

Link



Dipl.-Math. Wolfgang Woide

Nach dem Studium der Mathematik und Informatik Organisationsprogrammierung auf Nixdorf-Computern, achtzehn Jahre IT-Vertrieb, anschließend selbstständiger Vermittler von IT-Fachpersonal. Seit 2016 entwickelt er wieder – vor allem kaufmännische Lösungen in FileMaker.

woide@woisoft.de

Aus Eins mach Zwei

Trennung von Logik und Layout in Datenbanken

Eine FileMaker Datenbank enthält normalerweise die kompletten Definitionen und Inhalte von Layouts, Scripts und Tabellen. Das führt dazu, dass sie mit der Zeit einen riesigen Rucksack an Daten mit sich herumschleppt und immer unhandlicher wird, wenn es um die Einspielung von Updates beim Kunden geht.

Aus diesem Grund praktiziere ich schon länger eine Aufteilung meiner FileMaker Datenbanken in zwei Dateien:

- Datei 1 (**Kunden.fmp12**) enthält nur die Layouts, Scripts und die Definition der Beziehungen
- Datei 2 (**Kunden_Tabellen.fmp12**) beinhaltet die Tabellen, die Tabellendefinition und die gespeicherten Daten

Das Verfahren hat sich in der Zusammenarbeit mit meinen Kunden als praktikabel erwiesen und auch das notwendige Script für den Datenimport funktioniert seit mehreren Jahren absolut zuverlässig.

Der Vorteil dieser Vorgehensweise liegt darin, dass bei Layout- oder Scriptänderungen lediglich die Datei 1 beim Kunden per Copy eingespielt werden muss, da die Kundendaten nicht davon tangiert werden.

Es gibt allerdings auch einen Nachteil: Ist eine Änderung im Bereich der Tabellen bei Kunden zu installieren, müssen die Kundendaten in die neue Struktur der Datei 2 umkopiert bzw. importiert werden.

Damit wird ein Importprogramm benötigt, das für jede Tabelle vorhanden sein muss. Da der Import vom Aufbau her



identisch ist, kann er für die einzelnen Tabellen aber einfach kopiert und angepasst werden.

Wenn Sie Ihre Datenbank aufteilen möchten, sollten Sie die nachfolgenden Punkte unbedingt beachten:

- In Datei 1 muss unter dem Menüpunkt „Datenquellen“ die Datei 2 eingefügt werden.
- Datei 1 und Datei 2 müssen im selben Ordner gespeichert sein.
- Im Beziehungsdiagramm von Datei 1 werden sämtliche Tabellen grafisch dargestellt, die in Datei 2 definiert wurden.
- In Datei 1 werden zudem sämtliche Beziehungen definiert und grafisch dargestellt.

Das Import-Script

Eine wichtige Voraussetzung für die Verwendung des Import-Scripts ist die Definition der Datensatz-ID als fortlaufende Nummer.

Von **Kunden_Tabellen.fmp12** (= Datei 2) wird eine Kopie mit dem Namen **Kunden_Tabellen_ori.fmp12** erstellt und auf dem Desktop abgelegt (das Script unterscheidet zwischen Windows- und Mac-Version).

In unserem Beispiel wird anschließend in **Kunden_Tabellen.fmp12** das Script „Import Kundenstamm“ durch einen Klick auf die Schaltfläche „Import im Layout Kunden“ aktiviert.

Import Kundenstamm

```
◆ Script ausführen
[ Angegeben: Aus Liste ; "Pfad Desktopordner" ; Parameter: ]

◆ Variable setzen
[ $path ; Wert: $$Pfaddesktop&"Kunden_Tabellen_ori.fmp12" ]

◆ Gehe zu Layout
[ "Kunden" ; Animation: Ohne ]

◆ Alle Datensätze anzeigen

◆ Alle Datensätze löschen
[ Mit Dialog: Aus ]

◆ Datensätze importieren
[ Mit Dialog: Aus ; "$path" ; Hinzufügen; Mac Roman ]

◆ Sortieren
[ Wiederherstellen ; Mit Dialog: Aus ]

◆ Gehe zu Datens./Abfrage/Seite
[ Letzte(r) ]

◆ Nächste fortlaufende Nummer setzen
[ Kunden::ID_Kd ; Kunden::ID_Kd+1 ]
```

In diesem Beispiel verwende ich als Primärschlüssel eine fortlaufende Nummer je Tabelle.¹ Deshalb muss nach dem Import in der jeweiligen Tabelle mit „Nächste fortlaufende Nummer setzen“ die erste freie Nummer ermittelt und gesetzt werden. Beim Import von offline erfassten Daten kann diese Struktur zu Kollisionen führen, da jedes mobile Gerät seinen eigenen Nummernkreis verwaltet. Dieses Problem lässt sich mithilfe einer PersistentID lösen.²

Die Verwendung von UUIDs würde Dopplungen ausschließen; ein bestehendes System auf UUID umzurüsten bedeutet allerdings einen erheblichen Aufwand.

Wie man nachfolgend sieht, wird im Script zuerst der Desktoppfad ermittelt:

Pfad Desktopordner

```
◆ Variable setzen
[ $pfad ; Wert: Hole ( DesktopPfad ) ]

◆ Variable setzen
[ $pfad ; Wert: ZeichenRechts ( $pfad ; Länge ( $pfad ) -1 ) ]

◆ Wenn
[ Hole ( SystemPlattform ) = 1 ]

◆ Variable setzen
[ $$Pfaddesktop ; Wert: "filemac:/" & $pfad ]

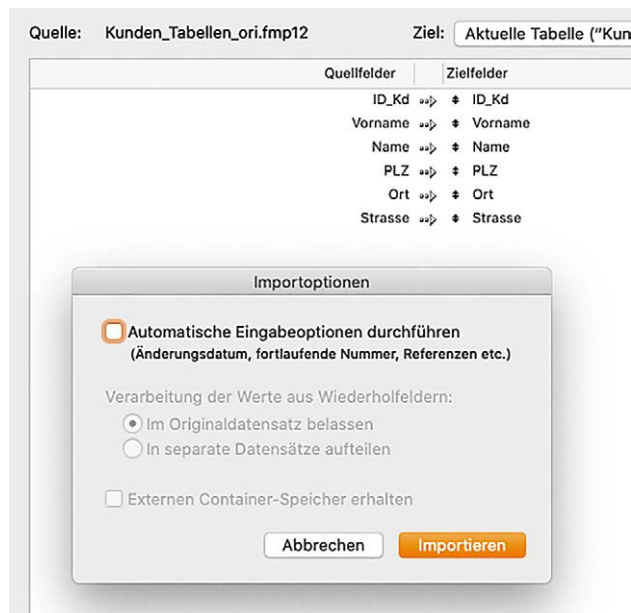
◆ Sonst, wenn
[ Hole ( SystemPlattform ) = -2 ]

◆ Variable setzen
[ $$Pfaddesktop ; Wert: "filewin:/" & $pfad ]

◆ Ende (wenn)
```

- In der Funktion „Datensätze importieren“ wird jedem Zielfeld ein Feld der Quelltable zugeordnet. Diese Zuordnung muss bei einer Änderung der jeweiligen Tabelle entsprechend angepasst werden.
- Bei den Importoptionen muss darauf geachtet werden, dass das Feld mit der eindeutigen Tabellen-ID aus der Quelldatei übernommen und nicht neu vergeben wird.

Das heißt, die automatische Eingabeoption darf nicht angekreuzt sein.



Am Ende des Scripts wird die höchste ID ermittelt, sodass dem System die nächste freie ID-Nummer zur Verfügung gestellt wird. Zuvor wird die Tabelle nach ID-Nummer aufsteigend sortiert.

Das beigefügte Beispiel enthält nur eine Tabelle zur Veranschaulichung. In der Praxis wird dieses Script mit den entsprechenden Änderungen für jede einzelne Tabelle enthalten sein. Anschließend wird ein Script definiert, das die einzelnen Importscrip ts nacheinander aufruft.

Ausblick

In einem weiteren Schritt habe ich eine zusätzliche Aufteilung vorgenommen: Ich verwalte die Anmeldung der einzelnen User in einer separaten FileMaker Datei, über die jeder User nach seiner Anmeldung per Script mit seiner Rolle und einem Admin-Passwort angemeldet wird. Auch diese Struktur hat sich in der Praxis bewährt, da die vielen individuellen Anmeldedaten immer beim Kunden verbleiben. In der nächsten Ausgabe werde ich diesem Verfahren einen separaten Artikel widmen. Ich wünsche viel Erfolg bei der Umsetzung und stehe gern für Rückfragen zur Verfügung. ■

Fußnoten

- 1) Für solche Operationen, aber auch um Datensätze anzulegen oder zu löschen, verwende ich eine Kopie der jeweiligen Tabelle, die in keiner Beziehung zu einer anderen Tabelle steht.
- 2) Die Technik mit PersistentID sowie das dazugehörige Import-Script werde ich in einem nachfolgenden Artikel erläutern.
- 3) https://de.wikipedia.org/wiki/Universally_Unique_Identifier

FileMaker Magazin



Das FileMaker Magazin

- Einzige deutschsprachige Fachzeitschrift zu FileMaker
- Wissen aus erster Hand von anerkannten FileMaker Fachautoren
- Große Themenvielfalt für Anwender und Entwickler

Exklusiv für Premium-Abonnenten

- Sechs FMM Ausgaben pro Jahr
- Kostenlose Nutzung des Abonnentenbereichs auf **www.filemaker-magazin.de**
- PDF-Archiv mit allen bisher veröffentlichten Ausgaben
- Jede Ausgabe mit kostenlosen Beispieldateien und Zusatzinfos zum Download

Unser Service

- Aktuelle Neuheiten, Tipps und Infos, Kleinanzeigen und vieles mehr jederzeit auf unseren Webseiten
- Hilfe bei allen Fragen zu FileMaker im FMM Forum
- Kompetente Beratung zum Kauf von FileMaker Lizenzen: Einfach anrufen +49 (0)40 589 65 79 70.

Wenn Sie sich für ein FileMaker Magazin
Abo interessieren, klicken Sie bitte hier!

Eine kostenlose **Leseprobe** des FileMaker Magazins erhalten Sie, wenn Sie hier klicken.

Hier finden Sie Aktuelles zu **FileMaker Lizenzen**, egal ob Sie kaufen, mieten oder sich einfach informieren möchten.