

9 Upsizing der Datenbank Bücher

In diesem Kapitel wird aus der Access-Datenbankanwendung *Bücher* mit Hilfe des Upsizing-Assistenten von Access eine waschechte Client/Server-Anwendung. Sie lernen hier den Umgang mit dem Upsizing-Assistenten und anschließend die in den allermeisten Fällen notwendige Fehlerbehebung im neu angelegten Access-Projekt kennen.

Upsizing: Hintergründe und Begriffe

Upsizing der Datenbank Bücher

Das Upsizing einer Access-Datenbank wird – wie schon erwähnt – in der Regel dann notwendig, wenn die Anwendungen zu groß geworden sind und/oder zu viele Anwender in einer Netzwerkumgebung gleichzeitig auf die Anwendungen zugreifen. Beides äußert sich letztlich durch identische Symptome:

Bei der Arbeit mit der Datenbankanwendung ist keine zufrieden stellende Geschwindigkeit mehr gewährleistet.

Upsizing: Hintergründe und Begriffe

Bei der Portierung einer Access-Datenbank auf den SQL Server bzw. die MSDE gibt es verschiedene Regeln zu beachten, die mit unterschiedlichen Datentypen, eingebauten Funktionen und der unterschiedlichen Funktionalität zusammenhängen. Außerdem verfügt ein Access-Projekt bzw. die MSDE über Objekte, die eine Access-MDB gar nicht kennt.

Damit der Benutzer bei Problemen, die nach dem Upsizing auftreten können, genau weiß, wonach er schauen muss, sind die vorherige Klärung einiger Begriffe und verschiedene Hintergrundinformationen notwendig.

Das Problem liegt einfach darin begründet, dass der Upsizing-Assistent einige Dinge schlicht und einfach nicht korrekt konvertieren kann. Deshalb ist im Prinzip nach jeder Konvertierung mehr oder weniger viel Handarbeit erforderlich.

Vergleich von Datenbankobjekten

Wir beginnen mit einer Gegenüberstellung von Datenbankobjekten, die Ihnen in einem Access-Projekt bzw. einer Access-MDB begegnen können.

Access-Datenbank	Access-Projekt
<i>Tabelle</i>	<i>Tabelle</i>
<i>Auswahlabfrage</i>	<i>Sicht (View)</i>
<i>Aktions- oder Parameterabfrage</i>	<i>Gespeicherte Prozedur</i>
<i>Beziehungsfenster</i>	<i>Datenbankdiagramm</i>
<i>Formular</i>	<i>Formular</i>
<i>Bericht</i>	<i>Bericht</i>
<i>Datenzugriffsseite</i>	<i>Datenzugriffsseite</i>
<i>Makro</i>	<i>Makro</i>
<i>Modul</i>	<i>Modul</i>

Tabelle 1: Vergleich von Datenbankobjekten

Weitere typische SQL Server-Objekte lernen Sie in einem separaten Abschnitt weiter unten in diesem Kapitel kennen.

Vergleich von Datentypen

In Access stehen neun Datentypen, wobei beim Datentyp *Zahl* über die Eigenschaft *Feldgröße* weitere Einstellungen vorgenommen werden. Die MSDE kennt einige weitere Datentypen, die es bei Access nicht gar nicht gibt.

Access-Datentypen	SQL Server-Datentypen
<i>Ja/Nein</i>	<i>Bit</i>
<i>Zahl (Byte)</i>	<i>tinyint</i>
<i>Zahl (Integer)</i>	<i>smallint</i>
<i>Zahl (Long Integer)</i>	<i>int</i>
<i>Zahl (Single)</i>	<i>real</i>
<i>Kein Gegenstück</i>	<i>bigint</i>

Upsizing: Hintergründe und Begriffe

Zahl (Double)	(Double) float
Währung	money, smallmoney
Dezimal/Numerisch	decimal, numeric
Datum/Uhrzeit	datetime, smalldatetime
AutoWert (Inkrement)	int (mit definierter Identität ID-Eigenschaft)
Text	varchar, nvarchar
Memo	Text
OLE-Objekt	image
Replikations-ID (GUID bzw. Globally Unique Identifier)	uniqueidentifier
Hyperlink	char, nchar, varchar, nvarchar (Hyperlink-Eigenschaft Ja)
Kein Gegenstück	varbinary
Kein Gegenstück	smallint
Kein Gegenstück	timestamp
Kein Gegenstück	char, nchar
Kein Gegenstück	sql_variant
Kein Gegenstück	Benutzerdefiniert

Tabelle 2: Vergleich der Datentypen

Vergleich von Trennzeichen und Operatoren

Der SQL Server und Access verwenden unterschiedliche Trennzeichen, Operatoren, Platzhalterzeichen in der **WHERE**-Klausel eines SQL-Strings und Zeichen für Datums- bzw. Textangaben. Oftmals werden diese Trennzeichen und Operatoren vom Assistenten nicht korrekt umgesetzt, so dass anschließend entsprechende Nacharbeiten erforderlich sind.

In der folgenden Tabelle werden die Unterschiede gegenübergestellt.

Vergleich von	Access	SQL Server
Datumsangaben	Leiterzeichen (#)	Einfaches Anführungszeichen (')
Zeichenfolge	Einfaches (') oder Doppelte Anführungszeichen (")	Einfaches Anführungszeichen (')
Mod-Operator	Mod	Prozentzeichen (%)
Verkettung von Zeichenfolgen	Ampersand (&)	Pluszeichen (+)
Platzhalter für ein einzelnes Zeichen	Fragezeichen (?)	Unterstrich (_)
Platzhalter für kein oder mehrere Zeichen	Sternzeichen (*)	Prozentzeichen (%)
Konstante für True	Ja, Ein, Wahr oder - 1	1
Konstante für False	Nein, Aus, Falsch oder 0	0

Tabelle 3: Vergleich von Trennzeichen und Operatoren

Vergleich des SQL-Befehlssatzes

Der SQL-Befehlssatz von Access und der MSDE bzw. dessen Implementierung ist in einigen Punkten doch recht unterschiedlich. Neben Zeichen und Operatoren (siehe vorheriger Abschnitt) gibt es weitere wichtige Unterschiede:

Ausgabefelder

Access erlaubt in einer Abfrage mehrere Ausgabefelder mit gleichem Namen. Der SQL Server erlaubt dies in einem View aber nicht.

Upsizing: Hintergründe und Begriffe

Das SQL-Element DISTINCTROW

Access unterstützt das Element DISTINCTROW, das Daten aus mehrfach auftretenden Datensätzen (und nicht nur aus mehrfach auftretenden Feldern) nur einmal auswählt.

Darüber hinaus können in Access eindeutige Datensätze durch entsprechende Eigenschaftseinstellungen innerhalb einer Abfrage ausgewählt werden.

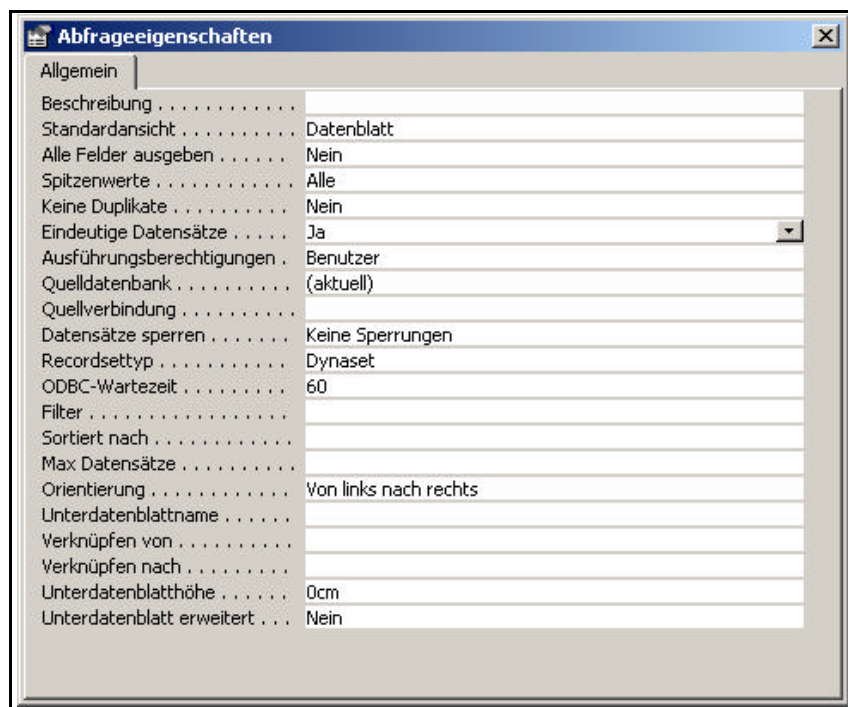


Abbildung 1: Eigenschaftsfenster einer Access-Abfrage

Es geht hier um die Eigenschaft *Eindeutige Datensätze*, die auf *Ja* eingestellt werden kann.

Der SQL Server dagegen unterstützt alle diese Einstellungen nicht.

SQL-Befehle COMMIT und LOCK

Diese SQL-Befehle, die für Zugriffsrechte verwandt werden, werden von Access nicht unterstützt. In Access können sie aber mit Hilfe von VBA-Code nachgebildet werden.

VBA-Anweisungen in SQL

Access erlaubt die Verwendung von beliebigen in VBA-Anweisungen im SQL-Code. Diese nicht unerhebliche Funktionserweiterung steht für den SQL Server bzw. die MSDE nicht zur Verfügung.

Weitere SQL-Elemente

In der folgenden Tabelle werden einige weitere wichtige SQL-Elemente, ihre Unterschiede und ihre Behandlung durch den Upsizing-Assistenten gegenübergestellt.

SQL-Element	Access	SQL Server	Upsizing-Assistent
TOP (Spitzenwerte)	Verfügbar	Verfügbar	
CREATE INDEX	Erlaubt auf- und absteigende Indizes		Konvertierung
DROP INDEX	Syntax: DROP INDEX <Indexname> ON <Tabellenname>	Syntax: DROP INDEX <Tabellenname>, <Indexname>	Keine Konvertierung
OWNER ACCESS	Verfügbar	Nicht verfügbar	Wird entfernt
ORDER BY in UNION-Abfragen	Mehrere ORDER BY-Klausel erlaubt	Eine ORDER BY-Klausel am Ende erlaubt	Keine Konvertierung
TRANSFORM	Verfügbar für Kreuztabellenabfragen	Nicht verfügbar	Keine Konvertierung

Upsizing: Hintergründe und Begriffe

PARAMETERS	Verfügbar	Nicht verfügbar	Parameter wird entfernt
Bezeichner	Begrenzt auf 64 Zeichen, Schlüsselwörter und spezielle Zeichen erlaubt	Mit den Access-Bezeichnern kompatibel	Konvertierung in SQL Server-Syntax

Tabelle 4: Vergleich von SQL-Elementen

Vergleich von VBA- und Transact-SQL-Funktionen

Wenn in einer Access-Abfrage Funktionen verwendet werden, ist es möglich, dass diese vom SQL Server nicht unterstützt werden.

In einem solchen Fall bleibt Ihnen nichts anderes über, als die betreffenden Funktionen per Hand anzupassen.

Im VBA-Code können die Access-Funktionen weiterhin verwendet werden.

Zeichenfolgefunktionen

Der Upsizing-Assistent konvertiert die folgenden VBA-Funktionen, falls sie als Gültigkeitsregeln für Tabellen und Felder bzw. in Standardwerten vorkommen, in die entsprechenden Transact-SQL-Funktionen um.

VBA-Funktion	Transact-SQL-Funktion
Asc(x)	ASCII(x)
Chr\$(x)	CHAR(x)
Lcase\$(x)	LCASE(x), LOWER(x)
Len(x)	DATALENGTH(x)
Ltrim\$(x)	LTRIM(x)
Mid\$(x, y, z)	SUBSTRIN(x, y, z)
Right\$(x)	RIGHT(x, y)

Rtrim\$(x)	RTRIM(x)
Space\$(x)	SPACE(x)
Str\$(x)	STR(x)
Ucase\$(x)	UCASE(x), UPPER(x)

Tabelle 5: Vergleich von Zeichenfolgefunktionen

Konvertierungsfunktionen

Der Upsizing-Assistent konvertiert die folgenden VBA-Funktionen, falls sie als Gültigkeitsregeln für Tabellen und Felder bzw. in Standardwerten vorkommen, in die entsprechenden Transact-SQL-Funktionen um.

VBA-Funktion	Transact-SQL-Funktion
Ccur(x)	CONVERT(money, x)
Cdbl(x)	CONVERT(float, x)
Cint(x)	CONVERT(smallint, x)
Clng(x)	CONVERT(int, x)
Csng(x)	CONVERT(real, x)
Cstr(x)	CONVERT(varchar, x)
Cvdate(x)	CONVERT(datetime, x)

Tabelle 6: Konvertierungsfunktionen

Datumsfunktionen

Der Upsizing-Assistent konvertiert die folgenden VBA-Funktionen, falls sie als Gültigkeitsregeln für Tabellen und Felder bzw. in Standardwerten vorkommen, in die entsprechenden Transact-SQL-Funktionen um.

VBA-Funktion	Transact-SQL-Funktion
Date(x)	CONVERT(DATETIME, CONVERT(varchar, GETDATE(x)))
Dateadd("<Access	DATEADD(<SQL Server datepart>, x, y)

Upsizing: Hintergründe und Begriffe

datepart>", x, y)	
Datediff(" <Access datepart>", x, y)	DATEDIFF(<SQL Server datepart>, x, y)
Datepart(" <Access datepart>", x)	DATEPART(<SQL Server datepart>, x)
Day(x)	DATEPART(dd, x)
Hour(x)	DATEPART(hh, x)
Minute(x)	DATEPART(mi, x)
Month(x)	DATEPART(mm, x)
Now(x)	GETDATE(x)
Second(x)	DATEPART(ss, x)
Weekday(x)	DATEPART(dw, x)
Year(x)	DATEPART(yy, x)

Tabelle 7: Datumsfunktionen

Mathematische Funktionen

Der Upsizing-Assistent konvertiert die folgenden VBA-Funktionen, falls sie als Gültigkeitsregeln für Tabellen und Felder bzw. in Standardwerten vorkommen, in die entsprechenden Transact-SQL-Funktionen um.

VBA-Funktion	Transact-SQL-Funktion
Int(x)	FLOOR(x)
Sgn(x)	SIGN(x)
Rnd(x)	ROUND(x)

Tabelle 8: Mathematische Funktionen

Spezielle SQL Server-Objekte

In diesem Abschnitt lernen Sie einige typische Objekte von Access-Projekten und ihre wichtigsten Eigenschaften im Unterschied zu Access-Datenbanken kennen.

Datenbankdiagramme

Datenbankdiagramme entsprechen im Prinzip dem Beziehungsfenster von Access, wofür wohl in erster Linie die Ähnlichkeit in der Darstellung verantwortlich ist.

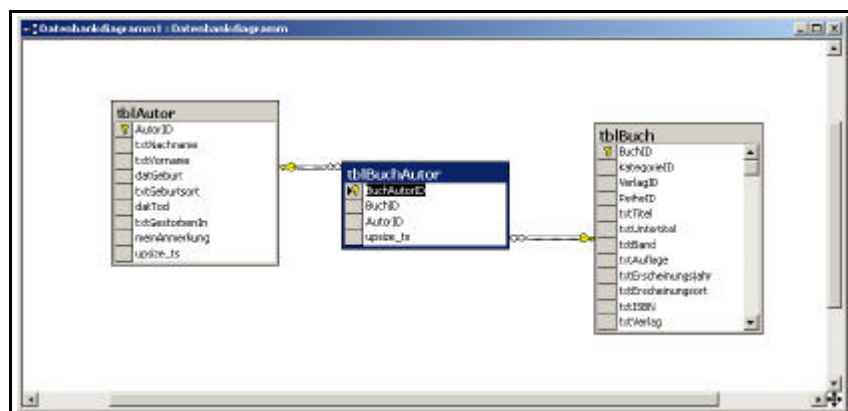


Abbildung 2: Datenbankdiagramm

Nur können in einem Datenbankdiagramm neben dem reinen Festlegen und Bearbeiten von Beziehungen noch folgende Aufgaben erledigt werden:

- Anlage und Bearbeitung von Tabellen- und Feldeigenschaften

Spezielle SQL Server-Objekte

- Anlage und Bearbeitung von Indizes

Views

Views entsprechen den Auswahlabfragen in Access mit einigen wichtigen Unterschieden:

- Views dürfen keine Parameter enthalten. Auch hier muss eine gespeicherte Prozedur verwendet werden, falls Parameter übergeben werden müssen.
- Views können keine Verweise auf Formulare oder Steuerelemente in Formularen enthalten.

Das folgende Beispiel einer Access-Auswahlabfrage, bei der als Kriterium ein Formular-Steuerelement angegeben ist, kann so in einem View nicht verwendet werden.

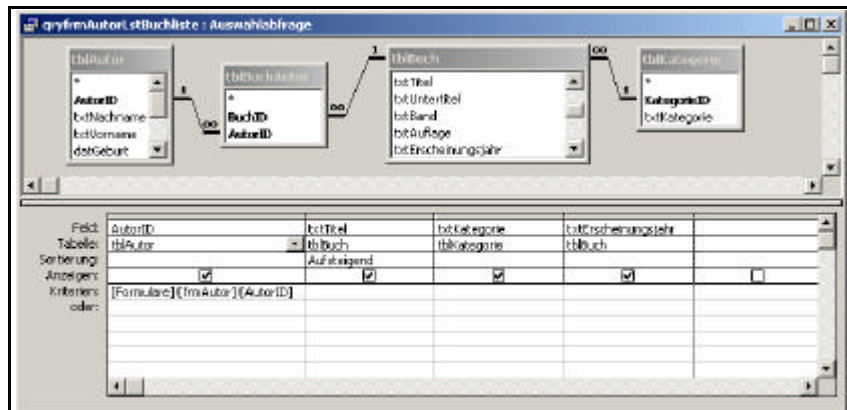


Abbildung 3: Formular-Steuerelement als Kriterium in einer Auswahlabfrage

Beim SQL Server 2000 ist erstmals die Verwendung der ORDER BY-Klausel in einem View möglich. Eine gewünschte Sortierung muss also nicht mehr mit Hilfe einer gespeicherten Prozedur realisiert werden.

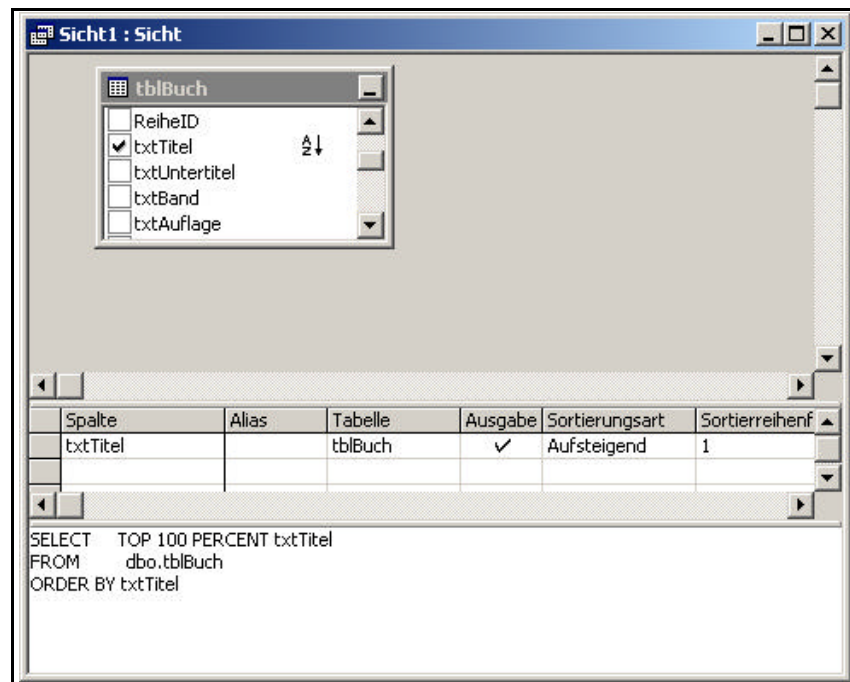


Abbildung 4: Verwendung der ORDER BY-Klausel in einem View

Gespeicherte Prozeduren bzw. Funktionen

Gespeicherte Prozeduren (englisch *Stored Procedures*) bzw. Funktionen sind vorkompilierter Code, der serverseitig ausgeführt wird.

Eine gespeicherte Prozedur bzw. Funktion kann neben reinen SQL-Befehlen auch aus Kommandos zur Programmsteuerung (Schleifen, Verzweigungen, Fehlerbehandlungsroutinen usw.) bestehen.

Spezielle SQL Server-Objekte



Abbildung 5: Transact-SQL-Funktion

Beim SQL Server findet hierfür die schon erwähnte Programmiersprache Transact-SQL Verwendung. Die Verwendung gespeicherter Prozeduren bzw. Funktionen bringt folgende Vorteile mit sich:

- Weil SQL-Befehle und Programmlogik direkt auf dem Server abgearbeitet werden, werden die Ausführungsgeschwindigkeit erhöht und die Netzwerkbelastung entsprechend reduziert.
- Transact-SQL erlaubt durch seinen Sprachumfang auch komplexe Programmabläufe.
- Gespeicherte Prozeduren bzw. Funktionen können durch die Vergabe von Ausführungsberechtigungen natürlich auch voll in das Sicherheitskonzept des SQL Servers integriert werden.

Die von Access her bekannten Aktionsabfragen (Anfüge-, Aktualisierungs-, Tabellenerstellungs- und Löschabfrage) können Sie bei SQL Server nur mit Hilfe einer gespeicherten Prozedur oder Funktion realisieren.

Trigger

Trigger sind eine Sonderform von gespeicherten Prozeduren mit all deren Vorzügen in der Praxis.

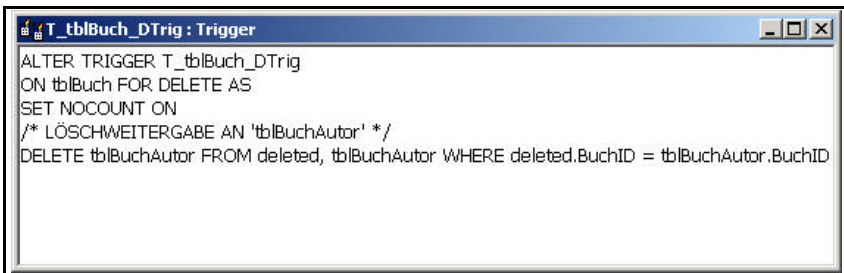
Trigger sind immer direkt an Tabellen gebunden und werden dann ausgeführt, wenn das Ereignis eintritt, für das sie programmiert worden sind.

Trigger können für Einfüge-, Lösch- und Aktualisierungsaktionen verwendet werden. Man spricht deshalb auch von *Insert*-, *Delete*- und *Update*-Trigger.

Ein typischer Einsatzbereich für Trigger ist die von Access her bekannte Lösch- bzw. Aktualisierungsweitergabe an eine Detailtabelle, die dort

bekanntlich im Beziehungsfenster mit Hilfe von Dialogfenstern festgelegt werden kann.

Die Umsetzung einer Löschweitergabe durch einen Trigger schaut beispielsweise folgendermaßen aus:



```
ALTER TRIGGER T_tblBuch_DTrig
ON tblBuch FOR DELETE AS
SET NOCOUNT ON
/* LÖSCHWEITERGABE AN 'tblBuchAutor' */
DELETE tblBuchAutor FROM deleted, tblBuchAutor WHERE deleted.BuchID = tblBuchAutor.BuchID
```

Abbildung 6: SQL-Code eines Trigger

Der Upsizing-Assistent: Vorbereitung

Der einfachste Weg bei der Umstellung einer Access-Datenbank auf die MSDE oder die Vollversion des SQL Servers ist der Einsatz des Upsizing-Assistenten von Access XP.

Der Upsizing-Assistent wandelt eine Access-Datenbank in ein Access-Projekt um, das auf einer SQL Server-Datenbank beruht.

Dabei werden Tabellenstrukturen, Beziehungen, Daten, Indizes, und Gültigkeitsregeln der Ausgangsdatenbank beibehalten. Der Assistent verwendet dabei, wo immer das möglich ist, die Vorteile der SQL Server-Funktionalität.

Dieser Assistent kann – vereinfacht ausgedrückt – eine ganze Menge, aber beileibe nicht alles.

Daraus ergibt sich logischerweise zweierlei:

Das Assistent erspart in jedem Fall eine ganze Menge Handarbeit.

Der Upsizing-Assistent: Vorbereitung

Sie müssen andererseits damit rechnen, an einigen Punkten nachbessern zu müssen.

Es hat sich also so ergeben, dass unsere Datenbankanwendung *Bücher* für den Einsatz in einem größeren Netzwerk fit gemacht werden muss.

Vorüberlegungen

Wie fast alles im Leben hat auch das Upsizing einer Access-Datenbankanwendung ihre Vor- und Nachteile. Vor dem Upsizing sollten Sie sich also diese Vor- und Nachteile einmal kurz vor Augen führen. Die Vorteile einer SQL Server-Datenbank sind an anderer Stelle schon erwähnt worden und werden hier nur noch einmal knapp zusammengefasst:

Vorteile

- Skalierbarkeit der Anwendung in Bezug auf zukünftige Erweiterungen
- Optimale Benutzersicherheit durch das Sicherheitssystem des SQL Servers
- Optimale Datensicherheit durch das Transaktionsprotokoll
- in einer Netzwerkumgebung bessere Leistung durch serverseitige Aktionen und Verarbeitungen

Nachteile

Wesentlicher Nachteil des Upsizing ist der damit verbundene Zeitaufwand. Und da Zeit in den allermeisten Fällen auch gleichzeitig Geld bedeutet ...

Der Zeitaufwand beim Upsizing ergibt sich aus folgenden Tätigkeiten:

- die eigentliche Umwandlung
- die notwendigen Nacharbeiten
- eine ausgiebige Testphase

Der Umfang des Zeitaufwands hängt natürlich vom Einzelfall ab. Ein wesentlicher Faktor ist beispielsweise, ob und in welchem Maße bei der Ausgangs-Datenbank mit DAO programmiert worden ist. Dieser VBA -Code muss ja bekanntlich komplett auf ADO umgestellt werden.

In einem solchen Fall sollten Sie die Code-Umstellung von DAO auf ADO vor dem Upsizen vornehmen und ausgiebig testen.

Die verschiedenen Upsizing-Methoden

Zu den Vorüberlegungen gehört auch, sich für eine der drei Möglichkeiten zu entscheiden, die der Upsizing-Assistent zur Verfügung stellt.

Möglichkeit 1: Access-Datenbank bleibt unverändert

Kopien der Tabellen aus der Access-Datenbank werden in SQL Server-Tabellen umgewandelt und dort gespeichert. Ein Zugriff auf die Tabellen mit Access ist natürlich nicht möglich. Dafür hat sich an Ihrer Access-Datenbank überhaupt nichts geändert.

Möglichkeit 2: Access-Datenbank mit eingebundenen Tabellen

Die Tabellen der Access-Datenbank werden in SQL Server-Tabellen umgewandelt und anschließend mit der Access-Datenbank verknüpft.

Das ist die klassische FrontEnd-BackEnd-Lösung in Access mit der Besonderheit, dass sich die eingebundenen Tabellen auf einem SQL Server befinden. Der Rest der Access-Datenbank bleibt unverändert.

Möglichkeit 3: Access-Projekt

Die Access-Datenbank wird vollständig in ein Access-Projekt konvertiert. Dazu zählen – neben den Tabellen, die sich anschließend auf dem SQL Server befinden – auch alle anderen Objekte der Access-Datenbank.

Die Access-Datenbank wird bei dieser Möglichkeit komplett verändert.

Der Upsizing-Assistent: Durchführung

Es liegt auf der Hand, dass wir nur bei der Verwendung der letzten Möglichkeit auch eine echte Client/Server-Anwendung als Ergebnis erhalten. Deshalb entscheiden wir uns auch für diese Möglichkeit.

Vorbereitungen

Vor den eigentlichen Konvertierung der Access-Datenbank mit dem Upsizing-Assistenten sind – auch aus Sicherheitsgründen – noch einige Schritte notwendig:

- Erstellen Sie eine Sicherungskopie der Access-Datenbank, die konvertiert werden soll.

Obwohl die ursprüngliche Access-Datenbank beim Upsizing erhalten bleibt, gehen Sie so in jedem Fall auf Nummer Sicher.

- Legen Sie auf jeden Fall einen Standarddrucker fest, damit Sie nach dem Upsizing den vom Assistenten erstellten Upsizing-Bericht auch ausdrucken können.
- Melden Sie sich bei einer abgesicherten Access-Datenbank als Administrator oder als anderer Benutzer mit allen Rechten an.
- Auch für den SQL Server benötigen Sie entsprechende Zugriffsrechte, um dort unter anderem Datenbanken und Tabellen anlegen zu können.
- Stellen Sie sicher, dass der SQL Server bzw. die MSDE auch gestartet ist.

Nach Klärung dieser Punkte kann das Upsizing beginnen.

Der Upsizing-Assistent: Durchführung

Öffnen Sie die Access-Datenbank, die konvertiert werden soll, und starten Sie den Upsizing-Assistenten. Dies geschieht mit dem Menübefehl *Extras / Datenbank-Dienstprogramme / Upsizing-Assistent*.

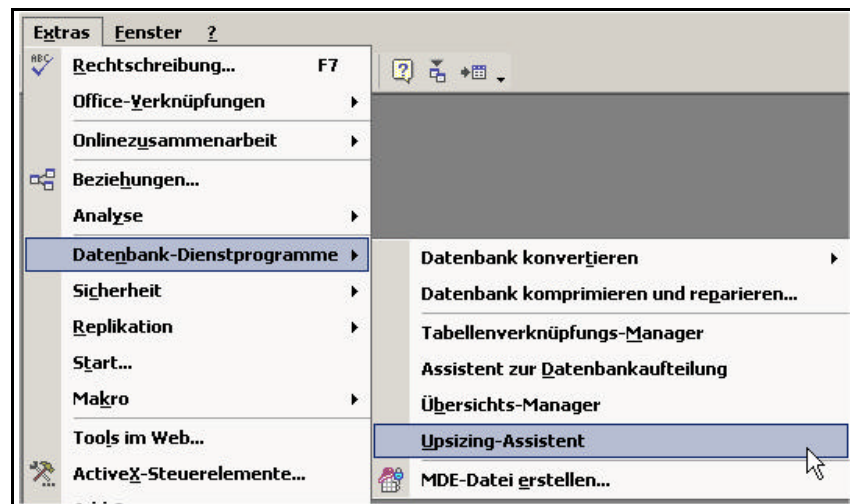


Abbildung 7: Start des Upsizing-Assistenten

Neue SQL Server-Datenbank erstellen

Im ersten Dialogfenster des Assistenten wählen Sie aus, ob Sie eine vorhandene SQL Server-Datenbank verwenden oder eine neue SQL Server-Datenbank erstellen wollen.

Der Upsizing-Assistent: Durchführung



Abbildung 8: Das Dialogfenster *Upsizing-Assistent*

Im Normalfall werden Sie die zweite Option *Neue Datenbank erstellen* verwenden, die auch voreingestellt ist.

Verbindungs- und Anmeldedaten

Im nächsten Schritt des Assistenten legen Sie den Namen des SQL Servers, den *Benutzernamen*, ein *Kennwort* und den Namen der neuen SQL Server-Datenbank fest.

Als Standard-Datenbanknamen schlägt Ihnen der Assistent den Namen der Access-Datenbank mit der Erweiterung *SQL* vor.

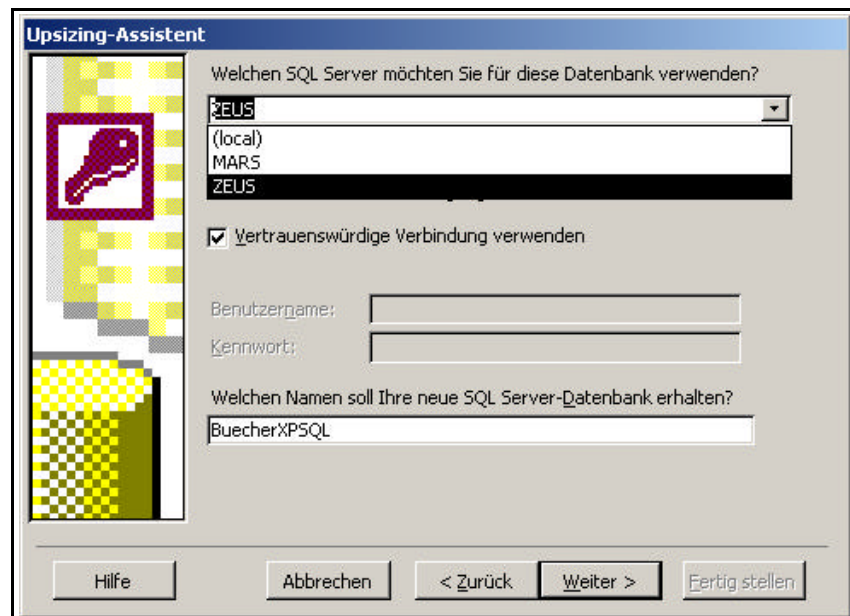


Abbildung 9: Verbindungs- und Anmeldedaten festlegen

Falls der SQL Server auf einem Windows NT/2000/XP-System läuft, können Sie das Kontrollkästchen *Vertrauenswürdige Verbindung verwenden* aktivieren. Das entspricht der Ihnen schon bekannten Windows-Authentifizierung.

Anderenfalls ist für den SQL Server immer der Benutzer *sa* (*system administrator*) eingerichtet, der mit allen Rechten ausgestattet ist. Schon deshalb ist es auf jeden Fall empfehlenswert, ein Anmeldekennwort für den Benutzer *sa* zu verwenden.

Auswahl der Tabellen

Im nächsten Schritt des Assistenten wählen Sie aus dem Bereich *Vorhandene Tabellen* die Tabellen aus, die auf den SQL Server portiert werden sollen.

Der Upsizing-Assistent: Durchführung

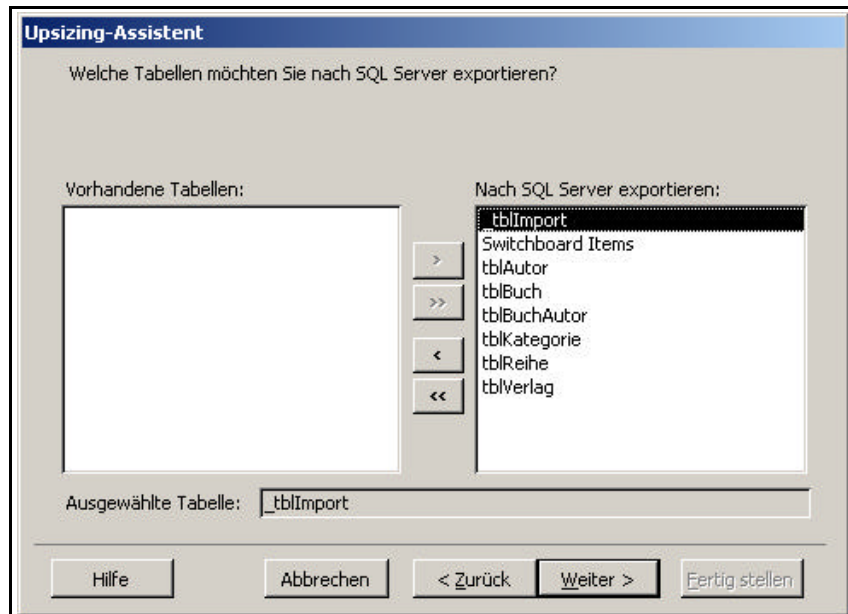


Abbildung 10: Auswahl der Tabellen

In einem Access-Projekt gibt es keine Möglichkeit, mit lokalen Access-Tabellen zu arbeiten. Falls Sie also hier eine Tabelle nicht auswählen, steht diese Tabelle anschließend in Ihrem Access-Projekt nicht mehr zur Verfügung.

Daraus ergibt sich, dass auch die Tabelle *Switchboard Items*, auf der das Übersichtsformular der Datenbank Anwendung basiert, auf den SQL Server portiert werden muss.

Tabellenattribute festlegen

Im nächsten Schritt des Assistenten wird Ihnen ein Dialogfenster mit verschiedenen aktivierten Kontrollkästchen angeboten.

Da es sich hierbei um die Umsetzung wichtiger Tabellenattribute handelt, ist es sinnvoll, die einzelnen Punkte Schritt für Schritt zu betrachten.

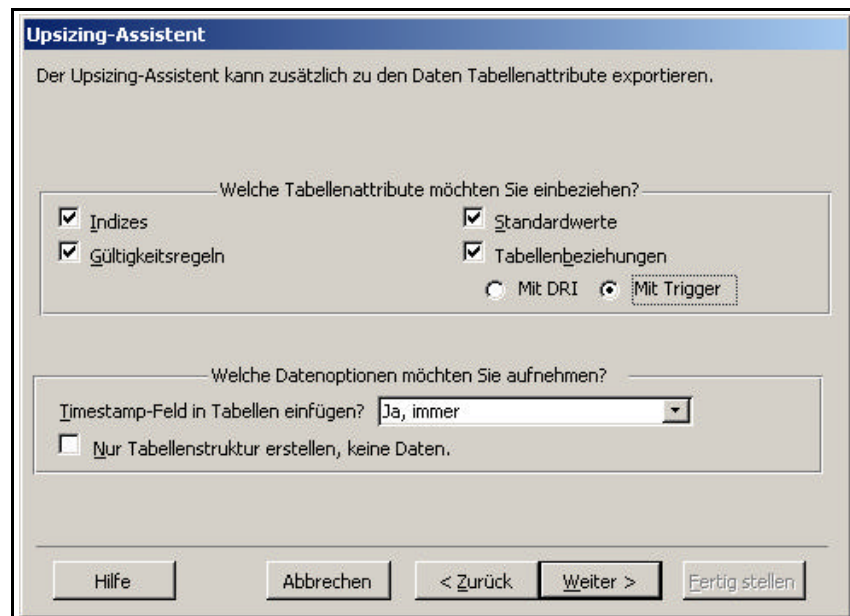


Abbildung 11: Festlegung von Tabellenattributen

Indizes

Bei aktiviertem Kontrollkästchen versucht der Assistent, auch die Indizes für den SQL Server zu erstellen. Dabei werden die vorhandenen Access-Indizes quasi abgebildet.

Das bedeutet, jeder Primärschlüssel in Access wird auch als Primärschlüssel in der SQL Server-Tabelle festgelegt. Entsprechendes gilt auch für die Fremdschlüssel.

Bei den SQL Server-Indizes kann – ab dem SQL Server 2000 – genau wie bei Access auch eine auf- bzw. absteigende Sortierung festgelegt werden.

Der Upsizing-Assistent: Durchführung

Gültigkeitsregeln

Falls das Kontrollkästchen *Gültigkeitsregeln* aktiviert ist, setzt der Assistent vorhandene Gültigkeitsregeln in Trigger auf dem SQL Server um – zumindest in der Theorie.

In der Praxis ist es durchaus üblich, dass dies nur teilweise oder gar nicht funktioniert. Dann muss die Umsetzung anschließend händisch erfolgen – oder Sie verzichten auf die Gültigkeitsregeln. Letzteres ist allerdings nicht so ganz im Sinne des Erfinders.

Standardwerte

Der Assistent versucht, in Access definierte Standardwerte in die SQL Server-Datenbank zu übernehmen.

Tabellenbeziehungen

Tabellenbeziehungen können entweder über *DRI (Declared Reference Integrity)* oder mit Hilfe von Triggern umgestellt werden.

Die Auswahl *Mit DRI* funktioniert wie die referentielle Integrität in Access.

Diese Funktionalität kann nur von Access XP, nicht jedoch von Access 2000 genutzt werden.

Der SQL Server kann Aktualisierungs- bzw. Löschweitergaben auch über Trigger realisieren. Dabei erhalten alle Detailtabellen *Insert*-Trigger, während die Mastertabellen *Delete*-Trigger erhalten. Wählen Sie also hier die Option *Mit Trigger* aus.

Timestamp-Felder

Der SQL Server speichert in Timestamp-Feldern die letzte Aktualisierung eines Datensatzes auf den Sekundenbruchteil genau.

Diese Auswahl ist sinnvoll. Damit kann beispielsweise in einer Mehrbenutzerumgebung genau festgestellt werden, wann zuletzt auf den betreffenden Datensatz zugegriffen wurde. Diese Auswahl ist also in jedem Fall empfehlenswert.

Tabellenstruktur ohne Daten

Die Auswahl dieser Option bietet sich beispielsweise für den Fall an, wenn – bei einem größerem Datenbestand – nur festgestellt werden soll, inwieweit das Upsizing problemlos funktioniert oder nicht.

Upsizing-Methode auswählen

Nach diesen Einstellungen wählen Sie im nächsten Dialogfenster die *Upsizing-Methode* aus.

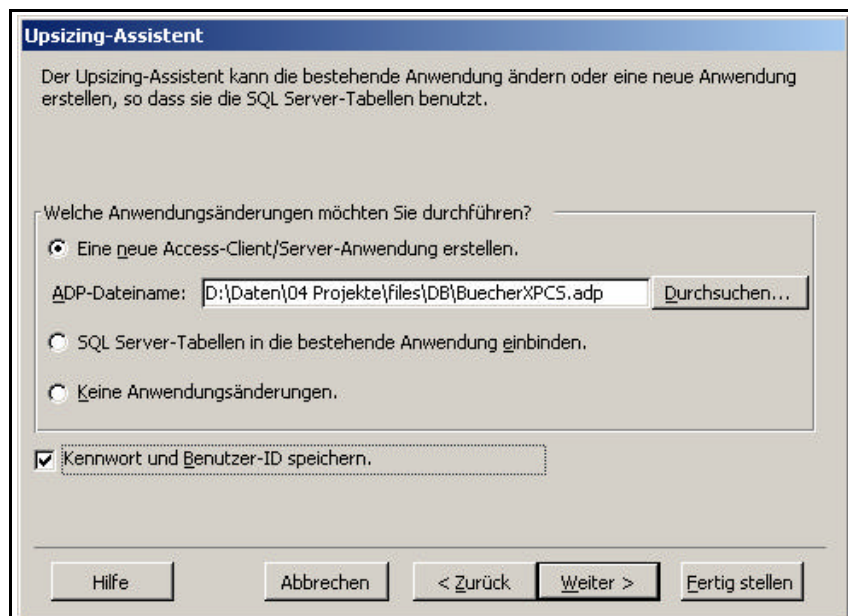


Abbildung 12: Auswahl der *Upsizing-Methode*

Die Optionen im Bereich *Welche Anwendungsänderungen möchten Sie durchführen?* stehen für folgende Unterschiede:

Der Upsizing-Assistent: Durchführung

Eine neue Access-Client/Server-Anwendung erstellen

Dies ist die Standardeinstellung und auch unsere Wahl.

Der Assistent legt in diesem Fall ein neues Access-Projekt an, dessen Namen Sie frei festlegen können. Die Standardbenennung ist der Datenbankname mit den angehängten Buchstaben *CS*. Die Dateiendung ist *adp*, wodurch ein Access-Projekt gekennzeichnet wird.

SQL Server-Tabellen in bestehende Anwendung einbinden

Bei dieser Option werden zunächst alle Access-Tabellen auf den SQL Server kopiert und per ODBC mit der Access-Datenbank verknüpft.

Die Tabellen in der Access-Datenbank sind durch ein angehängtes *_lokal* gekennzeichnet und können im Prinzip gelöscht werden.

Keine Anwendungsänderungen

Die Access-Tabellen werden in SQL Server-Tabellen kopiert. Die Access-Anwendung als solche bleibt allerdings unverändert erhalten.

Falls Sie das Kontrollkästchen *Kennwort und Benutzer-ID speichern* aktivieren, brauchen sich die Benutzer nicht beim SQL Server anzumelden.

Das bedeutet allerdings, dass jeder Benutzer, der Zugriff auf das Access-Projekt hat, gleichzeitig auch Zugriff auf den SQL Server hat.

Deswegen wird bei dieser Auswahl folgende Warnmeldung angezeigt.

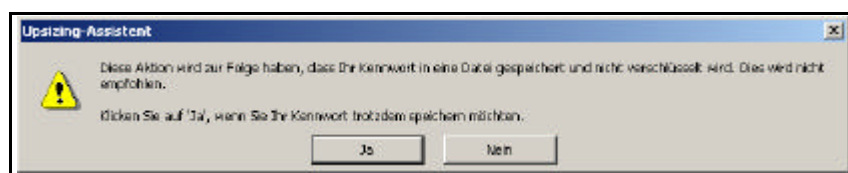


Abbildung 13: Warnmeldung

Durchführung

Damit hat der Assistent alle die Angaben, die zum Upsizing der Access-Datenbank notwendig sind.

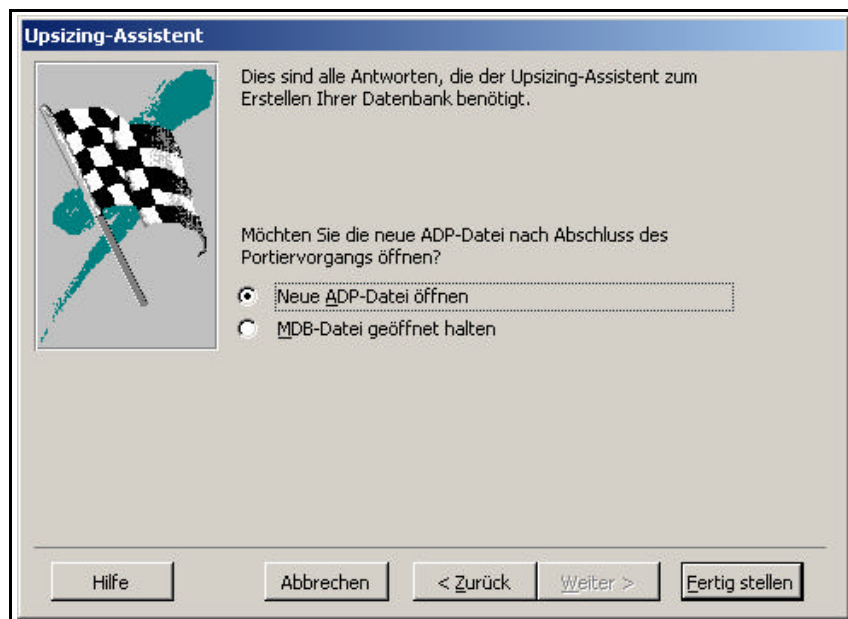


Abbildung 14: Das Dialogfenster *Upsizing-Assistent*

Der Vorgang des Upsizing kann schon eine gewisse Zeit in Anspruch nehmen. Über den Fortschritt werden Sie durch eine Verlaufsanzeige informiert.

Wenn der Vorgang abgeschlossen ist, wird ein Protokoll in Form eines Berichts erstellt: das *Upsizing-Protokoll*.

Das *Upsizing-Protokoll* wird im gleichen Ordner wie die Access-Projekt-Datei als Berichts-Snapshot mit der Dateiendung *.snp* gespeichert. Der Dateiname ist der gleiche wie der Dateiname der Access-Projekt-Datei.

Der Upsizing-Assistent: Durchführung

Das Upsizing-Protokoll

Das *Upsizing-Protokoll* enthält eine genaue Beschreibung aller vom Assistenten erzeugten Objekte, durchgeführten Aktionen und vor allem eine Aufstellung der beim Upsizing aufgetretenen Fehler.

Daran können Sie sich bequem bei den dann notwendigen Nacharbeiten orientieren.

Bei größeren Upsizing-Problemen sollten Sie das Protokoll auf jeden Fall ausdrucken, damit Sie alles schwarz auf weiß vor sich liegen haben.

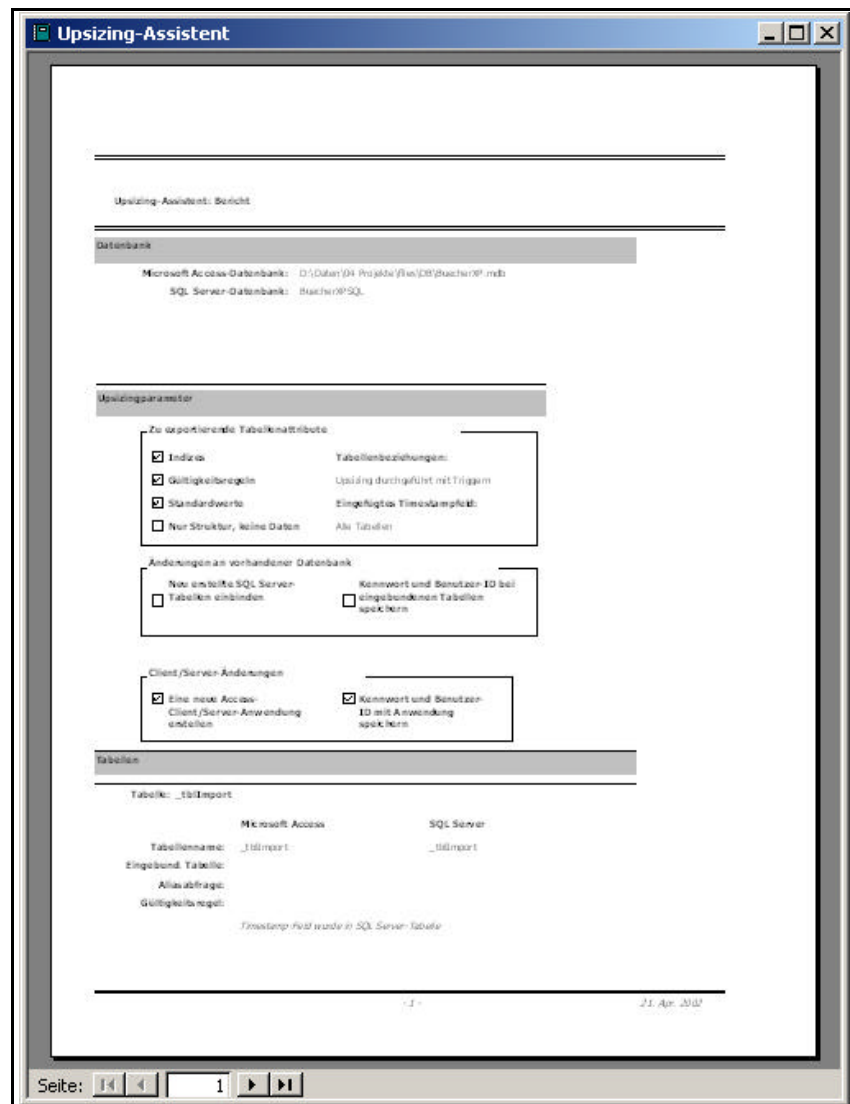


Abbildung 15: Das Upsizing-Protokoll

Im Einzelnen finden Sie im Upsizing-Protokoll Informationen zu folgenden Themen:

Access-Projekt: Bestandsaufnahme

- detaillierte Angaben zur SQL Server-Datenbank (Größe, Transaktionsprotokoll)
- Auflistung aller Parameter, die Sie in den Dialogfenstern des Upsizing-Assistenten ausgewählt haben
- Informationen zu den neu erstellten SQL Server-Tabellen (Datentypen, Indizes, Gültigkeitsregeln usw.)
- alle aufgetretenen Fehler

Access-Projekt: Bestandsaufnahme

Nachdem der Upsizing-Assistent seine Arbeit abgeschlossen hat, haben Sie das fertige Access-Projekt im Datenbankfenster vor sich.

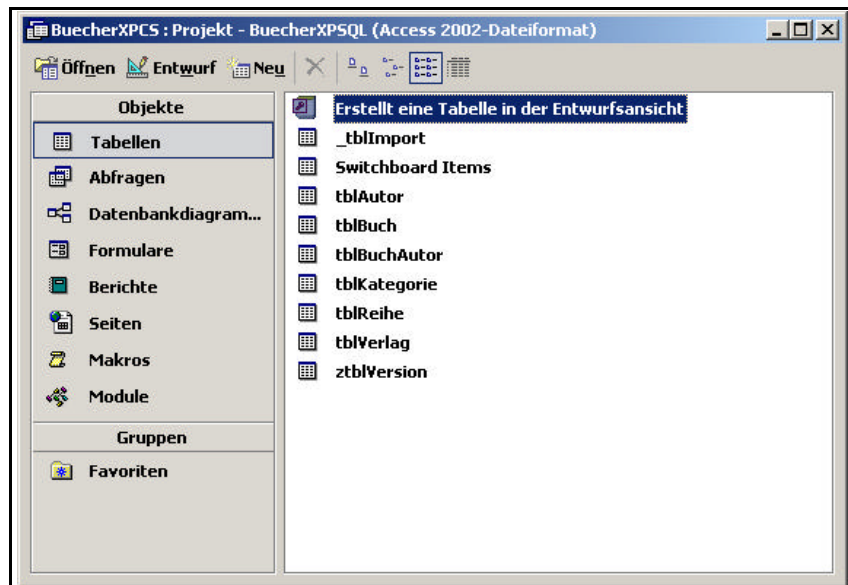


Abbildung 16: Access-Projekt im Datenbankfenster

Überprüfung des Access-Projekts

Als Erstes sollten Sie alle Formulare, Berichte usw. öffnen und überprüfen, was funktioniert und was nicht. Bei auftretenden Fehlern schauen Sie sich die zugrunde liegenden Abfragen an.

Es kann auch sinnvoll sein, die reine Anzahl der Abfragen in der Access-Anwendung und im Access-Projekt miteinander zu vergleichen. Manche ehemalige Abfragen aus der Access-Datenbank fehlen in dem Access-Projekt einfach.

Bei all dem ist das Upsizing-Protokoll äußerst hilfreich. Sie finden darin nicht nur einen Überblick über die notwendigen Nacharbeiten, sondern auch Listings der Fehlversuche, mit denen der Upsizing-Assistent versucht hat, bestimmte Funktionalitäten umzusetzen.

Ein Ausdruck dieses Protokolls, auch wenn es dreißig oder mehr Seiten sein sollten, ist auf jeden Fall zu empfehlen.

Fehlerliste

Halten Sie alle auftretenden Fehler in einer Fehlerliste fest oder markieren Sie diese im ausgedruckten Upsizing-Protokoll.

Beim Upsizing der Datenbankanwendung *Bücher* sind bei mir, das heißt in meiner Systemumgebung, die folgenden Fehler aufgetreten:

Upsizing-Fehler 1: Die Anfügeabfrage *ins_tblImport*

Die Anfügeabfrage *ins_tblImport* konnte vom Upsizing-Assistenten nicht umgesetzt werden und fehlt einfach im Access-Projekt.

Das Upsizing-Protokoll vermerkt zu diesem Punkt, dass das Upsizing nicht durchgeführt werden konnte. Außerdem ist hier der offensichtlich fehlgeschlagene Versuch dokumentiert, dieses Problem mit Hilfe einer gespeicherten Prozedur zu lösen. Hier ist also auf jeden Fall Handarbeit angesagt.

Access-Projekt: Bestandsaufnahme

Upsizing-Fehler 2: Die Abfrage qrySubfrmBuchAutorCboAutorID

Diese Abfrage ist ebenfalls nicht im Access-Projekt vorhanden. Beim Öffnen der Formulare *frmBuch* und *SubFrmBuchAutor* wird eine entsprechende Fehlermeldung angezeigt.

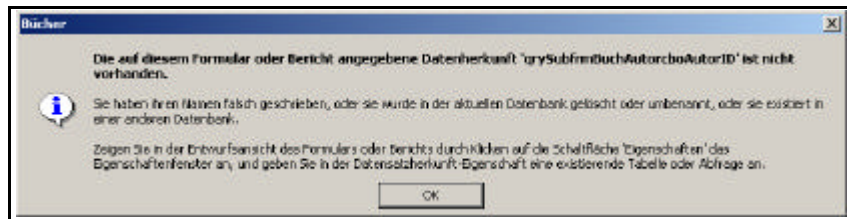


Abbildung 17: Fehlermeldung beim Öffnen des Formulars

Das Kombinationsfeld im Formular *SubFrmBuchAutor*, dessen Datenherkunft auf diese Abfrage eingestellt ist, zeigt auch folgerichtig keinen einzigen Datensatz an.

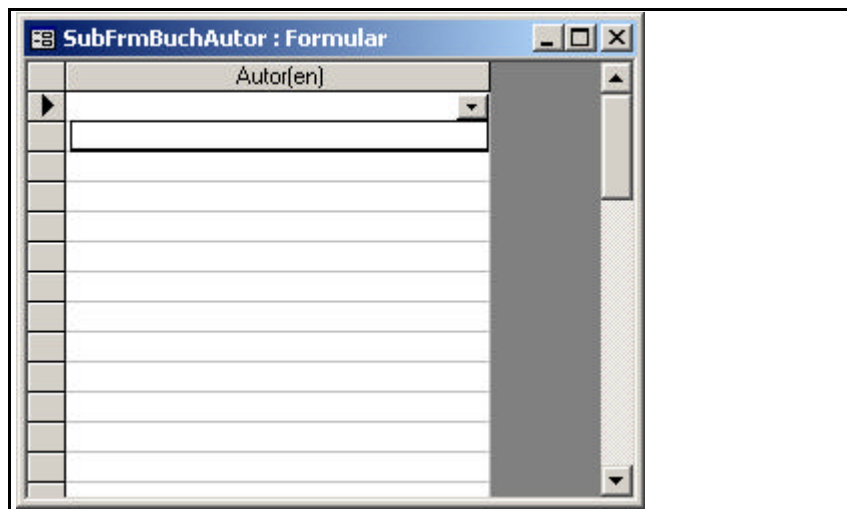


Abbildung 18: Leeres Formular

Das Upsizing-Protokoll vermerkt zu diesem Punkt ebenfalls, dass das Upsizing nicht durchgeführt werden konnte. Der Versuch, hier eine Funktion zu erstellen, hatte keinen Erfolg. Hier ist also ebenfalls Handarbeit angesagt.

Upsizing-Fehler 3: Die Abfrage qryfrmAutorLstBuchliste

Zur Erinnerung: Diese Abfrage ist die Datenherkunft für die Bücherliste im Autorenformular. Als Parameter wird dieser Abfrage die *AutorID* des aktuellen Datensatzes im Autorenformular übergeben.

Beim Aufruf des Autorenformulars erscheint dann auch eine Fehlermeldung:

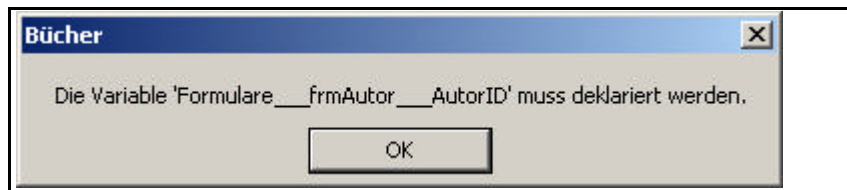


Abbildung 19: Fehlermeldung beim Aufruf des Autorenformulars

Nach einem Klick auf *OK* wird ein Dialogfenster angezeigt, dass zur Eingabe eines Parameters auffordert.

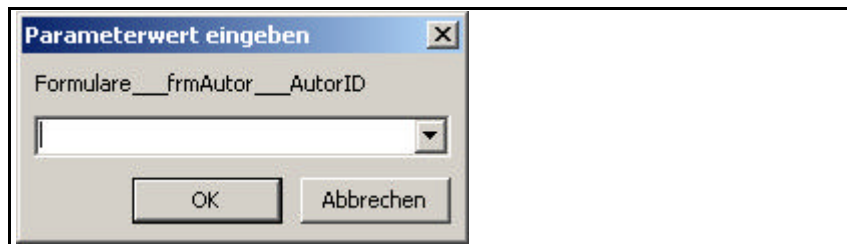


Abbildung 20: Das Dialogfenster *Parameterwert eingeben*

Nach der Eingabe einer vorhandenen Autoren-ID wird dann auch brav der betreffende Datensatz im Formular angezeigt.

Das Problem liegt auf der Hand: Die zur Synchronisation notwendige Autoren-ID aus dem Formular wird nicht an das Listefeld übergeben.

Auch dieser Fehler wird im Upsizing-Protokoll aufgeführt. Der erfolglose Upsizing-Versuch dieser Abfrage schaut in der Entwurfsansicht wie folgt aus:

Access-Projekt: Bestandsaufnahme

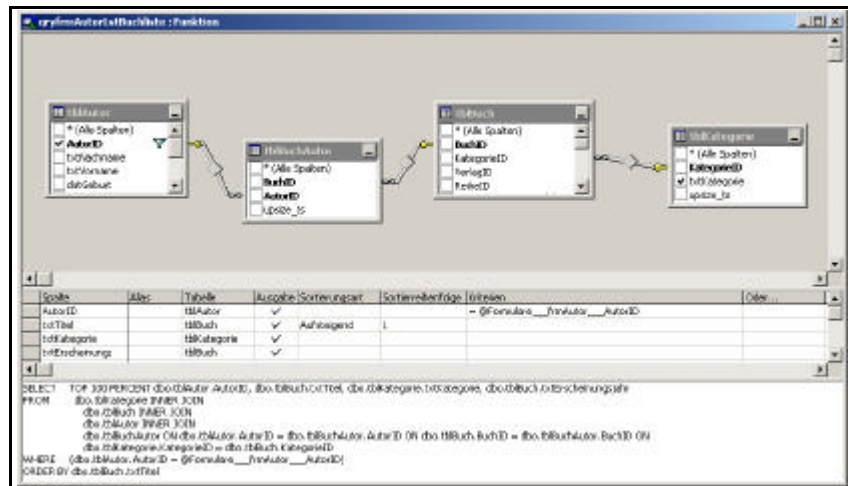


Abbildung 21: Entwurfsansicht der Funktion

Upsizing-Fehler 4: Das Übersichtsformular

Beim Übersichtsformular funktioniert der Menüpunkt *Übersichtseinträge ändern* nicht mehr. Das ist eigentlich im Grunde genommen kein Fehler, sondern eigentlich mehr eine Macke. Diese hängt damit zusammen, dass derartige Änderungen von einem eingebauten Access-Assistenten vorgenommen werden.

Dieser Assistent ist dazu ausgelegt, auf eine Tabelle mit dem Namen *Switchboard Items* zuzugreifen und Änderungen an dieser Tabelle vorzunehmen. Allerdings funktioniert das nicht, wenn es sich bei dieser Tabelle um eine SQL Server-Tabelle handelt.

Upsizing-Fehler 5: Das Kombinationsfeld im Autoren-Unterformular

Der letzte auftretende Fehler ist auf den ersten Blick mysteriös und nicht so ohne weiteres erklärbar:

Bei einem Doppelklick auf das Kombinationsfeld *Autor(en)* im Unterformular des Bücherformulars wird nach dem Upsizing folgende

Fehlermeldung angezeigt, deren Text auf den ersten Blick nicht sonderlich erhellend, sondern ausgesprochen kryptisch ist:



Abbildung 22: Fehlermeldung

Mysteriös ist dieser Fehler vor allem deshalb, weil er bei einem Doppelklick auf das Kombinationsfeld *Reihe*, bei dem eine identische Ereignisprozedur aufgerufen wird, nicht auftritt. Hier wird das Eingabeformular zur Anlage eines neuen Datensatzes genauso geöffnet, wie es auch vorgesehen ist. Auch die Eingabe und Speicherung eines neuen Datensatzes im Formular *frmReihen* funktionieren einwandfrei.

Das Autorenformular zur Anlage eines neuen Datensatzes wird dagegen nicht geöffnet. Das bedeutet, die Ereignisprozedur bricht durch den auftretenden Fehler bzw. die Fehlerbehandlungsroutine in der Ereignisprozedur bedingt, einfach ab:

```
On Error GoTo Err_AutorID_DblClick
Dim lngAutorID As Long

If IsNull(Me![AutorID]) Then
    Me![AutorID].Text = ""
Else
    lngAutorID = Me![AutorID]
    Me![AutorID] = Null
End If

DoCmd.OpenForm "frmAutor", , , , , acDialog
```

Durch dieses Verhalten kann allerdings schon auf die Stelle in der Ereignisprozedur geschlossen werden, an der der Fehler auftritt:

```
Me![AutorID] = Null
```

Access-Projekt: Behebung der Upsizing-Fehler

In Verbindung mit dem Text der Fehlermeldung *Sie wollten den Nullwert einer Variablen zuweisen ...* haben wir hier einen ersten Ansatzpunkt für das weitere Vorgehen.

Access-Projekt: Behebung der Upsizing-Fehler

Insgesamt sind beim Upsizing der Datenbankanwendung *Bücher* auf den SQL Server also fünf Fehler aufgetreten, die nun behoben werden müssen.

Die Behebung der aufgetretenen Fehler erfolgt genau in der Reihenfolge, wie sie im letzten Abschnitt aufgelistet wurden.

Behebung Upsizing-Fehler 1: Die Anfügeabfrage

Die Lösung zur Behebung dieses Fehlers besteht in der Erstellung einer gespeicherten Prozedur.

Eine der Neuerungen in Access XP ist, dass hier auch gespeicherte Prozeduren im Abfrage-Designer erstellt werden können. In Access 2000 musste zu diesem Zweck noch der reine SQL-Code in ein Editorfenster eingegeben werden.

Das ist eine gewaltige Arbeitsvereinfachung, weil – wie Sie gleich erfahren werden – das Erstellen einer gespeicherten Prozedur im Prinzip genauso erfolgen kann wie die Erstellung einer Anfügeabfrage in einer Access-Datenbank.

Die bisherige Methode zur Erstellung von gespeicherten Prozeduren ist in einem Access XP-Projekt natürlich weiterhin möglich.

Klicken Sie dazu im Register *Abfragen* des Datenbankfensters auf die Schaltfläche *Neu* und wählen Sie im Dialogfenster *Neue Abfrage* den Eintrag *Text-Gespeicherte Prozedur entwerfen* aus.



Abbildung 23: Auswahl einer Text-Gespeicherten Prozedur

Daraufhin wird ein Editorfenster angezeigt, das den Rumpf einer neuen gespeicherten Prozedur enthält. Dort können Sie dann die entsprechenden SQL-Anweisungen eingeben.

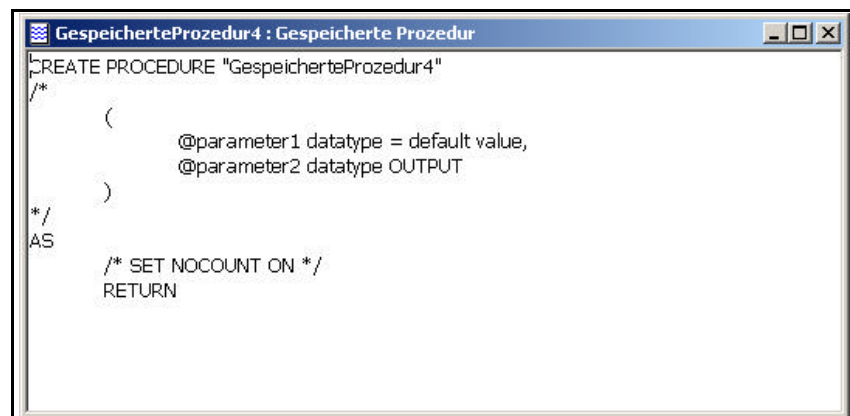


Abbildung 24: Rumpf einer gespeicherten Prozedur im Editorfenster

Wenn eine im Datenbankfenster abgelegte gespeicherte Prozedur erneut aufgerufen wird, beginnt der SQL-Code mit

```
ALTER PROCEDURE
```

Access-Projekt: Behebung der Upsizing-Fehler

Er kann dann im Editorfenster ganz normal bearbeitet und wieder gespeichert werden.

Mit den Zeichen `/*` und `*/` werden in Transact-SQL Kommentarbereiche gekennzeichnet.

Erstellung einer gespeicherten Prozedur im Abfragedesigner

Klicken Sie zuerst doppelt im Bereich *Abfragen* des Datenbankfensters auf den Eintrag *Erstellt eine gespeicherte Prozedur unter Verwendung des Designers*.

Wählen Sie dann im Dialogfenster *Tabelle hinzufügen* die Tabelle `_tblImport` aus.

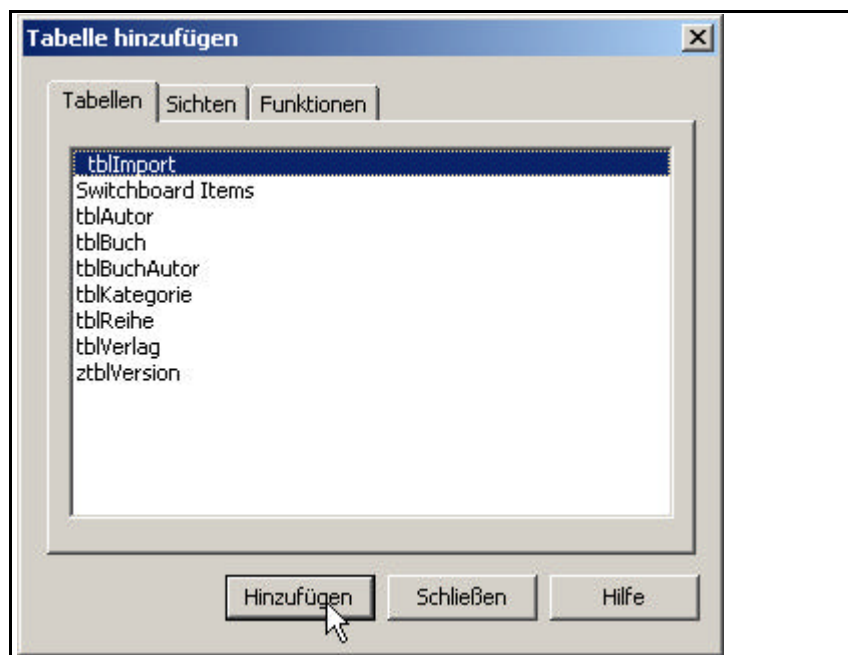


Abbildung 25: Das Dialogfenster *Tabelle hinzufügen*

Daraufhin wird der Arbeitsbereich für den Abfrageentwurf angezeigt.

Markieren Sie dort die drei für den Import benötigten Tabellenfelder *txtTitelImport*, *txtEinband* und *txtKategorie*.

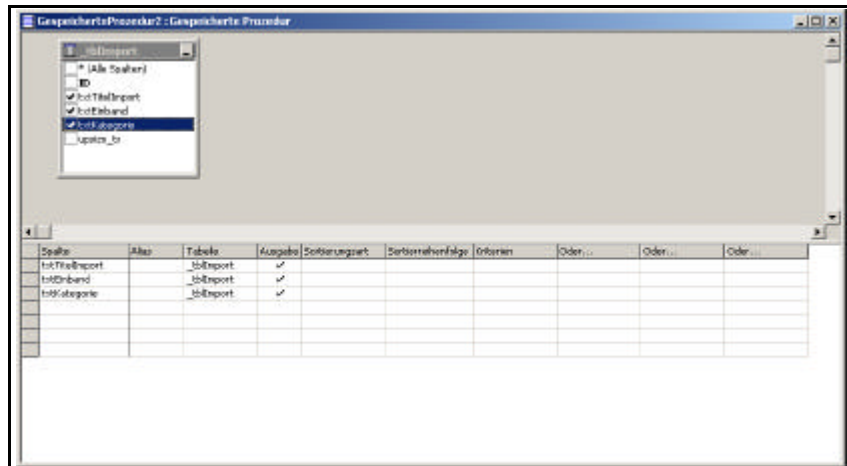


Abbildung 26: Entwurfsansicht einer gespeicherten Prozedur

Alternativ dazu können Sie die benötigten Felder auch aus der Liste in der Tabellenspalte *Spalte* auswählen.

Wählen Sie anschließend aus dem Menü *Abfrage* den Befehl *Anfügeabfrage* aus.

Access-Projekt: Behebung der Upsizing-Fehler

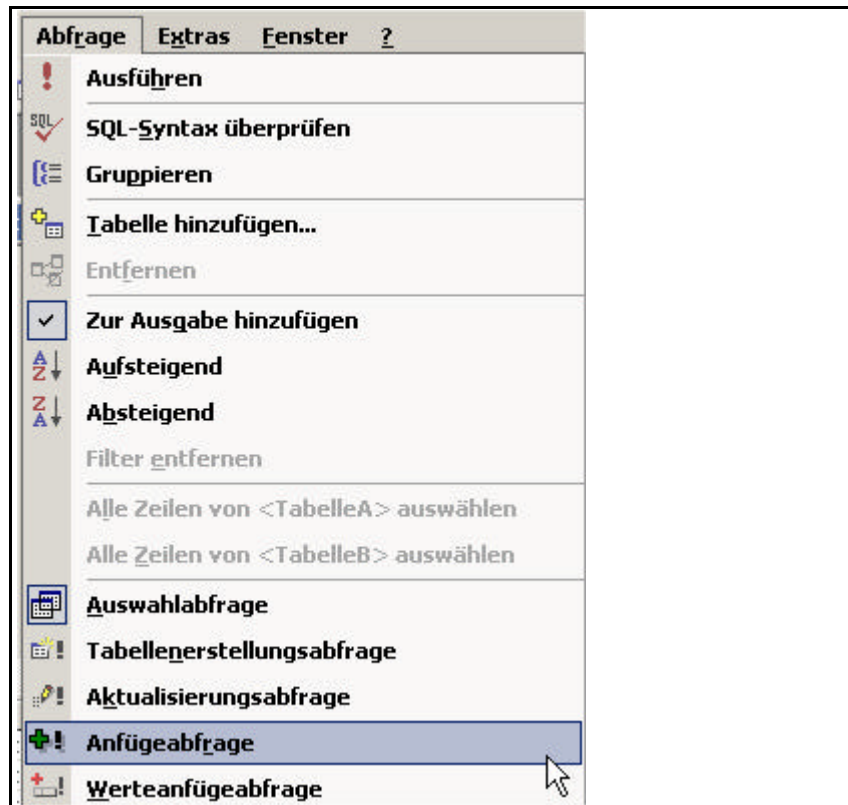


Abbildung 27: Der Menübefehl *Abfrage / Anfügeabfrage*

Daraufhin wird das Dialogfenster *Zieltabelle für Anfügeabfrage auswählen* angezeigt.

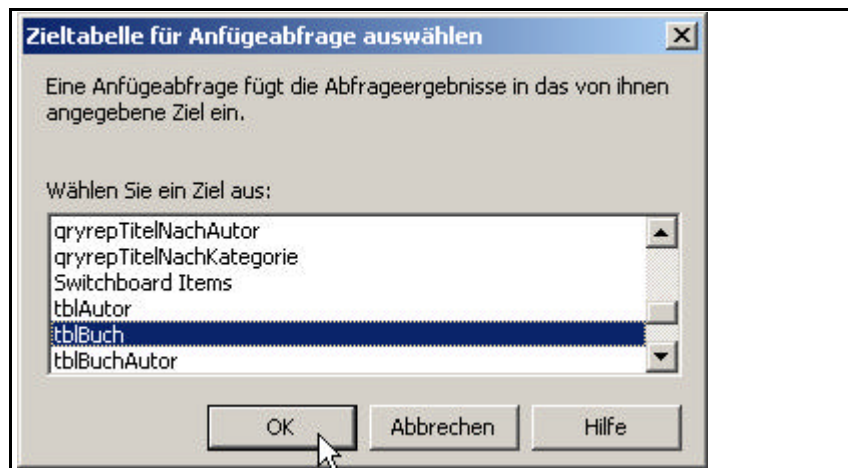


Abbildung 28: Das Dialogfenster *Zieltabelle für Anfügeabfrage auswählen*

Wählen Sie dort als Ziel die Tabelle *tblBuch* aus und klicken Sie auf die Schaltfläche *OK*.

Im Abfrageentwurf wird jetzt die Spalte *Anfügen* angezeigt. Wählen Sie dort die drei entsprechenden Felder aus der Tabelle *tblBuch* aus.

Access-Projekt: Behebung der Upsizing-Fehler

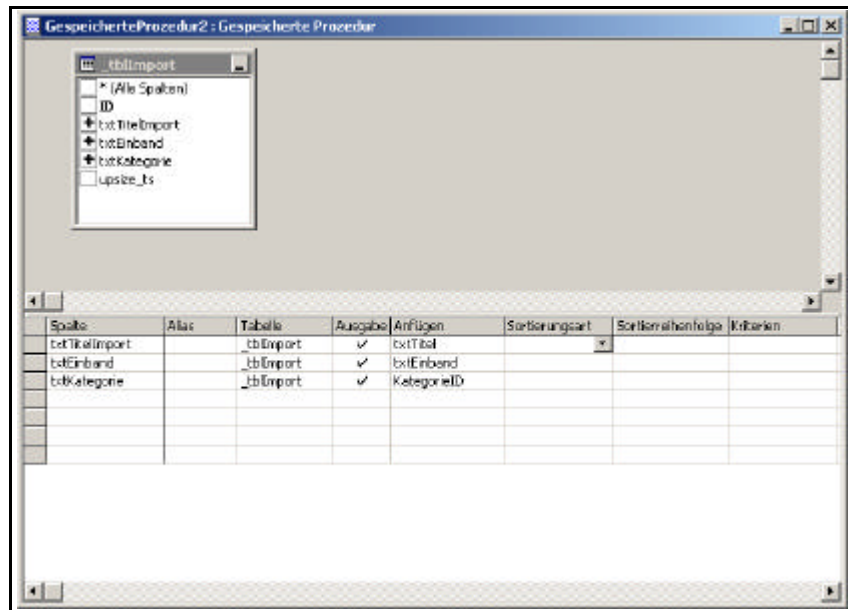


Abbildung 29: Auswahl der Anfügefelder

Damit ist die gespeicherte Prozedur im Prinzip fertig.

Als Nächstes sollten Sie sich zur Verdeutlichung den SQL-Code der gespeicherten Prozedur anschauen. Klicken Sie dazu in der Symbolleiste auf das Symbol SQL.

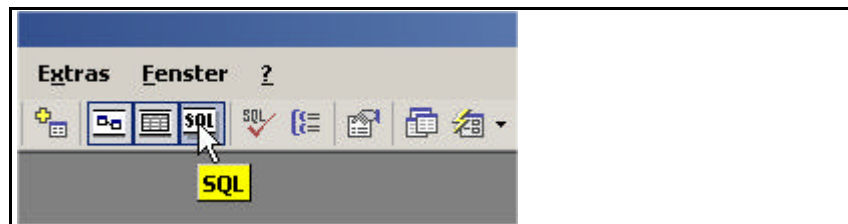


Abbildung 30: Auswahl des SQL-Bereichs

Im unteren Teil des Abfrageentwurfs wird daraufhin auch das SQL-Statement der gespeicherten Prozedur im *SQL-Bereich* des Abfragefensters sichtbar.

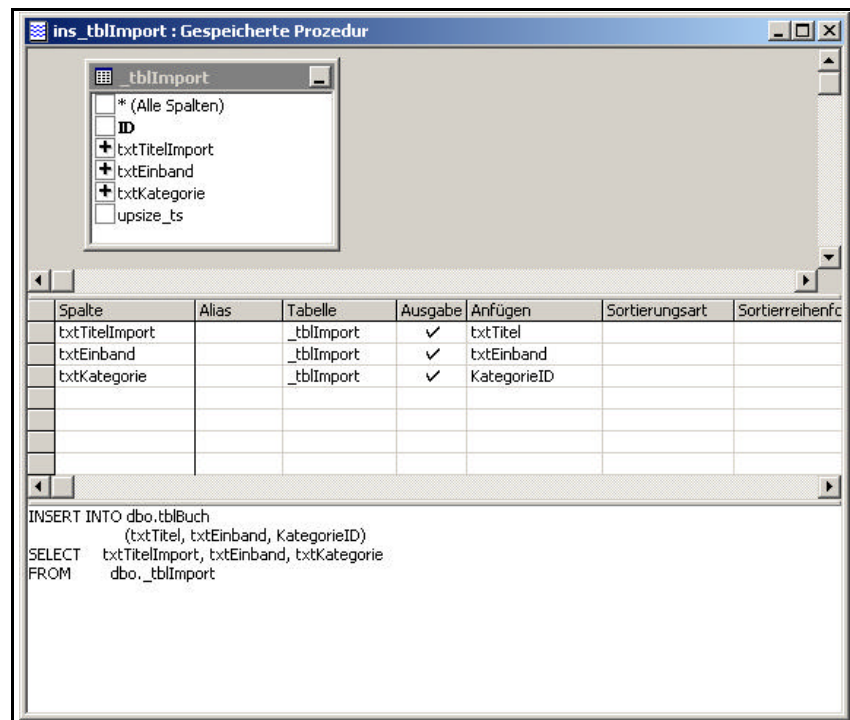


Abbildung 31: SQL-Statement der gespeicherten Prozedur

Das Präfix *dbo* vor dem Tabellennamen bezeichnet den Eigentümer der SQL Server-Tabelle.

Speichern Sie die gespeicherte Prozedur unter dem Namen *ins_tblImport*.

Behebung Upsizing-Fehler 2: Die Abfrage qrySubfrmBuchAutorCboAutorID

Zur Behebung des zweiten Fehlers benötigen wir eine Datenherkunft für das Kombinationsfeld *CboAutorID* im Formular *frmBuchAutor*.

Der einfachste Lösungsansatz für dieses Problem liegt in der Erstellung eines neuen Views. Denn der Grund für diesen Fehler besteht einfach darin, dass der Upsizing-Assistent Probleme bei der Umsetzung der Zeichenverkettung

Access-Projekt: Behebung der Upsizing-Fehler

Autor(en): [txtNachname] & ", " & [txtVorname]

hatte.

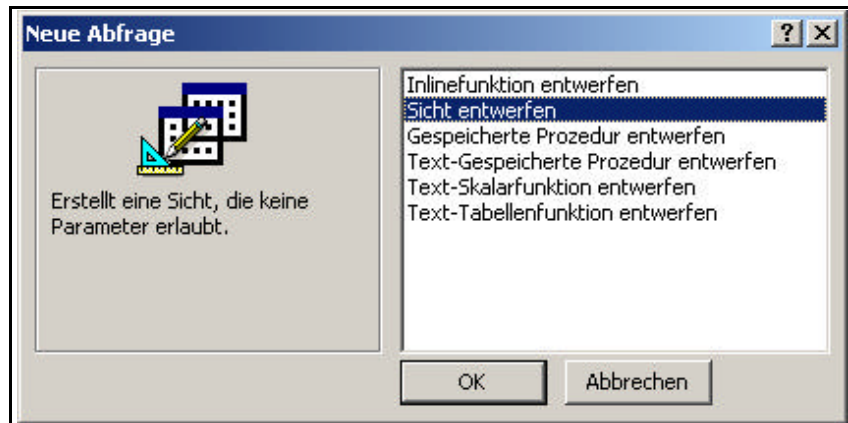


Abbildung 32: Das Dialogfenster *Neue Abfrage*

Fügen Sie dann die Tabelle *tblAutor* hinzu und wählen Sie zunächst das Feld *AutorID* aus.

Geben Sie dann in das Feld *Spalte* den Ausdruck

`txtNachname + ', ' + txtVorname`

und in das Feld *Alias* `Autor(en)` ein.

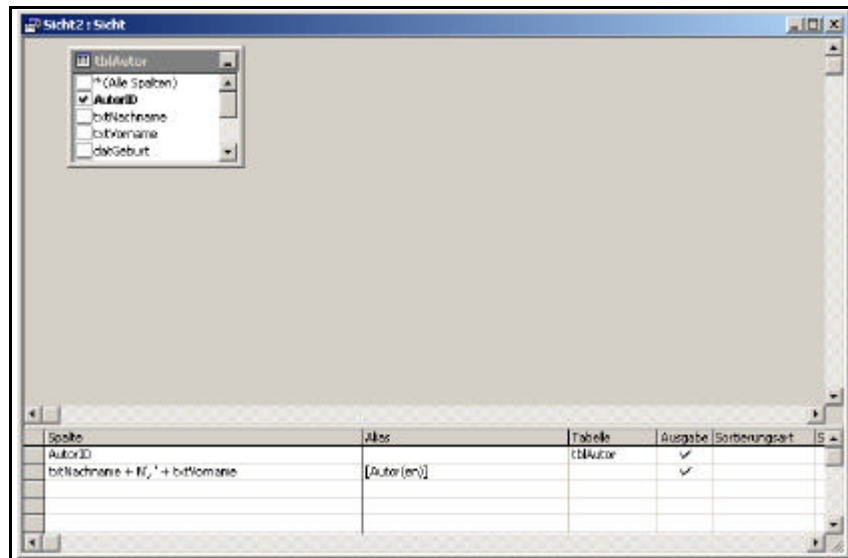


Abbildung 33: Abfrageentwurf

Der Abfragedesigner ändert den Spalteneintrag in `txtNachname + N' + txtVorname` ab und setzt um den Aliasbegriff zwei eckige Klammern.

Wählen Sie anschließend für dieses Feld unter *Sortierungsart* den Eintrag *Aufsteigend* aus und aktivieren Sie den SQL-Bereich.

Access-Projekt: Behebung der Upsizing-Fehler

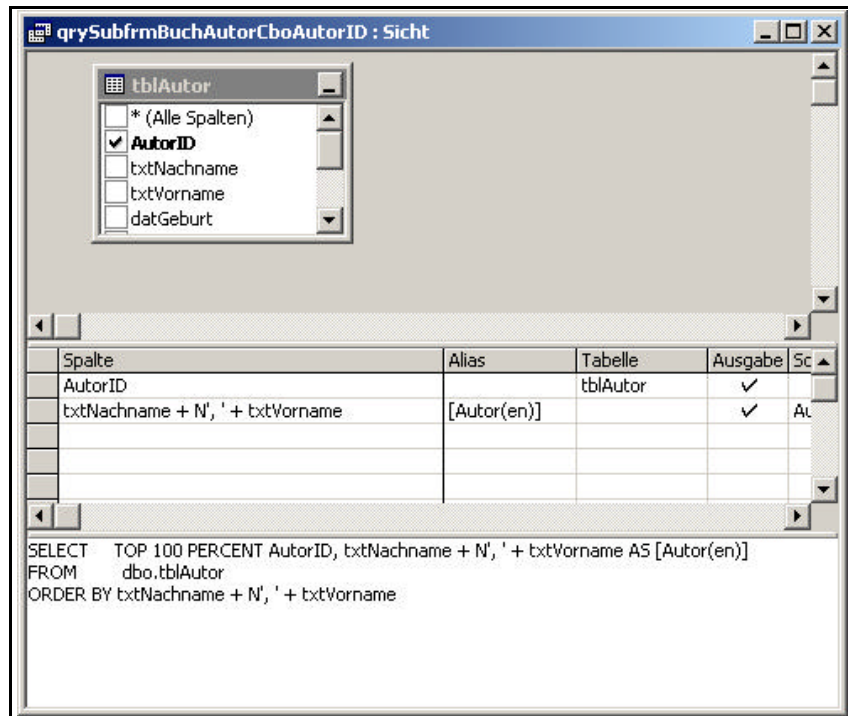


Abbildung 34: Abfrageentwurf mit SQL-Bereich

Der View ist nun fertig und sollte unter dem Namen *qrySubfrmBuchAutorCboAutorID* gespeichert werden, weil auf diesen Abfragenamen die Datenherkunft des Kombinationsfeldes *CboAutorID* eingestellt ist.

Anschließend funktioniert auch das Unterformular wieder fehlerfrei.

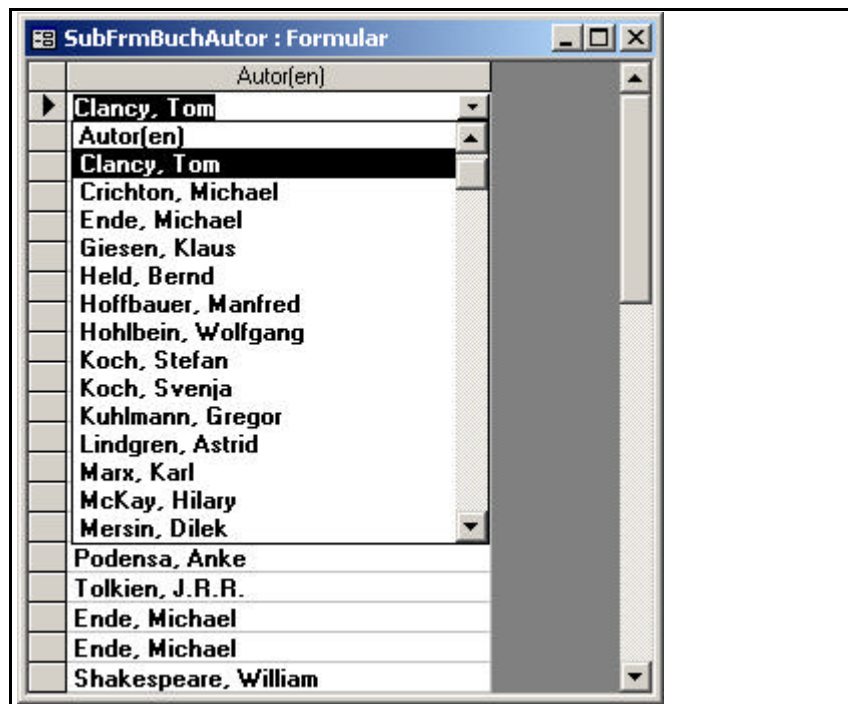


Abbildung 35: Das Formular *SubFrmBuchAutor*

Daraus kann ohne weiteres der Schluss gezogen werden, dass sich ab jetzt auch das Formular *frmBuch* wieder ohne Fehlermeldung öffnen lässt.

Behebung Upsizing-Fehler 3: Die Bücherliste

Der Grund für diesen Fehler liegt nach einiger Überlegung auf der Hand:

Die Funktion, die als Datenherkunft für das Listenfeld *LstBuchliste* im Formular *frmAutor* dient, wird auf dem SQL Server ausgeführt. Dort hat sie aber naturgemäß keinen Zugriff auf das Formularsteuerelement *AutorID* im Formular *frmAutor*, das sich im Access-Projekt befindet. Das ganze Problem kann also durchaus auch räumlich betrachtet werden.

Access-Projekt: Behebung der Upsizing-Fehler

Eine Lösung für dieses Problem ist die Erstellung einer gespeicherten Prozedur, mit der die gleiche Funktionalität erreicht werden kann. Diese gespeicherte Prozedur muss dann zwingend einen Parameter enthalten, der den gleichen Namen wie das Formularsteuerelement hat: *AutorID*. Das ist schon die komplette Lösung.

Erstellung der gespeicherten Prozedur

Die Erstellung der gespeicherten Prozedur erfolgt genauso, wie beim letzten Mal.

Klicken Sie im Bereich *Abfragen* des Datenbankfensters auf die Schaltfläche *Neu* und wählen Sie im Dialogfenster *Neue Abfrage* den Eintrag *Gespeicherte Prozedur entwerfen* aus und klicken Sie auf die Schaltfläche *OK*.

Fügen Sie folgende Tabellen zum Entwurfsfenster hinzu:

- *tblAutor*
- *tblBuch*
- *tblBuchAutor*
- *tblKategorie*

Stellen Sie anschließend die Spalten genauso wie in der folgenden Abbildung 36 ein:

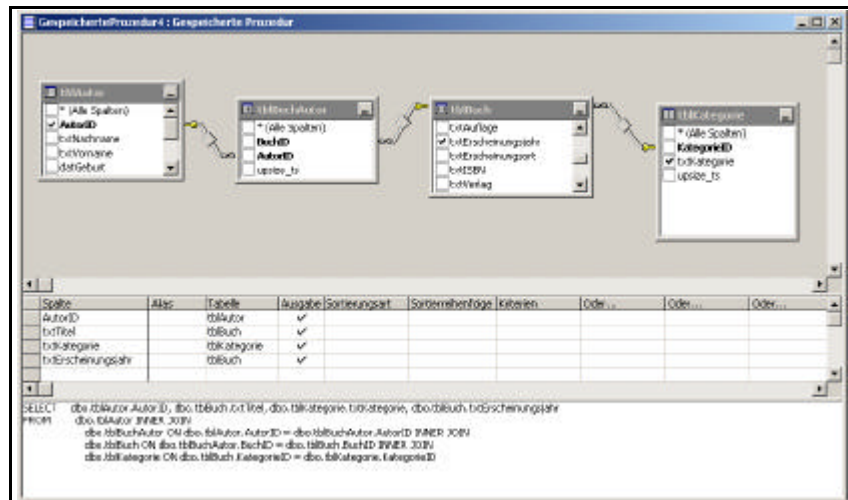


Abbildung 36: Die gespeicherte Prozedur in der Entwurfsansicht 1

Achten Sie dabei bitte genau auf die Reihenfolge der Felder, weil darauf im Listenfeld des Formulars Bezug genommen wird.

Geben Sie anschließend für das Feld *AutorID* in der Spalte *Kriterien* die Variable `=@AutorID` ein und stellen Sie für das Feld *txtTitel* die Sortierungsart auf *Aufsteigend* ein.

Access-Projekt: Behebung der Upsizing-Fehler

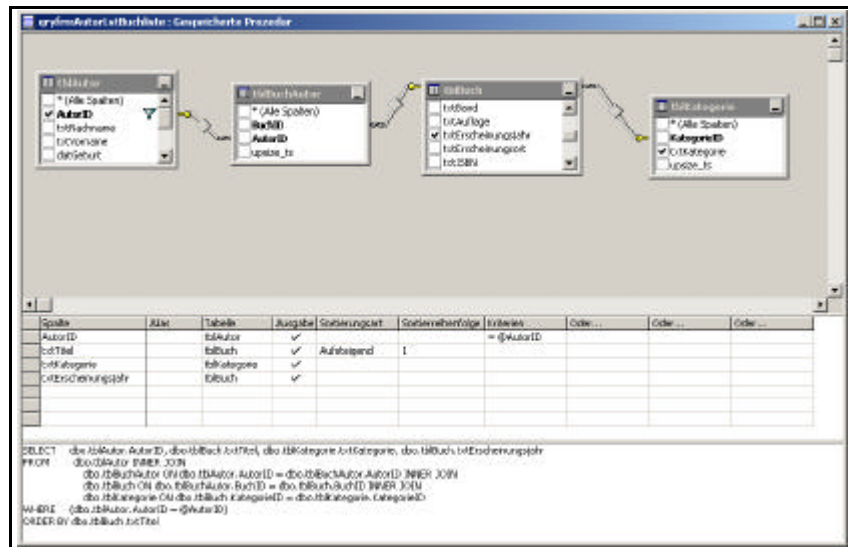


Abbildung 37: Die gespeicherte Prozedur in der Entwurfsansicht 2

Speichern Sie anschließend die gespeicherte Prozedur unter dem Namen *qryfrmAutorLstBuchliste*.

Anschließend kann auch das Autorenformular wieder ohne Fehlermeldung geöffnet werden und die Funktionalität der Bücherliste steht ebenfalls wieder vollständig zur Verfügung.

Behebung Upsizing-Fehler 4: Das Übersichtsformular

Die einfachste Lösung für die Behebung dieser Macke ist, den Eintrag für die Übersichtsänderung einfach aus der Tabelle *Switchboard Items* per Hand zu löschen.

Eine etwas kompliziertere Lösung würde die Änderung die Änderungen der Übersichtseinträge von dem Access-Assistenten abkoppeln. Dazu müsste allerdings die gesamte Programmlogik für den Zugriff auf die Tabelle *Switchboard Items* und die Änderung der Einträge in der Tabelle neu programmiert werden.

Das aber ist für diesen Zweck einfach zu aufwendig und deshalb ist die einfachere Lösung vorzuziehen.

Öffnen Sie die Tabelle *Switchboard Items* und markieren Sie den betreffenden Datensatz, der diesen Eintrag im Übersichtsformular erzeugt.

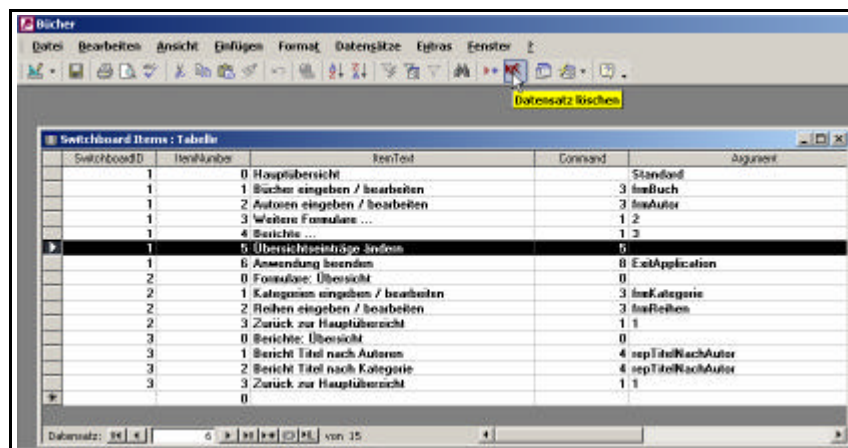


Abbildung 38: Datensatz aus der Tabelle *Switchboard Items* löschen

Löschen Sie dann diesen Datensatz. Anschließend wird dieser Eintrag nicht mehr im Übersichtsformular angezeigt.

Neue Einträge für das Übersichtsformular können weiterhin per Hand durch Hinzufügen der entsprechenden Einträge in der Tabelle *Switchboard Items* angelegt werden.

Behebung Upsizing-Fehler 5: Anlage neuer Autoren im Bücherformular

Die erste Fehleranalyse hat ergeben, dass der Fehler mit der Codezeile

```
Me![AutorID] = Null
```

zusammenhängen könnte.

In einem solchen Fall ist es durchaus legitim und auch üblich, das Problem empirisch – oder einfacher ausgedrückt durch Ausprobieren – anzugehen.

Access-Projekt: Behebung der Upsizing-Fehler

Bei diesem konkreten Beispiel braucht dazu lediglich der `Else`-Zweig in der `If`-Anweisung in der Ereignisprozedur auskommentiert zu werden.

```
    If IsNull(Me![AutorID]) Then
        Me![AutorID].Text = ""
    ' Else
    '     lngAutorID = Me![AutorID]
    '     Me![AutorID] = Null
    End If
```

Ein erneuter Versuch bestätigt die Vermutung:

Bei einem Klick auf das Kombinationsfeld tritt kein Fehler mehr auf und das Autorenformular zur Anlage eines neuen Datensatzes wird geöffnet.

Damit ist der Test allerdings noch nicht beendet: Ein neu eingegebener Datensatz muss auch gespeichert und anschließend im Kombinationsfeld zur Auswahl zur Verfügung stehen. Auch das funktioniert einwandfrei.

Damit ist allerdings noch nicht die eigentliche Ursache des Problems gefunden. Es liegt allerdings die Vermutung nahe, dass dieser Fehler irgendwie mit Schlüsseleigenschaften oder Beziehungen zusammenhängt. Dazu ist zunächst ein Blick auf den Tabellenentwurf aller Tabellen nötig, in denen das Feld *AutorID* vorkommt. Zum Glück sind das nicht allzu viele.

Die Fehlerursache

Die Lösung findet sich in der Tabelle *tblBuchAutor*.

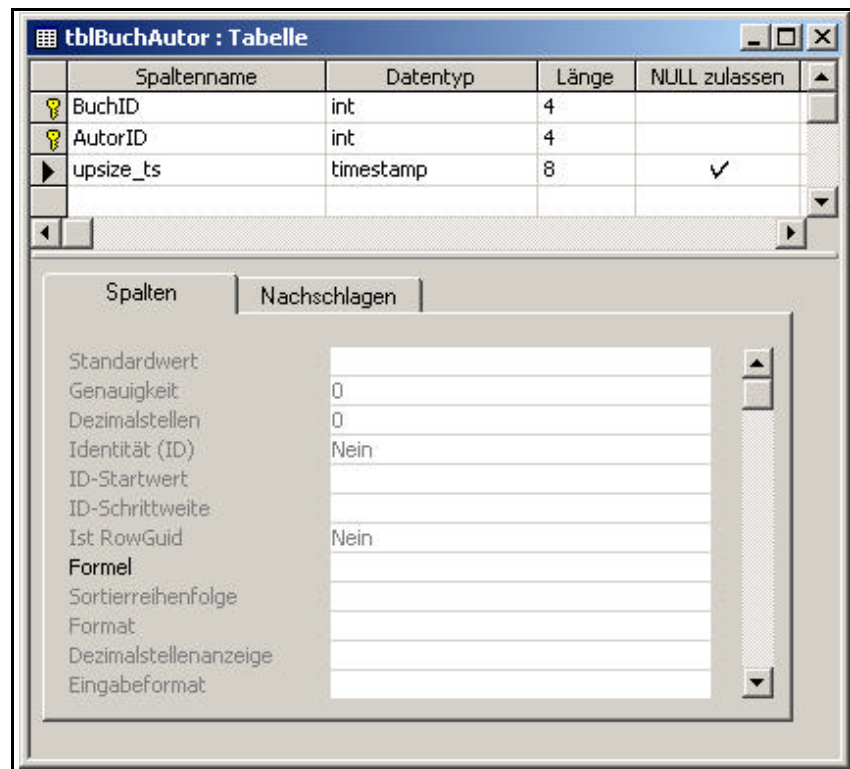


Abbildung 39: Die Tabelle *tblBuchAutor* im Access-Projekt

Bei den beiden Feldern des gemeinsamen Primärschlüssels *BuchID* und *AutorID* kann die Eigenschaft *NULL zulassen* nicht aktiviert werden. So ist nun einmal der Standard für Primärschlüsselfelder beim SQL Server.

Im Unterschied zu einer Access-Datenbank wird beim SQL Server der Wert für das Primärschlüsselfeld nicht schon vor dem Speichern des neuen Datensatzes ermittelt, sondern erst nach dem Speichervorgang. Deswegen führt die Zuweisung

```
Me![AutorID] = Null
```

zum dem bekannten Fehler.

Im Vergleich dazu schaut genau die Tabelle *tblBuchAutor* in der Access-Datenbank genau an dieser Stelle grundsätzlich anders aus.

Access-Projekt: Behebung der Upsizing-Fehler

Die Eigenschaft *Eingabe erforderlich* der gemeinsamen Primärschlüsselfelder *BuchID* und *AutorID* ist auf den Wert *Nein* eingestellt. Das ist auch der Grund dafür, dass in der Access-Datenbank die Ereignisprozedur zum Anlegen neuer Autoren funktioniert.

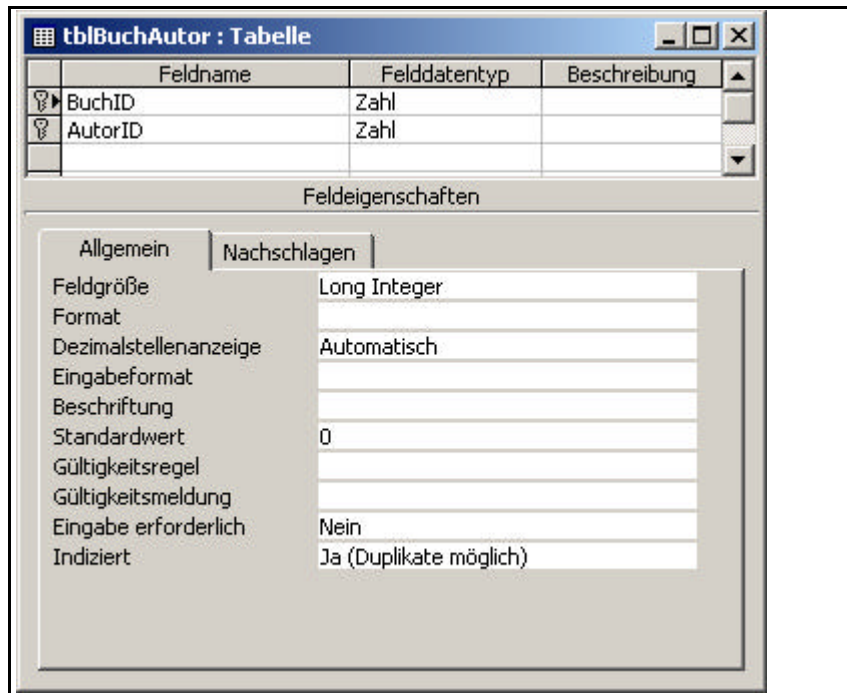


Abbildung 40: Die Tabelle *tblBuchAutor* in der Access-Datenbank

Lösungsmöglichkeiten

Wie könnte eine Lösung für dieses Problem ausschauen?

Als erste Möglichkeit könnte man einfach die drei Zeilen in der Ereignisprozedur auskommentiert lassen. Das ist – gelinde gesagt – nicht sonderlich elegant weil der *Else*-Zweig der *If*-Anweisung zu der Prozedurlogik hinzugehört.

Die bessere und programmatorisch korrekte Lösung in diesem Fall ist eine Änderung der Programmierlogik:

- Die Ereignisprozedur `Private Sub AutorID_Db1Click` wird komplett gelöscht.
- Die Ereignisprozedur beim Ereignis *Bei nicht in Liste* oder `NotInList` wird durch folgenden VBA-Code ersetzt, der im Prinzip genau die gleiche Funktionalität leistet:

```
Private Sub AutorID_NotInList(NewData As String,
Response As Integer)
On Error Resume Next
```

```
    If MsgBox("Der Autor ist noch nicht vorhanden.
Möchten Sie ihn anlegen?", vbYesNo) = vbYes Then
        Response = acDataErrContinue
        DoCmd.OpenForm "frmAutor", , , , acFormAdd
        Forms!frmAutor!AutorID = NewData
```

```
    Else      ' eventuellen Tippfehler abfangen
```

```
        Response = acDataErrContinue
        AutorID.Undo
    End If
```

```
End Sub
```

Allerdings muss in diesem Fall die Zuweisung des Werts für die *AutorID* an das Kombinationsfeld und das erneute Abfragen seiner Datenherkunft vom Autorenformular aus erfolgen.

Dazu bietet sich das Ereignis *Beim Schließen* des Formulars *frmAutor* an. Die dort schon vorhandene Ereignisprozedur braucht lediglich um zwei Zeilen ergänzt zu werden:

```
...
' Wenn das Bücherformular geladen ist
If IsLoaded("frmBuch") Then
```

Access-Projekt: Behebung der Upsizing-Fehler

```
Forms![frmBuch]![SubBuchAutor].Form![AutorID] =  
Me!AutorID
```

```
Forms![frmBuch]![SubBuchAutor].Form![AutorID].Requery
```

```
Exit Sub
```

```
...
```

Achten Sie bitte auf die Syntax für den Bezug auf das Steuerelement im Unterformular des Formulars *frmBuch*.

Die allgemeine Form dieser Syntax ist:

```
Forms![Hauptformular]![UnterformularSteuerelement im  
Hauptformular].Form![Steuerelement im Unterformular]
```

Damit sind alle Fehler und Macken in dem neu erstellten Access-Projekt beseitigt und die Anwendung funktioniert wieder genauso, wie vor dem Upsizing.

Jetzt ist es aber an der Zeit, sich mit den Einzelheiten und den Feinheiten eines Access-Projekts vertraut zu machen. Dabei geht es, wie nicht anders zu erwarten, in erster Linie um eine ganze Menge SQL Server-Know-how.