

.NET Frameworkで作る Windows サーバー

作ればわかる アプリケーションの動作とメカニズム

秋月 巖 AKIZUKI, Iwao <秋月巖ソリューション事務所> ▶ <http://www.akizuki.co.jp/>

第18回

SQL Serverとクライアントプログラムを中継する SQLプロキシサーバーを作る—その3—

Jet DB エンジンを DBサーバー化する

前号でも説明したようにJetデータベースサーバー (図1) とは、Jetデータベースエンジンをデータベースサーバーとして使用するものである。図2のクライアントプログラムからアクセスして動作を確認できるが、ADO.NETを使用したSQL Server用のクライアントからなら

ば、何でもアクセスできる。ただし、まだ制限が多いので既存のクライアントプログラムを使用した場合は、正常に動作しないことが多いだろう。

Jetデータベースをファイル共有型としてネットワーク経由で通常使用している場合、ファイルサーバーを利用してデータベースサービスを実現しているということもできる。この場合、もちろんSQL ServerやOracleといったデータベースサーバーとは違い、検索に必要なデータのすべてをファイルサーバーに要求することになる。最悪の場合、すべてのデータがネットワークを走ることになる。ここが、リクエスト (SQL文) に対して結果だけを返信するデータベースサーバーとの最大の違いである。もちろん、実際にはJetデータベースエンジンとSQL Serverの違いはこれだけではないのだが、構造的な最大の違いはここにあるということができる。

ちなみにJetデータベースエンジンを使ってWebアプリケーションを開発した場合、Webサーバーがサーバーとなることで、データベースサーバーと同じように動作する。Webブラウザが送信したリクエストに対し、Webサーバーは結果だけを返信する。Internet ExplorerのDHTMLの機能を使えば、サーバーからテーブルの内容を全部受け取り、クライアントで検索するというシステムが作れることを、ふと思いましたが、これはかなり特殊な例だといえるだろう。ちなみに、こ

合、すべてのデータがネットワークを走ることになる。ここが、リクエスト (SQL文) に対して結果だけを返信するデータベースサーバーとの最大の違いである。もちろん、実際にはJetデータベースエンジンとSQL Serverの違いはこれだけではないのだが、構造的な最大の違いはここにあるということができる。

レベル >>> Level

1 2 3 4 5

ツール >>> Tool

- Visual Studio 2005 Professional
- SQL Server 2000
- SQL Server 2005
- Access 2003

言語 >>> Language

- Visual Basic

サンプル >>> Sample

この記事で取り上げたソースコードおよびサンプルプログラムは、<http://www.shoeisha.com/mag/windev/>からダウンロード可能です。

の方法を用いたシステムは、経験的にいって案外うまく動作する。今日の回線状況なら、数十キロバイト程度のデータをダウンロードするのに大して時間はかからないし、デビュー以来ほとんど進歩しないDHTMLは、けっこう安定している。ただし、もちろんInternet Explorerに依存したシステムになってしまう。

さて、Jetデータベースサーバーは、次のような特徴をもつ。

- ①SQLプロキシサーバーのカスタマイズとして実装されている。
- ②クライアントのプログラミングにADO.NETを使うことができる。

まあ、①だから②なわけである。SQLプロキシサーバーは、前号まで作ったSQL Serverのプロキシ機能を実現するものである。前号では、カスタマイズのためのプロシージャを独立して用意した。だから、Jetデータベースサーバーの機能は、ほとんど、このプロシージャに記述することになる。つまり、SQLプロキシサーバーの実装に関する知識が十分になくても、カスタマイズされたコードを理解できるということである。Visual Studioの「Windows フォームデザイナーで生成されたコード」の部分の内容が理解できなくても、Visual Studioを使ってプログラムが作成できるようなものである。

前号では、今回のJetデータベースサーバーの原型をサンプルプログラムとして提供したが、半角の文字列型だけの対応だった。今回のものは、2バイト文字も対応しているし、数値型も扱うことができる。しかし、それ以外のデータ型には未対応である。日付型の場合、SQL Serverのヘルプにデータ形式が詳しく説明されているので、対応は案外と簡単かもしれない。いずれにしても、どれも完全に対応できるかどうかはわからないのだが、できる範囲で努力してみよう。

Jetデータベースサーバーは実用のためというより、SQL Serverの研究用に開発している部分が多い。実装するためにSQL Serverのプロトコルを調べる必要があるからである。また、そのノウハウは、SQLプロキシサーバーの別のカスタマイズに有効に活かすことができるはずである。

図1：SQLプロキシサーバーをカスタマイズして作ったJet DBサーバー

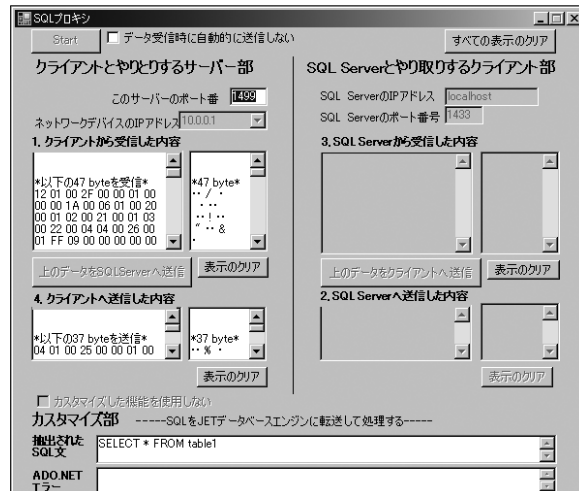


図2：クライアントプログラムの例



Jetデータベースサーバーの実装その1

～AppSQLPrxyStartプロシージャ

SQLプロキシサーバーのカスタマイズ用として「AppSQLPrxyStart」プロシージャと「AppSQLPrxyReceiveFromClient」プロシージャを用意したことは前回説明した。前者は初期化処理で、後者はクライアントからリクエストを受け付けたときに実行する処理である。通常、ほとんどのカスタマイズ処理は、この「AppSQLPrxyReceiveFromClient」プロシージャに記述することになるだろう。Jetデータベースサーバーの場合も同様である。

リスト1はJetデータベースサーバーの初期化処理をしている「AppSQLPrxyStart」プロシージャである。

ここではJetデータベースエンジンを使用するための準備をしている。OleDbConnectionオブジェクトのOpen