

## Das verbleibende Problem

Haben Sie vielleicht schon bemerkt, welches Problem mit der Technik der Datenbankaufteilung weiterhin ungelöst bleibt? Was ist denn, wenn die weitere Entwicklung Ihrer Datenbankanwendung nicht darin besteht, ein Formular oder einen Bericht zu verbessern oder ein neues Formular oder einen neuen Bericht zu erstellen? Was ist denn, wenn Sie feststellen, dass in einer Tabelle eine Spalte fehlt oder dass sogar eine ganze Tabelle fehlt? Dann müssten Sie ja eine neue Backend-Datei erstellen und dem Benutzer übergeben! Und was passiert dann mit seinen Daten? Dann haben wir ja wieder das Problem, das wir bereits für gelöst hielten!



**Wichtig** Investieren Sie unbedingt genügend Zeit in die Überlegungen zur Datenmodellierung.

Spätestens wenn Sie die Datenbankaufteilung durchführen, muss Ihr Datenmodell ziemlich perfekt sein und alle nur denkbaren Eventualitäten bereits berücksichtigen. Aber auch für den Fall, dass es trotzdem notwendig werden sollte, die Tabellenstruktur im Backend zu ändern, gibt es eine Lösung.

Der Lösungsansatz besteht darin, dass Sie es dem Benutzer auf keinen Fall zumuten können, ihm eine neue Backend-Datei zu geben, denn dann entstünde wieder das Problem der Rettung der vorhandenen Daten. Sie können dem Benutzer höchstens eine neue Frontend-Datei geben. Diese muss es dann irgendwie bewerkstelligen, die Backend-Datei des Benutzers zu manipulieren, ohne die vorhandenen Daten zu beschädigen.

Und das geht tatsächlich!

Sie benötigen in der Frontenddatei eine Tabelle SQLBEFEHLE mit den Spalten SQL\_KEY, SQL\_VERSION und SQL\_BEFEHL. Sie enthält – wie die Spaltennamen schon erahnen lassen – eine Liste aller SQL-Befehle, die notwendig sind, um die Backenddatei auf den neuesten Stand zu bringen.

Beispiele für die wichtigsten Befehle:

- Anlegen einer neuen Tabelle:  

```
create table XXX (XXX_KEY COUNTER PRIMARY KEY,  
XXX_SPALTE1 VARCHAR(50),  
XXX_SPALTE2 DOUBLE)
```
- Hinzufügen einer neuen Spalte zu einer existierenden Tabelle:  

```
alter table XXX add column XXX_SPALTE1 VARCHAR(50)
```
- Einfügen von Daten in eine existierende Tabelle:  

```
insert into XXX (XXX_SPALTE1, XXX_SPALTE2) values ('texttext', 5.8)
```
- Hinzufügen eines Fremdschlüssels zu einer existierenden Tabelle:  

```
alter table XXX add constraint FK_YYY foreign key (YYY_KEY) references  
YYY (YYY_KEY)
```

**Kapitel 10: Datenbankanwendungen 357**

Der letzte SQL-Befehl stellt durch den Zusatz »references« gleichzeitig die notwendige Beziehung zu der Tabelle YYY mit dem dazugehörigen Primärschlüssel YYY\_KEY her.

In der Form\_Open-Prozedur desjenigen Formulars, das sich beim Start der Datenbank als erstes öffnet, sorgen dann folgende VBA-Befehle dafür, dass die SQL-Befehle aus der Tabelle SQLBEFEHLE ausgeführt werden:

```
Set rs = CurrentDb.OpenRecordset("SQLBEFEHLE", dbOpenDynaset)
rs.MoveFirst
For VersNr = VorhandeneVersion + 1 To ErforderlicheVersion
    rs.MoveFirst
    Do Until rs.EOF
        If rs!SQL_VERSION = VersNr Then db_be.Execute rs!SQL_BEFEHL
        rs.MoveNext
    Loop
Next VersNr
```

Im obigen Code werden Ihnen die beiden Variablen VorhandeneVersion und ErforderlicheVersion auffallen. Damit sind wir bei der Frage:

## Welche Version ist es denn?

Ich mag es gar nicht sagen, aber mit dieser Lösung entsteht schon wieder ein neues Problem: Woher weiß denn die neue Frontend-Version, dass es sich bei dem von ihr vorgefundenen Backend um eine alte Version handelt?



**Profitipp** Erstellen Sie in der Frontend-Datei und in der Backend-Datei die Tabellen FE\_INFO bzw. BE\_INFO mit den beiden Spalten FE\_NAME und FE\_WERT bzw. BE\_NAME und BE\_WERT.

Dabei handelt es sich um »Universaltabellen«, die beliebige Informationen aufnehmen können – z.B.

- BE\_INFO.BEI\_NAME="version" und
- BE\_INFO.BEI\_WERT="3".

Dann können Sie mit dem SQL-Befehl

```
select BEI_WERT from BE_INFO where BEI_NAME="version"
```

herausbekommen, welche Versionsnummer die Backend-Datei hat! Darauf aufbauend können Sie dann VBA-gesteuert entscheiden, welche Veränderungen an der Tabellenstruktur der Backend-Datei evtl. erforderlich sind.