



特集 **1**  
SPECIAL FEATURE

初めての  
データベース  
アプリケーション

3

# 更新操作の工夫でデータベースを使いやすく クエリをカスタマイズして データベースとのやりとりを最適化

大澤 文孝 OSAWA, Fumitaka

## はじめに

ADO.NETでデータベース操作をするときにもっとも厄介な問題は、「データベースからのデータの取得」と「データベースへの更新操作」です。

ADO.NETの非接続型データアクセスでは、データベースからメモリへと読み込み、メモリ上で変更して、あとで書き戻すという手法をとります。

データベースに接続したままデータを変更するわけではないので、データベースへと書き戻すときに、誰かが「すでに変えていた」とか「削除していた」という事態は、容易に起こり

得ます。このとき、無理矢理書き込むと、上書きが起きたり、消されたはずのレコードが復活したりしてしまいます。

また、DataGridViewコントロールなどでは、メモリ上のデータとデータバインディングするので、データ量が多い場合には、いくつかの問題を抱えます。

たとえば数十万件のレコードが存在するデータベースでは、全データをメモリ上にとってくることはできず、ある程度に分割して読み取って出力するといった工夫が不可欠です。

本稿では、このようなデータベースの更新に関する問題と、その問題を解決するためのいくつかの方法を紹介します。

## ADO.NETの構成と問題点

レベル >>> Level

1 2 3 4 5

言語 >>> Language

- Visual Basic
- SQL

ツール >>> Tool

- Visual Studio 2005 Professional
- SQL Server 2005 Express

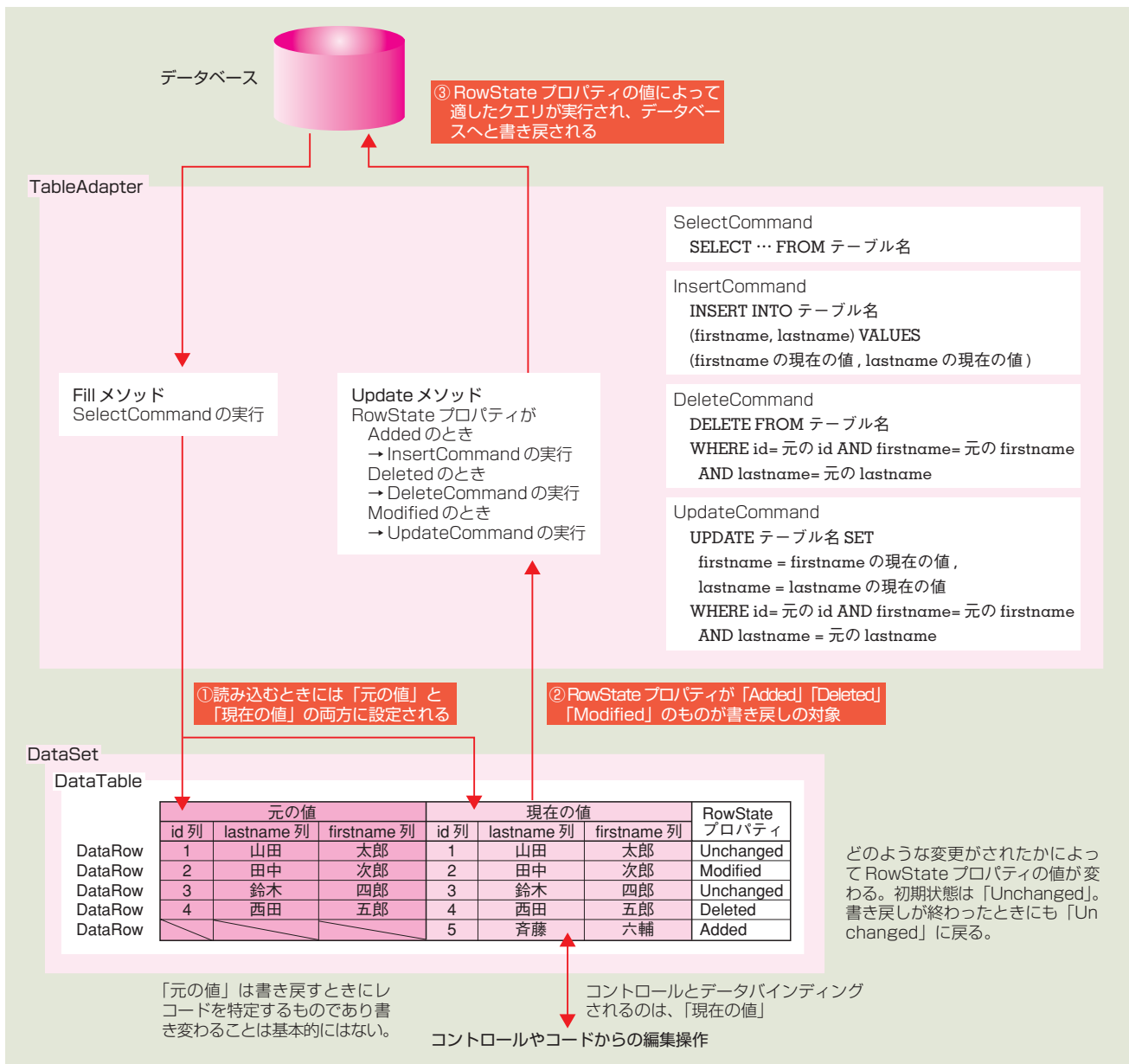
まずADO.NETでは、データベース更新処理がどのように実装されていて、何が問題となるのかという点から見ていきましょう。

### DataRowの詳細

ADO.NETにおけるデータベースの操作は、図1に示すオブジェクトが絡

SPECIAL FEATURE

図1：ADO.NETにおけるデータベース操作の流れ



みます。

Visual Studio 2005のWindows フォームを使って開発する場合、データベース操作の中心となるのは、TableAdapter オブジェクトです<sup>[注1]</sup>。

TableAdapter オブジェクトには、データベースから読み取るための Fill メソッドとデータベースへと書き戻すための Update メソッドがあります。

データベース操作で重要な役割をす

るのが、1レコードを表現する DataRow オブジェクトです。

### データベースからの読み取り

DataRow オブジェクトの内部には、「元の値 (Original値)」と「現在の値 (Current値)」の2つの値をもっています<sup>[注2]</sup>。

Fill メソッドを呼び出したときには、SelectCommand プロパティに設定され

注1) 本稿ではウィザードで作成された Table Adapter オブジェクトを活かした開発をしていきます。しかし.NET Frameworkにおいて、データベース操作の基本となるのは、DataAdapter オブジェクト (SqlDataAdapter オブジェクト) です。実際、TableAdapter に備わる各メソッドのデフォルトの動作では、DataAdapter オブジェクトの同名のメソッドを呼び出すだけの動作をしています。ヘルプを調べるときには、DataAdapter オブジェクトに関する情報も参照してください。

注2) BeginEdit メソッドを呼び出して編集を開始したときには、その編集をあとで取り消せるようにするために、さらに「編集中の値 (Proposed 値)」を使えます。