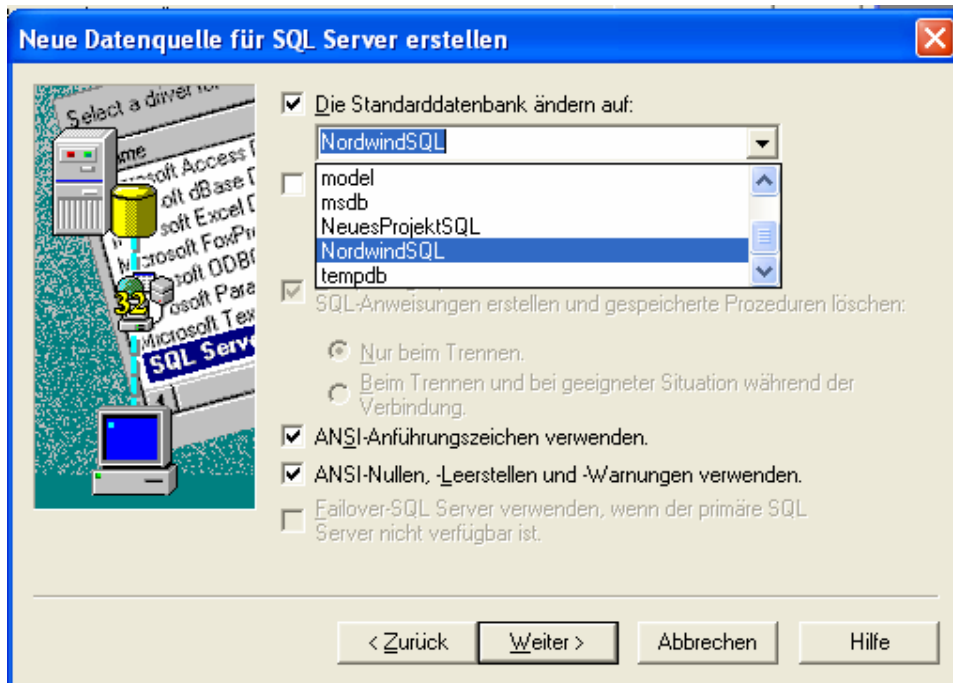


Abb. 3.4: Dialog zur Auswahl einer Datenbank bei Erstellung einer ODBC-Datenquelle

Dies sind die Basisinformationen, die Sie benötigen, um einen Zugriff auf eine ODBC-Datenquelle zu erhalten. Sie können noch eine Vielzahl weiterer Einstellungen vornehmen, die wir an dieser Stelle jedoch nicht behandeln können.

### 3.5.3 Eine kurze Einführung in die unterschiedlichen Datenquellen

Grundsätzlich unterscheiden wir Maschinendatenquellen und Dateidatenquellen. Maschinendatenquellen wiederum werden nach Systemdatenquellen und Benutzerdatenquellen unterschieden.

#### Maschinendatenquellen

Die Informationen von Maschinendatenquellen werden unter einem benutzerdefinierten Namen in der Windows-Registrierungsdatenbank gespeichert.

Maschinendatenquellen können nur auf dem Computer genutzt werden, auf dem Sie erstellt wurden. Während Benutzerdatenquellen wiederum nur dem Benutzer zur Verfügung stehen, der sie erstellt hat, stehen Systemdatenquellen für jeden Benutzer zur Verfügung, der sich auf diesem Computer anmelden kann.



### **Dateidatenquellen**

Die Informationen von Dateidatenquellen werden in einer Textdatei gespeichert. In Verbindung mit dem korrekten ODBC-Treiber können diese Datenquellen auf jedem beliebigen Computer eingesetzt und weitergegeben werden.

Der Vorteil von Dateidatenquellen liegt damit in ihrer größeren Flexibilität.

### **Zusammenfassung**

- ▶ Eine ACCESS-Datenbank besteht aus verschiedenen Objekten. Jeder Objekttyp hat ein genau umrissenes Aufgabenfeld.
- ▶ Gruppen dienen dazu, Ihnen die Verwaltung der Datenbankobjekte zu erleichtern.
- ▶ Sie haben die Möglichkeit, die Objekte neu zu erstellen oder aus anderen Datenbanken zu importieren. Sie können zusätzlich Tabellen aus anderen Datenbanken beziehungsweise aus anderen Formaten in eine ACCESS-Datenbank einbinden.
- ▶ Bei der Arbeit mit Fremdformaten besteht ein wesentlicher Unterschied im Umgang mit den Tabellen darin, wie ACCESS mit Fremdformaten umgehen kann.
- ▶ Es gibt unterschiedliche Konzepte, Datenbanken zu entwerfen. Ein gängiges Konzept beim Aufbau von ACCESS-Datenbanken ist es, mit geteilten Datenbanken zu arbeiten.

### **Übungsaufgabe 1**

Erstellen Sie eine geteilte Datenbank, indem Sie:

1. eine neue Datenbank mit dem Namen *FrontendDB* erstellen
2. die Tabellen der NORDWIND-Beispieldatenbank mit dieser Datenbank verknüpfen
3. die Abfragen, Formulare, Berichte, Makros und Module aus der NORDWIND-Beispieldatenbank in diese neue Datenbank importieren. Verzichten Sie auf den Import von Datenzugriffsseiten.

### **Übungsaufgabe 2**

Erstellen Sie in der Datenbank *FrontendDB* eine neue Gruppe mit dem Namen *Personalinformationen*. Fügen Sie dieser Gruppe die Tabelle **Personal** und das Formular **Personal** hinzu.



### Testaufgaben

- ▶ Frage 1: Bei welchem der folgenden Begriffe handelt es sich nicht um ein Objekt einer ACCESS-Datenbank.
  - A. Formular
  - B. Bericht
  - C. Makro
  - D. Modul
  - E. Gruppe
  
- ▶ Frage 2: Welche der folgenden Objekte können in ACCESS-Datenbanken eingebunden werden?
  - A. Tabellen
  - B. Abfragen
  - C. Formulare
  
- ▶ Frage 3: In welchem Format sollten Sie in geteilten Datenbanken die Tabellen speichern?
  - A. In einer Textdatei
  - B. In einer beliebigen ACCESS-Datenbankdatei
  - C. In der niedrigsten Version von ACCESS, mit der auf die Daten zugegriffen werden soll
  
- ▶ Frage 4: Können Sie durch den Import einer Tabelle eine bestehende Tabelle überschreiben?
  - A. Ja
  - B. Nein

