

# XML und Datenbanken

Die Schnittstellen von Access und SQL  
Server professionell nutzen

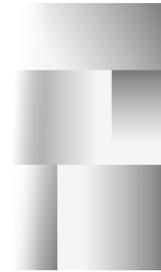
Uwe Hess

new technology

Markt+Technik Verlag

# Kapitel 3

## Access



3.1	Allgemeines	262
3.2	Access oder SQL Server	262
3.3	Erstellen einer Access-Datenbank	267
3.4	Arbeiten mit XML in Access	284
3.5	Verwenden des Application-Objekts	288
3.6	ReportML	302
3.7	Zugriff auf XML mit ADO	306
3.8	MDAC – Neues für ADO	325
3.9	Arbeiten mit dem DOM	332

## 3.1 Allgemeines

Mit MS Access steht Ihnen eine einfach zu programmierende Möglichkeit zum Speichern von Daten in einem relationalen Datenbanksystem zur Verfügung. Bereits mit der Version Access 2000 wurde diese Lösung um weitere Funktionalitäten zum Arbeiten mit XML erweitert. Allerdings erweisen sich die Möglichkeiten von Access für die Anforderungen der heutigen Zeit als nicht mehr ganz ausreichend. Die Ursachen dafür liegen unter anderem in der mangelhaften Netzwerkfähigkeit und in der zu schwachen Leistungsfähigkeit bei Mehrfachzugriffen. Seit der Version Access 2000 wendet sich Microsoft von der bisherigen Access-Strategie ab und beginnt mit der Implementierung von SQL Server. Eine einfache Erweiterung der bisher bekannten Access-Datenbanken um diese Möglichkeit würde langfristig gesehen keine befriedigende Lösung darstellen. Stattdessen wurde Access um die Möglichkeit der Erstellung von Access-Projekten, mit denen auf einfache Weise der Zugriff auf SQL Server möglich ist, erweitert. Dazu wurde der Umfang der Access-CD um die *SQL Server Desktop Edition* erweitert. Um ein Access-Projekt zu erstellen, ist die Installation der auf der Access-CD enthaltenen *SQL Server 2000 Desktop Edition* erforderlich. Langfristig gesehen werden hier sicherlich massive Änderungen in der Access-Strategie zu erwarten sein, die auf eine vollständige Ablösung von dem bisher verfolgten Access-Modell abzielen.

Wenn Sie sich nun speziell für das Arbeiten mit XML interessieren, dann eröffnen sich Ihnen damit zusätzliche Möglichkeiten. Zum einen stehen Ihnen die in Access enthaltenen Möglichkeiten zum Arbeiten mit XML zur Verfügung. Zum anderen können Sie auch viele der im weiteren Verlauf des Buches beschriebenen Möglichkeiten zum Arbeiten mit XML und SQL Server nutzen. Die in SQL Server enthaltenen Möglichkeiten zum Arbeiten mit XML werden im nächsten Kapitel ausführlich behandelt. Access selbst enthält verschiedene Möglichkeiten zum Arbeiten mit XML, die in den folgenden Abschnitten behandelt werden.

## 3.2 Access oder SQL Server

In diesem Buch erfolgt die Beschreibung der Möglichkeiten zum Arbeiten mit XML in Access und in SQL Server. Wer sich mit diesem Thema beschäftigt, weil er demnächst ein Projekt mit einer der beiden Datenbanken erstellen will, der wird sich zwischen einer dieser Datenbanken entscheiden müssen. Damit stellt sich nun auch die Frage über die Vor- und die Nachteile der beiden Produkte.

### 3.2.1 Access

Mit Access erhalten Sie eine preisgünstige Möglichkeit zum Speichern von Daten innerhalb eines relationalen Datenbanksystems. Der Umgang mit Access ist leicht zu erlernen und erfordert einen relativ geringen Aufwand. Allerdings ist die Leistungsfähigkeit von Access stark eingeschränkt. Hier ist eine Verwaltung von Daten mit einem Umfang von mehreren zehntausend Datensätzen pro Tabelle nicht sinnvoll und ergibt keine befriedigenden Antwortzeiten. Außerdem sind mit Access keine echten Transaktionen möglich. Access verwaltet eine Transaktion im Hauptspeicher des ausführenden Computers, womit keine absolute Sicherheit gewährleistet werden kann. Ein weiteres Manko besteht in der mangelhaften Realisierung von gleichzeitigen Zugriffen mehrerer Benutzer, so z.B. über das Internet.

Aufgrund des günstigen Preises stellt Access aber eine unschlagbar günstige Alternative für Projekte dar, die keine große Anzahl von gleichzeitigen Benutzerzugriffen bewältigen müssen. So gesehen kann sich unter Umständen sogar der Einsatz für ein Webprojekt lohnen. Wenn das Projekt mit der Zeit einen wesentlich größeren Umfang annimmt und die Anforderungen an die Leistungsfähigkeit steigen, dann ist immer noch eine Migration auf SQL Server möglich. Hier sollten Sie aber immer von Anfang an den bei einer Migration zu erwartenden Arbeitsumfang im Auge behalten.

### 3.2.2 Access und SQL Server Desktop Edition

Seit der Version Access 2000 erfolgt auf der Programm-CD zusätzlich die Auslieferung der SQL Server Desktop Edition. Dabei handelt es sich um eine Datenbanktechnologie, die kompatibel zu SQL Server ist. Diese Version kann unter Microsoft Windows 2000 oder höher und unter Windows 98 oder höher eingesetzt werden. Im Gegensatz zu SQL Server ist die Größe der Datenbank auf 2 GB beschränkt, was allerdings für Projekte mit durchschnittlichen Anforderungen kein Problem darstellen dürfte. Eine weitere Einschränkung besteht in der Begrenzung auf fünf gleichzeitige Abfragebatches. Weitere eintreffende Batchaufträge werden in eine Warteschlange gestellt und erst mit dem Freiwerden eines der bereits anstehenden Batchaufträge bearbeitet. Immerhin ist es aber damit ohne weiteres möglich, den Grundstein für ein professionelles Datenbankprojekt einer kleinen bis mittleren Arbeitsgruppe oder eines einfachen Webprojektes zu legen. Wenn die Anforderungen an das Datenbankprojekt wachsen, dann können Sie die Datenbank ohne weitere Probleme auf SQL Server 2000 migrieren. Allerdings sind in der mit Access ausgelieferten Version nicht die gemeinsam mit SQL Server ausgelieferten Verwaltungstools enthalten. Damit verfügen Sie nur über etwas eingeschränkte Möglichkeiten zur Nutzung aller Möglichkeiten von SQL Server. So sind Sie über Access nicht ohne größeren Aufwand in der Lage, eine Verwaltung der Sicherheit durchzuführen und müssen vorerst mit den Standardeinstellungen auskommen. Doch auch dieser Umstand stellt keine unüberwindbare Hürde dar. Die gesamte Funktionalität von SQL Server kann mit der Verwendung

von Transakt SQL-Befehlen gesteuert und verwaltet werden. Sie können damit Datenbanken und Objekte anlegen, die Datenbanksicherheit konfigurieren und alle Verwaltungsarbeiten erledigen. Wenn Sie bedenken, dass die Funktionalität der Verwaltungstools letztendlich auch nur über diese Befehle realisiert wird, dann werden Sie einsehen, dass auch Sie die wichtigsten Aufgaben mit diesen Befehlen erledigen können. Allerdings erfordert der Einsatz dieser Befehle eine genaue Kenntnis der Funktionsweise von SQL Server, was den Rahmen dieses Buches weit überschreiten würde.

#### **HINWEIS**

Die in SQL Server enthaltenen Möglichkeiten zum Arbeiten mit XML stehen auch in der SQL Server Desktop Edition zur Verfügung. Mit der Installation dieser Version können Sie viele der im Kapitel zu SQL Server und C# beschriebenen Möglichkeiten zum Arbeiten mit XML nutzen.

### **3.2.3 SQL Server**

Mit SQL Server erhalten Sie eine professionelle Datenbank, die auch die effektive Verwaltung sehr großer Datenmengen ermöglicht. Allerdings steht dem ein relativ hoher Preis gegenüber sodass der Einsatz von SQL Server ein gewisses Budget erfordert. Für den Anfang und für einfache Aufgaben reicht wie gesagt der Einsatz der in Access enthaltenen SQL Server Desktop-Version aus. Auf die Dauer werden Sie jedoch auf eine der höheren Versionen umsteigen müssen. Mit SQL Server können Sie ohne Probleme Daten für eine große Anzahl von gleichzeitigen Zugriffen über ein Netzwerk oder das Internet zur Verfügung stellen. Mit den in SQL Server enthaltenen Objekten wie z.B. gespeicherten Prozeduren können Sie die Leistungsfähigkeit der Datenbank erheblich verbessern und sind zugleich in der Lage, den Zugriff auf die Datenbank über ein ausgefeiltes Berechtigungssystem zu steuern.

### **3.2.4 Installation von Access**

Wenn Sie sich für das Arbeiten mit Access entscheiden, dann haben Sie die Wahl zwischen der Installation einer Version aus dem Office-Paket oder der Installation einer einzelnen Version. Beide Installationsarten unterscheiden sich nicht wesentlich voneinander. Mit dem Aufruf der Datei *setup.exe* erscheinen die üblichen Dialogfenster, in denen die Angabe der Registrierungsinformationen erfolgt. Anschließend müssen Sie sich zwischen der Standard- und der benutzerdefinierten Installation entscheiden. Mit der Entscheidung für eine benutzerdefinierte Installation können Sie anschließend die zu installierenden Komponenten auswählen.

Was das spätere Arbeiten mit XML betrifft, so ist hier nicht die explizite Auswahl bestimmter Komponenten erforderlich, die Installation dieser Funktionalität erfolgt standardmäßig.



Abbildung 3.1: Die Auswahl der zu installierenden Komponenten

### 3.2.5 Installation der SQL Server Desktop Edition

Die SQL Server Desktop Edition ist zwar auf der Access-CD enthalten, sie wird aber nicht standardmäßig mitinstalliert. Dazu rufen Sie im Verzeichnis *MSDE2000* die Datei *setup.exe* auf, mit der die automatische Installation erfolgt. Um die Installation auf einem Computer mit Windows NT 4 (nur ab SP6) oder Windows 2000 ausführen zu können, müssen Sie an diesem Computer als Administrator angemeldet sein. Nach der erfolgreichen Installation sollten Sie den Computer neu starten.

#### HINWEIS

Wenn Sie das Thema XML und Datenbanken mithilfe der Programmiersprache C# erschließen wollen, dann können Sie unter Umständen von der Installation der SQL Server Desktop Edition absehen. So erfolgt mit der Installation des Framework.SDK, welches Sie für das Arbeiten mit C# benötigen, auf Ihrem Computer automatisch die Installation der SQL Server Desktop Edition. Diese Version enthält im Gegensatz zu der mit Access ausgelieferten Version die beiden Beispieldatenbanken *northwind* und *pubs*, mit denen Sie auf bequeme Weise den Zugriff auf eine bestehende Datenbank testen können.

Unter Windows NT und Windows 2000 wird SQL Server als Dienst gestartet. Mit der erfolgten Installation wird dieser Dienst standardmäßig automatisch gestartet. Unter Windows 98 und Windows ME verwalten Sie den Startvorgang von SQL Server mit dem SQL Server Service Manager. Unter Umständen müssen Sie hier den Startvorgang manuell einleiten. Dazu führen Sie in der Symbolleiste der Windows-Taskleiste auf das Symbol *MSSQLServer* einen doppelten Mausklick aus. Hier aktivieren Sie das Kontrollkästchen *Dienst bei Betriebssystemstart automatisch starten*, womit der SQL Server Service Manager bei jedem Start von Windows automatisch gestartet wird.

### Verwalten der SQL Server Desktop Edition

Mit der Erstellung eines Access-Projektes sind im Menü *Extras/Datenbank Dienstprogramme* verschiedene zusätzliche Befehle zum Verwalten der SQL Server Desktop Edition verfügbar. Mit diesen zusätzlichen Befehlen sind Sie in der Lage, die wichtigsten Operationen zum Sichern der Daten auszuführen:

- *SQL-Datenbank sichern*. Mit diesem Befehl erstellen Sie eine Sicherungsdatei (DAT) der aktuellen SQL Server-Datenbank.
- *SQL-Datenbank wiederherstellen*. Mit dem Befehl erfolgt die Wiederherstellung einer SQL Server-Datenbank von einer Sicherungsdatei (DAT).
- *SQL-Datenbank löschen*. Der Befehl löscht eine SQL Server-Datenbank unwiderruflich. Die gelöschte Datei kann nicht wiederhergestellt werden.
- *Datenbankdatei kopieren*. Damit erfolgt das Kopieren einer SQL Server-Datenbank auf einen anderen Computer mit SQL Server.
- *Datenbanktransfer*. Damit wird eine SQL Server-Datenbankdatei (MDF) transferiert.

### Bearbeiten der Sicherheitseinstellungen

Wenn Sie die SQL Server Desktop Edition auf einem System mit NT4, Win2000 oder höher installieren, dann erfolgt die Installation mit der Windows NT-Authentifizierung des Benutzers, der die Installation durchführt. Bei dem Zugriff auf die Datenbank über ein Access-Projekt erfolgt dann die Authentifizierung an der Datenbank über das Systemkonto des aktuellen Benutzers. Daraus ergibt sich, dass der Zugriff über einen System-Account nur für Personen möglich ist, die ebenfalls über Administratorrechte verfügen.

SQL Server unterstützt neben der System-Authentifizierung eine eigene Authentifizierung. Mithilfe dieser Server-Authentifizierung ist z.B. ein Zugriff auf die Datenbank über das Internet möglich. Diese Server-Authentifizierung können Sie auch zum Zugriff auf die Datenbank innerhalb einer Arbeitsgruppe verwenden. Der Benutzername für diesen Server-Account lautet standardmäßig *sa* und enthält kein Passwort. Dieser Account beinhaltet alle administrativen Rechte an der Datenbank. Da dieser Account mit der Installation automatisch angelegt wird, stellt er ein deutliches Sicherheitsrisiko dar. Es wird aus diesem Grund dringend

empfohlen, für diesen Account ein Kennwort zu vergeben. Zum Vergeben eines Kennwortes stehen Ihnen zwei Möglichkeiten offen: Entweder Sie erledigen diese Aufgabe mithilfe entsprechender Transakt-SQL-Befehle oder Sie verwenden dazu die SQL Server-Tools. Sie können z.B. mit diesen Werkzeugen über ein Netzwerk auf diesen lokalen Computer zugreifen. Die Verwendung der Transact-SQL-Befehle erfordert ein umfangreiches Basiswissen, was an dieser Stelle aus Platzgründen nicht vermittelt werden kann. Wenn Sie die SQL Server Desktop Edition auf einem Computer unter Windows NT oder Windows 2000 installiert haben, dann können Sie im Menü *Ansicht* von Access den gemischten Sicherheitsmodus aktivieren. Damit wird der Zugriff auf die Datenbank über die System-Authentifizierung und über die Server-Authentifizierung ermöglicht.

#### HINWEIS

Wenn Sie den gemischten Sicherheitsmodus aktivieren, dann sollten Sie unbedingt ein Kennwort für den Server-Account vergeben. Verwenden Sie den Menüpunkt *Extras/Sicherheit/Login Kennwort*.

Um Einstellungen am Sicherheitsmodus der SQL Server Desktop Edition vornehmen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Sie müssen das Access-Projekt mit SQL Server 7.0 oder höher oder mit der hier besprochenen SQL Server 2000 Desktop Edition verbinden.
- Das Access-Projekt und SQL Server müssen gemeinsam auf einem Computer mit Windows NT oder Windows 2000 ausgeführt werden.
- Sie müssen dazu an dem Computer als Administrator angemeldet sein.

### 3.3 Erstellen einer Access-Datenbank

Um in Access eine neue Datenbank zu erstellen, öffnen Sie Microsoft Access, betätigen den Menüpunkt *Datei/Neu* und wählen aus dem Dialogfeld *Neue Datei* den Eintrag *Leere Datenbank*. In dem damit erscheinenden Dateibrowser geben Sie den Namen und den Speicherort der neuen Datenbank an. Mit dem Bestätigen des Dialoges erfolgt die Erstellung einer neuen Access-Datenbank, welche in einem eigenen Verwaltungsfenster angezeigt wird. Wer sich bisher ausschließlich mit SQL Server oder entsprechenden Datenbanken beschäftigte, dem wird hier sogleich der Unterschied in der Art und Weise der Verwaltung der Datenbank auffallen. In Access erfolgt die Verwaltung und die Bearbeitung einer Datenbank nicht über einen gemeinsamen Server, sondern jede erstellte Datenbank wird separat verwaltet. Die Verwaltung einer Access-Datenbank erfolgt zum einen über das Verwaltungsfenster und zum anderen über die im Access-Hauptfenster enthaltenen Menübefehle.

Im Access-Hauptfenster finden Sie alle Befehle zum Bearbeiten der in der Datenbank enthaltenen Objekte sowie zur allgemeinen Verwaltung der Datenbank. Im Verwaltungsfenster befinden sich alle Befehle zum Wechseln der Ansicht der in

der Datenbank enthaltenen Objekte sowie zum Erstellen neuer Objekte. Diese Befehle finden Sie auch in der Menüleiste des Hauptfensters wieder. Das Verwaltungsfenster besteht aus zwei Bereichen. Im linken Bereich sind alle Kategorien der in Access möglichen Objekte aufgelistet. Mit einem Mausklick auf einen der Einträge erfolgt im rechten Bereich die Anzeige aller erstellten Objekte dieser Kategorie. Im oberen Bereich befinden sich die Menübefehle *Öffnen* und *Entwurf*, mit denen ein Objekt entweder in seiner Arbeitsansicht angezeigt oder in der Entwurfansicht zum Bearbeiten geöffnet wird. Mit dem Befehl *Neu* können Sie ein neues Objekt erstellen und mit *Löschen* wird ein Objekt aus dem Verwaltungsfenster entfernt.

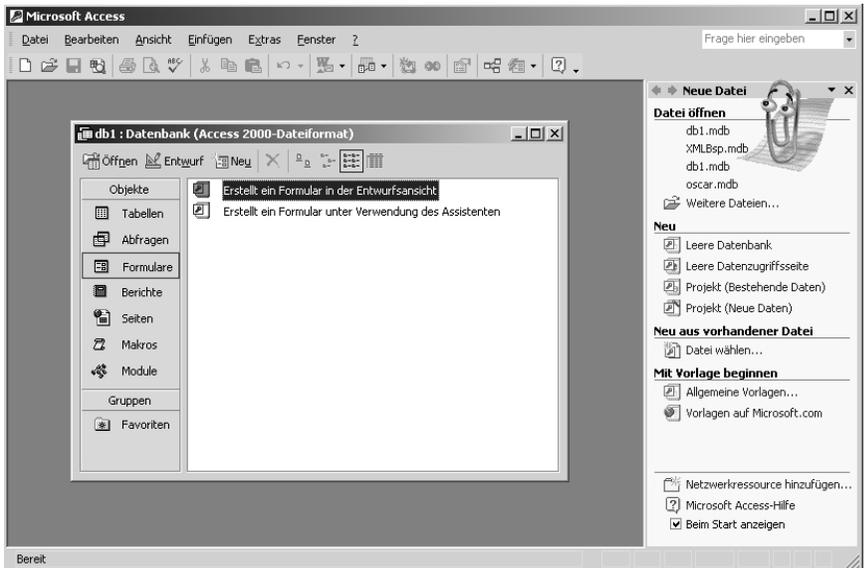


Abbildung 3.2: Das Verwaltungsfenster innerhalb des Access-Hauptfensters

### HINWEIS

Wenn Sie ein in dem Verwaltungsfenster enthaltenes Objekt löschen, dann wird dieses unwiderruflich entfernt und kann nicht wiederhergestellt werden.

## 3.3.1 Erstellen eines Access-Projektes

Um eine Datenbank in der SQL Server Desktop Edition oder zu einem SQL Server anzulegen, müssen Sie die Erstellung eines Access-Projektes wählen. Die Verwendung der XML-Möglichkeiten von SQL Server über Access ist zwar nicht gerade sinnvoll, Sie können aber auf diese Weise in der SQL Server Desktop Edition eine Datenbank einrichten und anhand dieser die Möglichkeiten zum Arbeiten mit XML und SQL Server nachvollziehen. Um also ein Access-Projekt zu

erstellen, gehen Sie anfangs in ähnlicher Weise vor wie bei der Erstellung einer Access-Datenbank. Dazu wählen Sie den Menüpunkt *Datei/Neu* und aus dem Dialogfeld *Neue Datei* den Eintrag *Projekt (leere Daten)*. In dem damit erscheinenden Dateibrowser geben Sie den Namen und den Speicherort des neuen Projektes an. In der Projektdatei selbst erfolgt lediglich die Speicherung von Informationen über das Projekt. Alle in dem Projekt erstellten Objekte (Tabellen, Sichten usw.) werden in SQL Server erstellt und gespeichert. Mit dem angelegten Projekt sind Sie nun in der Lage, diese neuen Objekte zu verwalten.

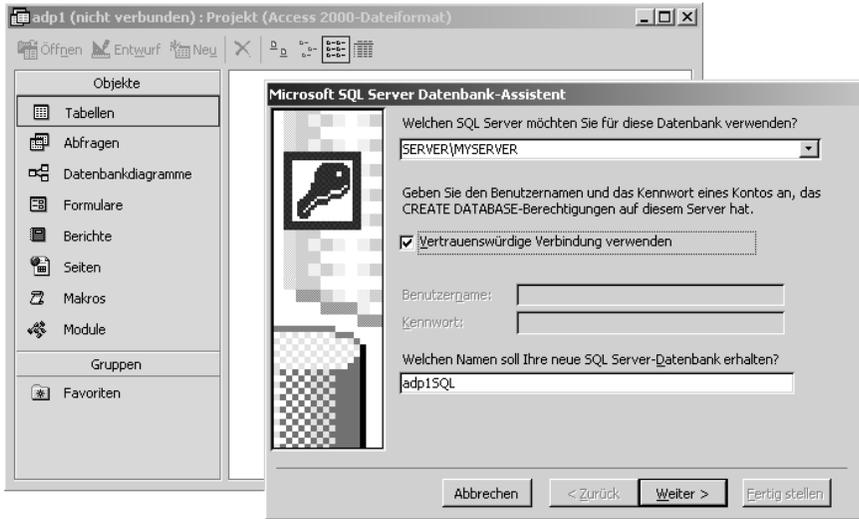


Abbildung 3.3: Anlegen eines Access-Projektes

### HINWEIS

Mit einem Access-Projekt erfolgt lediglich die Verwaltung einiger der in SQL Server enthaltenen Objekte. Dazu gehören u.a. Abfragen und Tabellen. Diese Aufgabe wird ansonst von den SQL Server-Verwaltungstools übernommen, welche wesentlich mehr Verwaltungsmöglichkeiten zur Verfügung stellen.

### Überprüfen der Verbindungseigenschaften

Wenn Sie mithilfe von Access-Projekten mehrere SQL Server-Datenbanken erstellt haben, dann können Sie auf einfache Weise innerhalb eines geöffneten Access-Projektes zwischen diesen Datenbanken wechseln und deren Objekte in dem Verwaltungsfenster anzeigen. Dazu verwenden Sie den Menüpunkt *Datei/Verbindung*. Hier geben Sie den zu verbindenden Datenbankserver, die Anmeldeinformationen sowie die Datenbank des ausgewählten Servers an. Hier können Sie sich auch mit einem SQL Server eines über das Netzwerk erreichbaren anderen Computers verbinden. Wenn Sie alle erforderlichen Angaben gemacht haben,

dann können Sie mit dem Betätigen der Schaltfläche *Verbindung testen* die getroffenen Einstellungen überprüfen. Mit der Schaltfläche *OK* übernehmen Sie diese Einstellungen. Anschließend werden im Verwaltungsfenster die Objekte der gewählten Datenbank angezeigt.



**Abbildung 3.4:** Hier überprüfen Sie die Verbindungseigenschaften

### 3.3.2 Erstellen einer Access-Tabelle

Um eine neue Tabelle zu erstellen, wählen Sie in der Objektleiste des Verwaltungsfensters den Eintrag *Tabellen* und anschließend den Menüpunkt *Entwurf*. Damit wird eine neue Tabelle in der Entwurfsansicht geöffnet. Anschließend können Sie die Feldnamen, die Datentypen und eine Beschreibung zu jedem Tabellenfeld eingeben, wobei die Angabe einer Beschreibung optional ist. Im unteren Bereich des Arbeitsfensters befindet sich eine Registerkarte, in der Sie weitere Eigenschaften zu jedem der Felder angeben können. Anschließend können Sie die Tabelle speichern und mit der Eingabe der Tabelleninhalte beginnen. Bei der Vergabe des Namens einer Tabelle müssen Sie verschiedene Regeln einhalten.

- Ein Name kann aus bis zu 64 Zeichen bestehen.
- Ein Name kann eine Kombination von Buchstaben, Zahlen, Leerzeichen und Sonderzeichen enthalten. Bei der Verwendung von Leerzeichen müssen Sie beachten, in einer SQL-Abfrage den Namen innerhalb von eckigen Klammern anzugeben. Nicht erlaubte Zeichen sind ein Punkt (.), Ausrufezeichen (!), Graviskakzent ( ` ) und eckige Klammern ( [ ] ).

- Der Name darf nicht mit vorangestellten Leerzeichen anfangen und keine Steuerzeichen enthalten (ASCII-Werte 0 bis 31).
- Es dürfen in Tabellen-, Ansichten- oder gespeicherten Prozedurnamen keine doppelten Anführungszeichen (") enthalten sein.

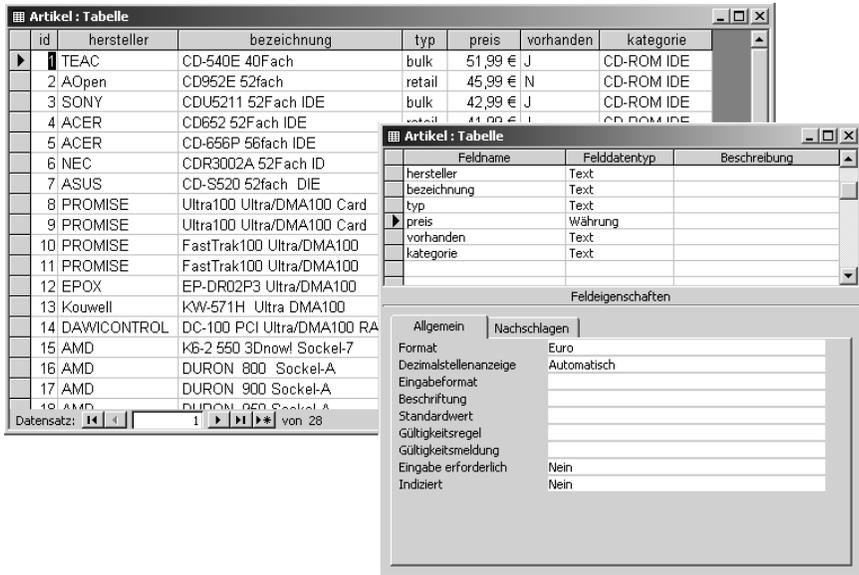


Abbildung 3.5: Eine Tabelle in der Entwurfsansicht und in der Tabellenansicht

## Datentypen in Access-Tabellen

Access stellt dem Benutzer eine umfangreiche Auswahl an verschiedenen Datentypen bereit, die im Prinzip alle Anforderungen für die zu speichernden Formate abdecken können. Bei der Erstellung einer Spalte geben Sie gemeinsam mit dem Spaltennamen die Obergruppe des zu verwendenden Datentyps an. Den genauen Bereich des angegebenen Typs geben Sie in der darunter befindlichen Registerkarte *Allgemein* an. Die Auswahlmöglichkeiten in dieser Registerkarte sind abhängig von dem zuvor ausgewählten Datentyp.

### HINWEIS

Sie können jederzeit einen Datentyp oder den Bereich eines Typs nachträglich ändern. Dabei müssen Sie aber beachten, dass bei der Änderung in einen anderen Typ oder einen kleineren Bereich für einen Typ bereits vorhandene Feldinhalte gelöscht oder abgeschnitten werden können. Auf diese Weise gelöschte oder abgeschnittene Feldinhalte können nicht wiederhergestellt werden und sind unwiederbringlich verloren.

- *Zahl.* Diesen Typ verwenden Sie für numerische Werte, bei denen normale Anforderungen an die Genauigkeit gestellt werden. Die Art und Größe der numerischen Werte stellen Sie im Feld *Feldgröße* der Registerkarte *Allgemein* ein.
- *Währung.* Bei Berechnungen mit hohen Anforderungen an die Genauigkeit, welche bei der Berechnung von Geldbeträgen bestehen, verwenden Sie den Typ *Währung*. Hier besteht nicht die Gefahr von Abrundungen bei Berechnungen. Im Feld *Format* der Registerkarte *Allgemein* stellen Sie den Bereich des Typs ein.
- *OLE-Objekt.* Mit diesem Typ speichern Sie in der Spalte ein OLE-Objekt. Dabei kann es sich z.B. um einen formatierten Text oder ein Dokument handeln.
- *Datum/Uhrzeit.* Mit dem Typ speichern Sie Datums- und Zeitangaben in verschiedenen Formaten. Sie können sich zwischen einem Standarddatum, verschiedenen vordefinierten Formaten oder auch einem benutzerdefinierten Format entscheiden.
- *Text.* Mit diesem Typ können Sie Texte bis zu einer maximalen Länge von 255 Zeichen speichern. Standardmäßig ist hier der Umfang von 50 Zeichen vorgesehen. Diese Einstellung können Sie im Feld *Feldgröße* ändern. An einem eingegebenen Text angehängte Leerzeichen werden nicht gespeichert. Werden die Daten in einer SQL Server-Datenbank gespeichert, sind auch mehr als 255 Zeichen möglich.
- *Memo.* Wenn Sie Text mit mehr als 255 Zeichen speichern wollen, dann verwenden Sie den Typ *Memo*. In einem solchen Feld können bis zu 65.536 Zeichen gespeichert werden. Dieser Typ kann, im Gegensatz zum Typ *OLE-Objekt*, nicht zum Speichern von Textformaten verwendet werden. Wenn Sie eine Spalte dieses Typs sortieren, dann werden bei der Sortierung nur die ersten 255 Zeichen eines Feldes berücksichtigt.
- *Byte.* Wenn Sie nur ganze Zahlen im Bereich von 0 bis 255 speichern wollen, dann entscheiden Sie sich für diesen Datentyp. Der Vorteil dabei ist der, dass dieser Typ nur ein Byte Speicherplatz belegt.
- *AutoWert.* Bei diesem Typ erfolgt die automatische Vergabe eines fortlaufenden Wertes für ein Feld. Dieser Wert ist immer eindeutig und kann zum sicheren Identifizieren eines Datensatzes verwendet werden.
- *Ja/Nein.* Bei der Verwendung dieses Typs erfolgt in der Spalte die Darstellung eines Kontrollkästchens, welches der Benutzer markieren kann. Ein markiertes Kontrollkästchen wird mit einem Häkchen dargestellt. Access speichert den Inhalt dieses Feldes intern mit dem Wert 0 bzw. 1 ab. Dieser Typ ist vorgesehen, um den Benutzer eine Auswahl treffen zu lassen bzw. eine Markierung festzulegen.

### Festlegen von Standardwerten

Mit einem Standardwert können Sie festlegen, dass beim Einfügen eines neuen Datensatzes für dieses Feld ein bestimmter Wert automatisch vergeben wird. Damit entfällt die Notwendigkeit, für dieses Feld in der SQL-Anweisung einen Wert festzulegen. Wenn Sie in der SQL-Anweisung für dieses Feld einen Wert festlegen, dann wird statt dem Standardwert dieser Wert verwendet. Bei einem Standardwert kann es sich z.B. um eine Zahl, eine Zeichenfolge oder auch um das Ergebnis einer Funktion handeln.

Ausdruck	Typ	Ergebnis
2	Zahl	2
"August"	Zeichenfolge	August
Date()	Funktion	18.12.2001
" "	Leerezeichen	

**Tabelle 3.1:** Beispiele für Standardwerte

### Festlegen von Gültigkeitsregeln

Mit einer Gültigkeitsregel können Sie festlegen, was für Daten ein Benutzer in ein Tabellenfeld eingeben darf. Damit können Sie auf einfache Weise sicherstellen, dass in ein Feld keine ungültigen Daten eingegeben werden und diese Werte sich innerhalb eines bestimmten Bereiches befinden.

Gültigkeitsregel	Beschreibung
<>0	Der Wert darf nicht Null sein
50 oder >70	Der Wert muss 50 oder größer als 70 sein
>#01.01.2001#	Das Datum muss größer als 01.01.2001 sein

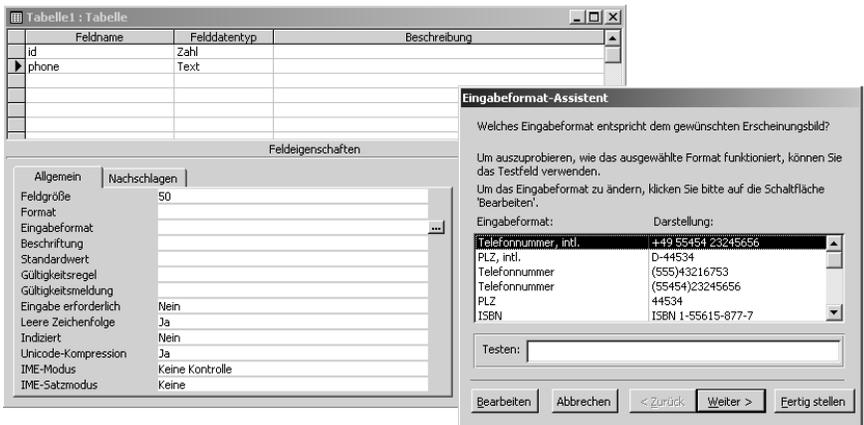
**Tabelle 3.2:** Beispiele für Gültigkeitsregeln

### Festlegen von Eingabeformaten

Mit einem Eingabeformat können Sie festlegen, in welcher Art und Weise ein Benutzer Daten in ein Tabellenfeld eingeben muss. Dabei kann es sich z.B. um ein Datums- oder Währungsformat handeln. Damit stellen Sie sicher, dass die Daten in einer einheitlichen Art und Weise vorliegen. Je nach verwendetem Datentyp sind hier verschiedene Einstellungen möglich. Bei den meisten Typen wird bei der Wahl eines Eingabeformates ein Assistent angezeigt, mit dem ein vordefiniertes Eingabeformat ausgewählt werden kann. In der Abbildung sehen Sie den Assistenten für das Eingabeformat des Datentyps *Text*. Der Assistent stellt nicht nur eine große Anzahl an unterschiedlichen Eingabeformaten zur Verfügung, er bietet auch die Möglichkeit zum Testen des gewählten Formates.

- *Zahl*. Felder mit diesem Typ können mit Regeln für eine Feldgröße und einem Bereich versehen werden.

- *Text*. Hier kann angegeben werden, wie viele Zeichen maximal eingegeben werden dürfen.
- Sie können für alle Typen festlegen, ob in ein Feld ein Wert eingegeben werden muss oder ob ein *NULL*-Wert verwendet werden kann. Wenn kein Wert angegeben wird, erfolgt automatisch das Einfügen des Wertes *NULL*. Diese Regel gilt allerdings nicht für den Typ *AutoWert*.
- *Text*, *Datum/Uhrzeit* und *Zahl*. Bei diesen Typen können Sie ein mit Leerstellen versehenes Eingabeformat definieren. Für diese Leerstellen können Sie festlegen, welche Werte dort zugelassen sind.



**Abbildung 3.6:** Der Assistent für ein Eingabeformat

### Festlegen eines Primärschlüssels

Beim Speichern einer neuen Tabelle werden Sie, falls Sie es noch nicht getan haben, aufgefordert, einen Primärschlüssel zu erstellen. Mit einem Primärschlüssel erhöhen Sie die Leistungsfähigkeit einer umfangreichen Datenbank beträchtlich. Dabei erfolgt im Hintergrund die Erstellung eines Index, mit dem das Auffinden von Werten bei Abfragen beschleunigt wird. Sie können in jeder Tabelle nur einen Primärschlüssel erstellen. Ein Feld, dem ein Primärschlüssel zugewiesen wird, darf zwingend nur einmal vorkommende eindeutige Werte enthalten, es sind also keine Duplikate möglich. Außerdem darf ein solches Feld keine Nullwerte enthalten.

#### **HINWEIS**

Mit der Vergabe von eindeutigen Werten sind Sie außerdem in der Lage, jeden Datensatz mit Sicherheit eindeutig identifizieren zu können. Aus diesem Grund ist es ratsam, in einer Tabelle generell eine Spalte mit einem ID-Wert zu führen.

Um einen Primärschlüssel zu einer Spalte hinzuzufügen, markieren Sie zuerst in der Entwurfsansicht die entsprechende Spalte und wählen anschließend den Befehl *Ansicht/Primärschlüssel*.

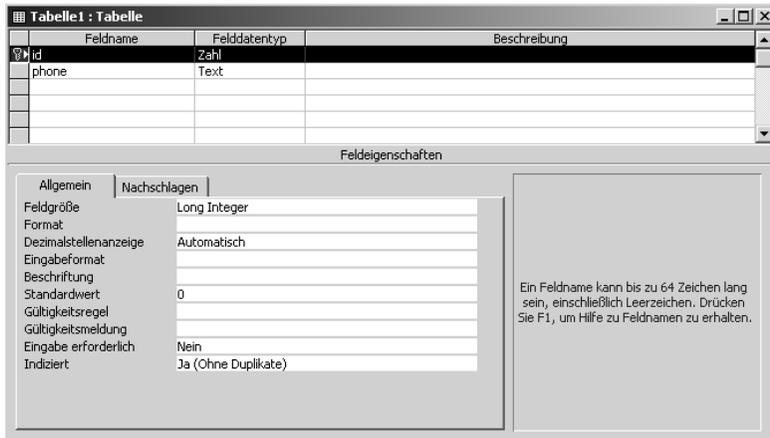


Abbildung 3.7: Eine Spalte mit Primärschlüssel

Mit der Verwendung eines Primärschlüssels muss beim Einfügen eines neuen Datensatzes in dieses Feld zwingend ein Wert eingegeben werden. Dabei ist es unerheblich, ob Sie als Feldinhalt eine Zeichenkette oder eine Zahl vorsehen. Da hier aber die Verwendung von Duplikaten ausgeschlossen ist, stehen Sie vor dem Problem immer einen eindeutigen Wert zu finden. Bei kleineren Tabellen lässt sich dieses Problem sicherlich mit einer vorhergehenden Abfrage der Spalte lösen, praktisch gesehen ist diese Vorgehensweise allerdings nicht empfehlenswert. Eine Lösung dieses Problems wird mit einem Blick auf die Eigenschaft *Felddatentyp* sichtbar. Hier können Sie unter anderem die Einstellung *AutoWert* auswählen. Mit der Entscheidung für diesen Feldtyp stehen Ihnen unter *Feldgröße* nur noch zwei Auswahlmöglichkeiten zur Verfügung. Dabei handelt es sich um die beiden Angaben *LongInteger* und *Replikations-ID*.

- *LongInteger*. Bei dieser Auswahl erfolgt das automatische Einfügen einer Ganzzahl. Dabei können Sie unter *Neue Werte* zwischen dem Hinzufügen eines Inkrementes oder eines Zufallswertes auswählen.
- *Replikations-ID*. Hier erfolgt das Hinzufügen eines eindeutigen Schlüssels, zu dessen Berechnung unter anderem die aktuelle Zeit und die Mac-Adresse einer vorhandenen Netzwerkkarte verwendet wird. Dieser Typ sollte vor allem dann verwendet werden, wenn die Datenbank zur Replikation vorgesehen ist. In dem Fall muss gewährleistet sein, dass die vergebene Zeichenkette absolut einmalig ist und nicht zufällig auf einem anderen System ebenfalls vergeben wurde. Eine solche Replikations-ID hat folgendes Format:  
{7BEC457C-8B0C-49A4-80C4-4A7FD963A87B}

### Zulassen von leeren Zeichenfolgen

Mit dieser Option legen Sie fest, dass in dieses Feld beim Einfügen eines neuen Datensatzes kein Wert eingegeben werden muss. In dem Fall wird automatisch der Wert NULL eingefügt. Diese Möglichkeit wird oft bei Daten genutzt, deren Angabe optional ist, wie z.B. eine zweite Telefonnummer bei Personendaten. Diese Möglichkeit kann nicht bei indizierten Spalten oder bei Spalten mit Fremdschlüsseln gewählt werden.

### Indizieren von Feldern

Mit dem Indizieren von Feldern können Sie die Leistungsfähigkeit von größeren Datenbanken verbessern. Dabei erfolgt die Erstellung eines Index, mit dessen Hilfe die Abfrage nach bestimmten Feldinhalten beschleunigt werden kann. Die Erstellung eines Index erweist sich nur bei großen Tabellen mit mehreren tausend Datensätzen als sinnvoll, bei kleineren Tabellen ist der mögliche Geschwindigkeitsvorteil kaum messbar.

Einstellung	Beschreibung
Nein (Voreinstellung)	Kein Index erstellen
Ja (Duplikate möglich)	Index lässt Duplikate zu
Ja (Ohne Duplikate)	Index lässt keine Duplikate zu

**Tabelle 3.3:** Die möglichen Einstellungen für Indexe

### Erstellen von Objekten eines Access-Projektes

Das Erstellen von Tabellen, Abfragen, Datenbankdiagrammen usw. erfolgt im Prinzip auf die gleiche Weise wie in einer Access-Datenbank. Allerdings erfolgt hier z.B. bei der Definition von Tabellenspalten die Verwendung von SQL Server-Datentypen sowie die Verwendung der SQL Server-spezifischen Tabelleneigenschaften. Das Gleiche gilt für die Erstellung und die Bearbeitung aller anderen Objekte in einer SQL Server-Datenbank über ein Access-Projekt. Wenn Sie also in einem Access-Projekt neue Objekte erstellen wollen, dann benötigen Sie Kenntnisse über SQL Server. Die wichtigsten Informationen über Tabellen, gespeicherte Prozeduren und Sichten in SQL Server finden Sie im Kapitel zu SQL Server.

#### HINWEIS

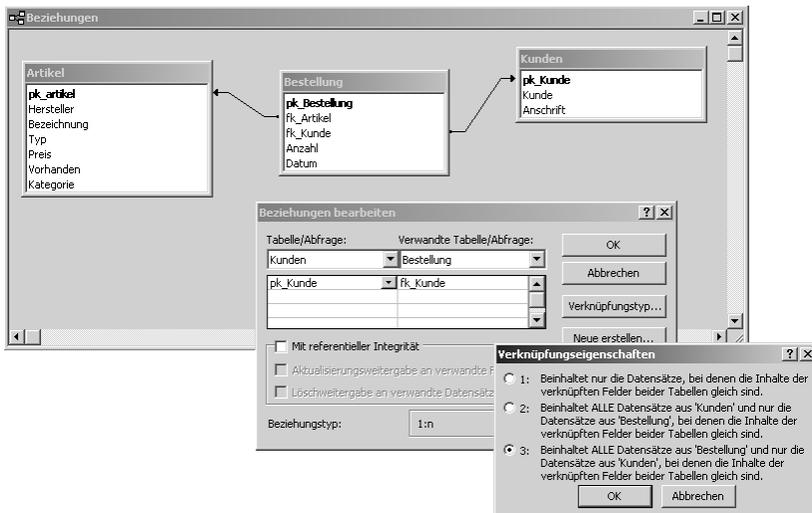
Die in dem Kapitel zu SQL Server enthaltenen Informationen über das Erstellen von Datenbankobjekten dienen lediglich als Orientierung für Neueinsteiger. Für tiefer greifende Informationen empfehle ich unbedingt das Studium weiterführender Fachliteratur.

### Erstellen von Beziehungen zwischen Tabellen

Beziehungen stellen ein wichtiges Merkmal in relationalen Datenbanken dar. Mit einer Beziehung erfolgt die Verknüpfung von Feldern zwischen zwei Tabellen. Dabei müssen beide Felder den gleichen Feldtyp aufweisen. Hier wird zwischen 1:1- und 1:n-Beziehungen unterschieden.

- *1:1-Beziehung*. Dabei kann ein Wert der einen Tabelle nur mit einem Wert der anderen Tabelle übereinstimmen.
- *1:n-Beziehung*. Bei dieser Beziehung kann ein Wert der einen Tabelle mit mehreren Werten der anderen Tabelle übereinstimmen. Dieses Modell wird wesentlich häufiger angewendet als eine 1:1-Beziehung. Ein typisches Beispiel dafür stellt eine Datenbank zur Speicherung von Kunden- und Rechnungsdaten dar. Ein Kunde kann in der Kundentabelle nur über einen Eintrag verfügen, aber in der Rechnungstabelle können logischerweise mehrere Datensätze zur Verwaltung der Rechnungen dieses Kunden enthalten sein. In diesem Fall ist immer von einem Primärschlüssel und einem Fremdschlüssel die Rede, wobei der Primärschlüssel den eindeutigen Datensatz der einen Tabelle und der Fremdschlüssel die n-Fach vorkommenden Daten der anderen Tabelle beschreibt.

Um eine Beziehung zwischen zwei Tabellen zu erstellen, wählen Sie den Menüpunkt *Extras/Beziehungen*. Damit erfolgt die Anzeige der Arbeitsoberfläche, in die Sie zuerst die entsprechenden Tabellen einfügen müssen. Beim ersten Start der Arbeitsoberfläche erscheint ein Dialogfeld, in dem Sie alle benötigten Tabellen auswählen und übernehmen können. Alternativ dazu können Sie auch weitere Tabellen über den Menüpunkt *Beziehung/Tabelle anzeigen* hinzufügen. Dieser Menüpunkt steht allerdings nur zur Verfügung, wenn die Arbeitsoberfläche zum Bearbeiten von Beziehungen angezeigt wird.



**Abbildung 3.8:** Bearbeiten von Beziehungen

Nachdem alle benötigten Tabellen auf der Arbeitsoberfläche angeordnet wurden, können Sie mit der Definition der Beziehungen beginnen. Dazu klicken Sie auf den Bezeichner der Spalte der einen Tabelle und ziehen den Mauszeiger auf den

Bezeichner der Spalte der anderen Tabelle. Anschließend erscheint ein Dialogfeld, in dem Sie weitere Einstellungen an der eben erstellten Beziehung vornehmen können. Dieses Dialogfeld können Sie auch später aufrufen. Dazu bewegen Sie den Mauszeiger über die Linie zwischen den beiden Feldern und betätigen die rechte Maustaste. In dem damit erscheinenden Dialogfeld wählen Sie dann den Eintrag *Beziehung bearbeiten*.

In dem Dialogfeld können Sie zum einen die Beziehung zwischen verschiedenen Feldern verändern. Außerdem können Sie hier die Integrität der Beziehung festlegen. Mit der Verwendung von Integrität stellen Sie sicher, dass Verknüpfungen zwischen Tabellen jederzeit ihre Gültigkeit behalten und die Daten in verknüpften Tabellen nicht versehentlich gelöscht werden. Mit der Markierung dieser Auswahl stehen Ihnen zwei weitere Auswahlmöglichkeiten zur Verfügung, mit denen Sie die Art und Weise der Datenintegrität einstellen können.

- *Aktualisierungserweitergabe*. Dabei werden bei der Aktualisierung von Primärschlüsseln in der einen Tabelle diese Veränderungen an den verknüpften Fremdschlüssel der anderen Tabelle weitergegeben und diese damit automatisch aktualisiert.
- *Löschweitergabe*. Dabei werden beim Löschen von Datensätzen in der Primärtabelle diese Löschungen automatisch an den durch die Beziehung verknüpften Datensätzen der Fremdschlüsseltabelle durchgeführt.

Zusätzlich können Sie hier den Beziehungstyp genauer festlegen. Dazu klicken Sie auf die Schaltfläche *Verknüpfungstyp*. In dem daraufhin erscheinenden Dialogfeld legen Sie fest, ob es sich bei der Beziehung um eine 1:1- oder eine 1:n-Beziehung handelt.

### 3.3.3 Erstellen einer Abfrage

Abfragen stellen ein wichtiges Werkzeug beim Arbeiten mit Access dar. Mithilfe einer Abfrage sind Sie in der Lage, dem Benutzer vordefinierte SQL-Anweisungen beim Arbeiten innerhalb der Datenbankoberfläche zur Verfügung zu stellen. Abfragen eignen sich ebenfalls zum schnellen Erzeugen und Testen von Abfragen. Dabei können Sie zwischen verschiedenen Arten von Abfragen wählen:

- *Auswahlabfragen und Kreuztabellenabfragen*. Mithilfe einer solchen Abfrage können Sie Daten aus einer oder mehrerer Tabellen abfragen. Sie können innerhalb einer Auswahlabfrage Gruppierungen oder Sortierungen vornehmen sowie Berechnungen ausführen lassen. Diese Art von Abfragen wird besonders häufig angewendet. Dieser Abfragetyp entspricht einer *SELECT*-Anweisung.
- *Aktualisierungsabfrage*. Mit diesem Abfragetyp können Sie auf einfache Weise umfangreiche Daten in einer oder mehreren Tabellen aktualisieren und ersetzen. Dieser Abfragetyp entspricht einer *UPDATE*-Anweisung.

- *Anfügeabfrage.* Mit einer Anfügeabfrage können Sie Daten an eine oder mehrere Tabellen anfügen. Die neuen Datensätze werden dabei an die bestehenden Datensätze angehängen. Dieser Abfragetyp entspricht einer *INSERT INTO*-Anweisung.
- *Löschabfrage.* Mit einer Löschabfrage erfolgt das Löschen von einem oder mehreren Datensätzen einer oder mehrerer Tabellen. Dieser Abfragetyp entspricht einer *DELETE*-Anweisung.
- *Tabellenerstellungsabfrage.* Mit dieser Abfrage können Sie Daten aus einer Tabelle abfragen und diese direkt in eine neue Tabelle einfügen. Dieser Abfragetyp entspricht einer *INSERT SELECT INTO*-Anweisung.
- Um eine Abfrage zu erstellen, wählen Sie in der Objektleiste des Datenbankfensters den Eintrag *Abfragen* und in der darüber befindlichen Leiste den Eintrag *Neu*. Es erscheint ein Dialogfeld, in dem Sie sich zwischen dem Erstellen einer neuen Abfrage oder der anschließenden Verwendung eines Assistenten entscheiden.

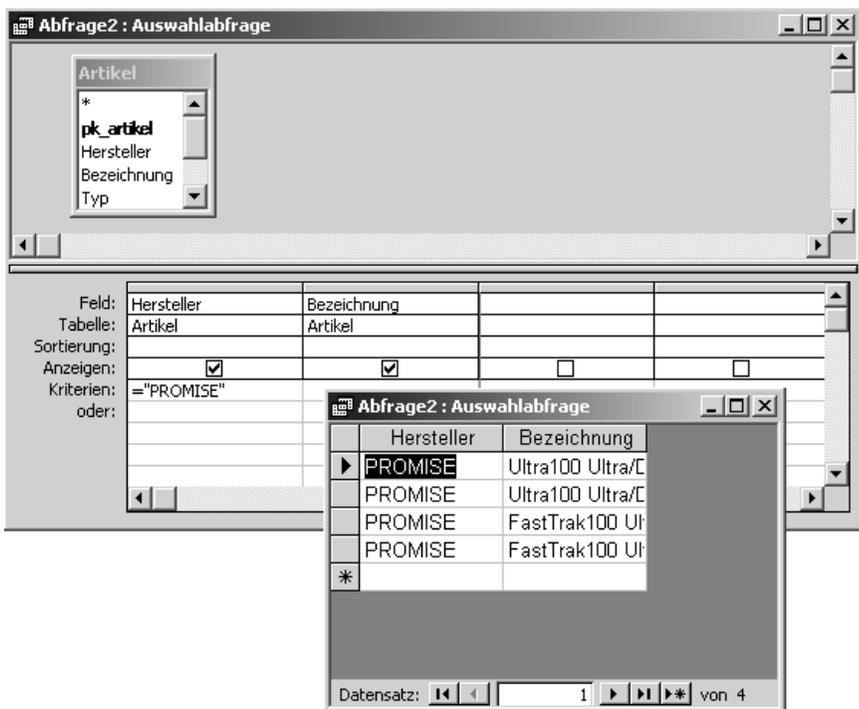


Abbildung 3.9: Eine Abfrage in der Entwurfsansicht und das Ergebnis

Wenn Sie noch keine Erfahrungen mit Access-Abfragen gesammelt haben, dann sollten Sie für die ersten Schritte durchaus die Verwendung des Assistenten in Betracht ziehen. Dabei können Sie in mehreren aufeinander folgenden Dialogfeldern nach und nach die gewünschte Abfrage definieren. In jedem der Dialogfelder wird Ihnen dabei die Möglichkeit gegeben, verschiedene Quellen sowie Ausgabeformate auszuwählen.

Wer sich mit der Anwendung von Abfragen bereits auskennt, der sollte stattdessen unbedingt die Anzeige der Entwurfsansicht wählen. Hier stehen Ihnen letztendlich wesentlich mehr Möglichkeiten zur Verfügung. Mit der Entscheidung für die Anzeige der Entwurfsansicht erfolgt die Anzeige eines Dialogfeldes, in dem Sie eine oder mehrere Quellen für die zu erstellende Abfrage auswählen können. Bei diesen Quellen kann es sich um Tabellen oder auch andere Abfragen handeln.

### HINWEIS

Um während der Bearbeitung einer Abfrage weitere Tabellen einzufügen, klicken Sie den Menüpunkt *Abfrage/Tabelle anzeigen*. Dieser Menüpunkt steht Ihnen nur in der Entwurfsansicht einer Abfrage zur Verfügung.

In der vorhergehenden Abbildung sehen Sie eine Abfrage in der Entwurfsansicht. Nach dem Anzeigen aller benötigten Tabellen müssen Sie die anzuzeigenden Spalten zu der Abfrage hinzufügen. Dazu markieren Sie die entsprechende Spalte in der angezeigten Tabelle und ziehen den Eintrag mit gedrückter Maustaste in eine der Spalten des unteren Bereiches mit der Bezeichnung *Feld*. Alternativ dazu können Sie auch auf eine Feld-Spalte des unteren Bereiches mit der Maus klicken. Damit verwandelt sich die Spalte in ein DropDown-Feld, in dem Sie die anzuzeigende Tabellenspalte auswählen können. In den darunter befindlichen Spalten können Sie festlegen, aus welcher Tabelle eine Spalte angezeigt und ob eine Sortierung des Ergebnisses vorgenommen werden soll. In der Spalte *Anzeigen* legen Sie fest, ob die entsprechende Spalte angezeigt wird. Am wichtigsten dürfte mit Sicherheit die Spalte *Kriterien* sein. Hier können Sie ein oder mehrere Auswahlkriterien für die Abfrage festlegen. Die hier getroffenen Einstellungen entsprechen der WHERE-Klausel einer SQL-Abfrage.

Wenn Sie alle Einstellungen getroffen haben, dann können Sie die neue Abfrage einem ersten Test unterziehen. Dazu speichern Sie die Abfrage zuerst mithilfe des Menübefehls *Speichern* und vergeben dabei einen möglichst aussagekräftigen Namen. Anschließend klicken Sie den Menüpunkt *Abfrage/Ausführen* oder Sie verwenden die gleichnamige Schaltfläche in der Menüleiste. Eine andere Möglichkeit zum Ausführen einer Abfrage besteht in der Umschaltung in die Datenblattansicht.

### HINWEIS

Die Umschaltung in eine andere Ansicht bietet Ihnen verschiedenste Möglichkeiten. So können Sie z.B. in eine Diagrammansicht umschalten und so auf einfache Weise eine grafische Auswertung erzeugen.

### Anzeigen des SQL-Codes

Ein besonders interessantes Merkmal von Abfragen besteht in der Anzeige des visuell erzeugten SQL-Codes. Auf diese Weise können Sie ohne tiefgreifende SQL-Kenntnisse umfangreiche SQL-Abfragen entwerfen und mit diesem Code anschließend eine Anwendung, die auf Ihre Datenbank zugreift, übernehmen. Die Anzeige des SQL-Codes erfolgt über die Umschaltung in die SQL-Ansicht über den Menüpunkt *Ansicht/SQL Ansicht*.



**Abbildung 3.10:** Die Anzeige des visuell erzeugten SQL-Codes

Sie können übrigens auch direkt in den hier angezeigten Code manuell eingreifen. Wenn der bearbeitete Code keine Fehler enthält, dann stellt Access die Abfrage im Arbeitsfenster automatisch visuell dar.

### 3.3.4 Verwalten einer Access-DB

Mit den bisher beschriebenen Funktionalitäten von Access verfügen Einsteiger immerhin über Grundkenntnisse für erste Schritte mit Access. Abschließend folgen noch einige kurze administrative Themen.

#### Konvertieren einer Access-Datenbank

Wenn Sie sich für das Arbeiten mit Access entschieden haben, dann stehen Sie unter Umständen mittelfristig vor der Aufgabe, diese Datenbank weiterzugeben oder diese anderen Benutzern zur Verfügung zu stellen. Damit treten aber auch unweigerlich früher oder später Versionskonflikte auf, die durch die Neuerungen der letzten drei Versionen von Access bedingt sind. Diese Neuerungen führen teilweise zu einer Inkompatibilität einiger Funktionalitäten, sodass eine Konvertierung in die gewünschte Version unumgänglich ist. Dabei handelt es sich um die Versionen Access 97, 2000 und 2002. Das Arbeiten mit Access 97 bringt Ihnen jedoch keinen Vorteil, da dort die hier beschriebenen XML-Funktionalitäten nur eingeschränkt zur Verfügung stehen.

**HINWEIS**

Sie können mit einer höheren Version immer auf Datenbanken einer niedrigeren Version zugreifen. Um Änderungen an einer solchen Version vorzunehmen und diese dauerhaft zu speichern, müssen Sie diese unbedingt in die aktuelle Version konvertieren.

Um eine Access-Datenbank in einer der drei genannten Versionen zu konvertieren, verwenden Sie den Menüpunkt *Extras/Datenbankdienstprogramme/Datenbank konvertieren* und wählen die entsprechende Version aus. Beim Konvertieren in das alte Access 97-Format gehen allerdings die ab Access 2000 standardmäßig integrierten XML-Fähigkeiten verloren. Dies betrifft nicht das Arbeiten mit programmtechnisch gelösten Aufgabenstellungen wie z.B. ADO. Hier müssen Sie dafür sicherstellen, dass auch in der konvertierten Version alle benötigten Klassen verfügbar und korrekt eingebunden sind.

**Komprimieren und Reparieren einer Datenbank**

Das Komprimieren und Reparieren stellt beim Arbeiten mit Access ein ganz besonders wichtiges Thema dar. Die Notwendigkeit einer regelmäßigen Komprimierung einer Access-Datenbank hat ihre Ursache in der Art und Weise der Speicherung der in den Tabellen enthaltenen Daten. Beim Ändern und beim Löschen von Datensätzen sind die alten Daten zwar für den Benutzer nicht mehr zugänglich, doch der von ihnen belegte Platz wird physikalisch nicht freigegeben. Stattdessen werden neue Daten an das Ende der Datendatei angehängt. Demzufolge erfolgt im Laufe der Zeit eine Fragmentierung der Datendatei. Beim Komprimieren wird diese Datendatei defragmentiert und der belegte Speicherplatz damit wieder freigegeben.

**HINWEIS**

Sehr häufig genutzte Access-Datenbanken können derart stark fragmentiert werden, dass sie bis auf ein Mehrfaches ihrer ursprünglichen Größe anwachsen. In dem Fall muss die Datenbank unbedingt defragmentiert werden, da zum einen deren Leistungsfähigkeit stark sinkt und zum anderen ihre Funktionssicherheit gefährdet wird.

Die Notwendigkeit einer Reparatur der Datenbank wird von Access in der Regel beim Öffnen der Datenbank automatisch erkannt. In einem damit erscheinenden Dialogfeld werden Sie aufgefordert, diesen Schritt ausführen zu lassen. Die Ursache für eine notwendige Reparatur kann unter anderem in fehlenden Verweisen auf Datenbankobjekte oder auch inkonsistente Tabellen und Verknüpfungen sein. Dieser Zustand kann unter anderem beim unsachgemäßen Schließen der Datenbank oder bei einem Systemabsturz eintreten. Schon aus diesem Grund empfiehlt es sich, die Datenbank regelmäßig zu sichern und zu komprimieren.

Um eine Datenbank zu komprimieren oder zu reparieren, verwenden Sie den Menüpunkt *Extras/Datenbank-Dienstprogramme/Datenbank komprimieren und reparieren*.

## Vergeben eines Datenbankkennwortes

Access verfügt über ein eigenes Sicherheitsmodell, mit dem der Zugriff auf eine Datenbank reglementiert werden kann. Die einfachste Möglichkeit der Verwendung eines Sicherheitsmodells besteht in der Vergabe eines Datenbankkennwortes. Dies erfolgt über den Menüpunkt *Extras/Sicherheit/Datenbankkennwort festlegen*. Damit wird der Zugriff auf die Datenbank nur mit der Eingabe des damit vergebenen Kennwortes ermöglicht.

### HINWEIS

Das hier vergebene Kennwort wird im Klartext in der Datenbankdatei abgelegt und kann jederzeit mit dem Öffnen der Datendatei durch einen einfachen Texteditor ermittelt werden. Wenn Sie einen wirklich wirksamen Schutz der Datenbank erzielen wollen, dann müssen Sie die Datenbank zusätzlich über den Menüpunkt *Extras/Sicherheit/Datenbank verschlüsseln* einer Verschlüsselung unterziehen.

## Benutzer- und Gruppenkonten verwenden

Statt eines allgemeinen Kennwortes können Sie auch Berechtigungen auf Gruppenebene vergeben. Diese Berechtigungen können Sie dann verschiedenen Objekten der Datenbank zuordnen, sodass jeder Gruppe nur bestimmte Funktionalitäten zur Verfügung stehen. Dazu müssen Sie zuerst unter *Extras/Sicherheit/Benutzer- und Gruppenkonten* Gruppen erstellen, denen Sie anschließend erstellte Benutzer zuordnen. Hier können Sie zugleich die Anmeldeinformationen für die erstellten Benutzer verwalten. Standardmäßig enthält Access die beiden Gruppen *Administratoren* und *Benutzer* sowie den Benutzer *Administrator*.

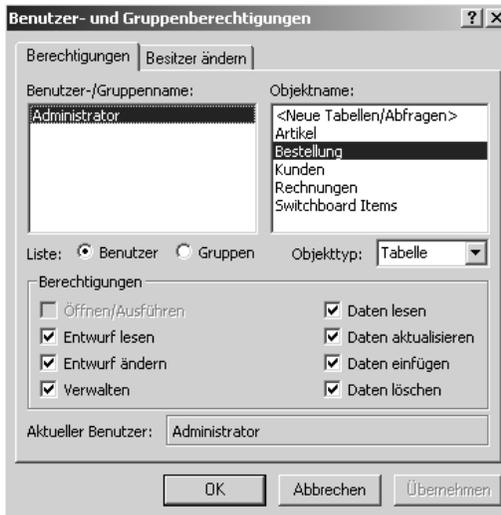


Abbildung 3.11: Hier vergeben Sie die Berechtigungen für einzelne Objekte

Unter dem Menüpunkt *Extras/Sicherheit/Benutzer- und Gruppenberechtigungen* können Sie den Benutzern und Gruppen verschiedene Datenbankobjekte zuordnen und spezielle Rechte für diese Objekte vergeben. So können Sie z.B. den Zugriff auf Formulare für bestimmte Benutzer unterbinden oder Tabellen nur zum Lesen zulassen.

## 3.4 Arbeiten mit XML in Access

Access bietet verschiedene Möglichkeiten zum Arbeiten mit XML. Dabei handelt es sich zum einen um bereits integrierte Lösungen für den Austausch von Daten. Diese Möglichkeiten stehen erst mit der Version Access 2000 und höher zur Verfügung. Zur Nutzung dieser Möglichkeiten stehen dem Benutzer verschiedene Assistenten zur Verfügung.

Sie können allerdings auch eigene Lösungen implementieren. Diese eigenen Lösungen müssen allerdings mit Programmcode realisiert werden. Eine programmierte Lösung ermöglicht in jedem Fall die Bereitstellung individuellerer Funktionalitäten. Dazu ist es notwendig, dass auf dem ausführenden Computer alle benötigten Programmbibliotheken installiert und registriert sind. Wenn Sie Ihre Access-Lösung vertreiben und weitergeben wollen, dann können Sie dieses Problem mithilfe des Assistenten zum Erstellen einer ausführbaren Access-Datenbank lösen. Dieser Assistent befindet sich in der Access Developer-Version und ermöglicht die Erstellung eines Installationspaketes. Damit wird die Installation und die Ausführung einer einzelnen Access-Datenbank auf einem Computer ermöglicht, auf dem kein Access- oder Office-Paket installiert ist.

### 3.4.1 Datenaustausch mithilfe von Assistenten

Die einfachste Möglichkeit zum Datenaustausch mit XML-Dokumenten bietet Access über die beiden Menüpunkte *Externe Daten* und *Exportieren*. Über die dahinter verborgenen Assistenten erhalten Sie auf einfachste Weise einen schnellen Zugriff auf Daten in den verschiedensten Formaten.

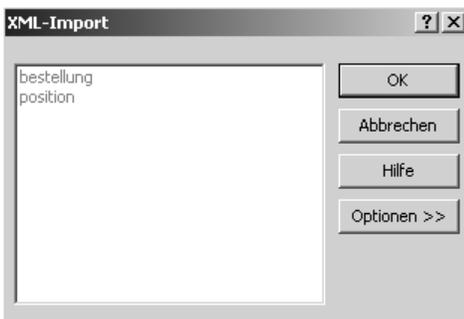
### 3.4.2 Importieren von XML-Daten

Access stellt dem Benutzer die Möglichkeit des Imports von XML-Dokumenten bereit. Dabei wird die hierarchische Struktur des XML-Dokumentes in eine relationale Struktur transformiert und die Daten in einer Tabelle abgelegt. Das Importieren von XML-Daten in eine Access-Datenbank erfolgt nach bestimmten Kriterien, welche von der Struktur der XML-Datei beeinflusst werden. Dabei erfolgt die Identifizierung der Struktur anhand der in der Datei befindlichen Tags. Damit wird auch ersichtlich, weshalb die Verwendung von wohlgeformtem XML so wichtig ist. Bei der Verwendung einer Syntax, welche nicht dem XML-Regelwerk entspricht, kann die Struktur der Datei nicht eindeutig interpretiert werden. Access führt beim Import automatisch eine Überprüfung der Dokumentstruktur durch. Mit der Erkennung einer fehlerhaften Struktur erfolgt die Ausgabe einer entsprechenden Fehlermeldung.



**Abbildung 3.12:** Fehlerhafte Dokumente können nicht importiert werden

Um ein XML-Dokument zu importieren, wählen Sie den Menüpunkt *Datei/externe Daten/importieren*. Wählen Sie in dem daraufhin erscheinenden Browser die zu importierende Datei, womit Access mit der Überprüfung des zu importierenden Dokumentes beginnt. Wenn die Überprüfung positiv verläuft, dann werden im nächsten Dialogfeld alle in dem Dokument enthaltenen Elementknoten angezeigt. Innerhalb dieser Liste können Sie keine Auswahl treffen. Die einzige Möglichkeit zum Beeinflussen des Datenimportes bietet die Schaltfläche *Optionen*. Mit dem Anklicken dieser Schaltfläche erfolgt eine Vergrößerung des Dialogfeldes. In dem damit zusätzlich angezeigten Bereich können Sie wählen, ob Sie nur die Struktur oder die Struktur und die Daten in einer neuen Tabelle oder die Daten an eine bestehende Tabelle anfügen wollen.



**Abbildung 3.13:** Das Dialogfeld für den Import der Daten

Nach dem erfolgreichen Import der Daten werden Sie sehen, dass für die unterschiedlichen Elementknoten des Dokumentes jeweils eine separate Tabelle angelegt wurde. Diese Vorgehensweise erklärt sich aus der automatisierten Umwandlung der hierarchischen Struktur in eine relationale Struktur. Außerdem ist Access nicht in der Lage, die in einem Schema oder einer DTD enthaltenen Informationen über die zu verwendenden Datentypen umzusetzen. Dies können Sie mit dem Öffnen der neuen Tabelle in der Entwurfsansicht nachvollziehen. Hier werden Sie sehen, dass für jedes Feld der Tabelle automatisch der Datentyp `Text` angelegt wurde. Um also die Eigenschaften eines XML-Dokumentes in einer Access-Datenbank weiter zu verwalten, müssen Sie wohl oder übel selbst eine Vergabe der gewünschten Datentypen vornehmen.

### 3.4.3 Exportieren von Daten nach XML

Während zum Import von Daten nur wenige Arbeitsschritte ausgeführt werden müssen, stehen beim Exportieren der Daten einige Möglichkeiten mehr zur Auswahl. Diese Möglichkeiten stehen aufgrund der verschiedenen Wege der Verwendung der exportierten Daten zur Verfügung. So ist es möglich, dass ein Benutzer die Daten entweder nur elektronisch weiter verarbeiten oder auch diese Daten ohne Umwege präsentieren will. Deshalb stehen beim Datenexport verschiedene Auswahlmöglichkeiten offen, die je nach Belieben auch untereinander kombiniert werden können.

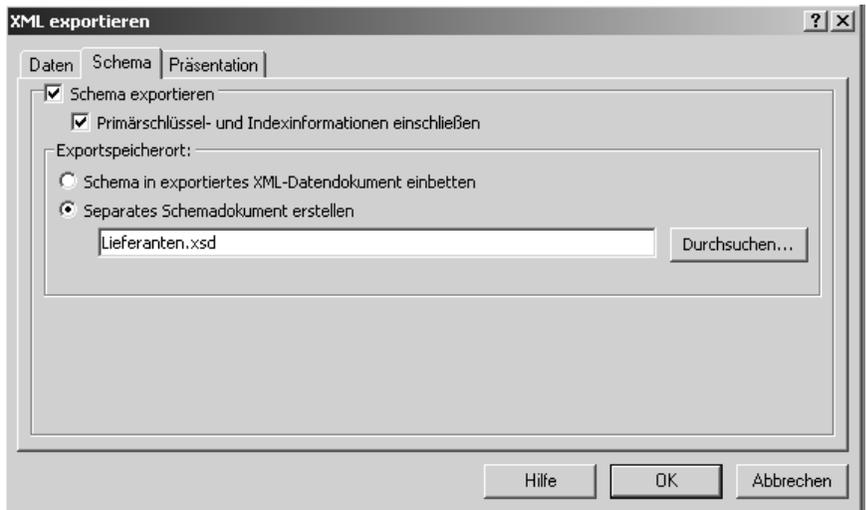


Abbildung 3.14: Das Dialogfeld zum Exportieren der Daten

Access unterstützt den Export von Daten nach XML in einer vielfältigen Art und Weise. So können Sie Daten aus einer Tabelle, Abfrage, einem Datenblatt, Formular oder Bericht in eine XML-Datei exportieren. Somit sind Sie in der Lage, auch bereits sortierte Daten oder eine bestimmte Ergebnismenge in einer XML-Datei zusammenzufassen.

Um den Export der Daten aus einer Tabelle durchzuführen, markieren oder öffnen Sie im Datenbankfenster die Datenquelle, deren Daten Sie für den Export vorgesehen haben. Anschließend klicken Sie auf den Menüpunkt *Exportieren* und wählen in dem sodann erscheinenden Browser den Speicherort sowie unter *Dateityp* den Eintrag *XML-Dokument (\*.xml)*. Anschließend erscheint ein Dialogfeld mit drei Auswahlkästchen. Hier können Sie festlegen, ob Sie die darin enthaltenen Daten in einer XML-Datei speichern, ein Schema erstellen oder die Daten präsentieren wollen. Mit dem Betätigen der Schaltfläche *Optionen* wird das Dialogfeld vergrößert. In dem nun sichtbaren Bereich können Sie alle benötigten Einstellungen vornehmen.

## Export in eine XML-Datei

Bei der Wahl dieser Möglichkeit erfolgt der Export der gesamten Daten in eine XML-Datei am gewählten Speicherort. Die Zieldatei enthält lediglich die ausgewählten Daten sowie den dazugehörigen XML-Code. Um der Datei weiteren Code hinzuzufügen, mit dem es möglich ist, die Daten formatiert darzustellen, müssen Sie mindestens eine der beiden folgenden Auswahlmöglichkeiten hinzufügen.

## Erstellen eines XSL-Schemas

Hier wird zusätzlich zu der gewählten Datenquelle eine XSL-Datei erstellt. Diese Möglichkeit steht nur zur Verfügung, wenn zugleich der Export von Daten erfolgt. Hier können Sie zusätzlich festlegen, ob das XSL-Schema in dem XML-Dokument eingebettet oder in einer eigenen Datei erstellt werden soll.

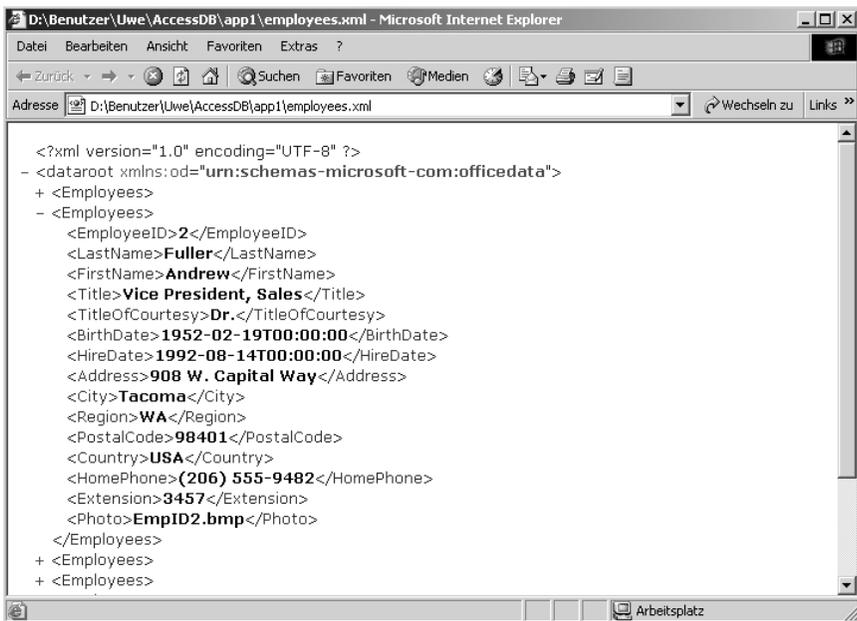


Abbildung 3.15: Eine erstellte XML-Datei im Browser

## Erstellen einer Präsentation

Mit dieser Auswahl können Sie die ausgewählten Daten direkt in einer HTML- oder ASP-Datei anzeigen lassen. Dabei werden die Daten jedoch nicht direkt in der erstellten Datei abgelegt. Hier enthalten die erstellten Dateien Code, welcher serverseitig (ASP) oder clientseitig (HTML) ausgeführt wird und der den Zugriff auf die XML-Datei ermöglicht.

**Listing:** Der Code einer erstellten ASP-Datei

```
<%  
Set objData = Server.CreateObject("MSXML.DOMDocument")  
objData.async = false  
  
if (false) then  
    Set objDataXMLHTTP =  
        Server.CreateObject("Microsoft.XMLHTTP")  
    objDataXMLHTTP.open "GET", "", false  
    objDataXMLHTTP.setRequestHeader "Content-Type", "text/xml"  
    objDataXMLHTTP.send  
    objData.load(objDataXMLHTTP.responseBody)  
else  
    objData.load(Server.MapPath("Kunden%20Abfrage.xml"))  
end if  
  
Set objStyle = Server.CreateObject("MSXML.DOMDocument")  
objStyle.async = false  
objStyle.load(Server.MapPath("Kunden%20Abfrage.xml"))  
Response.ContentType = "text/html"  
Response.Write objData.transformNode(objStyle)  
>
```

## 3.5 Verwenden des Application-Objekts

Die in dem vorherigen Abschnitt beschriebenen Möglichkeiten zum Import und Export von XML reichen in der Regel zwar aus, aber oft benötigt ein Anwender eine speziell auf seine Bedürfnisse zugeschnittene Anwendungsoberfläche. Das Erstellen einer solchen Oberfläche ist an sich gar nicht so schwierig – wenn man die Grundlagen für das Arbeiten mit Visual Basic beherrscht.

**HINWEIS**

In der aktuellen Version von Office XP und Access 2002 steht die Unterstützung durch VB.NET noch nicht zur Verfügung, sodass hier nach wie vor mit dem herkömmlichen Visual Basic gearbeitet werden muss

Die im vorhergehenden Abschnitt beschriebene Funktionalität der Menüpunkte zum Zugriff auf XML-Dokumente beruht auf den beiden Methoden `ImportXML()` und `ExportXML()` des Application-Objektes. Mithilfe dieser beiden Methoden sind Sie in der Lage, alle dort enthaltenen Funktionalitäten zu realisieren und einem Anwender speziell angepasste Dialoge zur Verfügung zu stellen. In der Abbildung sehen Sie ein Beispielformular, mit dem der Datenaustausch zu XML innerhalb einer einheitlichen Oberfläche realisiert wird.

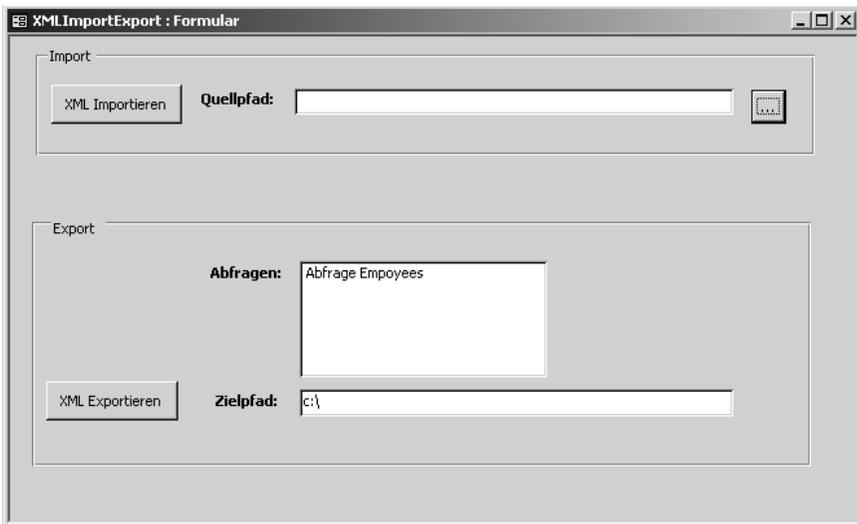


Abbildung 3.16: Eine eigene Oberfläche für den Datenaustausch

### 3.5.1 ExportXML

Die Methode `ExportXML()` ermöglicht den Export von Daten in ein XML-Dokument. Beim Export überprüft Access, ob ein Dokument mit diesem Namen schon vorhanden ist. Wenn dies der Fall ist, dann wird der Benutzer mit einem Dialogfeld darauf hingewiesen. Mit der Methode können Sie, wie bei den in Access enthaltenen Menüpunkten auch, eine XSL-Datei oder eine präsentationsfähige HTML- bzw. ASP-Seite erstellen. Die Methode erwartet zwingend einen gültigen Pfad, in dem das Zieldokument erstellt werden soll. Wenn dieser Pfad nicht vorhanden ist, dann wird dieser nicht erstellt, sondern es wird stattdessen ein Fehler ausgelöst.

**Listing:** Der Aufruf der Methode `ExportXML ()` mit allen Attributen

```
Ausdruck.ExportXML(ObjectType, DataSource, DataTarget, DataTransform,
SchemaTarget, SchemaFormat, SchemaTransform, PresentationTarget,
PresentationTransform, ImageTarget, LiveReportSource, Encoding, OtherFlags)
```

Die meisten der von der Methode `ExportXML()` erwarteten Attribute sind optional und müssen nicht angegeben werden. Bei der Zusammenstellung der Parameter müssen Sie deshalb besonders Acht geben. Um die gewünschte Funktionsweise zu erreichen, müssen Sie die Attribute in der richtigen Kombination anwenden. So ist es nicht möglich, eine XSL-Datei zu erstellen wenn nicht zugleich der Parameter `DataTarget`, welcher das XML-Dokument bezeichnet, angegeben wird.

- *Ausdruck*. Unbedingt erforderlich. Ein Ausdruck, der ein Application-Objekt referenziert.

- *ObjectType*. Erforderlicher *AcExportXMLObjectType*-Wert. Dieser Wert entspricht dem Typ des zu exportierenden Access-Objekts. Der hier angegebene Wert kann einer der folgenden *AcExportXMLObjectType*-Konstanten entsprechen. Diese Konstanten repräsentieren die Access-Objekte, aus denen der Export von Daten möglich ist. *acExportDataAccessPage* | *acExportForm* | *acExportFunction* | *acExportQuery* | *acExportReport* | *acExportServerView* | *acExportStoredProcedure* | *acExportTable*
- *DataSource*. Dieser Wert entspricht dem Namen des zu exportierenden Access-Objekts und muss unbedingt angegeben werden.
- *DataTarget*. Optionaler String-Wert. Der Dateiname und der Pfad für die exportierte Datei. Wird dieses Argument nicht angegeben, werden die Daten nicht exportiert.
- *DataTransform*. Der Name der XSL-Datei, die für die Daten ausgeführt werden soll, bevor sie in die Zieldatei geschrieben werden. Dieser Wert ist optional und muss in der richtigen Weise mit *SchemaTarget* kombiniert werden.
- *SchemaTarget*. Ein korrekter Dateiname und der existierende Pfad für die exportierten Schemainformationen. Dieses Argument ist optional. Wird dieses Argument nicht angegeben, werden die Schemainformationen in das Datendokument eingebettet.
- *SchemaFormat*. Optional anzugebender Wert vom Typ *AcExportXMLSchemaFormat*. Dieser Wert repräsentiert das Format, in dem Schemainformationen exportiert werden. Der Wert von *AcExportXMLSchemaFormat* kann einer der folgenden Konstanten entsprechen: *acSchemaNone* *Standard* | *acSchemaXSD*
- *SchemaTransform*. Diese Angabe ist optional und entspricht dem Namen der XSL-Datei, die für die Schemainformationen ausgeführt werden soll.
- *PresentationTarget*. Der Dateiname und ein existierender Pfad für die exportierten Präsentationsinformationen. Die Angabe dieses Wertes ist optional.
- *PresentationTransform*. Der Name der XSL-Datei, die für die Präsentationsinformationen ausgeführt werden soll. Die Angabe dieses Wertes ist optional.
- *ImageTarget*. Der Pfad für exportierte Bilder. Die Angabe dieses Wertes ist optional.
- *LiveReportSource*. Verbindungsinformationen für einen Bericht, der Live-Daten enthält. Hier kann es sich um einen Verweis auf eine ODBC-Datei oder eine XMLSQL-Anforderung handeln. Für die Verwendung dieses Argumentes muss das Argument *ObjectType* den Wert *acExportReport* enthalten. Die Angabe dieses Wertes ist optional.
- *Encoding*. Die für die exportierte Datei zu verwendende Textkodierung. *AcExportXMLEncoding* kann einer der folgenden Konstanten entsprechen. Die Angabe dieses Wertes ist optional.

*acEUCJ* | *acUCS2* | *acUCS4* | *acUTF16* | *acUTF8* *Standard*

- *OtherFlags*. Eine Bitmaske, die andere Verhaltensweisen im Zusammenhang mit dem Export in das XML-Format definiert. In der folgenden Tabelle ist das Verhalten als Folge bestimmter Werte beschrieben. Die Angabe dieses Wertes ist optional. Um die entsprechende Funktionalität zu erhalten, können Sie die Werte miteinander addieren.

Wert	Beschreibung
1	Enthält die »n«-Tabellen für das durch DataSource angegebene Objekt.
2	Erstellt relationale Schemaeigenschaften.
4	Erstellt einen ASP-Wrapper. Ansonsten wird standardmäßig ein HTML-Wrapper verwendet. Dies gilt nur beim Exportieren von Berichten.
8	Erstellt erweiterte Schemaeigenschaften.

**Tabelle 3.4:** Die möglichen Werte für das Attribut OtherFlags

Der Aufruf der Methode erfolgt auf die hier gezeigte Art und Weise. Die Angabe der benötigten Attribute erfolgt unter Verwendung deren Bezeichner und der gleichzeitigen Angabe von den entsprechenden Werten. In dem Beispiel sehen Sie die Anwendung der Methode für den Export der Abfrage *Auswertung*. Zusätzlich erfolgt die Erstellung einer XSL-Schemadatei mit dem Namen *AuswertungSchema.xml*.

**Listing:** Der Aufruf der Methode ExportXML ()

```
Application.ExportXML _
    ObjectType:=acExportTable, _
    DataSource:="Auswertung", _
    DataTarget:=" Auswertung.xml", _
    SchemaTarget:="AuswertungSchema.xml", _
    SchemaFormat:=acSchemaXSD, _
    OtherFlags:=1
```

### 3.5.2 ImportXML

Die Methode ImportXML() ermöglicht den Import von XML-Dokumenten in die aktuelle Datenbank. Für die zu importierenden Daten wird automatisch eine Tabelle erstellt, die den Namen des im Quelldokument enthaltenen Elementknotens enthält.

```
Ausdruck.ImportXML(DataSource, Data Transform, OtherFlags)
```

- *Ausdruck*. Ein Ausdruck, der ein Application-Objekt referenziert.
- *DataSource*. Der Name und der Pfad der zu importierenden XML-Datei. Diese Angabe ist unbedingt erforderlich.
- *DataTransform*. Der Name der XSL-Datei, die für die eingehenden XML-Daten ausgeführt werden soll. Die Angabe dieses Wertes ist optional.

- *OtherFlags*. Eine Bitmaske, die andere Verhaltensweisen im Zusammenhang mit dem Importieren aus dem XML-Format definiert. Um die entsprechende Funktionalität zu erhalten, können Sie die Werte miteinander addieren.

Wert	Beschreibung
1	Überschreiben. Die Importdatei überschreibt ggf. die Zieldatei, ohne darauf hinzuweisen.
2	Keine Struktur erstellen. Standardmäßig werden neue Strukturen erstellt. Ist das Überschreiben nicht festgelegt, werden Sie aufgefordert, das Überschreiben zu bestätigen.
4	Daten nicht importieren. Standardmäßig werden Daten importiert, wenn ein Datendokument zum Erstellen eines Schemas verwendet wird.

**Tabelle 3.5:** Die möglichen Werte für das Attribut *OtherFlags*

Beim Import von Daten ist das Attribut *OtherFlags* von besonderer Bedeutung. Mit den hier möglichen Einstellungen können Sie z.B. beeinflussen, ob eine bereits bestehende Tabelle durch die zu ersetzenden Daten erneuert werden soll. Wenn dies erfolgt, dann gehen die überschriebenen Daten unwiederbringlich verloren. Sie sollten sich also vor der Anwendung der Methode immer über die möglichen Konsequenzen im Klaren sein.

**Listing:** Der Aufruf der Methode `ImportXML`

```
Application.ImportXML _
    DataSource="C:\XMLData\Bestellungen.xml", _
    OtherFlags:=1
```

### 3.5.3 Realisierung des Datenexports

Das folgende Beispiel demonstriert den Export einer Tabelle aus einer Access-Datenbank über eine Abfrage in ein XML-Dokument. In dem Beispiel wird davon ausgegangen, dass sich innerhalb der Datenbank die zu exportierende Tabelle sowie eine Abfrage, welche diese Tabelle verwendet, befinden.

#### TIPP

Statt einer Abfrage können Sie selbstverständlich auch eine Tabelle oder einen Bericht verwenden. In der Praxis erweist es sich jedoch als sinnvoll, die zu exportierenden Daten vor dem Export mithilfe einer Abfrage zu sammeln und zu ordnen.

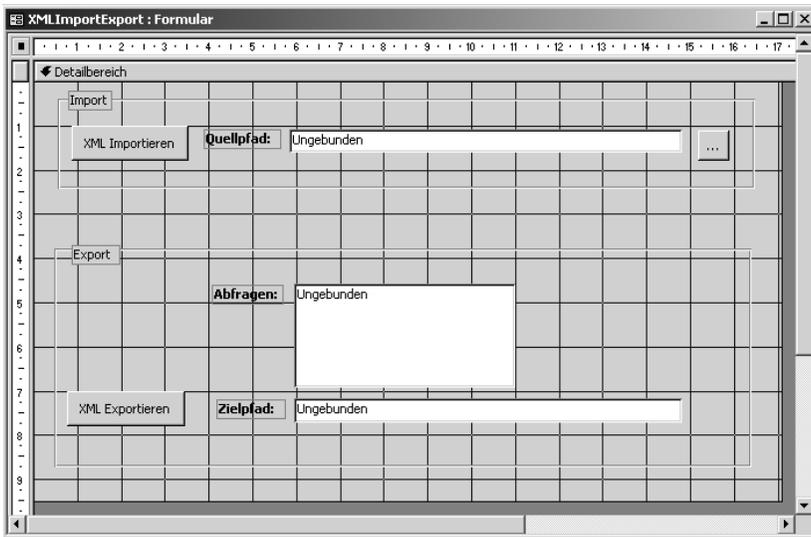
Das Beispiel soll dem Anwender zugleich eine Arbeitsoberfläche zur Verfügung stellen, über die er alle erforderlichen Schritte komfortabel ausführen und überwachen kann. Dafür benötigen Sie ein leeres Formular, auf dem alle benötigten Steuerelemente untergebracht werden. Dazu öffnen Sie ein neues Formular in der Entwurfsansicht. Standardmäßig enthält ein leeres Formular Elemente für die

Navigation innerhalb von gebundenen Datenquellen. Da diese Elemente hier nicht benötigt werden, öffnen Sie das Eigenschaftsfenster des neuen Formulars und nehmen in der Registerkarte *Format* folgende Einstellungen vor:

Eigenschaft	Einstellung
Bildlaufleisten	Nein
Datensatzmarkierer	Nein
Navigationsschaltflächen	Nein
Trennlinien	Nein
Größe anpassen	Nein

**Tabelle 3.6:** Veränderte Einstellungen in der Registerkarte *Format*

Nach dem Übernehmen der beschriebenen Einstellungen platzieren Sie auf dem Formular alle benötigten Steuerelemente. Dazu markieren Sie innerhalb der Entwurfsansicht das gewünschte Element in der Werkzeugsammlung und markieren anschließend dessen gewünschte Umrisse mit Maustaste auf dem Formular. Die Anordnung aller Elemente können Sie der folgenden Abbildung entnehmen.



**Abbildung 3.17:** Das Formular in der Entwurfsansicht

Die beiden Rahmen, welche die Elemente in zwei funktionelle Gruppen einteilen, können nach Belieben hinzugefügt werden, sie wirken nicht aktiv an dem Programm mit. Die Bezeichnungsfelder mit den fett hervorgehobenen Beschriftungen müssen Sie nicht explizit hinzufügen. Diese werden beim Einfügen der Textfelder und des Listenfeldes automatisch erstellt.

Steuerelement	Name	Beschriftung
Schaltfläche	btn_Import	XML Importieren
Schaltfläche	btn_Export	XML Exportieren
Schaltfläche	btn_Quelle	...
Textfeld	txt_Quelle	
Textfeld	txt_Ziel	
ListBox	lst_Abfrage	

**Tabelle 3.7:** Benötigte Steuerelemente

Widmen wir uns also zuerst der Funktionalität zum Import der Daten. Das Textfeld `txt_Quelle` dient zur Eingabe des Quellpfades des XML-Dokumentes. Da dies unter Umständen eine recht mühsame Aufgabe sein kann, soll dem Benutzer der Oberfläche ein Dateiauswahl-Dialog zur Verfügung gestellt werden. Bisher war dazu die Verwendung der Windows-API oder eines entsprechenden ActiveX-Controls erforderlich. Mit Office XP bzw. Access 2000 gehört dies der Vergangenheit an. Alle dazu benötigten Ressourcen befinden sich in der *Microsoft Office 10.0 Object Library*, welche auf einfache Weise in die Anwendung eingebunden werden kann.

Mithilfe der Schaltfläche `btn_Quelle` soll der Benutzer den Dateiauswahl-Dialog öffnen können. Um den Code zu der Schaltfläche zu bearbeiten, markieren Sie einfach in der Entwurfsansicht die Schaltfläche `btn_Quelle` und lassen sich den Eigenschaftsdialog anzeigen. In der Registerkarte *Ereignis* klicken Sie in das Eingabefeld neben *Beim Klicken*, womit sich das Eingabefeld in ein Auswahlfeld verwandelt. Wählen Sie hier den Eintrag *[Ereignisprozedur]* aus und klicken anschließend auf die Schaltfläche mit den drei Punkten. Damit öffnet sich der Visual Basic Editor, in dem automatisch die Ereignisprozedur `btn_Quelle_Click()` angelegt und angezeigt wird.

### Einbinden der Ressourcen

Als Erstes müssen Sie nun die bereits erwähnte Objektbibliothek einbinden, welche die Ressourcen für die Bereitstellung eines Dateialoges enthält. Dazu wählen Sie im Visual Basic Editor den Menüpunkt *Extras/Verweise*. In dem sodann erscheinenden Dialogfeld markieren Sie den Eintrag *Microsoft Office 10.0 Object Library und schließen den Vorgang mit der Schaltfläche OK ab*.

Jetzt können Sie den folgenden Code in das angelegte Codegerüst der Prozedur `btn_Quelle_Click()` eingeben. Um den Dateialog aufrufen zu können, müssen Sie zuerst eine Variable der Klasse erstellen, die diesen Dialog zur Verfügung stellt. Dazu kommt hier die Variable `fd` zum Einsatz. Bereits beim Deklarieren der Variablen `fd` werden Sie nach der Angabe von `As` die Anzeige eines frei schwebenden Auswahl Fensters bemerken, in welchem alle zur Verfügung stehenden Objekte und Klassen aufgelistet werden. Hier sollte auf alle Fälle auch die Klasse `FileDialog` enthalten sein. Wenn dies nicht so ist, dann überprüfen Sie unbedingt, ob die zuvor eingebundene Objektbibliothek auch wirklich vorhanden ist.

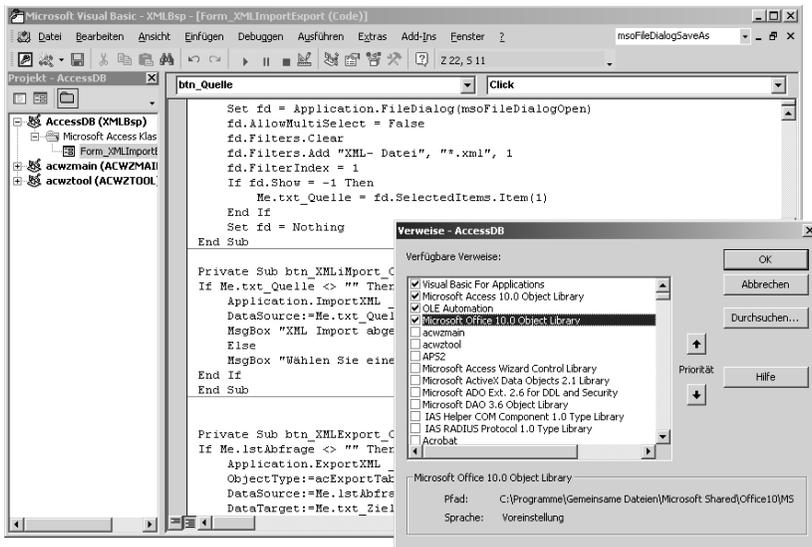


Abbildung 3.18: Das Einbinden einer Objektbibliothek

```

Private Sub btn_Quelle_Click()
    Dim fd As FileDialog
    Set fd = Application.FileDialog(msoFileDialogOpen)
    fd.AllowMultiSelect = False
    fd.Filters.Clear
    fd.Filters.Add "XML- Datei", "*.xml", 1
    fd.FilterIndex = 1
    If fd.Show = -1 Then
        Me.txt_Quelle = fd.SelectedItems.Item(1)
    End If
    Set fd = Nothing
End Sub

```

Mit der Initialisierung der Klassenvariable `fd` übergeben Sie eine von insgesamt vier möglichen Konstanten. Diese Konstanten repräsentieren die mit dem Dialogfeld zur Verfügung gestellte Funktionalität. Es gibt vier Typen von *FileDialog*-Objekten:

- **Öffnen-Dialogfeld:** *msoFileDialogOpen = 1*

Das Objekt ermöglicht dem Benutzer das Auswählen einer oder mehrerer Dateien. Es wird der Name der ausgewählten Datei und des entsprechenden Pfades zurückgegeben.

- **Speichern unter-Dialogfeld:** *msoFileDialogSaveAs = 2*

Ermöglicht das Auswählen des Speicherortes, in dem die aktuelle Datei dann gespeichert werden kann.

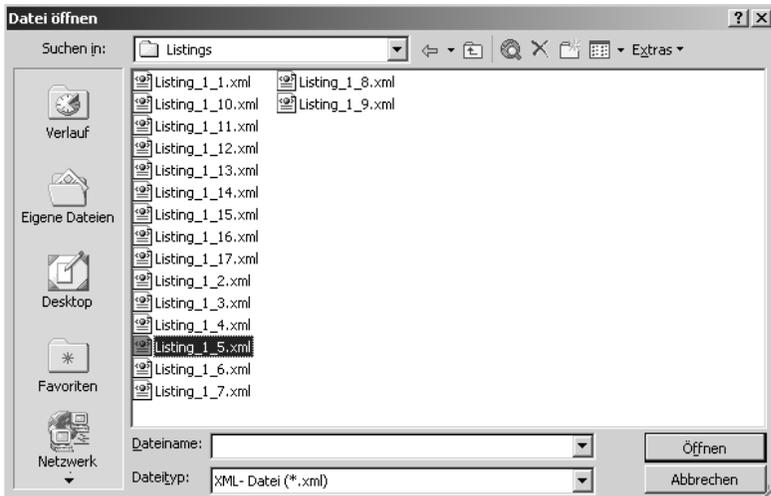
- *Dateiauswahl-Dialogfeld: `msoFileDialogFilePicker = 3`*

Ermöglicht dem Benutzer das Auswählen einer oder mehrerer Dateien. Die vom Benutzer ausgewählten Dateipfade sind in der Auflistung `FileDialog.SelectedItems` enthalten.

- *Ordnerauswahl-Dialogfeld: `msoFileDialogFolderPicker = 4`*

Ermöglicht das Auswählen eines Pfades. Der vom Benutzer ausgewählte Dateipfad ist in der Auflistung `FileDialog.SelectedItems` enthalten.

Der Aufruf des Dialogfeldes erfolgt mit der Methode `Show()` der Klasse `FileDialog`. Das Dialogfeld wird als modales Dialogfeld angezeigt, d.h. dass während dieser Zeit keine weiteren Aktivitäten in dem aufrufenden Programm möglich sind und die Ausführung von weiterem Code gestoppt wird. Mit dem Schließen des Dialogfeldes durch den Benutzer gibt die Methode einen numerischen Wert zurück, anhand dessen überprüft werden kann, ob eine Datei ausgewählt wurde. Wenn der Benutzer den Dialog ohne eine vorgenommene Auswahl schließt, dann gibt die Methode immer den Wert `-1` zurück. In dem hier verwendeten Code wird das Textfeld nur dann mit Daten versorgt, wenn eine Auswahl durch den Benutzer vorgenommen wurde.



**Abbildung 3.19:** Der Dialog zum Öffnen der XML-Datei

### Der Code zum Import der Daten

Mit der Auswahl einer Quelldatei wird also deren Name und Pfad in das Textfeld `txt_Quelle` übertragen. Alles andere ist jetzt nur noch ein Kinderspiel. Öffnen Sie jetzt wieder das Formular in der Entwurfsansicht und definieren Sie in der Eigenschaftsseite der Schaltfläche `btn_XMLImport` das Ereignis *Beim Klicken*. In die damit angelegte Prozedur `btn_XMLImport_Click()` geben Sie den folgenden Code ein:

**Listing:** Der Code zum Import eines XML-Dokumentes

```
Private Sub btn_XMLimport_Click()  
If Me.txt_Quelle <> "" Then  
    Application.ImportXML _  
        DataSource:=Me.txt_Quelle  
    MsgBox "XML Import abgeschlossen", vbInformation  
Else  
    MsgBox "Wählen Sie eine Quelldatei", vbExclamation  
End If  
End Sub
```

Statt dem Namen des Dokumentes wird hier der Eigenschaft `DataSource` der Inhalt des Textfeldes `txt_Quelle` übergeben. Dieses sollte also unbedingt auch einen gültigen Dateipfad enthalten. Mit dem Aufrufen der Methode `ImportXML` erfolgt anschließend der Import des XML-Dokumentes.

Aus Gründen der Übersichtlichkeit erfolgt hier der Aufruf der Methode nur mit einigen wenigen Attributen. Aus der zuvor erfolgten Beschreibung der möglichen Attribute können Sie jedoch ersehen, dass hier wesentlich mehr Möglichkeiten zum Anpassen der Methode vorhanden sind.

### Zugriff auf Datenbankobjekte für den Export

In dem hier vorgestellten Beispiel erfolgt der Export von Daten nicht aus einer Tabelle, sondern aus einer Abfrage heraus. Der Grund dafür ist der, dass in Access eine Abfrage die effektivste Möglichkeit bietet, alle benötigten Daten vor dem Export in der benötigten Art und Weise zusammenzustellen. Um dem Anwender eine einigermaßen komfortable Anwendung zur Verfügung zu stellen, sollte er über die Möglichkeit einer einfachen Auswahl aus den vorhandenen Abfragen verfügen. Dies wird hier mit einem `ListBox`-Steuerelement realisiert. Dabei werden alle in der Aufzählung `AllQueries` enthaltenen Objekte durchlaufen und diese mithilfe der Methode `AddItem()` der `ListBox` als Eintrag hinzugefügt.

```
Private Sub Form_Load()  
    Dim obj As AccessObject, dbs As Object  
    Set dbs = Application.CurrentData  
    For Each obj In dbs.AllQueries  
        Me.lstAbfrage.AddItem obj.Name  
    Next obj  
End Sub
```

Damit sind wir schon fast am Ende des Beispiels angelangt. Jetzt müssen nur noch die Inhalte der `ListBox` und des Textfeldes, welches die Informationen über den Zielpfad des zukünftigen Dokumentes enthält, als Werte auf die Attribute der Methode `XMLExport()` angewendet werden.

```

Private Sub btn_XMLExport_Click()
If Me.lstAbfrage <> "" Then
    Application.ExportXML _
        ObjectType:=acExportTable, _
        DataSource:=Me.lstAbfrage, _
        DataTarget:=Me.txt_Ziel & Me.lstAbfrage & ".xml", _
        SchemaTarget:=Me.lstAbfrage & ".xml", _
        OtherFlags:=1
    MsgBox "XML Export abgeschlossen", vbInformation
Else
    MsgBox "Wählen Sie eine Abfrage", vbExclamation
End If
End Sub

```

Auch hier gilt wieder, dass es sich nur um eine möglichst einfache und übersichtliche Demonstration der in Access enthaltenen Möglichkeiten handelt. Mit der individuellen Anwendung der Attribute der Methode `XMLExport()` sind Sie nun in der Lage, eine auf die verschiedensten Bedürfnisse zugeschnittene Lösung zu erstellen.

### 3.5.4 Elemente beim Import und Export

Beim Import eines XML-Dokumentes nach Access muss der Inhalt des Dokuments innerhalb einer Tabellenstruktur gespeichert werden können. Damit bei diesem Vorgang die Zuordnung aller Elemente des Dokuments möglich ist, erfolgt die Interpretation des Dokumentinhaltes auf eine bestimmte Art und Weise. Die Zuordnung basiert auf der Erkennung der Dokumentstruktur, was wiederum die Einhaltung der Richtlinien für wohlgeformtes XML erfordert.

XML-Objekt	Beispiel	Übernahme in Access
XML-Deklaration	<?xml version="1.0"?>	nein
Hauptelement	<auto>	Tabellenname
Element	<typ>	Feldname
Attribut	farbe="rot"	nein
Wert	"limousine"	Feldwert

**Tabelle 3.8:** Die Interpretation der Bestandteile eines XML-Dokumentes

So werden z.B. die Tags zur Auszeichnung der verwendeten XML-Version ignoriert. Der Name des Hauptelementes wird hingegen als der Name der zukünftigen Tabelle verwendet. In der Tabelle sehen Sie eine Gegenüberstellung verschiedener Bestandteile eines XML-Dokumentes und deren Umsetzung beim Import. Bei der Erkennung von Attributen beim Import eines XML-Dokumentes besteht eine kleine Einschränkung. Access 2002 erkennt nur Attribute in den XML-Schemadateien, die beim Export aus Access 2002 erstellt wurden.

Beim Export von Daten in ein XML-Dokument erfolgt die Handhabung der hier gezeigten Regeln in der umgekehrten Art und Weise. Der Name einer Tabelle bildet standardmäßig den Namen der Zieldatei und des Hauptelementes. Dieser Name wird ebenfalls als Name für ein optional zu erstellendes XSL-Schema verwendet. Die Namen der Spalten der zu exportierenden Tabelle bilden die Bezeichner der Elemente, welche die in den Tabellenspalten enthaltenen Werte aufnehmen.

### 3.5.5 Datentypen beim Import und Export

Wenn Sie in den vorhergegangenen Abschnitten einige der Beispiele nachvollzogen haben, dann sollten Sie unbedingt einmal einen Blick in die beim Import erstellten Tabellen bzw. XML-Dokumente werfen. Beim Vergleich der Ergebnisse wird Ihnen sicherlich auffallen, dass die Konvertierung bzw. die Übernahme der Datentypen unter Umständen etwas eigenwillig vonstatten geht. Der Grund dafür ist der, dass mit einem XSL-Schema nur eine bestimmte Anzahl von Datentypen beschrieben werden kann. Aus dem Grund erfolgt folgerichtig eine Typenkonvertierung anhand bestimmter Richtlinien.

#### HINWEIS

Beim Import von XML-Dokumenten kann eine Typenumwandlung nur dann erfolgen, wenn zu dem Dokument ein gültiges XSL-Schema zugeordnet werden kann. Dabei ist es unerheblich, ob dieses in dem Dokument eingebettet ist oder ob es sich in einer XSL-Datei befindet.

In der folgenden Tabelle finden Sie die äquivalenten XSD- und SQL-Typen, die beim Import und beim Export von Jet-Datentypen aus Access erstellt werden.

Jet-Datentyp (gebräuchlicher Name)	Jet-Datentyp (Access)	SQL-Datentyp	XSD-Datentyp
Text	Text	nvarchar	string
Memo	memo	ntext	string
Zahl (Byte)	Byte	tinyint	unsignedbyte
Zahl (Integer)	integer	smallint	short
Zahl (Long Integer)	longinteger	int	integer
Zahl (Single)	single	real	float
Zahl (Double)	double	float	double
Zahl (Repln ID)	replicationid	uniqueidentifier	binary
Zahl (Dezimal)	decimal	decimal	decimal
Datum/Uhrzeit	datetime	datetime	timeinstant

**Tabelle 3.9:** XSD- und SQL-Datentypen

Jet-Datentypen (gebräuchlicher Name)	Jet-Datentyp (Access)	SQL-Datentyp	XSD-Datentyp
Währung	currency	money	double
AutoWert	autonumber	int	integer
Ja/Nein	yesno	bit	Boolean
OLE-Objekt	oleobject	image	binary
Hyperlink	hyperlink	ntext	string
Datum (Legacy)	Date	datetime	date
Uhrzeit (Legacy)	Time	datetime	time
Binär	binary	varbinary image	binary

**Tabelle 3.9:** XSD- und SQL-Datentypen (Forts.)

### 3.5.6 In Access verwendete Namensräume

Ein weiteres interessantes Detail eines exportierten XML-Dokumentes finden Sie bei der Betrachtung des automatisch erstellten Namensraumes. Nehmen wir als Beispiel den beim Export der Tabelle *Employees* erstellten Namensraum.

```
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="employees.xsd">
```

Auf den ersten Blick wirkt die hier erstellte Deklaration des Namensraumes als unübersichtlich und verleitet den Anwender eher zum Überlesen der darin enthaltenen Bestandteile. Beim genaueren Hinsehen werden Sie jedoch bemerken, dass jeder Abschnitt eine ganz spezielle Funktion ausübt.

- `dataroot`  
Das Wurzelement, welches alle weiteren Elemente umschließt.
- `xmlns:od="urn:schemas-microsoft-com:officedata"`  
Die Deklaration des Namensraumes, welches das Dokument als Office-Bestandteil ausweist: Bei `xmlns:od` handelt es sich um den Verweis auf den Namensraum *Office Data*, mit dem die in dem Quelldokument enthaltenen Datentypen beschrieben werden.
- `xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"`  
Ein eindeutiger Wert, der es ermöglicht, den Namensraum eindeutig zuzuordnen
- `xsi:noNamespaceSchemaLocation="employees.xsd"`  
Die Angabe einer Referenz auf das zu dem Dokument gehörende XSL-Schema. Hier handelt es sich um ein in einer externen Datei befindliches Schema.

Nachdem nun der Aufbau und der Inhalt eines von Access erstellten Namensraumes etwas verständlicher geworden ist, lohnt sich auf alle Fälle ein Blick in das dazu erstellte XML-Schema. Die XSL-Datei befindet sich nach dem erfolgreich verlaufenen Export standardmäßig im gleichen Verzeichnis wie das erstellte XML-Dokument. Hier finden Sie die Deklaration der Namensräume für das XSL-Schema, wobei das Wurzelement immer mit dem Bezeichner `Schema` ausgezeichnet wird.

```
<xsd:schema xmlns:xsd=http://www.w3.org/2000/10/XMLSchema
  xmlns:od="urn:schemas-microsoft-com:officedata">
```

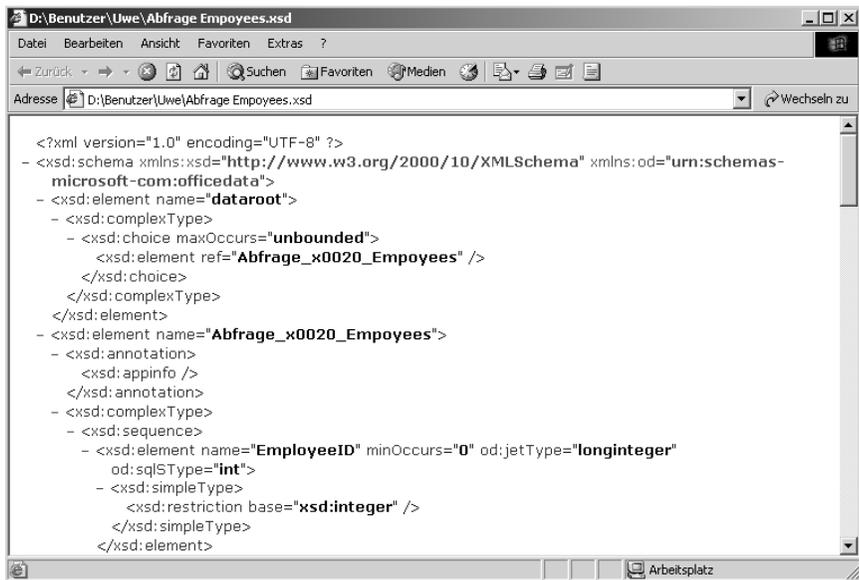


Abbildung 3.20: Die XSL-Schemadefinitionen

Mit den in dem Namensraum enthaltenen Elementen erfolgt die Beschreibung der Daten des Dokumentes in der folgenden Art und Weise:

```
<xsd:element name="City" minOccurs="0" od:jetType="text"
  od:sqlSType="nvarchar">
```

Hier erfolgt die Zuordnung des Elementes zu dem XSD-Namensraum und die Zuordnung der Datentypbeschreibung zu dem Office Data-Namensraum `od`. Die hier verwendeten Beschreibungen für Datentypen finden Sie in der Tabelle des letzten Abschnittes wieder.

Die beiden hier gezeigten Namensräume der XML-Schemadefinition (XSD) und Office Data (OD) verweisen auf die beiden einzigen von Access 2002 unterstützten Schemadefinitionen. Ungeachtet dessen ist es möglich, ein erstelltes Dokument

beliebig um eigene weitere Schemadefinitionen zu erweitern, ohne dass dabei die automatisch erstellte Dokumentstruktur zerstört wird. Mit einer solchen Erweiterung sind Sie z.B. in der Lage, ein Dokument mit eigenen Kommentaren zu versehen, mit denen Sie zusätzliche Informationen an den Empfänger hinzufügen können. Dazu fügen Sie einfach einen weiteren Bezeichner hinzu, der auf Ihre eigene XML-Schemadefinition verweist.

```
<dataroot xmlns:meinkomentar="urn:komentar">
  <xsd:element xsd:name="Bestelldatum" xsd:datum="01.02.01"
    komentar:date="Die Bestellung wurde storniert">
    ...
</dataroot>
```

## 3.6 ReportML

Der bisher mögliche Weg zum Exportieren von Daten aus Access nach XML besteht im Konvertieren einer Ergebnismenge in ein XML-Dokument. In den bisher behandelten Abschnitten haben Sie gesehen, dass die Quelle der Ergebnismenge eine Tabelle, eine Abfrage, ein Formular oder auch ein Bericht sein kann. Dabei erfolgte aber lediglich die Speicherung aller enthaltenen Daten und bis zu einem gewissen Punkt auch die Speicherung von Formaten und Typen. Damit ist es zwar möglich, die gespeicherten Informationen anderen Anwendungen zur Verfügung zu stellen, aber die Quellobjekte sind dort nicht mehr verfügbar.

Mit Access 2002 ist es nun erstmals möglich, auch Access-Objekte in ein XML-Dokument zu exportieren und diese somit anderen Anwendungen zur Verfügung zu stellen. Das Ziel ist es, die exportierten Access-Objekte innerhalb eines Webrowsers anzeigen zu lassen. Der Vorteil dieser Technik liegt klar auf der Hand: Bisher mussten die Empfänger eines Access-Berichtes entweder Access auf ihren Rechner installiert oder einen entsprechenden Viewer zur Verfügung haben, was damit zur Vergangenheit gehören soll.

### HINWEIS

Die hier beschriebenen Möglichkeiten sind zwar offiziell bereits in Access 2002 implementiert, aber zum Zeitpunkt der Entstehung des Buches standen Sie noch nicht vollständig zur Verfügung. Mit der endgültigen Verfügbarkeit wird mit dem Erscheinen des nächsten verfügbaren Service-Packs gerechnet. Wer diese Möglichkeiten in Zukunft nutzen will, der kann sich hier schon damit vertraut machen.

ReportML stellt eine von Microsoft entwickelte Auszeichnungssprache dar, mit der es möglich ist, Eigenschaften von Access-Objekten mithilfe von Tags zu beschreiben. Dabei müssen die verwendeten Tags letztendlich den Regeln für wohlgeformtes XML entsprechen.

Beim Speichern von Access-Formularen nach XML gelten folgende Regeln:

- Die angezeigten Daten ReportML-Dokument sind rein statisch und können nicht bearbeitet oder überschrieben werden.
- Angezeigte Formulare verfügen über keine Navigationsschaltflächen.
- Als eingebettete OLE-Objekte enthaltene Grafiken gehen beim Export verloren. Bei der Verwendung von Grafiken sollten diese stattdessen in Image-Controls verwaltet werden.
- Da die angezeigten Daten rein statischer Natur sind, erfolgt konsequenterweise eine Umwandlung von Auswahlfeldern in Textfelder. Bei Auswahlfenstern, die nur einen Eintrag anzeigen und deren restliche Spalten ausgeblendet sind, erfolgt nur die Übernahme der ersten angezeigten Spalte.
- Der Code, welcher zu einem Formular gehört, wird nicht exportiert und bleibt damit dem Empfänger des Dokumentes verborgen.

Beim Speichern von Access-Berichten nach XML gelten folgende Regeln:

- Es sind nur zwei Basistypen von Berichten in der Spalten- und Tabellenansicht zum Speichern in XML vorgesehen.
- Ein Bericht kann nur in XML gespeichert werden, wenn er geschlossen ist.
- Als eingebettete OLE-Objekte enthaltene Grafiken gehen beim Export verloren.
- Auswahlfelder werden als Textfelder umgewandelt.
- Die Eigenschaften zum automatischen Anpassen von Feldern an die Berichtsgröße stehen nicht zur Verfügung.
- Es erfolgt keine Unterstützung und Ausführung von Aggregatfunktionen.
- Die Sortierung und die Gruppierung der Daten eines Berichtes bleibt erhalten.
- Bei mehrseitigen Berichten erfolgt nur die einmalige Anzeige von Kopf- und Fußbereichen.
- Der zu einem Bericht gehörende Code wird nicht exportiert und bleibt somit dem Empfänger des Dokumentes verborgen.

In der folgenden Tabelle sehen Sie die Umsetzung der XML-Tags in die neuen ReportML-Tags.

Objekte	Tagtyp	Beispiele
Formular- oder Berichtsabschnitte, z.B. Kopf- und Fußzeilen	Element	<SECTION>...</SECTION>
Primäre Objekteigenschaften, z.B. Visibility, CanGrow, GroupLevel	Element	<CAN-GROW>true</CAN-GROW> <VISIBILITY>visible</VISIBILITY>
Zusätzliche Eigenschaftsinformationen, z.B. Datentyp	Attribute	<NAME type="last"> Berge </NAME>
Objektname (»id«) oder Objekttypen, wie Bestellungen_Unterbericht	Attribute	<REPORT id="Bestellungen_Unterbericht"> <SECTION type="report_header">

**Tabelle 3.10:** Tags für die Verwendung in einem ReportML-Dokument

Mithilfe der hier gezeigten Tags kann nun ein bestehendes Access-Objekt vollständig beschrieben werden, wobei wie bereits gesagt verschiedene Einschränkungen in Bezug auf die Darstellung der darin enthaltenen Elemente vorliegen. In dem folgenden Beispiel sehen Sie einen Lösungsvorschlag, mit dem die Anzeige eines Formulars erfolgen kann. In dem erzeugten Code erfolgt die Angabe des Formularobjektes innerhalb einer Baumstruktur, welche die Eigenschaften und Attribute des Berichtobjekts sowie die Eigenschaften und Attribute der in dem Bericht enthaltenen Abschnitte und Steuerelemente beschreibt. Alle Informationen über Elemente des Berichts, wie z.B. Drucker-Objektinformationen, werden als eigener Bereich mit Tags exakt definiert. Das Gleiche trifft auch auf Informationen über die Verbindungseigenschaften zur Datenbank zu. Aktuelle Werte befinden sich wie gewohnt zwischen dem Start- und dem End-Tag eines Elementes. Hierbei wird auf die Verwendung von Attributen gänzlich verzichtet. Wenn ein Element über keine Eigenschaften verfügt, also leer ist, dann wird auf deren Darstellung ebenfalls verzichtet. Innerhalb des Bereiches, der das Hauptdokument beschreibt, kann im Prinzip eine beliebige Anzahl von Unterelementen, also Unterberichte und Unterformulare, enthalten sein.

**Listing:** Ein Beispiel für die Vorgehensweise zur Verwendung von ReportML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Wurzelement -->
<RPTML>
<SYSTEM-SETTINGS>
  <LOCALE>1031</LOCALE>
</SYSTEM-SETTINGS>
<!-- Definition des Formulars "Bestellung"-->
<REPORT ID="Orders">
  <TITLE>Bestellungen</TITLE>
```

```

<LAYOUT>absolute</LAYOUT>
<OBJECT-TYPE>form</OBJECT-TYPE>
<!-- Druckinformationen -->
<PRINTER-ROW-SPACING>0</PRINTER-ROW-SPACING>
<PRINTER-ORIENTATION>portrait</PRINTER-ORIENTATION>
<!-- weitere Formulareigenschaften -->
<FETCH-DEFAULTS>>true</FETCH-DEFAULTS>
<MOVEABLE>>false</MOVEABLE>
<ALLOW-DESIGN-CHANGES>>false</ALLOW-DESIGN-CHANGES>
<!-- Eigenschaften der Verbindungszeichenfolge -->
<RECORD-SOURCE>Orders Qry</RECORD-SOURCE><ID>Orders</ID>
<CONNECTION-STRING>PROVIDER=Microsoft.Jet.OLEDB.4.0;
    DATA SOURCE=\\zaw-pro-1\mydocs2\frice\Northwind.mdb;
    USER ID=Admin;PASSWORD=;PERSIST SECURITY INFO=FALSE;
</CONNECTION-STRING>
<!-- Globale Formatvorlagendefinitionen fuer Komponenten
    und Abschnitte -->
<STYLE id="Style0">
    <BORDER-COLOR>#000000</BORDER-COLOR>
    <BORDER-WIDTH>1px</BORDER-WIDTH>
</STYLE>
<!-- Eigenschaften Formularabschnitte -->
<SECTION id="ReportHeader" type="report_header">
    <BORDER-STYLE>none</BORDER-STYLE>
    <BACKGROUND-COLOR>#008080</BACKGROUND-COLOR>
</SECTION>
<REPORT-ITEM id="OrderID" type="text-box">
    <CLASS>Style0</CLASS>
    <BOUND-TO-HYPERLINK>>false</BOUND-TO-HYPERLINK>
</REPORT-ITEM>
<REPORT-ITEM id="OrderID Label" type="label">
    parent="OrderID">
    <CLASS>Style1</CLASS>
    <HEIGHT>0.177in</HEIGHT>
</REPORT-ITEM>
<!-- Ende des Abschnittes, der das Formular "Bestellung"
    beschreibt -->
</REPORT>
<!-- Unterberichtsdefinition -->
<REPORT ID="Bestellungen_Unterbericht">
    ...
</REPORT>
</RPTML>

```