

# Die ultimative Power und Geschwindigkeit von VFP

*Valdis Matison*

Die Verwaltung des Reiseverkehrs durch den Kanaltunnel mit Hilfe einer FoxPro-Anwendung von Valdis Matison ist einer der Klassiker unter den FoxPro-Lösungen. Zusätzlich werden hier einige Probleme und deren Lösung bei der praktischen Umsetzung beschrieben.



Immer, wenn Kunden über Projekte reden, kommt die Frage auf, die viele Daten die größte Tabelle und die gesamte Datenbank aufnehmen kann. Dieser Artikel behandelt die Probleme, die auftauchen, wenn Ihr Kunde ausführt, dass einzelne Tabellen mehr als zwei Gigabyte an Daten aufnehmen müssen und die gesamte Datenbank aus vielen Gigabyte an Informationen besteht. Die hier behandelten Vorgehensweisen wurden beim Design der Anwendung berücksichtigt, die weltweit mit den größten Datenmengen umgeht, die eine einzelne FoxPro-Anwendung verwaltet. Diese Datenbank enthält 128 Gigabyte an Informationen. FoxPro limitiert die Größe der Tabellen auf jeweils 2 GB. Dies ist aber keine Limitierung durch das Betriebssystem. Auch wenn Ihre Daten auf einem Betriebssystem gespeichert werden, das große Dateien erlaubt, beispielsweise UNIX oder Windows NT unter NTFS, sind Ihre FoxPro-Tabellen auf 2 GB beschränkt.

Da die einzelne Tabelle auf eine maximale Größe beschränkt ist, gibt es auch eine praktische Grenze der Datensätze, die sie enthalten kann. Die beiden Faktoren, durch die die Größe der Tabelle bestimmt wird, sind die Anzahl der darin enthaltenen Datensätze und die physikalische Größe jedes Datensatzes. FoxPro ermöglicht der einzelnen Tabelle, eine Milliarde Datensätze zu enthalten, aber dies macht nicht viel Sinn, da diese Tabelle nur ein einzelnes Feld mit einer Breite

von einem Byte erlauben würde. Daher sind nur 256 eindeutige Datensätze möglich. Eine sinnvolle Begrenzung der Datensätze liegt bei 15 oder 20 Millionen mit einer durchschnittlichen Länge des Datensatzes von 100 bis 150 Byte. Außerdem können Informationen in Memofeldern gespeichert werden, was zusätzliche 2 GB Datenvolumen für eine einzelne Tabelle ermöglicht, da die Memodatensätze in einer separaten Datei gespeichert werden.

---

## A) Große logische Partitionen

---

Eine Möglichkeit des Teilens einer Tabelle besteht in der Erstellung einer logischen Datei mit mehr als zwei GB und der Partitionierung der Daten in physikalische Dateien, die sowohl von FoxPro als auch vom Betriebssystem akzeptiert werden. Dies impliziert, dass eine Methode für die Integration existiert, damit der Anwender mit den Daten arbeiten kann. Eine Abfrage beispielsweise kann sich aufgrund der Partitionierung über mehrere Tabellen erstrecken und es muss Code existieren, um auf die erforderlichen Tabellen zuzugreifen und gegebenenfalls die Ergebnismengen zu kombinieren.

Die Teilung der Daten kann an einer willkürlichen physikalischen Grenze geschehen, in logischen Verbindungen oder die Daten können in vertikale Partitionen aufgeteilt werden. Eine physikalische Grenze ist bei 2 GB erreicht und es muss eine neue Datei erstellt werden, die die zusätzlichen Informationen aufnimmt. Die logische Ansicht auf die

Daten ist einfach eine Verbindung zweier Tabellen. Diese Art der Teilung könnte in einer Transaktionsumgebung genutzt werden, in der Daten für die Archivierung gespeichert, aber nur selten geändert werden. Ein Beispiel dafür ist eine wissenschaftliche Datensammlung. In Abbildung 1 wird aus drei physikalischen Tabellen eine logische Tabelle erstellt.

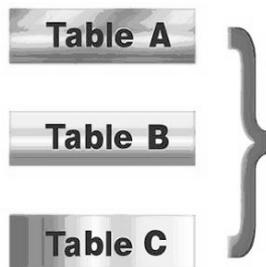


Abbildung 1. Horizontale Partitionierung - drei physikalische Tabellen können horizontal partitioniert werden, um eine logische Tabelle zu erstellen.

Eine logische Bindung ist einfach zu implementieren, da die meisten großen Datenmengen von selbst eine Teilung in Gruppen anbieten. Dabei kann es sich um geographische, demographische, chronologische oder andere logische Gruppen handeln, die in den Daten bestehen. Ein Vorteil des Teilens der Daten auf diese Weise besteht darin, dass kleinere Gruppen von Anwendern auf einer einzelnen Datei aktiv arbeiten können. Dadurch wird die Ausgabe des Systems insgesamt beschleunigt.

Viele alltägliche Befehle von FoxPro, beispielsweise SEEK oder LOCATE funktionieren mit Daten, die auf diese Weise aufgeteilt wurden, nicht und SQL-Abfragen müssen vorsichtig geschrieben werden, so dass UNIONS zwischen allen physikalischen Tabellen erstellt werden, die die logische Einheit bilden. Es kann sich dabei um eine funktionsfähige Lösung handeln, da Abfragen häufig nur bestimmte Teile der Daten erfordern. Informationen werden in der Regel für eine bestimmte Region abgefragt, oder es werden nur häufig genutzte Daten abgefragt, so dass die regulären Befehle von FoxPro gut funktionieren. Werden zusammengesetzte Informationen benötigt, kön-

nen Abfragen auf die einzelnen Tabellen ausgeführt und anschließend integriert werden, wenn alle Abfragen beendet sind.

Die vertikale Partitionierung einer Tabelle erfordert ein Teilen der Daten in mehrere Tabellen, wobei mehrere Schlüsselfelder in einer Tabelle verbleiben und die verbleibenden Supportinformationen in der anderen Tabelle (wie in Abbildung 2 dargestellt). Für jeden Datensatz in der linken Tabelle befindet sich in der rechten Tabelle ein Fremdschlüssel, der dem Primärschlüssel links entspricht. Diese Methode der Partitionierung bevorzugt Systeme, auf denen Abfragen nur auf indizierten Feldern ausgeführt und die Ergebnisdaten erst empfangen werden, wenn die Abfrage beendet ist. Die Tabelle und die dazu gehörende .CDX, die abgefragt werden, können auf einer sehr schnellen Festplatte gespeichert werden, während die verbleibenden Informationen auf einer CD-ROM oder einem optischen Datenträger gespeichert werden. Eine Abfrage wird gegen die Tabelle auf der linken Seite ausgeführt und es wird ein spezieller Datensatz zurückgegeben. Will der Anwender den Rest der Informationen angezeigt bekommen, wird der Primärschlüssel eingesetzt, um auf den entsprechenden Datensatz auf der CD-ROM zuzugreifen. Dieses Vorgehen ist bei der Speicherung von Grafiken in Memofeldern zu empfehlen. Der erforderliche Overhead konzentriert sich auf die Verwaltung der Verbindungen der Tabellen.

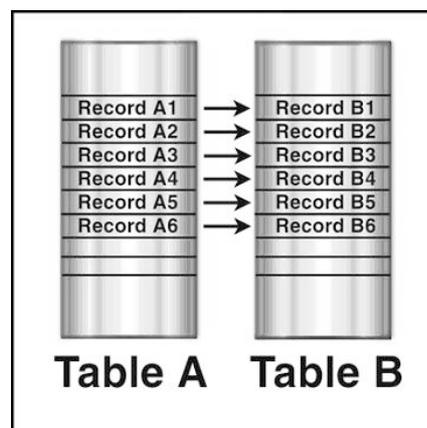
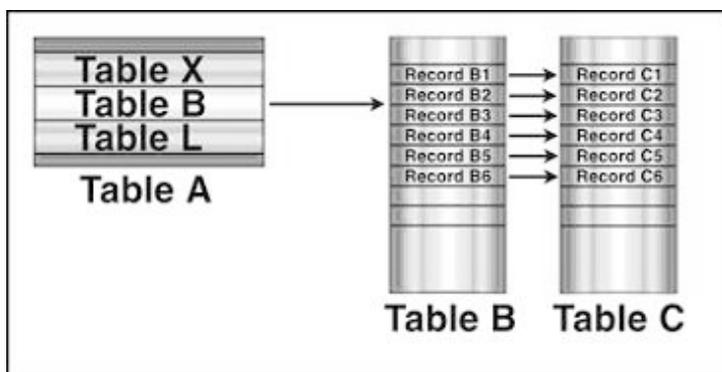


Abbildung 2. Vertikale Partitionierung - Zwei physikalische Tabellen sind vertikal partitioniert. Jeder Datensatz in Tabelle A ist mit dem dazu gehörenden Datensatz in Tabelle B über den Primärschlüssel verbunden.

Der Gedanke der vertikalen Aufteilung einer Tabelle kann noch einen Schritt weiter geführt werden, indem Referenzen auf mehrere verbundene Tabellen aufgenommen werden. Die Haupttabelle enthält nicht nur Schlüsselwerte, sondern auch getrennte Attribute für den Laufwerksbezeichner, Pfad und den Aliasnamen der Tabelle, die die detaillierten Informationen für einen bestimmten Datensatz enthält (dies wird in Abbildung 3 dargestellt). Die unterstützenden Tabellen sind durch einen logischen Wert, beispielsweise der Provinz, dem Monat oder der Altersgruppe aufgeteilt.



**Abbildung 3.** Partitionierung über Zeiger - Jeder Datensatz in Tabelle A verfügt über eine logische Referenz auf den Namen der Tabelle, in der die detaillierten Daten enthalten sind, sowie die Primärschlüsselinformation wie in Abbildung 2. Bei Tabelle C kann es sich wie in Abbildung 2 um eine Partition oder um eine getrennte Memodatei handeln.

Das Aufnehmen eines logischen Wertes für die Pfadinformation ist entscheidend, da dies die Verwaltung vereinfacht. Sowohl NetWare als auch NT ermöglichen den Einsatz logischer Zuweisungen von Laufwerksbezeichnern. Auch der Einsatz des Aliasnamens ist wichtig, da sich der physikalische Dateiname ändern kann. Der Einsatz einer Tabelle für die Datenverwaltung oder eines Datadictionary vereinfacht die Zuordnung der logischen Namen - der physikalische Speicherort jeder Tabelle ist im Datadictionary gespeichert.

Jetzt ist es einfacher, zu erkennen, wie eine extrem große Datenbank erstellt wird. Eine Primärtabelle könnte in einer 2 GB-Datei

über 15 Millionen Datensätze verfügen. Nehmen wir an, dass jeder Datensatz mit den Daten in Feldern und Memos 50 KB groß ist. Damit sind für jede Detailtabelle 40.000 Datensätze möglich. Die gesamte Datenbank könnte maximal 375 GB an Informationen enthalten.

## B) Die physikalische Datenspeicherung

Sie haben sich vermutlich keine allzu großen Gedanken über den Speicherort Ihrer Dateien gemacht. In einer Netzwerkumgebung befinden sich die Daten der Anwendung in der Regel auf einem einzelnen Netzlaufwerk und die Festplatte jedes Anwenders wird für temporäre Dateien genutzt, die während der Abfragen, Indexerstellung usw. erstellt werden.

Auch auf eine Datenbank, die aus sehr großen Tabellen besteht, greifen Sie über einen einzelnen logischen Laufwerksbezeichner zu. Dies ist möglich, da Netzlaufwerke partitioniert werden können, um ein einziges logisches Laufwerk zu bilden. Es ist einfacher, mit einem einzelnen Laufwerksbezeichner zu arbeiten, es gibt aber Vorteile in der Geschwindigkeit und der Fehlertoleranz, wenn für große Systeme mehrere Laufwerksbezeichner genutzt werden.

Das Risiko, dass auf die gesamte Datenbank nicht mehr zugegriffen werden kann, wenn eine Festplatte einen Defekt aufweist, ist geringer, wenn die Tabellen sich auf mehr als einer Partition befinden. Werden mehrere Laufwerke zu einer logischen Partition zusammengefasst und ein Laufwerk ist defekt, kann auf die gesamte Partition nicht mehr zugegriffen werden, unabhängig davon, wie viele Daten dort gespeichert sind. Große Partitionen benötigen länger für die Reparatur, da die gesamte Partition repariert werden muss, nicht nur ein einzelnes Laufwerk.

Es gibt noch einen anderen Vorteil beim Einsatz mehrerer Partitionen - es ist einfacher, den Zugriff auf einige Tabellen einzuschränken und die Verwaltung einer gegebenen Festplatte durchzuführen. In diesem

Fall werden die Partitionen mit einer Festplatte im Verhältnis eins zu eins dargestellt.

Die Performance wird auch erhöht, wenn sich die Tabellen auf einer Festplatte befinden und die dazu gehörenden Indizes auf einer anderen. Der Lesekopf der Festplatte muss sich nicht zwischen der Tabelle und den Daten hin- und herbewegen, wenn diese getrennt sind.

## **C) Bereitstellen von Plattenplatz**

Verwaltungsprozeduren wie Backup und Restore, die Indizierung, Sortierung, die Reservierung neuen Speicherplatzes usw. benötigen große Mengen freien Plattenplatzes. Dieser Platz sollte einige Gigabyte umfassen und die Größe bemisst sich nach Ihren größten Dateien. Wenn Sie die Anforderungen für ein großes System definieren, planen Sie einige Gigabyte freien Platz ein, die nur für die Verwaltung genutzt werden.

Werden die eben beschriebenen Partitionierungsschemata eingesetzt, muss der Datenbankadministrator entscheiden, wann den Arbeitstabellen zusätzlicher Plattenplatz zugeordnet wird. Der zu belegende Plattenplatz muss bereitgestellt werden, bevor in erheblichem Umfang Daten erfasst werden. Da die Tabellen größer werden, müssen Sie sicherstellen, dass kein anderer Prozess auf den bereitgestellten Platz auf der gesamten Festplatte Ihrer Anwendung zugreifen kann. Es gibt keine Möglichkeit, in FoxPro automatisch Plattenplatz zu reservieren, so wie dies in Oracle oder Sybase geschieht.

Besonders wichtig ist die Bereitstellung von Platz für binäre Daten. Wenn Ihr System normale Informationen in Feldern nutzt, ist beträgt die größte Informationsmenge in einem einzelnen Datensatz 65 KB. Wenn Sie mit Memo- oder Generalfeldern arbeiten, gibt es ein anderes Problem. Ein einzelner Datensatz kann aus einem 2 GB-Block bestehen. Der Datenbankadministrator muss die Größe der Datenbank aktiv überwachen, um Fehler aufgrund fehlenden Festplattenplatzes vorzubeugen.

Es gibt mehrere Methoden, um sicherzustellen, dass Platz vorhanden ist, wenn Sie ihn

benötigen. Die einfachste Methode, die auch ich bevorzuge, besteht darin, den Anwendern den Zugriff auf die Festplatte zu verwehren, es sei denn, sie arbeiten mit der Datenbankanwendung. Dies ist bereits durch Skripts des Betriebssystems ausgeführt. Leider ist dies nicht immer möglich, so dass andere Methoden eingesetzt werden müssen, um Plattenplatz zu reservieren.

Da FoxPro mit festen Feldlängen arbeitet, können wir an die Datenbank gelöschte leere Datensätze anhängen, um den Platz aufzufüllen. Es ist einfach, ein Schema zu entwickeln, das den nächsten verfügbaren freien Datensatz findet und diesen nutzt, wenn neue Informationen eingegeben werden. Viele Entwickler nutzen diese Methode bereits, um nicht genutzten Platz innerhalb der Dateien zu vermeiden (vergleichen Sie dazu Bill Lawders Artikel in der Ausgabe Mai 1994 des FoxPro Advisors für Ideen, gelöschte Datensätze erneut zu verwenden). Der Nachteil dabei ist, dass Ihre Dateien immer über die maximale Größe verfügen. Diese Methode hat auch Nachteile in der Performance. Sie müssen eventuell auch zusätzlich die Anzahl der physikalischen Datensätze mitschreiben, da RECCOUNT() auch die gelöschten Datensätze berücksichtigt.

Eine andere Möglichkeit, Plattenplatz zu reservieren, besteht darin, auf dem Laufwerk Dateien zu platzieren, deren einzige Aufgabe darin besteht, Platz einzunehmen. Steht der Anwendung kein Platz mehr zur Verfügung, können durch die FoxPro-Anwendung einzelne Dateien gelöscht werden. Eine 2 GB große Partition kann durch zwanzig 100 MB große Dateien reserviert werden. Dies kann Probleme mit dem Betriebssystem des Netzwerks hervorrufen, da Ressourcen belegt werden, die benötigt werden, um die Daten nach dem Löschen für ein eventuelles Wiederherstellen zu verwalten. Dies gibt auch Probleme, wenn Sie ein Backup ausführen wollen und die Backupsoftware verfügt nicht über ausreichenden Speicher für die Ausführung. Mir ist dies vor kurzem bei einem Test geschehen, den ich durchgeführt habe.

## D) Geschwindigkeit

Ich weiß, was Sie jetzt denken. Es ist möglich, eine wirklich große Datenbank zu erstellen, aber wie schnell kann dann auf die Daten zugegriffen werden? Hier läuft Fox-Pro zur Hochform auf. Abfragen auf indizierte Felder sind immer noch sehr schnell – auch in Tabellen mit 10 Millionen Datensätzen. Die Abfragen müssen selbstverständlich Rushmore-optimiert sein, entscheidend ist aber, wie die Daten auf der Festplatte gespeichert sind. Die Antwortzeiten werden direkt von der Art beeinflusst, in der die Daten physikalisch sortiert, nicht nur indiziert, sind. Die Daten sollten entsprechend dem am häufigsten gebrauchten Indextag sortiert sein. Die Antwortzeiten für einen einfachen Test sehen Sie in Abbildung 4. Dieser Test misst die Zeitspanne, die erforderlich ist, eine Tabelle mit zwei Millionen Datensätzen zu scannen. Beachten Sie, dass bei den kurzen Balken die Daten bereits sortiert sind. Der lange Balken repräsentiert eine Tabelle, die indiziert war, in der die Daten sich aber in einer zufälligen Reihenfolge befinden.

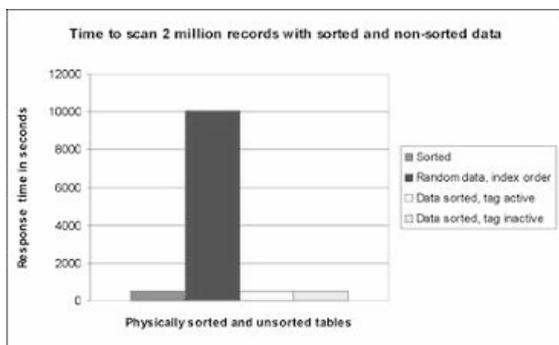


Abbildung 4. Verarbeiten großer Tabellen - Die Grafik zeigt, dass das Bewegen durch die Datensätze einer großen Tabelle schnell durchgeführt wird, wenn die Datensätze sich physikalisch in der richtigen Reihenfolge befinden.

Abbildung 5 zeigt die Zeiten, die benötigt werden, um einen Join von zwei Tabellen aufzubauen, die zwei Millionen bzw. 100.000 Datensätze enthalten. Beachten Sie, dass die Aufgabe fast um den Faktor 10 schneller ausgeführt wird, wenn beide Tabellen physikalisch sortiert sind, wobei der Index gleich bleibt.

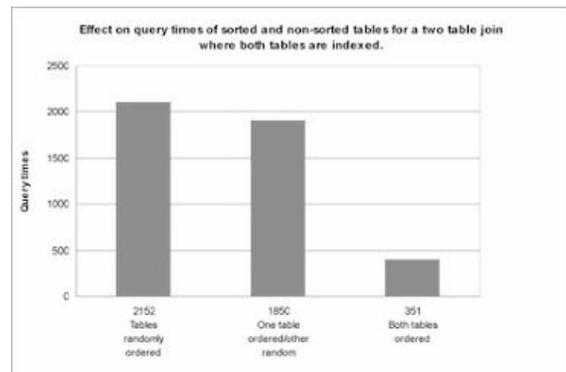


Abbildung 5. Verbinden großer Tabellen - die physikalische Sortierung der Tabellen bewirkt einen extremen Unterschied in der Geschwindigkeit des Join.

## E) Datenkomprimierung

Wir nutzen häufig einen selbst entwickelten Algorithmus für die Komprimierung von Daten, um die Anforderungen an die Datenspeicherung zu verringern. Die Datenkomprimierung wird in der Regel für Daten eingesetzt, die in Memofeldern gespeichert sind. Daten in Datensätzen zu komprimieren würde einfach einen zu großen Overhead erfordern. Populäre Kompressionsmechanismen wie DbfSpace oder Stacker kommen aus Gründen der Performance und der Fehlertoleranz nicht in Betracht. Softwarepakete, die im Stapelbetrieb genutzt werden können, beispielsweise lha oder PkZip, sind akzeptabel, da sie auf Basis Datei für Datei arbeiten und die Datenspeicherung auf der Festplatte nicht ändern.

Ein Datenkompressionssystem kann sehr effizient sein, wenn man die Natur der gespeicherten Daten verstanden hat. Die Kenntnis eines Musters oder der definierten Struktur kann eingesetzt werden, um das effizienteste Kompressionsschema für diese Information zu erstellen. Ein Nachteil des Einsatzes der Kompression ist, dass die Performance bei Lese- und Schreiboperationen nachlässt und exportierte Daten sind für die steuernde Anwendung nicht lesbar, es sein denn, sie werden zuvor dekomprimiert. Für Sie hat der Einsatz eines eigenen Schemas für die Datenkompression den Vorteil, dass Sie die vollständige Kontrolle darüber haben. Dieses Schema kann geändert werden, wenn sich die Daten ändern und kann auch je

nach Bedarf ein- und ausgeschaltet werden. Außerdem sind Sie in diesem Punkt nicht von den Launen des Betriebssystems oder von Herstellern von Fremdprodukten abhängig.

---

## F) Datenzugriff

---

Routinen für das Öffnen von Dateien müssen verlässlich funktionieren und es ermöglichen, jeder einzelnen Datei einen vollständigen Pfad zuzuordnen. Laufwerksbezeichner und Pfade sollten separat gespeichert und zur Laufzeit zusammengefügt werden, um den korrekten Pfad zu den Tabellen zu ermitteln. Dies funktioniert am Einfachsten über ein Datadictionary oder eine Tabelle für die Administration der Daten. Novell ermöglicht es Ihnen, logische Laufwerksbezeichner dynamisch zuzuordnen, was Ihnen die Verwaltung erleichtert.

Im Fall eines Problems mit einer Festplatte haben Sie die Möglichkeit, die Daten auf ein anderes Laufwerk zu kopieren – entweder von Band oder vom defekten Laufwerk – und anschließend den Pfad zum Laufwerk in Ihrer Dateiöffnungsroutine zu ändern. Beim Einsatz logischer Laufwerksbezeichner können Sie Ihre Daten auch auf einem anderen Dateiserver wiederherstellen. Sie können für den Zugriff auf andere Speicherorte auch Aliasnamen einsetzen.

Dateiöffnungsroutinen müssen die Intelligenz eingebaut haben, dem Anwender den aktuellen Status einer Datei mitzuteilen. Ist es möglich, auf die Daten zuzugreifen, stellt dies kein Problem dar, aber wenn die Daten gerade gesichert oder wiederhergestellt werden, muss dem Anwender mitgeteilt werden, dass Teile der Datenbank nicht verfügbar sind. Nachdem die Daten auf dem System wiederhergestellt sind, sollte der Anwender die Möglichkeit haben, die gerade aktualisierten Tabellen erneut zu öffnen.

---

## G) Pflege

---

Datenbanksysteme erfordern eine regelmäßige Pflege. Strukturen könnten geändert werden, Daten müssen auf einer schnelleren

Hardware gespeichert werden oder Informationen müssen archiviert werden. Ein gutes System enthält die Pflegearbeiten bereits und lässt den Anwender so weit wie möglich bei der produktiven Arbeit – auch wenn nicht geplante Pflegearbeiten durchgeführt werden müssen. Ein Flag in der Tabelle für die Datenbankadministration kann anzeigen, welche Dateien für das System verfügbar sind und den Status der anderen Dateien mitteilen. Ihr System sollte in der Lage sein, dem Anwender zu ermöglichen, mit Teilen der Anwendung zu arbeiten – auch wenn einige Tabellen nicht verfügbar sind. Es ist nicht die Regel, dass die Anwender jederzeit den Zugriff auf sämtliche Daten benötigen, so dass die Deaktivierung von Teilen der Anwendung Sinn macht.

Es ist wichtig, zu verstehen, wie Ihre Anwendung die Komponenten der Dateien beeinflusst, beispielsweise die Memodatei oder den Index. Indizes wachsen, wenn ein Datensatz neu angelegt wird. Sie wachsen auch, wenn ein Teil des Schlüssels geändert wird, der im Indexausdruck enthalten ist. Viele Entwickler nutzen gelöschte Datensätze neu, indem jedes Feld eines gelöschten Datensatzes mit Leerzeichen aufgefüllt wird. Wird dem System ein neuer Datensatz hinzugefügt, wird der Datensatz erneut genutzt. Mit jeder Kombination von Delete und Replace wird der Index größer. Um den kleinstmöglichen Index zu erhalten, müssen sie sicherstellen, dass Ihre Daten in der gleichen Reihenfolge wie Ihr Indexausdruck sortiert sind.

Der PC, der die Pflegeaufgaben übernimmt, muss über eine ausreichend große Festplatte, einen schnellen Prozessor und viel Arbeitsspeicher verfügen, um die Aufgabe effizient erledigen zu können. Wenn wir Indizes für Tabellen erstellen, die Millionen Datensätze enthalten, bekommen wir Probleme mit einigen unserer Festplatten, die nicht über ausreichenden Platz für die Erstellung der temporären Dateien verfügen, die während der Indizierung benötigt werden.

---

## H) Backup und Restore

---

Wenn Computer im Spiel sind, ist auch Murphys Gesetz immer präsent. Es kann immer etwas schief laufen, daher ist es wichtig, dass Sie Ihre Daten gesichert haben. Die beste Möglichkeit für das Sichern von Daten ist, die Anwender aus dem System auszuloggen und anschließend die Backup-Software aufzurufen. Dies stellt ein Backup sicher, bei dem alle Transaktionen abgeschlossen sind, die Indizes sowie die Relationen zwischen den Tabellen aktualisiert sind. Der Nachteil dieses Vorgehens ist, dass die Anwender während des Backups nicht arbeiten können. Diese Art des Backups ist nicht angebracht, wenn ein System ständig arbeiten muss.

Die folgende Behandlung des Themas geht davon aus, dass zwischen der Datenbank-Software und der Backup-Software eine Kommunikation möglich ist. Befehle müssen gesandt werden, um das Backup zu starten und zu beenden und um den Status des Backup-Prozesses zu empfangen. Die Möglichkeit, Dateinamen mit vollständigen Pfaden zu senden, ist äußerst wichtig. Alle Transaktionen zwischen der Backup-Software und Ihrer Datenbank müssen von beiden Systemen festgehalten werden. Ist ein Restore erforderlich, muss das entsprechende Band sowie die Backup-Session über das Logsystem einfach aufzufinden sein. Ein aktives Backup oder Restore muss auch für die FoxPro-Anwendung sichtbar sein.

Probleme tauchen auf, wenn Systeme während des Betriebs gesichert werden sollen. Die Backup-Software überspringt die geöffneten Dateien und kehrt während des Sicherungslaufs später zurück, um erneut zu versuchen, diese Dateien zu sichern. Dateien, die dann noch geöffnet sind, werden vollständig ignoriert. Daher ist es häufig zu beobachten, dass Backups von Datenbanken nicht vollständig sind. Eine einfache und effektive Lösung dieses Problems besteht darin, ein zweites Set der Daten zu erstellen, während das System genutzt wird. Ein temporärer Arbeitsbereich wird gekennzeichnet, so dass er während der Backup- und Restore-Prozeduren ausgelassen wird. Unter FoxPro werden die Daten satzweise in diesen Bereich kopiert, so dass die normalen Akti-

vitäten auf den Tabellen durchgeführt werden können. Nachdem die Kopie vollständig ist, wird ein Befehl an die Backup-Software gesandt, damit mit der Sicherung der Kopie der Daten begonnen werden kann.

Diese Backup-Methode ist nicht idiotensicher. Das satzweise Kopieren mit den entsprechenden Sperrungen bietet eine Abbildung des Status Ihrer Datenbank zu einem gegebenen Zeitpunkt. Zu diesem Zeitpunkt können aber noch nicht abgeschlossene Transaktionen vorhanden sein. Dies ist alles, was von der Backup-Software sinnvoll abgefragt werden kann, die auch arbeiten muss, wenn die Tabellen geöffnet bleiben. Unvollständige Transaktionen können vorkommen, wenn die Tabelle, auf der aktuell ein Backup durchgeführt wird, häufig geändert wird oder von Werten in verbundenen Tabellen abhängig ist. Realtime-Backup-Systeme, die mit Oracle oder dem SQL Server eingesetzt werden, funktionieren auch nicht viel besser, wenn sie mit der gleichen Aktivität konfrontiert werden. In unseren Systemen nutzen wir unterschiedliche Stufen der Redundanz, um sicherzustellen, dass, wenn Daten während des Backup-Prozesses aktualisiert werden, irgendwo im System ein Datensatz mit der Aktivität vorhanden ist. Damit wird gewährleistet, dass für einen festen Zeitpunkt ein akkurates Backup gemacht werden kann.

Wie ich bereits weiter vorn bemerkt habe, besteht die beste Möglichkeit, eine Datenbank zu sichern, darin, alle Anwender auszuloggen und anschließend das Backup durchzuführen. Aber – weshalb sollten die Anwender mit dem gesamten System nicht mehr arbeiten können, weshalb sollte man sie nicht nur von Teilen der Datenbank ausschließen? Eine Datenbank kann aus vielen Tabellen bestehen und Tabellen können mehrere Partitionen aufweisen. Durch den Einsatz einer Anministrationstabelle kann der Datenbankadministrator Anwender effektiv von der Arbeit auf verschiedenen Tabellen ausschließen. Dies geschieht durch die Markierung der entsprechenden Tabelle als inaktiv, so dass die Anwender nicht mehr auf diese Tabelle zugreifen können. Nach der Beendigung des Backup wird die Markierung wieder auf aktiv zurückgesetzt.

Das Restore funktioniert auf die gleiche Weise wie das Backup. Die Daten werden zunächst im temporären Arbeitsverzeichnis des Backup-Systems wiederhergestellt. Die FoxPro-Anwendung kontrolliert die Daten im Arbeitsverzeichnis und entscheidet satzweise, wie diese Information der aktiven Tabelle hinzugefügt wird.

---

## I) Datenintegrität

---

FoxPro ist ein wirklich stabiles Produkt, trotzdem können defekte Tabellen vorkommen, auch wenn dies selten geschieht – und dies häufig ohne Ihr Wissen. Auch eine perfekt entworfene Anwendung kann falsche Daten enthalten. Meine Erfahrung hat gezeigt, dass defekte Daten in der Regel das Ergebnis inkompatibler Treiber, defekter Hardware oder apathischer Anwender sind. Nur die ersten beiden Gründe können Sie kontrollieren.

Kompatible Treiberversionen sind besonders in einer Netzwerkkumgebung wichtig und es ist erforderlich, dass Sie vollständig testen, damit Sie sicherstellen können, dass Sie über ein stabiles System verfügen. Die Probleme treten nicht immer sofort auf. Es gibt Treiber für das Netzwerk, für die Grafikkarte, das Betriebssystem der Workstation, die Festplatte der Workstation, das Netzwerklaufwerk, das Bandlaufwerk usw. Das Erstellen einer stabilen Konfiguration kann ein echtes Abenteuer sein, da es selten ist, dass ein Hersteller alle Kombinationen anderer Geräte berücksichtigt, die mit dem eigenen Produkt eingesetzt werden. Die Installation einer neuen Version einer Komponente führt häufig zu Problemen mit allen anderen Komponenten eines Systems, so dass ich eine ganzheitliche Lösung der Systemintegration empfehlen würde. Es ist ratsam, ein neues Servicepack des Betriebssystems zunächst auf einem Testsystem vollständig durchzutesten, bevor Sie alle Ihre PCs damit ausrüsten. In der Regel werden durch einen Patch Fehler beseitigt, manchmal aber auch neue hervorgerufen.

Defekte Hardware ist in der Regel einfacher auszumachen und zu reparieren, obwohl einige nebensächliche Ausrüstungen schwierig

zu diagnostizieren sein können. Netzwerkübergänge und Schnittstellenkarten können zu unkorrekten Paketen führen, die gesendet oder empfangen werden. Mein Rat im Hinblick auf Treiber und Hardware ist, nur mit Markenprodukten zu arbeiten, und sei es nur, um bessere Unterstützung durch den Hersteller zu erhalten. Von größter Bedeutung ist die Verkabelung. Es ist absolut notwendig, für Ihr Netzwerk die richtigen Kabel zu verwenden. Auch die Kabel innerhalb Ihres PCs sowie in Ihrem Dateiserver sind wichtig. SCSI-Kabel sind eine häufige Fehlerursache, einschließlich des Verschwindens ganzer Netzwerklaufwerke. Einer meiner Kunden hat das 6 Dollar-Kabel, das mit dem Laufwerk geliefert wurde, durch ein 75 Dollar-Kabel ersetzt. Seitdem läuft die Maschine problemlos.

---

## J) Defekte Daten

---

Defekte Daten treten auf, wenn unbeabsichtigte Zeichen Teil einer Tabelle werden. Diese irrtümlichen Zeichen sind keine ungültigen Daten wie ein unkorrektes Datum oder ein falscher numerischer Wert. Defekte Daten können auch in Tabellen auftreten, für die die Geschäftsregeln korrekt geschrieben und die Datenüberprüfung durchgeführt wurde. Ein einfaches Beispiel für defekte Daten ist, wenn ein Zeichen Teil eines Datums oder eines numerischen Feldes wird.

Jede .DBF besteht aus einem Headerbereich, der die Dateistruktur beschreibt, sowie den Daten, mit denen die Datei vervollständigt wird. Ein Fehler im Header kann dazu führen, dass die Tabelle vollständig unbrauchbar wird, da FoxPro die darin enthaltenen Informationen benötigt, um die Dateistruktur, die Anzahl der Datensätze usw. festzustellen. Die Fehlermeldung von FoxPro „Keine Datenbankdatei“ wird häufig durch einen Datensatzzähler im Header hervorgerufen, der größer ist als die Anzahl der Datensätze in der physikalischen Datei. Eine vollständige Beschreibung des Headers der .DBF finden Sie in der Dokumentation zu FoxPro.

Alle Daten werden auf eine bestimmte Art und Weise gespeichert und es ist möglich,

einige Probleme zu isolieren, indem jede Komponente Byte für Byte geprüft wird. Ein Datumsfeld wird beispielsweise physikalisch als YYYYMMDD gespeichert, wobei jedes Byte eine Ziffer repräsentiert. Ist eines der Bytes ein Zeichen, ist dieses Feld defekt. Das folgende Datum erzeugt keine Fehlermeldung, aber die Speicherung „1996C106“ für den 6. November 1996 ist offensichtlich falsch. Um festzustellen, ob in die .DBF eine falsche Information geschrieben wurde, können Sie die Low-Level-Funktionen einsetzen, um die korrekten Daten zu prüfen. Beginnen Sie damit, den Header und anschließend die Daten der .DBF zu prüfen, um festzustellen, ob die Headerinformation den Daten entspricht und ob die Daten vom richtigen Typ sind. Einige Datentypen sind schwieriger auf Korrektheit zu prüfen, beispielsweise Character- und besonders Memofelder, da sie die Speicherung der meisten unterschiedlichen Zeichen ermöglichen.

Datenfehler treten häufig auf, wenn Datensätze von einer Workstation auf den Dateiserver übertragen werden. Prüfsummen können helfen, sicherzustellen, dass der Datentransfer korrekt über die Bühne gegangen ist und sollten einen großen Datensatz immer begleiten, wenn er von oder zu der Workstation gesandt wird. FoxPro verfügt über eine Prüfsummenfunktion, Sie können sich aber auch Ihre eigene erstellen, um Ihre

eigenen wiederholbaren Ergebnisse zu erhalten. Routinen wie diese müssen so schnell wie irgend möglich ausgeführt werden und sollten in C geschrieben und über eine .PLB oder eine .FLL in FoxPro integriert werden.

---

## **K) Sicherheit**

---

Es ist wirklich möglich, große Datenmengen sicher zu speichern. Bei Gesprächen über solche Techniken sollten Sie auch unsere Lösung bedenken. Da FoxPro selbst über keine Sicherheitsmechanismen verfügt, müssen Sie experimentieren und kreativ sein, wenn Sie ein Sicherheitssystem implementieren. Nutzen Sie die Sicherheitsmechanismen des Betriebssystems vollständig aus – und fügen Sie anschließend Ihre eigenen hinzu. Dafür sind viele Schichten erforderlich, beginnend mit dem Administrator.

---

## **L) Power und Geschwindigkeit**

---

FoxPro ist eine mächtige Datenbank und Programmiersprache. Geschwindigkeit, eine offene Architektur sowie die Fähigkeit, C-Programme einzubinden, machen dieses Werkzeug auch für die größten Datenbankprojekte tauglich.