



Externe SQL- Quellen im Detail

Inhaltsverzeichnis

Über diese Technische Anleitung.....	4
Überblick über ESS	4
Wie ESS funktioniert	4
Über ODBC	4
Wie FileMaker Pro ESS-Tabellen verwaltet	6
Öffnen einer ESS-Tabelle	9
ESS Daten-Zwischenspeicherung	11
Suche in ESS-Tabellen.....	12
Über die Indizierung.....	12
Wie FileMaker Pro eine ESS-Tabelle durchsucht.....	12
Optimierung von ESS-Suchläufen.....	13
Sortieren in ESS-Tabellen.....	14
ESS im Beziehungsgraph	15
Bearbeiten von Daten in ESS-Tabellen	16
Erstellen von Daten.....	16
Lesen von Daten.....	17
Bearbeiten von Daten.....	17
Löschen von Daten	18
Information über Änderungen.....	18
ESS Mapping von Datentypen	19
Über MS SQL Server DateTime Felder	20
Über Datenbereiche und Validierung.....	20
Über Views	22
Schreibfähige Views	23
Views und Primärschlüssel	24
Komplexe Views	24
Verwaltung der ODBC-Verbindungen.....	25
Sicherheit und Authentifizierung.....	25
Über Verschlüsselung	26
Fehlerbehandlung	27
Weitere Beschränkungen von ESS	28
Binärdaten.....	28
Rich Text.....	28
Blättern in großen Datensatzmengen.....	28
Vorgänger/Nachfolger Lookup.....	28
Aus Index einfügen	29
Auto-Vervollständigen	29
Tabellenbasierte Wertelisten.....	29
Scriptverweise auf ESS-Felder	29
Verwalten von ESS-Feldern in Custom Web Publishing	30
Einige ESS-Verwendungsszenarien	30
1. 1. Open Source e-Commerce Lösung	30
2. Daten mit einer kommerziellen Anwendung teilen	31



3. Einblick in das Enterprise Finanzsystem.....	31
Optimale ESS-Verfahren für Entwickler	32
Integration statt Skalierung	32
Machen Sie sich mit der SQL-Sprache vertraut.....	32
Machen Sie sich mit einem ESS-Produkt vertraut.....	32
Halten Sie den Kontakt mit großen Datenmengen klein	33
Zusammenfassung.....	33



Über diese Technische Anleitung

Diese Technische Anleitung soll FileMaker® Pro Entwicklern und anderen Informationssystemexperten helfen, die technischen Einzelheiten der neuen Externen SQL Datenquellen (ESS)-Funktionalität der FileMaker Pro 9 Produktreihe besser zu verstehen. Mit Hilfe dieses Dokuments können Entwickler und Informationsarchitekten die Hauptvorteile des ESS-Funktionsumfangs bewerten und Einblicke in die Anwendungsbereiche und Integrationsmöglichkeiten gewinnen, für die die Funktionalität entwickelt wurde.

Dieser Artikel soll Entwicklern technische Hilfestellung und optimale Verfahren beim Bau von Lösungen mit ESS anbieten. Er ergänzt und erweitert den Artikel Einführung zu Externen SQL-Quellen, der auf der FileMaker-Webseite zur Verfügung steht. Leser sollten mit dem Inhalt der Einführungsanleitung vertraut sein, bevor Sie dieses Dokument lesen. Dieses einführende ESS-Dokument kann von folgender Seite heruntergeladen werden:

http://filemaker.custhelp.com/cgi-bin/filemaker.cfg/php/enduser/std_adp.php?p_faqid=6566

Überblick über ESS

Hauptmotivation und Entwurfsziele hinter der ESS-Funktionalität sind:

ESS ist eine Technologie, die es FileMaker Pro Lösungen ermöglicht, Datentabellen aus weit verbreiteten SQL-Speichersystemen zu integrieren, während gleichzeitig die Arbeit im Hinblick auf bestehende Metaphern und Konventionen so transparent wie möglich ist. ESS bietet eine einfache Verwendbarkeit und Transparenz und wurde nicht speziell entwickelt, um die Skalierbarkeit oder Leistungsfähigkeit von Lösungen auf FileMaker-Basis zu steigern.

Viele der speziellen technischen Entscheidungen innerhalb der ESS-Funktionalität sind unter Rückgriff auf diese Entwurfsziele besser verständlich.

Wie ESS funktioniert

Über ODBC

ESS kommuniziert mit ausgewählten externen Datenquellen (momentan Oracle Database 9g und 10g, MySQL Community Edition 5.0 und Microsoft® SQL Server 2000 und 2005) über Open Database Connectivity (ODBC) Aufrufe. ODBC ist eine Datenbank-Abstraktionsebene, die versucht, eine relative gleich bleibende Schnittstelle für eine Vielzahl an Datenquellen anzubieten, ob SQL-basiert oder anderweitig. (FileMaker Pro kann auch selbst eine ODBC-Datenquelle sein).

ODBC ermöglicht die Kommunikation mit einer Datenquelle über einen ODBC-Treiber: Eine kleine Software, die relative generische ODBC-Anfragen in entsprechende datenbankspezifische Anfragen übersetzt. Bei der Arbeit mit MS SQL Server wird der ODBC-Befehl „zeige mir alle Tabellen der Kundendatenbank“ in eine Anfrage wie „SELECT * FROM information_schema.tables“ umgewandelt. Der SQL-Server ODBC-Treiber ist für die Umwandlung der generischen Anfragen in eine speziell für MS SQL Server geeignete Anfrage und für die Rückübermittlung der Daten an den Klienten verantwortlich.



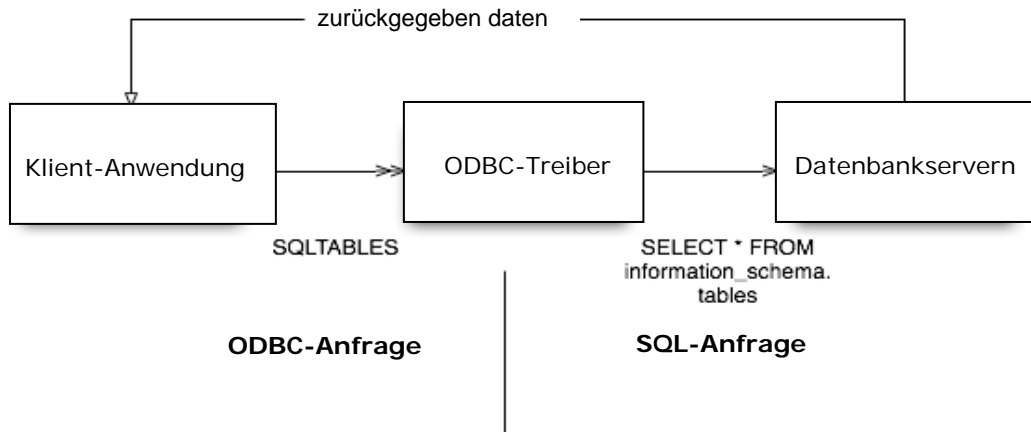


Abbildung 1

Bei typischen ODBC-Konfigurationen wird die Treibersoftware auf demselben Rechner wie die Klient-Anwendung installiert:

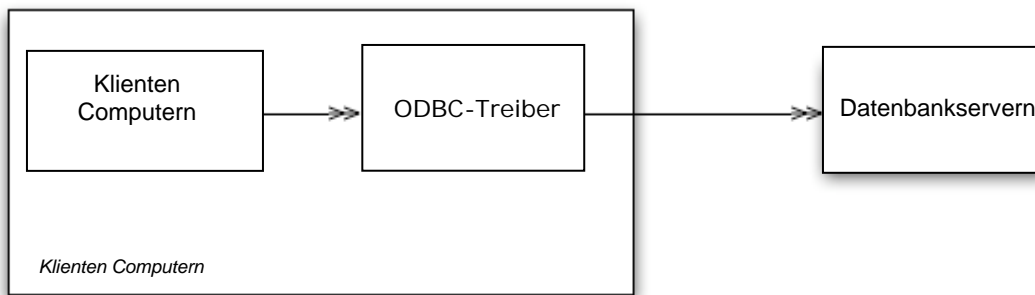


Abbildung 2

Da es mühsam sein kann, ODBC-Treiber auf vielen einzelnen Computern zu pflegen, ist es auch möglich, eine Anwendung in einer „ODBC Server“-Konfiguration aufzubauen, wobei die Treiber auf einem zentralen Server gespeichert und konfiguriert werden und die einzelnen Klienten dann mit dem Server interagieren, um Zugriff auf die ODBC-Daten zu erhalten.



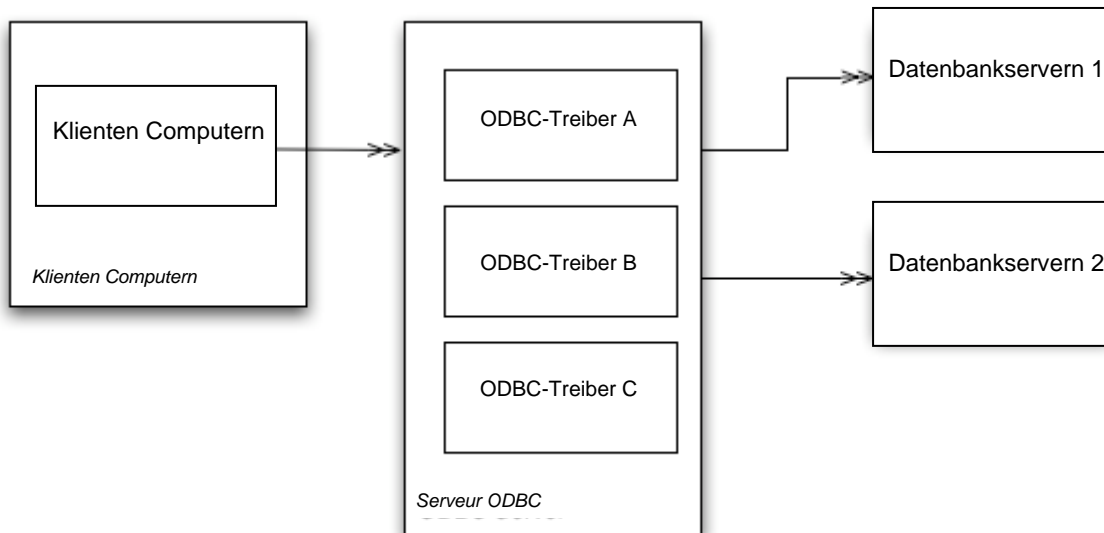


Abbildung 3

ESS in FileMaker Pro kann in jeder dieser Konfigurationen genutzt werden. Mit FileMaker Pro ESS ist es möglich, alle ODBC Verbindungs- und Konfigurationsinformationen auf einem zentralen FileMaker Server Rechner zu speichern, was die jeweilige Konfiguration von ODBC auf den einzelnen Arbeitsrechnern erspart. (Dies gilt nur, wenn die fraglichen FileMaker Pro Dateien auf einem FileMaker Server verwaltet werden).

Serverseitiges ODBC ist eine elegante Möglichkeit von FileMaker Pro ODBC, wo immer möglich als optimales Verfahren unterstützt werden sollte.

Wie FileMaker Pro ESS-Tabellen verwaltet

ESS-Tabellen im Beziehungsgraph

ESS-Tabellen können wie eigene FileMaker Pro Tabellen zum Beziehungsgraph hinzugefügt werden. Wie bei FileMaker Pro-eigenen Tabellen kann auch eine ESS-Tabelle mehrmals im Graph erscheinen und jedes Auftreten wird wieder durch eine andere Tabelleninstanz dargestellt. Wie bei externen FileMaker Pro-Tabellen erscheinen ESS-basierte Tabelleninstanzen mit kursiv geschriebenen Namen als Hinweis auf den externen Charakter. Im Gegensatz zu FileMaker Pro-Tabellen erscheinen ESS-Tabellen aber in der Tabellenliste der lokalen FileMaker Pro-Datei (ein Thema, dem wir uns im nächsten Abschnitt widmen).

Wie bei jeder anderen im Graph erscheinenden Tabelle können ESS-Tabellen Beziehungen mit eigenen FileMaker Pro-Tabellen oder auch mit anderen ESS-Tabellen haben. Es besteht sogar die Möglichkeit, Tabellen aus verschiedenen ESS-Quellen einzubinden und Beziehungen unter diesen zu erzeugen. (Man sollte dafür jedoch einen guten Grund haben, da so viele Anfragen an verschiedene Server generiert werden, was wiederum zu Leistungsproblemen führen kann).



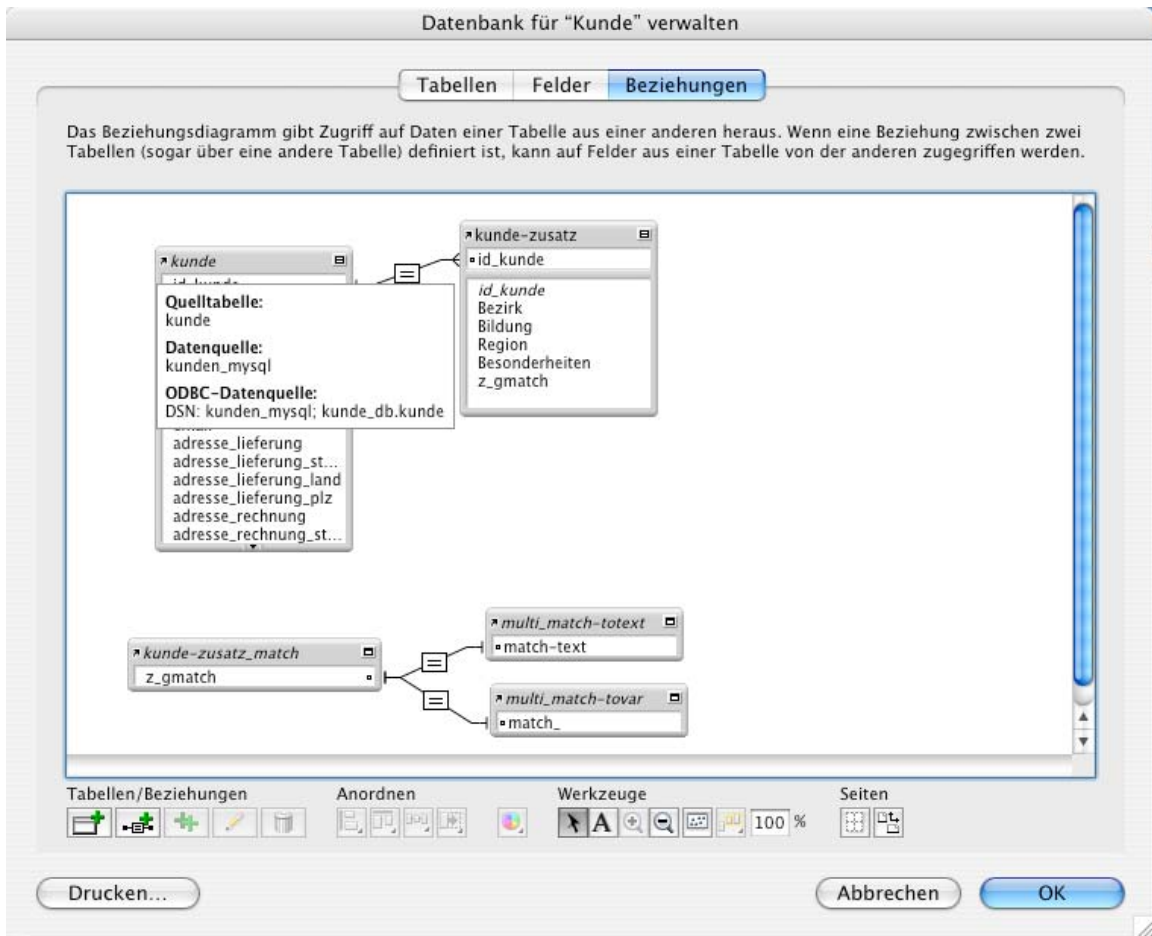


Abbildung 4

ESS-Tabellen im FileMaker Pro Schema

FileMaker Pro verwaltet ESS-basierte Tabellen, indem ein Schattenschema innerhalb einer FileMaker Pro Datenbank gepflegt wird. FileMaker Pro repräsentiert jede referenzierte ESS-Tabelle über eine entsprechende Schattentabelle, die zu der Liste der lokalen, eigenen FileMaker Pro-Tabellen hinzugefügt wird. Jede Schattentabelle enthält wiederum eine Anzahl von Schattenfeldern, die einigen der in der entsprechenden SQL-Tabelle definierten Feldern entsprechen (manche SQL-Felder haben eventuell nicht unterstützte Datentypen, wie Binärdaten, und diese SQL-Felder werden nicht in FileMaker Pro abgebildet. Diese Anleitung beschäftigt sich später noch genauer mit Problemen beim Mapping von Daten). Die Schattentabelle und die Schattenfelder werden erzeugt, wenn eine ESS-basierte Tabelle zum ersten Mal dem Beziehungsgraph in einer Datei hinzugefügt wird.

Das Schattenschema kann in einigen Punkten bearbeitet werden. Keiner dieser Schritte führt tatsächlich eine Änderung der Schemaelemente auf der SQL-Seite durch. Die ESS-Funktionalität von FileMaker Pro kann nicht direkt Tabellen oder Felder in einer SQL-Datenquelle löschen oder verändern. Es können jedoch einige nützliche Schritte auf der FileMaker Pro Seite durchgeführt werden.

Die Schattenfelder können in ihrer Funktionsweise erweitert werden, indem ihnen FileMaker Pro-Feldoptionen hinzugefügt werden, insbesondere Auto-Entry und Validierung. Mit wenigen Ausnahmen können alle FileMaker Pro Auto-Entry und Validierungsoptionen auf die



Schattenfelder angewendet werden. Dies hat keine Änderungen der entfernten SQL Datenbank zur Folge, sondern stellt nur Optionen dar, die innerhalb von FileMaker Pro verwaltet werden. Sie können beispielsweise einem ESS-Zeitfeld die Option hinzufügen, dass das Erstellungsdatum des Datensatzes automatisch eingetragen wird. Sie können auch einen automatischen Ersetzungsfiler anwenden, um eine Telefonnummer mit einem gegebenen Format zu maskieren.

Auf ähnliche Weise können Sie FileMaker Pro-Validierungseinschränkungen auf ESS-Felder über die Feldoptionen eines ESS-Schattenfeldes anwenden. Die einzige Beschränkung dieser Option resultiert aus der Tatsache, dass die ESS-Funktion Validierung auf Feldebene nutzt, um die Konformität der Datentypeinschränkungen auf dem ESS-Server zu verwenden. Zum Beispiel wird ein ESS-Feld, das auf dem Server als varchar(30) (Zeichenketten mit einer maximalen Länge von 30 Zeichen) definiert ist, mit einem FileMaker Pro-Textfeld gemappt, das Feld erhält jedoch eine Beschränkung auf 30 Zeichen über die FileMaker Pro-Feldvalidierung und dieser Grenzwert kann nicht geändert oder angehoben werden. (Siehe Abschnitt "ESS-Mapping von Datentypen" für weitere Informationen). Die Validierungsbeschränkungen "unique" (eindeutig) und „existing value“ (existierender Wert) können ebenfalls nicht auf ESS-Felder angewendet werden, da diese einen Zugriff auf die ESS-Indizes erfordern würden. (Als allgemeine Regel kann ein Befehl, der einen Zugriff auf Tabellenindizes erfordern würde, für ESS-Daten nicht ausgeführt werden. Eine Liste der wenigen anderen Einschränkungen dieser Art finden Sie weiter unten im Abschnitt „Weitere ESS-Einschränkungen“).

Speicheroptionen können für ESS-Felder nicht eingestellt werden. Die Speicherung von ESS-Feldern wird vollständig vom ESS-Server verwaltet und kann nicht überschrieben oder erweitert werden.

Das Löschen einer Schattentabelle löscht deren Verfügbarkeit für eine FileMaker Pro Lösung. Es entfernt die Tabelle aus der Tabellenliste und möglicherweise aus dem Beziehungsgraph. Es löscht die Tabelle aber nicht aus der SQL-Quelle. Wenn die zugrunde liegende ESS-Tabelle wieder zu dem Beziehungsgraph hinzugefügt wird, werden die Schattentabelle und deren Felder neu erzeugt.

Innerhalb einer Schattentabelle können Sie einzelne Schattenfelder löschen. Auch dies betrifft die ESS-Datenquelle nicht, sondern entfernt die Felder nur aus der Betrachtung durch FileMaker Pro, was kleinere Anfragen und schnellere Reaktionen möglich macht. Es ist eine gute Idee, alle Schattenfelder zu entfernen, die Sie mit Sicherheit nicht benötigen werden. Es ist jedoch immer möglich, die entfernten Felder „wiederherzustellen“, indem Sie das Schema der Schattentabelle mit seinem ESS-Gegenstück über die Sync-Schaltfläche im Dialog „Datenbank verwalten“ synchronisieren.

Abschließend ist es möglich, ergänzende Felder zu einer Schattentabellendefinition hinzuzufügen. Ergänzende Felder bestehen aus Ergebnisfeldern und nicht gespeicherten Formelfeldern, die zu einem Teil der Schattentabellendefinition werden und das ESS-Schema erweitern. Diese Felder sind streng auf nicht gespeicherte Logik beschränkt und erlauben die Einrichtung eines „abgeleiteten Feldes“, ein Feld, das seinen Wert aus anderen gespeicherten Feldern einer Tabelle berechnet, selbst aber nicht gespeichert oder indiziert wird.



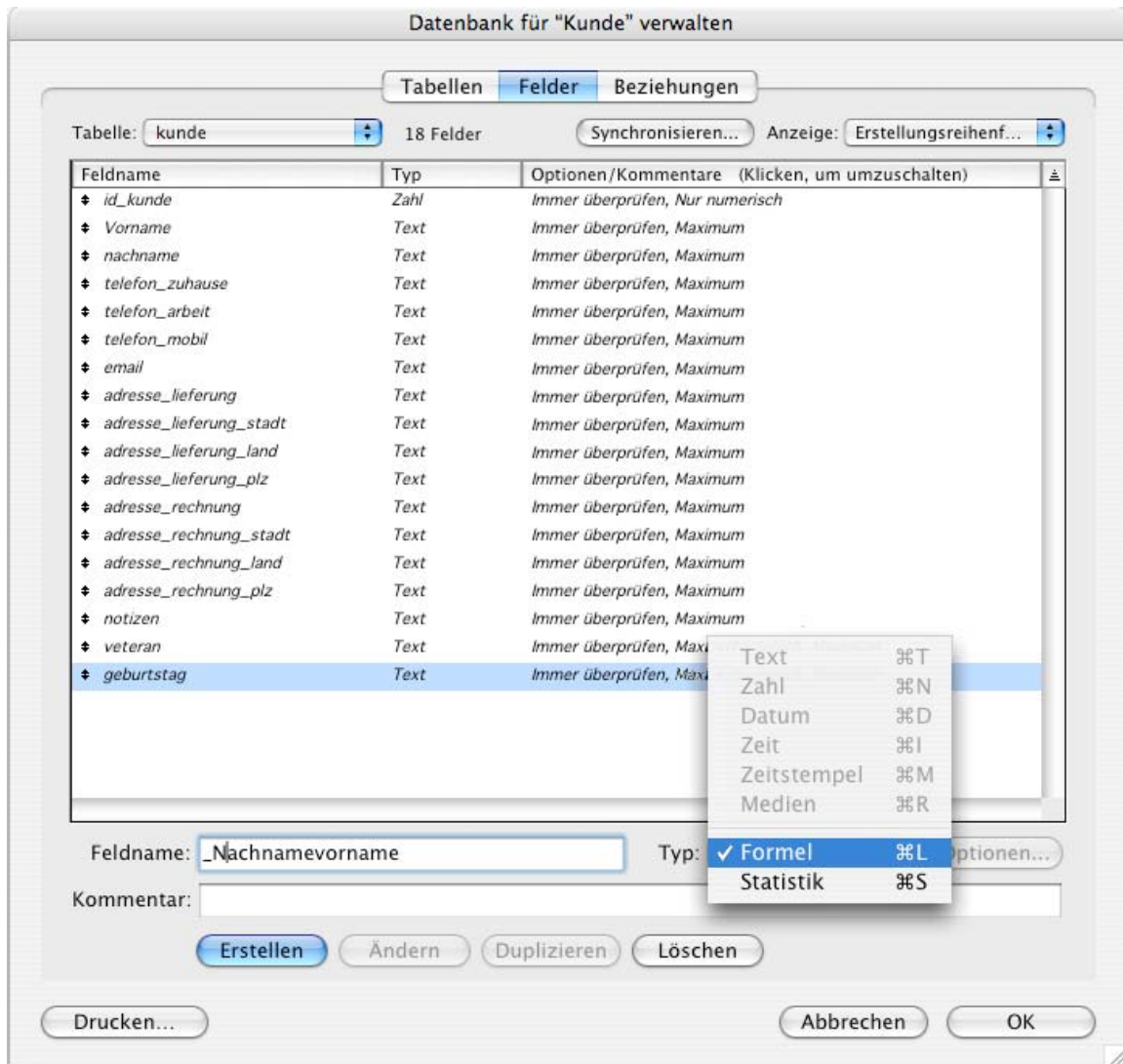


Abbildung 5

Öffnen einer ESS-Tabelle

Was passiert nun tatsächlich, wenn ein FileMaker Pro-Benutzer versucht, auf Daten einer ESS-Tabelle zuzugreifen?

Eines der Entwicklungsziele für die ESS-Funktionalität war der Erhalt der Klient-Server-Logik (soweit wie möglich), die die Interaktionen zwischen einem FileMaker Pro-Klient und einem FileMaker Pro-Host verwaltet. Mit anderen Worten, ein FileMaker Pro-Klient sollte Daten aus einer ESS-Tabelle auf dieselbe Weise anfragen und verwalten, wie die Daten einer eigenen FileMaker Pro-Tabelle verwaltet werden. Ziel war, dass keine komplett separate Interaktion für die Verwaltung ESS-basierter Daten geschrieben werden muss. Aus diesem Grund müssen ESS-Tabellen einige gleiche Merkmale wie eigene FileMaker Pro-Tabellen aufweisen, damit der FileMaker Pro-Host und Klient (z.B. FileMaker Server und FileMaker Pro) einem Gleich-Interaktionsmodell folgen können.



Primärschlüssel

Eine der wichtigsten Anforderungen für die Arbeit mit einer ESS-Tabelle ist, dass die Tabelle einen Primärschlüssel hat oder zumindest ein oder mehrere Felder, die als Primärschlüssel dienen können, selbst wenn sie nicht als solche in der SQL-Datendefinition ausgewiesen sind. FileMaker Pro-eigene Tabellen haben immer eine verborgene interne ID für jede Tabellenzeile, eine sog. Datensatz-ID, ungeachtet, ob der Entwickler ein Primärschlüssel-ähnliches Feld in der Tabelle definiert hat oder nicht. Die Klient-Server-Kommunikation in FileMaker Pro basiert auf der Fähigkeit, Datensätze über die Datensatz-ID abzufragen. Um ein ähnliches Interaktionsmodell für ESS-Tabellen zu erhalten, muss dieselbe Anforderung gelten: Jede ESS-Tabelle muss eine oder mehrere Spalten haben, die FileMaker Pro als eindeutigen Identifikationsschlüssel behandeln kann – die Datensatz-ID für die SQL-Zeile, wenn Sie so wollen.

Wenn eine Tabelle einer ESS-Datenquelle zum ersten Mal zum Beziehungsgraph hinzugefügt wird, versucht FileMaker Pro alles, um einen angemessenen Primärschlüssel zu finden. Dies geschieht durch Abfrage der SQL-Quelle, ob ein Primärschlüssel für die Tabelle definiert ist oder ob es andere geeignete eindeutige Felder oder Indizes gibt. Wenn FileMaker Pro selbst keinen angemessenen Primärschlüssel finden kann, werden Sie aufgefordert, eine oder mehrere Spalten der ESS-Tabelle als verwendbaren Schlüssel anzugeben. Die Daten dieser Spalten werden dann gescannt und es wird geprüft, ob diese wirklich (zu diesem Zeitpunkt) nur eindeutige Daten enthalten. Dieses Mapping eines Primärschlüssels kann später zurückgesetzt werden, wenn Sie einen anderen Primärschlüssel ausweisen möchten. Sie werden zur Eingabe eines Primärschlüssel-Mappings immer aufgefordert, wenn Sie das Schema einer Schattentabelle mit dem ESS-Server über die Schaltfläche Sync synchronisieren. (Dies alles gilt nur für den Fall, dass FileMaker Pro nicht selbst einen geeigneten Primärschlüssel feststellen kann).

Wenn eine ESS-Tabelle keine geeignete Primärschlüsselspalte oder –spalten enthält, ist es nicht möglich, die Tabelle dem Beziehungsgraph hinzuzufügen. FileMaker Pro betrachtet viele SQL-Feldtypen als geeignete Wahlmöglichkeit für Primärschlüssel, einige wenige SQL-Feldtypen werden jedoch nicht akzeptiert, darunter CLOB-Typ-Textfelder. (Es wäre ebenfalls davon abzuraten, einen numerischen Näherungswert wie eine Gleitkommazahl als Primärschlüssel festzulegen, obwohl FileMaker Pro dies nicht speziell untersagt).

Vorbereitung für die Verwaltung von ESS-Datensätzen

Eines der Entwurfsziele von ESS war, die Verwaltung von ESS-Datensätzen innerhalb eines Rahmens ähnlich dem, der für FileMaker Pro-Datensätze verwendet wird. Aus diesem Grund pflegt ESS ein Mapping zwischen dem Primärschlüssel einer ESS-Tabelle und der internen FileMaker Pro Datensatz-ID für die Zeile, wie sie in FileMaker Pro dargestellt ist.

FileMaker Pro versucht nicht, diese Mapping-Struktur auf einmal zu füllen, sondern während auf die ESS-Datensätze zugegriffen wird. Wenn die ESS-Datensätze gelesen werden, fügt FileMaker Pro diese seiner Mapping-Struktur hinzu, die in einer lokalen temporären Dateistruktur gespeichert wird und keinen großen Speicheraufwand auf dem Klient-Rechner erfordern sollte.

Es kann eine kleine Verzögerung beim Öffnen großer Dateien von bestimmten Datenbankservern geben. Genauer gesagt hängt die Zeit zum Öffnen einer FileMaker Pro-Datei, die auf eine Datenbank mit Kundendatensätzen zugreift, sehr von der Ausführungszeit dieser beiden Anfragen ab:

```
SELECT count(*) FROM kunde;
```

```
SELECT primary_key FROM kunde ORDER BY 1;
```



Unterschiedliche ESS-Plattformen können darin variieren, wie schnell solche Anfragen ausgeführt werden und die Ausführungszeit schwankt mit Sicherheit mit der Tabellengröße.

Daten abrufen

Wie bei der Arbeit mit eigenen FileMaker Pro-Tabellen ruft FileMaker Pro die Daten aus einer ESS-Tabelle nach Bedarf ab, wobei die Datensätze in Stapeln abgerufen werden, wann immer ein Benutzer die Daten anzeigt oder mit diesen arbeitet. Eine typische Stapelgröße beinhaltet 25-100 Datensätze und FileMaker Pro wird versuchen, zusätzliche Datensätze abzurufen, um zu erwartende Benutzeranfragen vorzubereiten.

Nehmen wir an, ein Benutzer öffnet eine FileMaker Pro-Datei und navigiert zu einem Layout, das Daten aus einer ESS-Tabelle anzeigt. Aufgrund von Größenbeschränkungen von Bildschirm und Layout kann er 55 Datenzeilen sehen. FileMaker Pro durchläuft dann die folgenden Schritte:

1. Füllen des Primärschlüssel-FileMaker-ID Mappings durch Abruf der Schlüssel aus den ungefähr 125 ersten ESS-Datensätzen.
2. Abrufen der tatsächlichen Zeilendaten aus der ESS-Tabelle über Abfragen basierend auf den Primärschlüsseln.
3. Speichern der Zeilendaten im lokalen Datenzwischenspeicher, wie für FileMaker Pro.
4. Anzeige der Daten für den Benutzer.

Aufgrund der stapelbasierten, bedarfsgerechten Weise, in der FileMaker Pro-Daten, ob über ESS oder anders, abrufen, kann diese Aktivität eine große Anzahl an Abfragen an den SQL Backend erzeugen. Da diese auf Primärschlüsseln basieren, sind die Abfragen sehr effizient, DBAs können jedoch bei der Beobachtung der Datenbankprotokolle die Anzahl der individuellen Abfragen erkennen.

Sobald Datenzeilen gemappt, abgerufen, zwischengespeichert und angezeigt wurden, sind diese für die Verwendung innerhalb von FileMaker Pro bereit. Sie verhalten sich ähnlich den FileMaker Pro-Zeilen, es gibt jedoch Unterschiede, die noch erläutert werden. Siehe Abschnitt „Daten in ESS-Tabellen bearbeiten“ für weitere Informationen zu den speziellen Verhaltensweisen von ESS-Datensätzen.

ESS Daten-Zwischenspeicherung

Wie bei der Interaktion mit eigenen FileMaker Pro-Tabellen wird FileMaker Pro ESS-Daten lokal zwischenspeichern, um eine bessere Reaktionsfähigkeit zu gewährleisten und die Anzahl der Anfragen an den Server zu reduzieren. Dabei müssen zwei unterschiedliche Arten von zwischengespeicherten Daten berücksichtigt werden.

Die erste ist die Datensatzkarte: die Datenstruktur, in der ESS-Primärschlüsselwerte mit den FileMaker Pro Datensatz-ID Werten gemappt werden. Diese Datenstruktur wird verwendet, um Fragen zu beantworten wie „Wie viele Zeilen existieren in der ESS-Tabelle?“ und „Haben wir diese spezielle Zeile schon gesehen?“ Die zweite ist der Zeilendatenspeicher, der die Daten für Zeilen zwischenspeichert, die bereits vom ESS-Server abgerufen wurden.

Bevor die tatsächlichen Datensatzdaten einer Zeile abgerufen werden können, muss es einen Eintrag für diese Zeile in der Datensatzkarte geben. Wenn Sie sich die SQL-Abfrageprotokolle ansehen, sehen Sie wahrscheinlich eine Mischung von Abfragen: Manche fragen nur den Primärschlüssel der Tabelle ab, um die Datensatzkarte weiter zu füllen, während andere die Primärschlüsselwerte nutzen, um Zeilendaten für Datensätze abzurufen.



Der Datenzwischenspeicher wird bei Bedarf aktualisiert - wenn der Benutzer Anfragen stellt oder Änderungen an Datensätzen durchführt und wenn sich der Datensatz auf dem ESS-Server ändert. Einzelheiten zu diesem Verhalten sind im Folgenden in den entsprechenden Abschnitten dieser Befehle ausgeführt.

Suche in ESS-Tabellen

Über die Indizierung

Eine effiziente Datenbanksuche ist nur über Indizes möglich. (Denken Sie an die Schwierigkeiten, ein Wort in einem Lexikon zu finden, wenn die Wörter in zufälliger Reihenfolge stehen würden, anstatt dass sie sortiert und mit Reitern indiziert wären). Bei der Arbeit mit FileMaker Pro-eigenen Tabellen müssen sich Entwickler normalerweise keine Gedanken um Indizes machen. In den meisten Fällen wird FileMaker Pro, wenn nicht explizit in einer Felddefinition angegeben, Indizes auf Feldebene selbst erstellen, wenn diese notwendig werden (beispielsweise wenn ein Feld als Matchfeld in einer Beziehung verwendet wird oder wenn ein Benutzer eine Suche innerhalb eines Feldes durchführt).

Dasselbe gilt nicht für SQL-basierte Datenquellen, mit denen wir über ESS reden. Diese Datenquellen führen keine Bedarfsindizierung durch, sondern verlassen sich auf die Existenz von Indizes, die von einem DBA erzeugt wurden. Bestimmte Typen von Indizes werden mehr oder weniger automatisch in einer SQL-Datenquelle erstellt, wenn ein Feld zum Beispiel als eindeutig ausgewiesen oder als Primärschlüssel gekennzeichnet wird. Die meisten Felder, die nicht als Schlüssel welcher Art auch immer behandelt werden, werden auch nicht indiziert, wenn nicht ein DBA speziell darum gebeten wurde. Die Indizierung wird bei SQL-Tabellen im Allgemeinen vorsichtig durchgeführt, da ein Index für eine große Tabelle viel Speicherplatz erfordern kann.

Die Suche über ein nicht indiziertes Feld per ESS wird wahrscheinlich einen Leistungsausfall verursachen, wie dies bei FileMaker Pro geschehen würde, was wenn möglich zu vermeiden ist.

Wie FileMaker Pro eine ESS-Tabelle durchsucht

Wenn ein Benutzer eine Suche durchführt, die eine ESS-Tabelle beinhaltet, wird FileMaker Pro so weit wie möglich versuchen, den SQL-basierten Backend-Server den Großteil der Sucharbeit machen zu lassen. Die Alternative wäre, alle relevanten Datensätze in FileMaker Pro abzurufen und die Suche innerhalb von FileMaker durchzuführen, was offensichtlich keine gute Leistung ergeben würde: Es ist besser, den Server die passenden fünfzig aus zehntausend Datensätzen auswählen zu lassen, als die gesamten zehntausend Datensätze in FileMaker Pro zu übernehmen und dann FileMaker Pro die Arbeit tun zu lassen.

FileMaker Pro-Suchanfragen können im Allgemeinen mit mindestens kleinen Erfolgen in SQL-Abfragen übersetzt werden. Es kann sein, dass eine komplexe Suchoperation in FileMaker keine entsprechende SQL-Anfrage erzeugen kann. Wenn dies der Fall ist, wird FileMaker Pro die Ergebnismenge vom Server in Stapeln von ungefähr 1000 Datensätzen abfragen und den Rest der Suche selbst durchführen. (Da FileMaker Pro niemals Indizes in ESS-Tabellen erzeugt oder speichert, wird dieser abschließende Suchdurchlauf als linearer Durchgang durch die SQL-Ergebnismenge genau wie bei der Suche in einem nicht gespeicherten Feld oder einer Formel durchgeführt).

Wenn FileMaker Pro eine Suche durchführt, werden im Allgemeinen zunächst nur die Primärschlüsseldaten abgerufen. Sobald eine Liste der Primärschlüssel für die Daten abgerufen wird, die der Anfrage entsprechen, werden die zusätzlichen Spalten nach Bedarf



abgerufen. Dies ist eine der vielen Möglichkeiten, mit denen FileMaker Pro versucht, die Anzahl der auf einmal abgerufenen Daten von einer ESS-Quelle zu beschränken.

Eine erfolgreiche Suche in einer Tabelle löscht den Datenzwischenspeicher für diese Tabelle und füllt diesen neu mit den Ergebnissen der Suche.

Optimierung von ESS-Suchläufen

Von Zeit zu Zeit werden Sie bemerken, dass eine ESS-Tabelle nur langsam abgerufen werden kann, und Sie fragen sich vielleicht warum. Obwohl die ESS-Funktionalität Entwicklern keine Möglichkeit bietet, SQL-Anfragen direkt zu schreiben oder zu optimieren, finden Sie hier einige Informationen dazu, welche Art von SQL-Abfragen von welchen FileMaker Pro-Anfragen erzeugt werden.

Die beste Möglichkeit, etwas über die Arten von Anfragen zu lernen, die die FileMaker Pro ESS-Funktion erzeugt, ist die Anfrageprotokolle des Datenbankservers anzusehen, mit dem Sie arbeiten. Die Protokollierungstechniken variieren je nach unterstützter ESS-Datenquelle und werden in diesem Artikel nicht vollständig analysiert.

Als Beispiel für eine "Optimierung durch Experimentieren" betrachten wir eine MySQL-basierte Tabelle, die Daten zu drei Millionen Städten der Welt enthält. Die Tabelle hat eine Spalte **Region**, die eine **Text**-Spalte ist und deren Werte „00“, „01“ usw. lauten. Angenommen Sie haben FileMaker Pro-Benutzern erlaubt, dieses Feld zu durchsuchen, diese beschwerten sich jedoch, dass die Suche langsam läuft. Sie wissen, dass das Feld **Region** in der MySQL-Tabelle indiziert ist, daher ist das Problem nicht einfach nur ein fehlender Index.

Wenn Sie das MySQL-Abfrageprotokoll öffnen, sehen Sie, dass Benutzersuchen in FileMaker Pro eine Anfrage auslösen, die ähnlich der Folgenden aufgebaut sind:

```
SELECT id_stadt FROM staedte WHERE region LIKE '%00%'
```

Das Problem dabei ist, dass FileMaker Pro MySQL aufgefordert hat, eine Wildcard-basierte Suche durchzuführen. (Dies wird durch die '%'-Zeichen angezeigt). Leider kann eine solche Suche nicht den MySQL-Index verwenden.

Sie entscheiden sich dafür, in FileMaker Pro eine andere Suche zu verwenden, in der Sie die Region in Anführungszeichen einfügen (dies zeigt eine Suche nach einem ganzen Wort in FileMaker Pro an). Die Überprüfung des MySQL-Anfrageprotokolls ergibt:

```
SELECT id_stadt FROM staedte WHERE region LIKE '00%'
```

Dies ist viel besser: Es gibt keine Wildcard am Anfang, daher wird der MySQL-Index verwendet, was das größte Problem war. Diese Abfrage läuft viel schneller als die Vorherigen. Aber das ist immer noch nicht das, was Sie wollten. Sie wollen den Literalwert "00" und keinen anderen suchen. Diese Suche würde 001 und 002 ergeben, wenn dies passende Regionalcodes wären. Versuchen Sie doch dazu, den "genau gleich" Operator (==) von FileMaker Pro zu verwenden und sehen Sie sich die daraus entstehende SQL-Abfrage an:

```
SELECT id_stadt FROM staedte WHERE region LIKE '00'
```

Dies ist perfekt: Der Index wird benutzt und es ist eine Genau-Gleich-Suche. In diesem Fall ist der Unterschied zwischen den letzten beiden Suchanfragen nicht wichtig: Es gibt nur einen Regionalcode beginnend mit "00", daher kann die mittlere Suche keine falschen Daten finden. Der größte Unterschied liegt zwischen der ersten und der zweiten Suche. Sie finden beide dieselben Daten, die zweite Suche nutzt jedoch den MySQL-Index und läuft so viel schneller.



In diesem Fall könnten Sie je nach Erfahrung entweder den Benutzern erklären, dass sie den „=“ Operator bei der Suche nach der Region verwenden sollen, oder Sie können einfach eine Script-Suchroutine erzeugen, die die entsprechenden Operatoren zu dieser oder anderen Suchen hinzufügt.

Relationale Anfragen erzeugen eine andere Art von Suchanfragen. Wenn Sie ein globales Feld erstellen müssten, in das der Benutzer einen Regionalcode eingeben kann und anschließend eine Beziehung mit der MySQL-Städte-tabelle basierend auf dem Feld erzeugt werden soll, könnten Sie sehen, dass die MySQL-Abfragen die folgende Form haben würden:

```
SELECT id_stadt FROM staedte WHERE region IN ('00')
```

Die IN Klausel in SQL erlaubt dabei eine Liste möglicher Suchkriterien, wie

```
SELECT id_stadt FROM staedte WHERE region IN ('00','04','07')
```

FileMaker Pro ESS verwendet die IN Klausel, um die Multikey-Funktionalität von FileMaker Pro zu unterstützen: Die Möglichkeit, eine Beziehung basierend auf einem mehrwertigen Schlüssel zu bilden. Die IN Klausel ist im Allgemeinen sehr effizient, daher sollten relationale Anfragen allgemein effizient strukturiert sein, zumindest wenn die korrekten Indizes existieren. Ad-hoc Suchen von Benutzern sind die einzigen, die Sie sorgfältiger betrachten müssten.

Die Anfragen in diesem Abschnitt dienen nur als Beispiel. Insbesondere ist die Verwendung von MySQL nur beispielhaft. Andere Datenquellen können dieselben Anfragen mehr oder weniger effizient behandeln. FileMaker Pro hat keine Spezifikationen veröffentlicht, welche Art von Anfragen ESS genau erzeugt und die hier gezeigten Anfragen können sich in zukünftigen Versionen der Datenbankserver, der ODBC-Treiber oder natürlich der FileMaker Pro-Produktreihe ändern. Die optimale Strategie ist, sich mit dem Prüfverfahren Ihrer SQL-Anfrageprotokolle vertraut zu machen, um bessere Informationen zu ESS-Verhaltensweisen zu bekommen.

Sortieren in ESS-Tabellen

Es gibt nur wenig zur Sortierung von ESS-Datensätzen zu sagen, da der Mechanismus hier einfacher ist. Die komplette Sortierung von ESS-Daten wird innerhalb von FileMaker Pro durchgeführt. Die Reihenfolge, in der ein SQL-Datenbankserver eine Ergebnismenge sortiert, hängt von dem Zeichensatz der Tabelle bzw. Tabellen ab, die dem Ergebnis zugrunde liegen. In FileMaker Pro werden alle Daten intern als Unicode gespeichert und die Sortierung basiert auf der FileMaker pro Indizierungsauswahl und der im FileMaker Pro-Klient eingestellten Sprache. In unterstützten ESS-Quellen kann der verwendete Zeichensatz von einem Server zum anderen, von einer Datenbank zur anderen, von einer Tabelle zur anderen oder sogar von einer Spalte zu einer anderen innerhalb derselben Tabelle variieren. Die einzige Möglichkeit die einheitliche Sortierung von FileMaker Pro sowohl bei ESS und FileMaker Pro-Daten zu erhalten, ist die Durchführung aller Sortierungsoperationen in FileMaker Pro, nachdem die Daten abgerufen wurden.

Daher wird die gesamte Ergebnismenge in FileMaker Pro geladen (in Gruppen von ungefähr 1.000) und dann sortiert, wenn ein Benutzer ESS-basierte Datensätze überprüft und sich entscheidet, die Ergebnismenge zu sortieren. (Manche Datensätze können bereits im Datenzwischenspeicher vorhanden sein und müssen so evtl. nicht erneut abgerufen werden).

Dies kann bei großen Datensatzmengen zeitaufwendig sein. Ein paar Hundert oder Tausend ESS-Datensätze sollten aber sehr schnell sortiert werden können, nur Datensatzmengen mit mehreren zehner- oder hunderttausend Datensätzen können nur sehr langsam sortiert werden.



ESS im Beziehungsgraph

ESS-Tabellen im Beziehungsgraph verhalten sich grundsätzlich wie eigene FileMaker Pro-Tabellen. ESS-Tabellen können an Beziehungen mit FileMaker Pro-Tabellen teilhaben, natürlich wie immer mit einigen wichtigen Unterschieden.

Alle FileMaker Pro-eigenen Beziehungstypen, einschließlich der Mehrprädikat-Beziehungen und Ungleich-Beziehungen werden unterstützt. Wie beim Suchen ist es möglich, dass eine Beziehungsanfrage nicht vollständig vom ESS-Server verarbeitet wird – in diesem Fall werden eventuell weitere Verarbeitungsschritte der zusammengesetzten Datensätze innerhalb von FileMaker Pro durchgeführt, sobald diese abgerufen werden. ESS-Tabellen können sowohl in Multi-Datensatz-Beziehungen (die Art, bei der man den Scriptbefehl Gehe zu Bezugsdatensatz „basierend auf allen Datensätzen in der Ergebnismenge“) und mehrwertigen Beziehungen (die Art, die entsteht, wenn eines der Matchfelder in der Beziehung mehrere, rückgabegetrennte Werte hat). Es gibt jedoch einen großen Unterschied im Verhalten von mehrwertigen Beziehungen, je nachdem ob die mehrwertigen Daten auf der FileMaker Pro Seite oder auf der ESS-Seite anfallen und ob diese im „parent“ (übergeordneten) oder „child“ (nachgeordnete) Datensatz vom Standpunkt einer speziellen Beziehungsanfrage her vorhanden sind.

Dies soll durch folgendes Beispiel näher erläutert werden: Denken Sie an einen übergeordneten Datensatz (ESS oder FileMaker Pro, hier macht dies keinen Unterschied), der mehrere Werte enthält, wie unten aufgeführt. Für einen gegebenen Wert in einem untergeordneten Datensatz zeigt die folgende Tabelle, ob es eine passende Beziehung gibt oder nicht, je nachdem, ob der nachgeordnete Datensatz ESS-basiert oder FileMaker Pro basiert ist.

Wert des übergeordneten Datensatzes: Rot¶Grün¶Blau

Wert des nachgeordneten Datensatzes	Join erfolgreich, wenn nachgeordnete Tabelle ESS basiert ist	Join erfolgreich, wenn nachgeordnete Tabelle FileMaker Pro basiert ist
Rot	Ja	Ja
Grün	Ja	Ja
Blau	Ja	Ja
Blau¶Rot	Nein	Ja
Blau¶Grün¶Rot	Nein	Ja
Rot¶Grün¶Blau	Nein	Ja
Gelb¶Rot	Nein	Ja
Gelb	Nein	Nein
Blau	Nein	Ja

Wie lässt sich dieses Verhalten erklären? Mit einem rückgabegetrennten übergeordneten Datensatz wird FileMaker Pro immer mehrere Anfragen durchführen, egal ob der übergeordnete Datensatz ESS-basiert oder FileMaker Pro eigen ist. Daher wird FileMaker Pro bei einem übergeordneten Wert von Rot¶Grün¶Blau, einer rückgabegetrennten Liste von



drei Werten, eine SQL-Anfrage bilden, die nach Rot OR (oder) Grün OR (oder) Blau sucht, ähnlich Folgender:

```
SELECT pk FROM multi_match WHERE match_text IN ('Rot','Grün','Blau')
```

Diese Anfrage durchsucht die SQL-Datenbank nach exakten Matches der Werte "Rot", "Grün" und "Blau". Nur nachgeordnete Datensätze mit einem dieser genauen Werte im Matchfeld werden zurückgegeben. Selbst der letzte Eintrag "Blau␣" passt nicht, da das am Ende stehende Carriage Return das exakte Match verhindert.

Die weiteren Einzelheiten des Join-Verhaltens hängen vom in der SQL-Tabelle (wenn überhaupt) vorhandenen Index ab. Der Join kann beispielsweise abhängig oder unabhängig von Groß-/Kleinschreibung sein, je nach Indizierung des Backend. Und manche ESS-Quellen können Indizierungstypen unterstützen (wie Volltextindizierung), die die oben angezeigten Ergebnisse verändern könnten. Der wichtige Punkt ist, dass ESS-Quellen im Allgemeinen bei mehrwertigen Matchfeldern ein anderes Join-Verhalten als FileMaker Pro Tabellen zeigen.

Außerdem ist es beim Aufbau von Beziehungen mit ESS-Tabellen nicht erforderlich, dass das Matchfeld oder die Felder auf der ESS-Seite überhaupt indiziert sind (dies ist ein wichtiger Unterschied zum Verhalten, wenn das Zielmatchfeld in einer FileMaker Pro Tabelle liegt). Exakt dieselben Anfragen werden auf dem Backend erzeugt. Natürlich laufen diese ohne unterstützende Indizes nicht so schnell, als wenn die Felder indiziert wären.

Bearbeiten von Daten in ESS-Tabellen

Die vorherigen Abschnitte haben erläutert, was passiert, wenn eine ESS-Tabelle zum ersten Mal in einer Benutzersitzung geöffnet wird und was während der Suche, dem Sortieren und bei Beziehungsoperationen geschieht. Dieser Abschnitt beschäftigt sich mit dem Mechanismus zum Erstellen, Lesen, Bearbeiten und Löschen von Daten in ESS-Tabellen.

Im Allgemeinen wird FileMaker Pro zur Minimierung der Anzahl Anfragen an die ESS-Quelle versuchen, seine Aktivitäten in einzelne Anfragen zu bündeln. So wird zum Beispiel.

```
SELECT a, b, c, FROM kunden_sql WHERE id_kunde IN (17, 100, 900)
```

dem folgenden vorgezogen

```
SELECT a, b, c, FROM customers_sql WHERE id_customer = 17  
SELECT a, b, c, FROM customers_sql WHERE id_customer = 100  
SELECT a, b, c, FROM customers_sql WHERE id_customer = 900
```

Darüber hinaus wird FileMaker Pro versuchen, so weit wie möglich Datenbanktransaktionen zu verwenden. Datenbanktransaktionen sind eine Befehlsgruppe, die ein vollständiges Zurücknehmen mehrerer Befehle ermöglicht, wenn ein einzelner Befehl in der Befehlsgruppe fehlschlägt. Transaktionen sind nützlich, da so bestimmte FileMaker Pro Befehle wie die Übertragung einer Menge von Bezugsdatensätzen (wie aus einem Ausschnitt) aus der Perspektive des ESS-Servers „gleichzeitig“ erscheinen. Transaktionen ermöglichen auch das Rückgängig machen einer Gruppe von Befehlen, wenn ein Teil des Befehls fehlschlägt (beispielsweise wenn eine Ausschnittzeile von insgesamt fünf übertragenen einen Validierungsfehler auslöst).

Erstellen von Daten

Wenn ein Benutzer einen Datensatz zu einer ESS-Tabelle in FileMaker Pro hinzufügt, wird keine ESS-Aktivität ausgelöst, bis der Benutzer den neuen Datensatz überträgt. Zu diesem Zeitpunkt wird der Datensatz als neuer Datensatz an den ESS-Server gesendet. Wenn die Erstellung des Datensatzes erfolgreich abläuft, löscht FileMaker Pro sofort diesen Datensatz



aus dem Zwischenspeicher und liest ihn neu aus der ESS-Quelle. Der Grund dafür ist, dass es Standardwerte oder automatisch gefüllte Werte auf der ESS-Seite geben kann (wie beispielsweise durch Datenbanktrigger erzeugte Werte), die ein Füllen von leeren Feldern oder ein Überschreiben von übermittelten Feldern verursachen können. Ein Beispiel dafür wäre ein Zeitmarkenfeld, in dem automatisch der Erstellungszeitpunkt auf der Serverseite eingetragen wird oder ein laufendes Nummernfeld, das automatisch vom ESS-Server erzeugt wird.

Es ist möglich, dass beim erneuten Lesen des Datensatzes mehr als ein Datensatz vom Server zurückkommt. Dieser Fall kann eintreten, wenn das/die von FileMaker Pro als Primärschlüssel verwendete Feld(er) nicht wirklich eindeutige Daten enthalten. Erinnern Sie sich daran, dass FileMaker Pro beim ersten Hinzufügen einer ESS-Tabelle zum Beziehungsgraph sein Bestes tut, um einen geeigneten Primärschlüssel zu finden. Wenn aber kein offensichtlich geeigneter Schlüssel gefunden wird, fordert FileMaker Pro den Benutzer oder Entwickler auf, einen solchen Schlüssel auszuwählen. Wenn die gewählte(n) Schlüsselspalte(n) nicht tatsächlich eindeutig sind, ruft FileMaker Pro eventuell mehrere Datensätze ab, wenn er nach dem „neuen“ Datensatz sucht: Alle anderen Datensätze mit demselben Primärschlüssel werden ebenfalls zurückgegeben. Diese Überlegung gilt auch für alle anderen Befehle und unterstreicht die Tatsache, dass besonders darauf geachtet werden muss, dass FileMaker Pro wirklich mit einem tatsächlich eindeutigen Schlüssel in jeder ESS-Tabelle arbeiten kann.

Lesen von Daten

Die vorherigen Abschnitte haben sehr ausführlich behandelt, auf welche Art und Weise FileMaker Pro Daten liest und seine Zeilenkarte und den Datensatzzwischenspeicher aufbaut. Ein Lesen verursacht im Allgemeinen kein Sperren oder andere Störungen auf der SQL-Seite. FileMaker Pro liest Daten in Blöcken, die ausreichend sind, um sowohl die aktuellen Datenanfragen eines Klient zu bedienen als auch um genug Datensätze im Zwischenspeicher zu halten, um den Klient-Aktivitäten 50-100 Datensätze „voraus“ zu sein.

Bearbeiten von Daten

Die meisten Datenbanksysteme verfügen über Prüfmechanismen für den Fall, dass zwei Benutzer versuchen, gleichzeitig ein und denselben Datensatz zu bearbeiten. Eigene FileMaker Pro Tabellen nutzen die pessimistische Sperrung: Wenn ein Benutzer beginnt, einen Datensatz zu bearbeiten, wird dieser als gesperrt gekennzeichnet und andere Benutzer können nicht versuchen, den Datensatz zu bearbeiten, bis der ursprüngliche Benutzer den Datensatz übermittelt oder verworfen hat, was die Sperre freigibt.

Bei Interaktionen mit ESS-Daten verwendet FileMaker Pro ein optimistisches Sperrmodell. Dies bedeutet, dass keine Prüfung durchgeführt wird, wenn ein Benutzer beginnt, einen Datensatz zu bearbeiten. Die Überprüfung findet erst dann statt, wenn der Benutzer den Datensatz übermittelt: Wenn sich die Daten in einem der Felder des Datensatzes geändert haben, seit der Benutzer mit dessen Bearbeitung begonnen hat, erhält der Benutzer eine Warnung und kann wählen, ob er die Daten des anderen Benutzers überschreiben oder seine eigenen Änderungen zurücknehmen will:



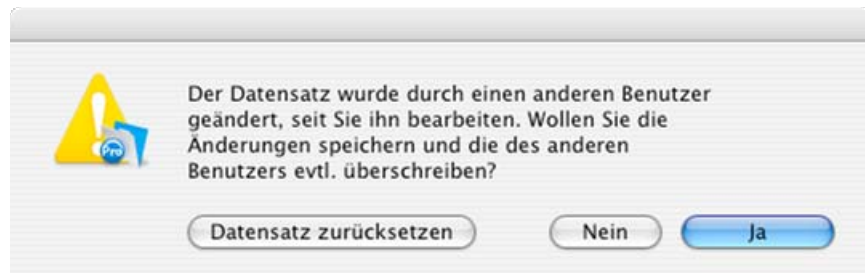


Abbildung 6

Beachten Sie, dass innerhalb von FileMaker Pro die Feststellung, ob der Datensatz überhaupt geändert wurde nur als Vergleich der „neuen“ Daten mit den „ursprünglichen“ Daten in den Schattenfeldern durchgeführt wird. Felder, die in der FileMaker Pro-Schattentabelle nicht als Schattenfelder vorhanden sind, werden bei dieser Prüfung nicht berücksichtigt. (Diese Felder können fehlen, weil deren SQL-Datentypen momentan nicht unterstützt werden oder weil der Entwickler entschieden hat, die Schattenfelddefinitionen zu löschen).

Wie bei der Datensatzerstellung wird der gesamte Datensatz neu in den FileMaker Pro Datenzwischenspeicher abgerufen, wenn eine Datensatzbearbeitung erfolgreich übermittelt wurde; dies trägt allen automatischen Einträgen Rechnung, die eventuell in der ESS-Tabelle in Kraft sind.

Löschen von Daten

Das Löschen eines ESS-Datensatzes ist relativ einfach. Der Datensatz wird aus der ESS-Tabelle und aus dem lokalen Datenzwischenspeicher gelöscht. Zu berücksichtigen ist jedoch, dass Zeilen eine nach der anderen über ihre Primärschlüsselwerte gelöscht werden. Dies bedeutet, dass FileMaker Pro zum Löschen einer Zeile den Wert des Primärschlüssels kennen muss. Aus diesem Grund kann der Befehl „Alle Datensätze löschen...“ sehr zeitaufwendig sein, da FileMaker Pro zuerst alle Primärschlüssel abrufen muss und anschließend die entsprechenden Zeilen eine nach der anderen löscht.

Information über Änderungen

Bei der Arbeit mit eigenen FileMaker Pro Daten ermöglicht die ständige Klient-Server-Verbindung von File Maker Pro eine nahezu direkte und vollständige Benachrichtigung aller verbundenen Benutzer über geänderte Daten. Gelöschte Datensätze verschwinden von den Bildschirmen der anderen Benutzer und Datenänderungen fließen direkt in die Datensätze anderer Benutzer ein. Vieles davon geschieht, weil der FileMaker Pro-Host Benachrichtigungen über Datenänderungen an verbundene Klienten übermittelt.

Bei ESS-Tabellen laufen die Dinge anders. FileMaker Pro verfügt über keine Möglichkeit, Benachrichtigungen von einem ESS-Server zu erhalten, selbst wenn der ESS-Server solche Benachrichtigungen anbietet, was diese aber nicht tun. FileMaker Pro-Klienten können nur von Änderungen an ESS-Daten erfahren, indem Daten vom Server angefordert und diese Serverdaten mit den aktuell in FileMaker Pro zwischengespeicherten Daten verglichen werden. Dies bedeutet, dass ESS-Daten in FileMaker Pro im Allgemeinen das Problem haben, dass sie leicht veraltet sein können, was bei eigenen FileMaker Pro Daten niemals ein Problem ist.

FileMaker Pro kann bestimmte Arten von Änderungen an den ESS-Datensätzen schnell erkennen. Alle Befehle, die Änderungen an der Anzahl der ESS-Datensätze oder an einem ihrer Primärschlüssel durchführen, werden effizient erkannt und in FileMaker Pro wiedergegeben. Eine Erstellung und Löschung von Datensätzen sowie die Änderung des



Primärschlüsselwertes einer Zeile (falls überhaupt möglich) werden relativ schnell erkannt und die erforderlichen Aktualisierungen werden zu FileMaker Pro übertragen.

Datensatzbearbeitungen sind da schon etwas anderes. Wenn ein anderer FileMaker Pro-Benutzer oder ein nicht-FileMaker Pro-Benutzer, Daten in einer ESS-Zeile ändert, werden andere FileMaker Pro-Benutzer, die diese Zeile ansehen, nicht direkt über die Änderungen informiert. Die Änderung wird erst erkannt, wenn FileMaker Pro einen Grund hat, die Zeile aus dem Datenzwischenspeicher zu löschen und neu abzurufen, beispielsweise wenn eine neue Suche durchgeführt wird.

Da die auf dem Bildschirm sichtbaren Daten mit den Serverdaten nicht mehr synchron sein können, bietet FileMaker Pro 9 jetzt den Befehl „Fenster aktualisieren“ im Menü Datensätze an. „Fenster aktualisieren“ verwirft alle angezeigten Datensätze aus dem Datenzwischenspeicher und aktualisiert deren Werte vom ESS-Server. Der Scriptbefehl „Fenster aktualisieren“ verfügt nun auch über eine entsprechende „Flush Cached SQL Data“ („Zwischengespeicherte SQL Daten verwerfen“) Option.

ESS Mapping von Datentypen

Wenn eine ESS-Tabelle zum ersten Mal zum Beziehungsgraph hinzugefügt wird, muss FileMaker Pro entscheiden, wie die in der ESS-Tabelle vorhandenen ESS-Datentypen repräsentiert werden. Für jedes ESS-Feld muss dessen Datentyp mit einem der FileMaker Pro-Kerndatentypen gemappt werden: Text, Number (Zahl), Date (Datum), Time (Zeit) oder Timestamp (Zeitmarke). Gleichzeitig ist es notwendig, alle Unterschiede im Bereich oder Verhalten zwischen den ESS-Datentypen zu berücksichtigen sowie des FileMaker Pro Typs, mit dem diese gemappt werden. Dies wird normalerweise durch Hinzufügen von Validierungsprüfungen zur ESS-Schattenfelddefinition in FileMaker Pro erreicht.

Folgende Tabelle beschreibt die aktuellen Mappings von unterstützten ESS-Datentypen zu FileMaker Pro Datentypen. Beachten Sie, dass einige ESS-Datentypen, insbesondere Binärdaten, momentan nicht unterstützt werden. ESS-Felder mit solchen Datentypen werden nicht gemappt.

FileMaker Pro	Oracle Database	MySQL	MS SQL Server
Text	char nchar varchar2 nvarchar2 clob nclob	char varchar tinytext mediumtext longtext text enum set	char nchar ntext nvarchar text varchar varchar(max) nvarchar(max)
Number	number raw float	boolean int tinyint smallint mediumint integer real double float	Int tinyint smallint bigint float decimal numeric money smallmoney



		decimal numeric year	
Date	date interval_year	date	
Time	interval_day	time	
Timestamp	timestamp	timestamp datetime	datetime smalldatetime
Nicht unterstützt	raw xmltype blob	"blob" data types spatial data types binary varbinary bit	timestamp binary varbinary image

Über MS SQL Server DateTime Felder

Die Arbeit mit Datums- und Zeitangaben ist eine sehr häufige Anforderung von Datenbanksystemen, daher ist es wichtig, hier einige mögliche Probleme mit den MS SQL Server Feldtypen **datetime** und **smalldatetime** zu erläutern. Im Gegensatz zu den anderen unterstützten ESS-Servern hat MS SQL-Server keine separaten reinen Datums- oder Zeit-Datentypen. Es verlässt sich rein auf die **datetime** und **smalldatetime**-Feldtypen, die alle eingegebenen Daten als Zeitmarke mit variierender Präzision speichern. MS SQL-Server akzeptiert reine Datums- oder Zeitdateneinträge, obwohl es die Daten umwandelt und intern als **timestamp** speichert. Momentan mappt FileMaker Pro ESS diese Felder als Zeitmarken-Felder. Dies bedeutet, dass im FileMaker Pro Kontext, reine Datums- oder Zeitdateneinträge *nicht* zulässig sind, da der Timestamp-Feldtyp in FileMaker Pro sowohl ein Datum als auch eine Uhrzeit benötigt, daher sind die FileMaker Pro Validierungsregeln für diesen Feldtyp strenger als die von SQL Server. Falls diese Einschränkung ein Problem für Ihre Anwendung ist, können Sie mit Hilfe eines Scripts den Benutzern die Möglichkeit geben, reine Daten- oder Zeitangaben in diese SQL-Server-Feldtypen einzugeben.

Über Datenbereiche und Validierung

Häufig hat ein ESS-Datentyp einen anderen Bereich zulässiger Werte als der FileMaker Pro-eigene Datentyp, mit dem dieser während ESS-Befehlen gemappt wird. Ein FileMaker Pro Textfeld kann beispielsweise hundert tausende von Zeichen enthalten, aber ein SQL-Feld mit **char(30)** speichert maximal 30 Zeichen. In diesem Fall wird die Beschränkung auf 30 Zeichen dem FileMaker Pro Schattenfeld als Validierungsbeschränkung hinzugefügt:





Abbildung 7

In anderen Fällen kann das ESS-Feld auch einen *größeren* Datenbereich aufnehmen als sein FileMaker Pro Gegenstück. Ein Datumsfeld in MySQL kann zum Beispiel Daten bis zu 12/31/9999 aufnehmen, aber die Datenspeichermöglichkeiten von FileMaker Pro enden mit 12/31/4000. Der Versuch, ein Datum wie 1/1/7000 in das ESS-Feld mit Hilfe von FileMaker Pro einzugeben, löst die Validierungsmeldung von FileMaker Pro aus:

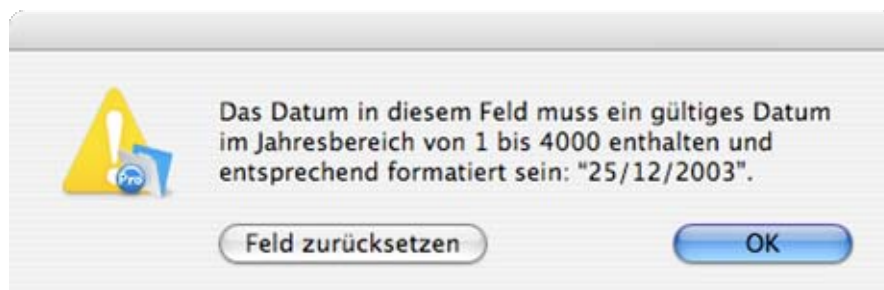


Abbildung 8

Dies ist gut, aber sagen wir einmal, dass es Zeilen mit Daten außerhalb des gültigen Bereichs gibt, die mit einem anderen Programm als FileMaker Pro eingetragen wurden? Diese Zeilen machen FileMaker Pro Probleme: Die Daten außerhalb des Bereichs werden als ? angezeigt und können nicht bearbeitet werden. (Trotzdem können Suchen nach Daten außerhalb des Bereiches eventuell noch funktionieren, da diese an den ESS-Server weitergegeben werden).

In manchen Fällen kann FileMaker Pro alle notwendigen Typ-basierten Validierungen eines Feldes nicht korrekt umsetzen. In diesem Fall werden über FileMaker Pro übermittelte Daten



außerhalb des gültigen Bereiches vom ESS-Server abgefangen, was wahrscheinlich zu einer ODBC-Fehlermeldung ähnlich dieser führt:

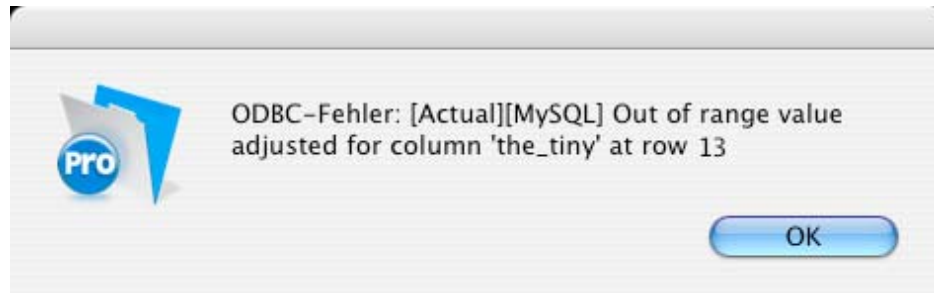


Abbildung 9

Solche Meldungen kommen direkt vom ODBC-Server selbst und benötigen häufig etwas Eingewöhnungszeit, um herauszufinden, was solche Fehler bedeuten und wann diese ausgelöst werden. Im Allgemeinen müssen Sie besonders auf die Feldtypen der ESS-Tabellen achten und sich der Fälle bewusst sein, in denen ESS-Feldtypen mögliche Befehle in FileMaker Pro beschränken und umgekehrt.

Über Views

Views sind ein leistungsstarkes Konzept SQL-basierter Datenbanksysteme. Wenn die Erstellung von Views in einem SQL-System für Sie eine Möglichkeit ist, werden Sie erkennen, dass Views eine leistungsstarke Ergänzung der ESS-Funktionalität bieten können.

Eine View kann als virtuelle Tabelle angesehen werden, die von einer Anfrage erzeugt wird. Angenommen Sie haben eine riesige Kundentabelle und wollen nur die Kunden aus Pennsylvania sehen, die im Jahr 2007 oder später hinzugekommen sind. Sie könnten eine Anfrage ähnlich dieser schreiben:

```
SELECT id_customer, name_first, name_last, join_date FROM customers WHERE  
join_date > '2007-01-01' AND address_state = 'PA'
```

In vielen SQL-Datenbanksystemen können Sie diese Anfrage mit Hilfe des folgenden Statements zu einer View machen

```
CREATE VIEW pa_neue_kunden AS SELECT id_kunde, Vorname, Nachname, beitrtrittsdatum  
FROM kunden WHERE beitrtrittsdatum > '2007-01-01' AND adresse_staat = 'PA'
```

Dies erzeugt eine neue View mit Namen **pa_neue_kunden**, die genauso aussieht und sich verhält wie eine normale Tabelle, nur mit weniger Feldern und Datensätzen als die ursprüngliche Kundentabelle.

Eine "Tabelle" wie diese, ist oft nicht physikalisch existent. Mit einem Satz, es ist einfach eine gespeicherte Anfrage. Aber sie verhält sich in vielen wichtigen Punkten wie eine echte Tabelle. Man kann zum Beispiel Anfragen direkt an die View stellen:

```
SELECT * FROM pa_neue_kunden WHERE Nachname = 'Miller'
```

Was die Datenbank in die folgende Anfrage an die "echte" Tabelle umsetzt:

```
SELECT id_kunde, Vorname, Nachname, beitrtrittsdatum FROM kunden WHERE  
beitritttsdatum > '2007-01-01' AND adresse_staat = 'PA' AND Nachname = 'Miller'
```

Darüber hinaus kann die View Klienten zur Verfügung gestellt werden, die eine Liste der verfügbaren Tabellen abfragen. Dies bedeutet, dass aus FileMaker Pro ESS-Perspektive, eine verfügbare View genauso wie eine Tabelle dem Beziehungsgraph hinzugefügt werden kann.



Hinweis: Um Views per ESS hinzuzufügen, müssen Sie eine Option in Ihrem ODBC-Treiber aktivieren, die die Weiterleitung von Views zusammen mit den Basistabellen ermöglicht. Auf ähnliche Weise sollten Sie das „Views“ Kästchen in der Datenquellendefinition von FileMaker Pro aktivieren:

The screenshot shows the 'Datenquelle bearbeiten' (Edit Data Source) dialog box. The 'Name' field contains 'MS SQL Server Kunden'. The 'Typ' (Type) is set to 'ODBC' (selected with a radio button). The 'DSN' field contains 'ss_kunden'. Under the 'Authentifizierung' (Authentication) section, the option 'Benutzername und Passwort angeben (gilt für alle Benutzer)' is selected. The 'Benutzername' (Username) field contains 'fmp' and the 'Passwort' (Password) field contains '<Zum Bearbeiten klicken>'. Under the 'Tabellen filtern' (Filter Tables) section, there are three empty text boxes for 'Katalogname', 'Schemaname', and 'Tabellenname'. To the right, under 'Nach Typ filtern' (Filter by Type), there are three checkboxes: 'Tabellen' (checked), 'Ansichten' (checked), and 'Systemtabellen' (unchecked). At the bottom, there are 'Abbrechen' (Cancel) and 'OK' buttons.

Abbildung 10

Obwohl die View in der Hinsicht dynamisch ist, dass sie sehr sensibel auf Datenänderungen reagiert und neue Kunden aus Pennsylvania anzeigt, sobald diese in das System eingegeben werden, ist sie auch etwas starr in dem Sinn, dass die Suchkriterien nicht einfach geändert werden können. Wenn Sie stattdessen die Kunden aus Utah sehen wollen, benötigen Sie eine andere View.

Der große Vorteil bei der Verwendung von Views auf diese Weise ist, dass sie damit große Tabellen „filtern“ können, bevor Sie mit diesen in FileMaker Pro arbeiten – im wahrsten Sinne des Wortes ein optimiertes Verfahren.

Schreibfähige Views

Bei vielen Datenquellen können Views auch aktualisiert werden. Ob dies der Fall ist, hängt jedoch von der Datenquelle ab. Alle unterstützten ESS-Quellen unterstützen auch aktualisierbare Views, aber die Mechanismen unterscheiden sich je nach Datenquelle und



können manchmal komplexen Einschränkungen unterliegen. Wenn schreibfähige Views in Ihrer Umgebung möglich sind, sollten Sie sich damit vertraut machen, wie das von Ihnen gewählte ESS-Produkt dies implementieren kann.

Views und Primärschlüssel

Wie richtige Tabellen unterliegen per ESS verwendete Views der Einschränkung, dass die View eine eindeutige Spalte haben muss, die als Primärschlüssel verwendet werden kann. Und wie bei richtigen Tabellen kann ein Entwickler diesen Schlüssel direkt festlegen. Im Gegensatz zu den meisten Basistabellen kann eine View aber auch einen nicht sorgfältig definierten Primärschlüssel enthalten, wobei Sie in diesem Fall eine Kombination einer oder mehrerer Spalten als Schlüssel auswählen müssen.

Komplexe Views

Views sind eine hervorragende Möglichkeit, große Ergebnismengen zu filtern, damit diese eine besser verwaltbare Größe bekommen und einfacher in FileMaker Pro verwendet werden können. Sie können ebenfalls verwendet werden, um Befehle auszuführen, die innerhalb von FileMaker Pro schwierig zu realisieren oder sehr zeitaufwendig wären.

Zusammenfassende Berichterstellung ist ein Beispiel. Angenommen Sie haben eine sehr große Kundendatenbank in einer ESS-Tabelle und wollen eine Zusammenfassung der Kunden gruppiert nach Staat, Region und Land sehen. Wir könnten die entsprechenden Ergebnisfelder in FileMaker Pro erstellen, aber die Erstellung des Berichtes würde das Laden aller Datensätze aus der ESS-Tabelle erfordern – wahrscheinlich keine gute Wahl.

Eine einzelne SQL-Anfrage kann jedoch etwas Ähnliches erreichen:

```
SELECT land, region, staat, count(*) FROM kunden GROUP BY land, region, staat
```

Diese Anfrage gibt eine Zeile pro unterschiedlichem Staat mit einer Zählung der Städte in dem jeweiligen Staat aus. Wenn diese Anfrage zur Erstellung einer View verwendet wird, hat die daraus resultierende Tabelle nur ein paar Tausend Datensätze zusammengefasst auf Staatenebene. Die verbleibenden Zusammenfassungen (auf Regional- und Landesebene) können leicht in FileMaker Pro erledigt werden. Für Operationen wie Durchschnittswerte oder Standardabweichungen berechnet dieser "Aggregat-der-Aggregate"-Ansatz keine korrekten Werte und es wäre ein anderer Ansatz erforderlich).

Diese Anfrage muss trotzdem noch ordnungsgemäß auf ESS-Seite optimiert werden, um schnell abzulaufen. Sie benötigt wahrscheinlich Indizes für Länder, Regionen und Staaten und kann trotzdem noch langsam ablaufen, wenn sehr große Datensatzzahlen zusammengefasst werden müssen. In vielen Szenarien kann diese Art von Operation jedoch weitaus effizienter auf dem ESS-Server durchgeführt werden.

Es gibt noch einen weiteren wichtigen Punkt zu komplexen Views. Aufgrund der Art und Weise, wie FileMaker Pro ESS-Datensätze in Stapeln anfragt, muss FileMaker Pro möglicherweise mehrere Anfragen stellen, um alle Datensätze aus der View abzurufen. Wenn dies der Fall ist, wird die View mehrere Male ausgeführt und wenn die View eine komplexe Anfrage darstellt, die eine bestimmte Laufzeit benötigt, kann dieser zeitaufwendige Vorgang mehrmals hintereinander ablaufen. Auf der anderen Seite kann dieses Verhalten auch verringert werden, wenn der ESS-Server einen Anfragezwischenspeicher nutzt, wie es manche unterstützte ESS-Server können. Oder Sie könnten die Möglichkeit haben, mit einer "materialisierten View" (oder "gespeicherten View") zu arbeiten, die in Wahrheit eine gespeicherte Tabelle ist, die aktualisiert wird, wenn die Tabelle bearbeitet wird. Bei großen Installationen kann eine solche Art von View die Leistung von Aggregatanfragen sehr verbessern.



Diese Diskussion soll einen Punkt unterstreichen, der später noch aufgegriffen wird: Es ist gut, so viel wie möglich darüber zu lernen, wie sich Ihr spezielles ESS verhält.

Verwaltung der ODBC-Verbindungen

Bei einem normalen Ablauf werden die Verbindungen mit speziellen ODBC-Quellen automatisch hergestellt, wenn ein Benutzer mit ESS-basierten Daten arbeiten muss. Es ist jedoch möglich, dass seine Verbindung zu einer ODBC-Quelle manchmal nicht wie erwartet durchgeführt wird oder dass ein Entwickler eine absichtliche Trennung von einer ODBC-Quelle durchführen möchte.

In FileMaker Pro 9 wurden die Scriptbefehle "Datei öffnen" und "Datei schließen" erweitert, um auch mit ODBC-Datenquellen zu funktionieren. "Datei öffnen" versucht, wenn es für eine ODBC-Datenquelle aufgerufen wird, eine neue Verbindung mit der Datenquelle herzustellen, selbst wenn es momentan keine offene Verbindung gibt. "Datei schließen" versucht die Verbindung zu der Datenquelle zu schließen. Nach jedem dieser Schritte verhält sich die Datenquelle wie normal: Wenn eine Verbindung geöffnet wird, steht die vollständige ESS-Funktionalität zur Verfügung und wenn die Verbindung geschlossen wird, löst die nächste Anfrage von Daten aus dieser Quelle einen neuen Verbindungsversuch aus.

Diese zusätzliche Funktionalität macht es möglich, eine Verbindung zu einer ODBC-Quelle zu schließen und einen Benutzer aufzufordern, sich mit anderen Zugangsdaten analog zu dem Scriptbefehl „Erneut anmelden“ (der nur bei FileMaker Pro eigenen Quellen funktioniert) anzumelden. Damit dies funktioniert, muss die Datenquelle so konfiguriert sein, dass sie nach Zugangsdaten fragt, statt diese zu speichern.

Es ist eine gute Idee, das Schließen einer ESS-Verbindung zu vermeiden, wenn das aktuelle Layout auf der Datenquelle basiert. Am besten finden Sie eine Möglichkeit, zuerst zu einem anderen Layout zu navigieren oder der Benutzer erlebt ein merkwürdiges Verhalten, trotz dessen der Befehl erfolgreich sein sollte.

Sicherheit und Authentifizierung

Berechtigungen und Zugangsdaten innerhalb einer SQL-Umgebung funktionieren im Allgemeinen auf ähnliche Weise wie in FileMaker Pro. Ein Datenbankadministrator kann ein oder mehrere Benutzerkonten für eine ESS-Datenbank definieren, die alle ihr eigenes Passwort haben. Jedes Konto kann über breit gefächerte oder rudimentäre Berechtigungen bis zu einzelnen Datenbank-, Tabellen- oder Feldberechtigungen verfügen, die unterschiedlichste Arten von Operationen wie Lesen, Bearbeiten und Löschen umfassen.

Wie in FileMaker Pro kann ein SQL DBA unterschiedliche Strategien bei der Kontoverwaltung verfolgen:

- Ein Konto je individueller Benutzer
- Ein Konto je Gruppe oder Benutzertyp
- Ein Konto je breit gefasster Zugriffsklasse mit anderweitiger Kontrolle der zugrunde liegenden Berechtigungen (eine Datenbank, die einer Webanwendung zuträgt, kann beispielsweise einen einzelnen „Web“-Benutzer definieren und feinere Berechtigungen der Webanwendung selbst überlassen)

Die ESS-Implementation von FileMaker Pro unterscheidet nicht zwischen diesen Szenarien und unterstützt deren Umsetzung gleichermaßen.



Ein FileMaker Pro Entwickler, der ESS verwendet, hat zwei Wahlmöglichkeiten, wie Benutzerzugriffsdaten gespeichert und verwaltet werden:

1. Speicherung eines einzelnen Benutzernamens und Passwortes in der Definition der ODBC-Datenquelle. Diese Zugangsdaten werden für alle Verbindungen über den speziellen DSN verwendet, den die Datenquelle nutzt. Die Benutzer werden nicht nach Zugangsdaten gefragt.
2. Keine Speicherung der Zugangsdaten in der Datenquellendefinition: Dabei werden die Benutzer beim ersten Zugriff auf die ESS-Datenquelle während einer aktuellen Benutzersitzung nach den Zugangsdaten gefragt.

Wenn der Administrator des ESS-Servers nur einen einzelnen Satz Zugangsdaten für alle FileMaker Pro Benutzer verwalten möchte, wäre die erste Option eine angemessene Wahl. Wenn die Datenbank über einzelne Benutzerkonten verwaltet wird, wäre die zweite Option besser.

FileMaker Pro gibt die FileMaker Pro-Zugangsdaten des aktuellen Benutzers nicht an die ESS-Quelle weiter, daher gibt es keine Möglichkeit für eine „Einzelanmeldung“. Es kann Umstände geben, bei denen etwas sehr Ähnliches erreicht werden kann: Wenn beispielsweise sowohl FileMaker Pro als auch der ESS-Server auf dieselbe Authentifizierungsquelle zugreifen (z.B. Active Directory oder Windows Authentication for SQL Server). Hier können die Zugangsdaten der Benutzer identisch sein, aber wenn die zweite Option gewählt wurde, müssen Benutzer immer noch mindestens einmal pro Sitzung Ihre ESS-Zugangsdaten eingeben.

Der wichtige Punkt zu Zugangsdaten ist, dass die Verwaltung derselben mit ESS außerordentlich einfach sein kann. Der einfachste Fall ist der, in dem es einen einzelnen Satz Zugangsdaten für alle FileMaker Pro ESS-Benutzer gibt und diese Zugangsdaten sind in der FileMaker Pro Datenquellendefinition gespeichert.

Über Verschlüsselung

Bevor wir zur Verschlüsselung hinsichtlich ESS kommen, betrachten wir zunächst die Rolle, die Verschlüsselung bei der normalen FileMaker Pro Kommunikation spielt.

FileMaker Pro Passwörter werden innerhalb der FileMaker Pro Dateien mit Hilfe einer Einwegverschlüsselung gespeichert. Einmal verschlüsselt, kann ein Passwort anschließend niemals wieder entschlüsselt werden. Um die Zugangsdaten zu überprüfen, wird das Passwort, das ein Benutzer eingibt, mit derselben Methode verschlüsselt und anschließend mit dem gespeicherten verschlüsselten Passwort verglichen. Wenn beide übereinstimmen, wird der Benutzer authentifiziert. Diese Einwegverschlüsselung ist sehr schwer zu knacken und kann nicht umgekehrt werden.

Es ist auch möglich, den gesamten Datenstrom zwischen einem FileMaker Pro Klient und dem FileMaker Pro Server zu verschlüsseln. Dies wird durch die Aktivierung von SSL (Secure Socket Layer) Verschlüsselung auf Serverseite erreicht. In diesem Fall werden alle Daten verschlüsselt, die zwischen dem Klient und Server ausgetauscht werden. Dies ist nur bei Einsatz von FileMaker Server möglich.

Bei der Arbeit mit ESS gelten einige Einschränkungen. Wenn ESS-Zugangsdaten innerhalb der Definition einer ODBC-Datenquelle in FileMaker Pro gespeichert sind, ist es nicht möglich, die sehr starke Einwegverschlüsselung auf diese Zugangsdaten anzuwenden, da diese Zugangsdaten zu einem Zeitpunkt vor der Übergabe an den ODBC-Treiber wieder entschlüsselt werden müssten. Daher unterliegen diese gespeicherten Zugangsdaten nicht demselben hohen Einwegsicherheit, wie dies bei FileMaker Pro Passwörtern der Fall ist.



SSL kann ebenfalls bestimmte Einschränkungen bei der Arbeit mit ESS haben. Wenn eine Lösung mit Hilfe von FileMaker Server verwaltet wird und SSL auf dem Server aktiviert ist, wird der gesamte Datenverkehr zwischen FileMaker pro Klienten und dem Server per SSL verschlüsselt. Dieser Schutz erstreckt sich nicht notwendiger Weise auf die Verbindung zwischen FileMaker Server und einem ESS-Server. Die Erzeugung eines SSL-Tunnels aus FileMaker Server zu einem ESS-Server würde eine SSL-Unterstützung im ODBC-Treiber erfordern oder mit anderen Worten die Erstellung eines Point-to-Point SSL Tunnels.

Fehlerbehandlung

In Übereinstimmung mit dem Entwurfsziel, ESS-Daten soweit wie möglich in eine FileMaker Pro Umgebung zu integrieren, wurde die ESS-Funktionalität entwickelt, um Benutzer wo immer möglich vor SQL-basierten Fehlern zu schützen. Es war auch gewünscht, das Hinzufügen neuer Fehlercodes auf der FileMaker Pro Seite allein auf die Behandlung von ODBC-Fehlern zu beschränken. Viele der während ESS-Operationen auftretenden Fehler werden daher auf existierende FileMaker Pro Fehlerzustände abgebildet. Alle Validierungsfehler, die FileMaker Pro abfängt, werden beispielsweise auf dieselbe Art dargestellt, wie reguläre FileMaker Pro Validierungsfehler. Das Nichtauffinden einer angegebenen Tabelle in einer ESS-Quelle würde zum Beispiel einen Fehlercode 106 („Tabelle fehlt“) auslösen.

Es gibt wenige neue Fehlercodes, die beim ODBC-Betrieb auftreten können:

Fehlercode	Fehlermeldung
117	Angegebene Datenquelle nicht vorhanden
416	Primärschlüssel der Datenquellentabelle erforderlich
417	Datenbank unterstützt keine ODBC-Operationen
512	Datensatz wurde bereits von einem anderen Benutzer verändert
513	Datensatz muss zur Erstellung mindestens in einem Feld einen Wert aufweisen
1409	Fehler (ODBC)
1413	Kommunikationsverbindung fehlgeschlagen (ODBC)

FileMaker Pro 9 fügt der Funktion „Hole(LetztenODBCFehler)“ eine neue Funktionalität hinzu. In FileMaker Pro 9 gibt „Hole(LetztenODBCFehler)“ den Fehlerstring des ODBC-Treibers aus, wenn „Hole(LetzteFehlerNr)“ den Code 1408 „Erweiterter Fehler (ODBC)“ gemeldet hat. „Hole(LetzteFehlerNr)“ gibt nur dann 1409 aus, wenn der ODBC-Treiber einen Fehler meldet, aber keinen speziellen Fehlerstring zurückliefert.

Die Behandlung von ODBC-Fehlern setzt dieselben Überlegungen voraus, wie sie bei regulären FileMaker Pro Fehlern auszuführen sind. Es ist möglich, die Fehler mit Hilfe des Scriptbefehls „Fehleraufzeichnung setzen“ [Ein] zu verbergen. In diesem Fall ist es notwendig, selbst auf Fehler zu prüfen und diese mit Hilfe von „Hole(LetzteFehlerNr)“ und „Hole(LetztenODBCFehler)“ zu beheben. Es ist auch möglich, die Fehler in FileMaker Pro direkt dem Benutzer anzeigen zu lassen, in diesem Fall erhält der Benutzer direkt die Fehlermeldungen der ODBC Datenquelle.



Weitere Beschränkungen von ESS

Obwohl die ESS-Funktionalität entwickelt wurde, um eine so weit wie möglich gehende Einbindung nicht-eigener Datentabellen in die FileMaker Pro Umgebung zu ermöglichen, gibt es einige unvermeidbare Unterschiede in der Funktionsweise solcher Tabellen. Dieser Abschnitt führt die prinzipiellen Beschränkungen von ESS-Tabellen im Vergleich zu eigenen FileMaker Pro Tabellen auf.

Binärdaten

Die unterstützten ESS-Quellen unterstützen wie viele SQL-Datenbankserver auch verschiedenste Binärdatentypen. Binärdaten sind Daten, die als reiner Datenstrom behandelt und nicht als Text, Zahl oder Datenwert interpretiert werden. Das Container-Feld von FileMaker Pro ist ein Beispiel für einen Binärdatentyp, der in FileMaker Pro eingebaut ist. Binäre Datenfelder können zur Speicherung von Bildern, Klängen oder anderen solchen Daten verwendet werden.

Binärdaten werden momentan nicht von ESS unterstützt und binäre Datenspalten in ESS-Tabellen werden nicht in das FileMaker Pro-Schattenschema übernommen. Obwohl FileMaker Pro bei Daten in FileMaker Pro Container-Feldern verstehen kann, welche Art von Daten dargestellt wird (Bild, Film, Tabellenkalkulation), hat FileMaker Pro im Fall von Binärdaten einer ESS-Quelle keine Möglichkeit festzustellen, was die Binärdaten darstellen und daher weiß das Programm nicht, wie diese Daten dem Benutzer dargestellt werden sollen.

Diese Einschränkung gilt auch für den Timestamp-Datentyp in SQL-Server, der ein Binärdatentyp ist.

Rich Text

FileMaker Pro Textfelder ermöglichen die Eingabe von formatiertem Text, der Formatierungsinformationen zusätzlich zu den reinen Textdaten enthält. Dies ist in ESS-Tabellen nicht möglich, da ESS-Tabellen die Speicherung formatierter Informationen gemeinsam mit Text nicht unterstützen. Textformatierungsoptionen sind für ESS-Felder deaktiviert.

Blättern in großen Datensatzmengen

Die Blätter Funktion, wie sie beispielsweise bei Verwendung des Buch-Symbols im Statusbereich oder mit Hilfe der Scroll-Leiste in einer Listen-Ansicht ausgelöst wird, kann in großen Datensatzmengen problematisch sein. Der Grund dafür ist, dass FileMaker Pro nicht einfach zu den gewünschten Datensätzen "springen" kann, sondern alle Datensätze zwischen dem aktuellen und dem Blätter-Sprungpunkt abrufen muss. Bei großen Datensatzmengen führt dies zur Übertragung von großen Datenmengen zwischen dem ESS-Server und dem FileMaker Pro-Klient, was eine Verlangsamung zur Folge hat.

Vorgänger/Nachfolger Lookup

Die meisten Auto-Eintrags-Optionen basierend auf einer Beziehung, wie bei Lookups oder automatisch gefüllten Bezugsfeldern, funktionieren weiterhin, wenn die Bezugstabelle ESS-basiert ist. Eine Ausnahme ist die Möglichkeit, den nächst höheren oder nächst niedrigeren Wert nachzusehen, wenn keine Bezugsentsprechung gefunden wird, was nicht funktioniert, wenn die Bezugstabelle ESS-basiert ist.



Aus Index einfügen

FileMaker Pro hat keinen Zugriff auf die Indizes, die auf dem ESS-Server gespeichert sind, daher ist die Funktion „Aus Index einfügen“ nicht für ESS-Felder verfügbar.

Auto-Vervollständigen

Die Fähigkeit, automatisch vervollständigte Optionen basierend auf zuvor eingegebenen Werten für ein Feld anzuzeigen, basiert auf Indizes und kann daher nicht als Anzeigeeption für ESS-Felder gewählt werden.

Tabellenbasierte Wertelisten

Wertelisten können nicht auf Daten einer ESS-Tabelle basieren. Der Grund dafür ist konzeptionell ähnlich dem Grund dafür, dass „Aus Index einfügen“, „Auto-Vervollständigen“ und „Vorgänger/Nachfolger Lookup“ nicht funktionieren: FileMaker Pro müsste auf die ESS-Indizes zugreifen können, um diese Operation effizient durchführen zu können.

Scriptverweise auf ESS-Felder

Wenn Ihr Script auf Felder einer ESS-Tabelle verweist (beispielsweise über die Scriptbefehle „Gehe zu Feld“ oder „Feldwert setzen“) können bestimmte Operationen dazu führen, dass diese Verweise veraltet sind. In einem typischen Szenario wird das ESS-Schema auf eine beliebige Art und Weise verändert und dann mit der FileMaker Pro-Schattentabelle neu synchronisiert. Wenn Sie beispielsweise ein oder mehrere Felder aus dem FileMaker Pro-Schattenschema löschen wollen, einige Felder zu der ESS-Tabelle hinzufügen möchten und dann neu synchronisieren, können Ihre Scriptbefehle nicht länger auf den Feldern operieren, die Sie beabsichtigt hatten.

Der Grund dafür ist: Wenn FileMaker Pro eine Schattenfelddefinition erstellt, entweder wenn eine ESS-Tabelle zum ersten Mal zum Beziehungsgraph hinzugefügt oder wenn das Schattenschema neu mit dem ESS-Schema synchronisiert wird, weist FileMaker Pro jedem ESS-Feld einen eindeutigen Schlüssel zu. Wie bei FileMaker Pro-Feldern wird dieser eindeutige Schlüssel und nicht der Feldname für Verweise auf dieses Feld verwendet. Beim Verhalten normaler FileMaker Pro-Felder ist nun anders, dass der eindeutige Schlüssel dieses Felds neu verwendet werden kann, wenn das Schattenfeld in FileMaker Pro (nicht das entsprechende Feld in der ESS-Tabelle) gelöscht wird. Wenn die Schemata nie wieder synchronisiert werden, kann der Schlüssel des gelöschten Feldes neu für ein anderes Feld auf der ESS-Seite verwendet werden und Scriptbefehle, die auf das gelöschte Schattenfeld verwiesen haben, zeigen nun zu dem neu zugewiesenen ESS-Feld.

Hier unterscheidet sich das Verhalten, das Sie beim Löschen eines Filemaker Pro Feldes erwarten würden. In diesem Fall würde betroffene Scripts, eine <Feld fehlt> Warnung auslösen. In diesem Szenario könnte ein ESS-Feld ohne Warnung durch ein anderes ersetzt werden.

Die zwei optimalen Verhaltensweisen lassen sich an diesem Punkt ableiten: Erstens kann es immer eine gute Idee sein, Script-Kommentare zu verwenden, um zu notieren, mit welchen ESS-Feldern ein Scriptbefehl zu arbeiten versucht; und zweitens ist es immer eine gute Idee, Ihre ESS-Integration nach der Neusynchronisierung Ihres Schemas sorgfältig zu prüfen, besonders wenn Sie Änderungen am Schattenschema und am ESS-Schema durchgeführt haben.



Verwalten von ESS-Feldern in Custom Web Publishing

ESS-Felder verhalten sich nicht wie erwartet bei bestimmten Custom Web Publishing (CWP) Befehlen. CWP bezeichnet hier speziell den Webzugriff mit Hilfe von XML, XSLT und PHP Technologien, die in FileMaker Server Advanced integriert sind.

Die Suche und Anzeige von Daten aus ESS-Quellen ist bei allen Arten von Custom Web Publishing möglich. (Dies kann ein relativ kostenintensiver Schritt sein, da FileMaker Pro den Verbindungskontext jedes Mal neu aufbaut, wenn die Seite angezeigt wird, einschließlich der Zeilenkarte, dem lokalen Datensatzzwischenspeicher und anderen Parametern). Aber andere Aktionen, die Daten verändern, einschließlich dem Löschen und Bearbeiten von Datensätzen, sind nicht zuverlässig. Diese Operationen verlassen sich darauf, dass FileMaker Pro Datensätze über Ihre FileMaker Pro Datensatz-IDs identifiziert werden können. Eigene FileMaker Pro-Datensätze erhalten bei der Erstellung eine dauerhafte Datensatz-ID und diese Datensatz-ID bleibt auch gültig, wenn mehrere Zugriffe auf den Datensatz erfolgen. Aber ESS-Datensätze werden jedes Mal dynamisch mit FileMaker Pro Datensatz-IDs gemappt, wenn sie geladen werden, und daher ist eine ESS Datensatz-ID in mehreren Benutzersitzungen nicht identisch. Bei Zugriff über XML oder PHP kann ein ESS-Datensatz in einem Moment eine Datensatz-ID aufweisen und im nächsten Moment eine andere, da aufeinander folgende Zugriffe auf die Daten auch aufeinander folgende Sitzungen erzeugen, was eine potenzielle Neuzuweisung von ESS Datensatz-IDs verursacht. Dieses erneute Mapping ändert aber nicht zwingender Weise die erscheinende Datensatz-ID – ein Datensatz kann bei mehreren Zugriffen nacheinander dieselbe ID aufweisen, jedoch kann man sich darauf nicht verlassen. Versuche, die ESS-Datensätze über ihre Datensatz-IDs zu verändern, können so zu falschen Ergebnissen oder wirklichen Fehlern führen.

Diese Einschränkung gilt nicht für Webtechnologien, die über andauernde Datenbanksitzungen auf ESS-Daten zugreifen, wie bei Instant Web Publishing oder Custom Web Publishing/XSLT mit Hilfe von Datenbanksitzungen.

Einige ESS-Verwendungsszenarien

In diesem Abschnitt werden einige der wahrscheinlichsten Szenarien für ESS besprochen und einige der Überlegungen, die in jedem Fall notwendig sein können. Dies sind natürlich nicht die einzigen Fälle, in denen es Sinn ergeben kann, ESS zu verwenden, jedoch sollten dies einige der am häufigsten auftretenden Szenarien sein.

Alle diese Fälle sind komplett hypothetisch und dienen nur zur Veranschaulichung einer Reihe verschiedener Einsatzmöglichkeiten.

1. Open Source e-Commerce Lösung

Eine Organisation nutzt FileMaker Pro für das Kundenmanagement und betreibt einen Webshop, der eine MySQL-basierte e-Commerce Lösung einsetzt. Die Firma verarbeitet ungefähr 200 Bestellungen pro Monat und jede Bestellung besteht im Durchschnitt aus zwei Artikelzeilen, daher erzeugt das System ungefähr 2.400 Bestellungen und 4.800 Artikelbestellungen jährlich.

Dies sind nur kleine Datensatzmengen. Als Entwickler könnten Sie diese My-SQL-basierenden Tabelle mehr oder weniger direkt in Ihren Beziehungsgraph einbinden und sie wie reguläre FileMaker Pro-Tabellen behandeln. Der wichtige Punkt hierbei ist, wie bei den meisten Integrationsszenarien, dass es wahrscheinlich sicherer ist, die Einträge nur zu lesen anstatt sie auch zu schreiben. Wenn das Ziel ist, die momentan in Bearbeitung befindlichen und abgeschlossenen Bestellungen eines Kundendatensatzes zu sehen, könnte es kaum einfacher sein: Erstellen Sie eine oder mehrere Beziehungen der FileMaker Pro-



Kundentabellen mit den MySQL Bestelltabelle und zeigen Sie die Bestellungen und Artikelbestellungen mit Ausschnitten auf der Kundenseite an.

Wenn Sie aus irgendeinem Grund die e-Commerce Lösung aktualisieren müssen, müssten Sie die Rolle aller Tabellen und Felder auf der MySQL Seite sehr gut verstehen. Die e-Commerce Schicht erwartet wahrscheinlich nicht, dass andere Anwendungen die Möglichkeit haben, die Basistabellen zu aktualisieren. Da die Anwendung Open-Source ist (in diesem hypothetischen Szenario), haben Sie eine gute Chance, den Anwendungscode einsehen und prüfen zu können, um mehr über die Bedeutung der zu Grunde liegenden Datenelemente zu erfahren.

2. Daten mit einer kommerziellen Anwendung teilen

Ein großes Projektmanagementbüro verwaltet sein vollständiges Projektportfolio mit Hilfe von Microsoft Project Server und einem MS SQL Server Backend. Eine Arbeitsgruppe innerhalb der Organisation würde gerne Ihre Projektdaten mit einer FileMaker Pro Datenbank der Kunden und Vertragspartner verbinden und ebenfalls ein paar zusätzliche Berichte zu Projekten erstellen, ohne den Project Server Administrator belästigen zu müssen, der bereits sehr viel zu tun hat.

Die gesamte Projektdatenbank ist sehr groß: Über fünfhundert aktive Projekte gleichzeitig, mit unzähligen Aufgaben und Abhängigkeiten. Diese Arbeitsgruppe benötigt nur die Projektkern Daten und keine Details und das auch nur für ihre eigenen Projekte.

Der Project Server Administrator ist bereit, der Arbeitsgruppe einen Satz SQL Server Zugangsdaten einzurichten. Dieser ermöglicht das Lesen aller Projekttabellen aber keine Schreiboperationen. Dies ist genau richtig: Es wäre viel zu riskant, den Datenspeicher einer kommerziellen Anwendung zu aktualisieren, ohne ein tief greifendes technisches Verständnis für die Art und Weise zu haben, wie die Datenbank verwendet wird. Unsachgemäße Datenänderungen würden das Risiko beinhalten, dass Projektdaten der gesamten Organisation beschädigt werden und da Microsoft Project einer kommerzielle Anwendung ist, gibt es relativ wenig Hoffnung herauszufinden, wofür die ganzen Datenbankelemente verwendet werden und wie sich eine gültige Transaktion zusammensetzt. Der Verkäufer behält sich natürlich das Recht vor, diese speziellen Verhaltensweisen jederzeit zu ändern. Wenn Sie also diese Tabellen aktualisieren wollen, sollten Sie es am besten über die verfügbaren Microsoft Project APIs tun.

Der reine Lesezugriff ist für die Anforderungen dieser Arbeitsgruppe genau richtig. Obwohl die SQL Server-Tabellen groß sind, achtet die Gruppe darauf, immer nur die Projektdaten in Beziehungen über Ausschnitte anzusehen, was die Anzahl der gleichzeitig angezeigten Datensätze zusätzlich einschränkt. Es gibt einen zusammenfassenden Bericht, der ausgeführt wird und detailliertere Daten ihrer Projekte zusammenstellt. Die ersten Versuche mit diesem Bericht waren nicht sehr erfolgreich, mit Hilfe des DBA konnten Sie jedoch ihre Suchanfrage verfeinern und die Berichterstellung sehr beschleunigen.

3. Einblick in das Enterprise Finanzsystem

Ein sehr großes Unternehmen hat eine umfangreiche, Oracle-basierte Rechnungsverarbeitungsanwendung. Eine Arbeitsgruppe innerhalb des Unternehmens erzeugt seine eigenen Aufträge und erhält Rechnungen, sendet diese Rechnungen jedoch zur Verarbeitung und Bezahlung an eine zentrale Stelle. Daher existiert der einzige zeitgemäße Datensatz, ob eine Rechnung bezahlt wurde oder nicht, innerhalb des Oracle Systems.

Diese Arbeitsgruppe würde gerne ihr FileMaker basiertes Auftrags- und Rechnungssystem in das Firmensystem so einbinden, dass sie sehen können, ob



eine bestimmte Rechnung schon bezahlt wurde. Dies umfasst einen Lesezugriff auf ein paar Datenhäppchen und ist theoretisch sehr einfach: Zugriff auf die richtige Oracle-Tabelle, dann eine Beziehung zwischen dem FileMaker Pro Rechnungsdatensatz und dem in Oracle herstellen.

Dieser letzte Schritt erweist sich jedoch als einzig schwieriger Punkt. Die Verkäufer-Rechnungsnummern sind nicht eindeutig. Die Primärschlüssel der Rechnung werden von FileMaker Pro erzeugt und haben keine Bedeutung in Oracle und die Oracle und FileMaker Pro Systeme haben unterschiedliche Verkäuferlisten und damit unterschiedliche Verkäufer-IDs. Es scheint keine wirklich gute Methode zu sein, die FileMaker Pro Rechnungsdatensätze mit dem Oracle-Datensatz zu verbinden, bis die Arbeitsgruppe davon erfährt, dass das Unternehmenssystem einen „zusätzlichen Schlüssel“ für jede Rechnung speichern kann. Solange die Schlüsseldaten zusammen mit der Rechnung versendet werden, können die Rechnungsbearbeiter diese eingeben. Mit nur einer kleinen Änderung des Arbeitsablaufs kann die Arbeitsgruppe jetzt die FileMaker Pro Rechnungsschlüssel versenden und im nächsten Monatslauf erscheinen die Schlüssel in der Oracle-Tabelle. Jetzt ist es endlich möglich, eine saubere Beziehung zwischen FileMaker Pro und Oracle zu erzeugen und die Arbeitsgruppe kann jetzt einfach feststellen, ob ihre Verkäufer bezahlt wurden oder nicht.

Optimale ESS-Verfahren für Entwickler

Die in diesem Abschnitt skizzierten Ideen stellen keine bindenden Regeln dar. Es handelt sich vielmehr um Überlegungen zu den wahrscheinlichen Stärken und Schwächen einer neuen komplexen Funktionalität. Der tatsächliche Einsatz wird diese Ideen mit Sicherheit noch verändern und erweitern. Wie in anderen Fällen wird die Realität am besten zeigen, was funktioniert. Hoffentlich bieten die folgenden Richtlinien eine solide Grundlage für den Beginn der Arbeit mit ESS.

Integration statt Skalierung

Die ESS-Funktionalität wurde in erster Linie entwickelt, um es FileMaker Pro Lösungen zu ermöglichen, Daten aus SQL-basierten Lösungen zu integrieren. ESS ist nicht als Hilfsmittel gedacht, um Lösungen über die Grenzen einer rein auf FileMaker Pro basierenden Lösung hinaus auszudehnen. Der Zugriff auf große Datenmengen per ESS muss vorsichtig verwaltet werden. Die Einbindung großer ESS-Tabellen in eine FileMaker Pro Lösung und der Versuch diese wie normale FileMaker Pro-Tabellen zu behandeln, könnte zu erheblichen Leistungseinbußen führen. Experimentieren ist immer der sicherste Weg. Beachten Sie, dass die Idee hinter der ESS-Funktionalität die Integration über die Skalierung stellt.

Machen Sie sich mit der SQL-Sprache vertraut

Mit ESS muss der Entwickler keine SQL-Anfragen mehr schreiben und es gibt de facto auch gar keine direkte Möglichkeit, dies zu tun (beachten Sie dagegen den vorherigen Abschnitt zu Views). Wenn Sie als DBA tätig sind, müssen Sie SQL sowie die Eigenheiten einer ESS-Datenbankplattform kennen. Wenn Sie mit einem DBA oder eine IT-Gruppe zusammenarbeiten müssen, werden Sie davon profitieren, die Arbeitsweise von Backend zu verstehen.

Machen Sie sich mit einem ESS-Produkt vertraut

Wenn Sie ESS verwenden wollen, sollten Sie sich auch mit dem ESS-Produkt vertraut machen, mit dem Sie eine Verbindung herstellen wollen. Sie werden wahrscheinlich einige oder alle der folgenden Punkte über ein spezielles Produkt in Erfahrung bringen müssen:



- In welchem Rahmen ein bestimmtes Produkt den SQL-Standard unterstützt (oder nicht)
- Verfügbare Administrationswerkzeuge
- Datentypen, ihre Bereiche und Beschränkungen
- Anfrageprotokollierung
- Indizierung
- Views (siehe den vorherigen Abschnitt zu Views)

Halten Sie den Kontakt mit großen Datenmengen klein

Diese Anleitung hat sich intensiv mit dem Mechanismus der FileMaker Pro ESS Interaktion befasst, um deutlich zu machen, dass große ESS-Datensatzmengen in FileMaker Pro sehr unhandlich werden können. Unkontrollierte Suchanfragen und Sortierungen von großen Datenmengen können zu großen Anfrageaktivitäten auf dem Server führen oder zu großen Mengen an ESS-Daten, die zur abschließenden Verarbeitung in FileMaker Pro übertragen werden. Denken Sie an die Verwendung von Datenbank-Views oder gut optimierte Anfragen, um die Gesamtgröße der ESS-Datensatzmenge, mit der FileMaker Pro arbeitet, zu beschränken.

Verwenden Sie optimierte Suchen

Wie dieser Artikel gezeigt hat, bevorzugt FileMaker Pro standardmäßig eine FileMaker Pro-ähnliche Suchsemantik. Oft führt dies zu Fuzzy-Match-Anfragen, die keine Vorteile gegenüber bestehenden Datenbankindizes haben. FileMaker Pro Anfrageoperatoren wie == oder einige der Beziehungsoperatoren können eine bessere Wahl sein, da diese dazu neigen, effiziente indizierbare Anfragen auf der SQL-Seite zu erzeugen. Denken Sie über die Kontrolle der Benutzer-Suchinteraktion mit ESS-Datenquellen nach. Geeignete Strategien können sein:

- Erzeugung von eigenen Suchbildschirmen, die den Benutzer darin einschränken, komplexe Anfragen an den ESS-Server zu senden.
- Erzeugung von Script-Suchroutinen, die die Suchkriterien des Benutzers vorverarbeiten und effizientere Operatoren wie == einfügen.
- Verwendung einer „relationalen Suche“-Technik, bei der die Suchkriterien eines Benutzers eine Beziehung in einer ESS-Tabelle auslösen, um die Vorteile der effizienteren relationalen Anfragen zu nutzen, die FileMaker Pro erzeugt.

Seien Sie vorsichtig mit Sortierungen

Denken Sie immer daran, dass die gesamte Sortierung von ESS-Daten innerhalb von FileMaker Pro durchgeführt wird. Daher ist es in Ihrem Interesse die Menge der ESS-Daten zu beschränken, die ein Benutzer mit einem Aufruf sortieren kann.

Zusammenfassung

ESS ist eine sehr leistungsstarke neue Funktionalität der FileMaker Pro Produktreihe. Wie die meisten leistungsstarken Funktionen erfordert sie eine zusätzliche Lernbereitschaft auf Seiten der Entwickler, um ihr Potenzial maximieren zu können. Wie die meisten neuen Funktionen müssen erst Tests durchgeführt und Erfahrungen gesammelt werden, damit weitere Richtlinien und optimale Verfahrensweisen entstehen können.



Der beste Rat, den wir Ihnen abschließend zu ESS geben können, ist: benutzen Sie es! Fangen Sie direkt an, befassen Sie sich einige Zeit mit den Funktionen und ihren Einsatzmöglichkeiten und sehen Sie, wie es Ihnen und Ihren Lösungen am besten helfen kann.

Über den Autor:

Steve Lane ist Vizepräsident von Soliant Consulting, wo er die FileMaker- und Webentwicklungsteams in den letzten acht Jahren geleitet hat. Er hat fünf Bücher zu FileMaker mitgeschrieben und spricht häufig bei der FileMaker Developer Conference, bei der er 2003 den FileMaker Fellowship Preis für die „Erweiterung der Grenzen von FileMaker Pro“ erhalten hat. Steve Lane ist autorisierter FileMaker-Trainer und war an der Entwicklung der letzten drei Versionen der Materialien der FileMaker Professional Training Foundation Series beteiligt. Er hat eine Reihe von FileMaker-Klienten beraten - von kleinen Privatunternehmen bis hin zu großen öffentlichen Firmen - und hat sich auf die Integration von FileMaker Systemen mit Web- und SQL-basierten Systemen spezialisiert. Er hat einen PhD in mittelalterlicher Geschichte der University of Chicago, an der er vor Jahren seine erste relationale Datenbank mit Hilfe des Macintosh-basierten Double Helix Systems erstellt hat.

Dieser Artikel hat sehr von den Ratschlägen und Beiträgen anderer profitiert. Insbesondere haben Galt Johnson und Clay Maeckel wichtige technische Hinweise gegeben, während sich Ray Cologon mit einem wichtigen Aspekt des SQL-Server Verhaltens in Verbindung mit Date/Time Feldern befasst hat.

©2007 FileMaker, Inc. Alle Rechte vorbehalten. FileMaker ist ein Warenzeichen von FileMaker, Inc., eingetragen in den USA und anderen Ländern, und das Dateiordnerlogo ist ein Warenzeichen von FileMaker, Inc. Alle anderen Warenzeichen sind Eigentum ihrer entsprechenden Eigentümer. FileMaker gewährt keine ausdrücklichen oder stillschweigenden Garantien hinsichtlich der Leistung oder Zuverlässigkeit aller hierin genannten Produkte, die von unabhängigen Händlern hergestellt werden. Alle Absprachen, Vereinbarungen oder Garantien gelten zwischen den Händlern und den möglichen Benutzern. Produktspezifikationen und Verfügbarkeit unterliegen Änderungen ohne vorherige Mitteilung. (Docv2)

DIESES DOKUMENT WIRD OHNE MÄNGELGEWÄHR OHNE GARANTIE IRGENDNEINER ART ZUR VERFÜGUNG GESTELLT. FILEMAKER LEHNT ALLE AUSDRÜCKLICHEN UND STILLSCHWEIGENDEN GARANTIE AB, EINSCHLIESSLICH ABER NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDEN GARANTIE DER ALLGEMEINEN GEBRAUCHSTAUGLICHKEIT ODER TAUGLICHKEIT FÜR EINEN BESTIMMTEN ZWECK ODER DIE GARANTIE DER NICHTVERLETZUNG. IN KEINEM FALL HAFTEN FILEMAKER UND SEINE LIEFERANTEN FÜR SCHÄDEN, EINSCHLIESSLICH UNMITTELBARE, DIREKTE, BEILÄUFIG ENTSTANDENDE SCHÄDEN ODER FOLGESCHÄDEN, VERLUST VON GESCHÄFTSGEWINNEN, ENTSCHÄDIGUNG MIT STRAFCHARAKTER ODER KONKRETE SCHÄDEN, SELBST WENN FILEMAKER ODER SEINE LIEFERANTEN ÜBER DIE MÖGLICHKEIT SOLCHER SCHÄDEN INFORMIERT WERDEN. EINIGE LÄNDER ERLAUBEN KEINEN AUSSCHLUSS ODER KEINE BESCHRÄNKUNG DER HAFTUNG. FILEMAKER KANN ZU JEDER ZEIT UND OHNE BENACHRICHTIGUNG ÄNDERUNGEN AN DIESEM DOKUMENT VORNEHMEN. DIESES DOKUMENT IST MÖGLICHERWEISE VERALTET UND FILEMAKER VERPFLICHTET SICH NICHT, DIESE INFORMATIONEN ZU AKTUALISIEREN.

