

Accessシステム 超リフォーム術

ポイントを抑えて楽々リフォーム

第
2
回

データベース分割と ネットワーク共有化

T'sWare

星野 努 HOSHINO, Tsutomu

<http://homepage1.nifty.com/tsware/>

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Level



Samples



はじめに

今回は、レガシーシステムからAccess 2002へのバージョンアップ（コンバート）について簡単に説明しました。これは、Access 95/97から2002という、いわばAccessという製品のアップデートに追従するための“縦方向”へのリフォームといえるものです。

今回は、単一のデータベースファイル、つまり「スタンドアロン」で運用しているデータベースシステムを複数のユーザーがネットワーク上で共有できるようにする、いわば“横方向”の拡張ともいえるリフォーム術について取り上げます。

単一データベース ファイルの問題点

まず、単一データベースファイル(.MDB)に、テーブルやクエリをはじめ、フォーム、レポート、マクロ、モジュールといったAccessのすべてのオブ

ジェクトを収めた場合の問題点について考えてみます。

この場合、ほとんどが実務に導入されてからの運用/保守上の問題点であり、基本的には開発する際の問題点はあまりありません。もし、みなさんが次ページに列挙した「単一データベースファイルの問題点」のいずれにも該当しないのであれば、あえて今回説明するリフォームを施す必要はありません。

データベースを分割する

さて、この単一データベースファイルによるシステムの管理/運用の問題点を克服する手段が「データベース分割」です。

これは、簡単にいうと、データ部分(テーブルのみ)とアプリケーション部分(クエリ、フォーム、レポート、マクロ、モジュール)を別々のデータベースファイルに分割し、アプリケーション部分のデータベースファイルから、

単一データベースファイルの問題点

問題点1 ▶ アプリケーションの変更が必要となった場合、すでにユーザーが使っているデータベースファイルを使って変更やテストを行なわなければならない。その間、ユーザーの業務を停止させる必要がある（あるいはあなたが残業や休日出勤する）。

問題点2 ▶ 開発者用のデータベースファイルまたは運用ファイルのコピーを使ってアプリケーションの変更を行なった場合、その都度、完成時点の運用データを取り込んで、本番用に移行する必要がある。

問題点3 ▶ データベースファイルを複数のユーザーに配布している場合（たとえばデータ共有してなくても）、システム変更があったときに上記の問題点1または問題点2の作業をすべてのユーザーに対して行なわなければならない。

問題点4 ▶ Accessデータベースファイルのサイズの上限はすべてのオブジェクトの総量によって決められるが、アプリケーション部分の増大によってデータ部分の領域が狭められる。実際、フォームやレポートはデータに比べて意外とサイズが大きい。

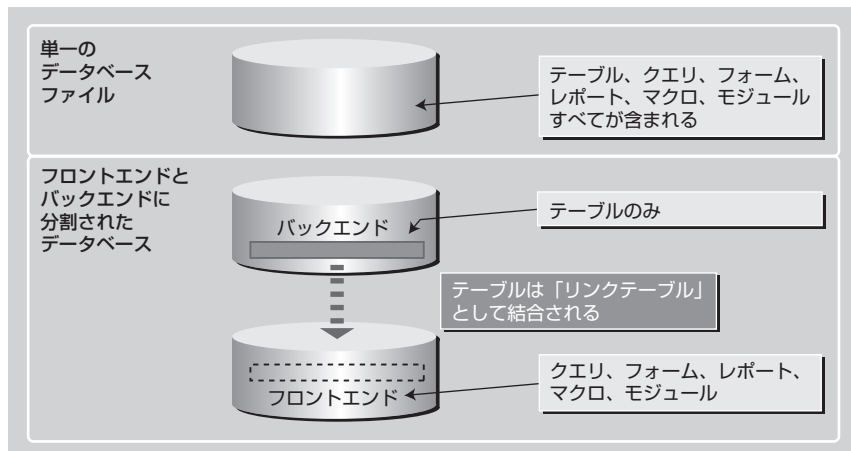
問題点5 ▶ 本来、定期的にバックアップすればよいのは随時更新されるデータ部分だけなのに、バックアップツールではアプリケーションも含めたデータベースファイル全体を毎回バックアップしてしまう。

問題点6 ▶ アプリケーション側でのトラブルでデータベースファイルが破損した場合、データも一緒に壊れる可能性が大きい。

問題点7 ▶ 基本的にネットワークでの共有はできない。ファイルサーバー上にデータとアプリケーションの両方を持った単一データベースファイルにおいてファイル共有するという強引な方法もあるが、パフォーマンスやネットワーク上の他のトラフィックへの悪影響などから一般にはご法度とされる。

*) 「データ」はテーブルを、「アプリケーション」はフォーム、レポート、マクロ、モジュールを指します。

図1：データベース分割の概念



注1) Accessではこのような分け方で表現していますが、ユーザーインターフェイスという意味合いからすると、クエリや、テーブルを操作するプロシージャなどもバックエンドといったほうが正しいかもしれません。SQL Serverではそれらもビューやストアドプロシージャと呼ばれるサーバー側のオブジェクトになっており、文字通りバックエンドとして扱われます。これについては本連載でも詳しく取り上げる予定です。

データ部分のデータベースファイルのテーブルを「リンク」することによってシステム全体を構成する方法です(図1)。

以降では、分割後の2つのデータベースファイルを区別するため、データ部分を「データ データベース」、アプリケーション部分を「アプリケーション データベース」と呼ぶことにします。

一般に「データ」データベースを「バックエンド」、「アプリケーション」データベースを「フロントエンド」と呼びます。システムを操作するユーザーから見た場合、ユーザーインターフェイスとなるフォームやレポートが前面にくるので、このような呼び方をします^[注1]。

フロント/バックエンド

データベース分割のもっとも大きなメリットは、Accessのスタンドアロンシステムを「LANでのデータベース共有システム」へと拡張できる点にあります。問題点7で述べた強引な方法を除けば、このデータベース分割した形がネットワーク共有の基本形となります。先月号で提示したロードマップにおいても、この分割した形態がC/Sでも応用されていることが確認できると思います。

具体的には、フロントエンドとバックエンドの各ファイルは図2のようにネットワーク上に配置することになります。