

# 完全掌握

DBアプリケーションなんて  
カンタンだ!

# SQL Server プログラミング

第

4

回

## DataSetオブジェクト を使ってみよう

# 再入門

株式会社システムインテグレータ

湯尾 守 YUO, Mamoru

<http://www.sint.co.jp/>

### Technology Tools

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

### Level



### Samples

・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、付録CD-ROMの¥DOTNET¥SQLServerディレクトリに収録しています。

#### ¥REGISTEMP2

今回作成したアプリケーションのソースファイル

#### ¥SQL

テーブルとサンプルデータを作成するSQLスクリプトの収められたファイル

### はじめに

ADO.NETが、かつてのADOから進化した点でもっとも特徴的な部分は「非接続環境におけるデータの扱いが考慮されている」という点です。つまり、データベースなどのデータ元に依存しない環境でのプログラミングをサポートする機能を持っているということです。このことによって、データベースであってもXMLデータやWebサービスであってもデータの種類の柔軟なプログラミングが可能になります。その中心となるオブジェクトがDataSetオブジェクトです。今回のテーマは、

DataSetオブジェクトの基礎を身につける  
です。

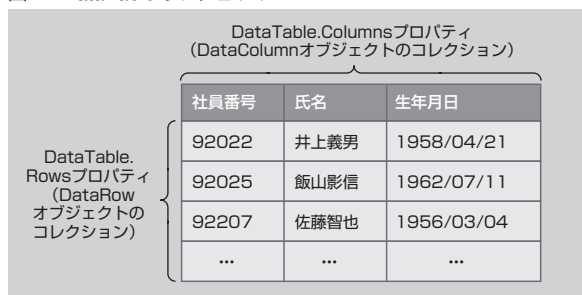
### 非接続環境で利用する オブジェクト

非接続環境とはいってもADO.NETのオブジェクト群はSQL Serverなどのリレーショナルデータベースが強く意識されています。たとえばテーブルに相当するのがDataTableオブジェクト、参照整合性制約に相当するのがDataRelationオブジェクトといった具合です。つまり、リレーショナルデータベースの機能を“データベースに接続することなく使用する”といったイメージになります。それでは主なオブジェクトについて見てゆきましょう。

### ■ DataTableとDataColumn

先ほど述べたとおり、DataTableオブジェクトはリレーショナルデータベースの「テーブル」に相当します。つまり、データを表形式で格納するオブジェクトです。そしてテーブルの「列」に当たるのがData

図1: DataTableオブジェクト



Columnオブジェクトです(図1)。

データベースではCREATE TABLE文で列とテーブルを同時に作成しますが、DataTableオブジェクトではまずコンストラクタでDataTableオブジェクトを作成した後、列のコレクションであるColumnsプロパティを利用してテーブルに列を追加します。

```
Dim employeesTable As New DataTable("社員")
employeesTable.Columns.Add("社員番号", GetType(String))
employeesTable.Columns.Add("氏名", GetType(String))
employeesTable.Columns.Add("生年月日", GetType(Date))
```

このコードでは「社員」という名前のテーブルを作成した後、そのテーブルに「社員番号」「氏名」というString型の列と「生年月日」というDate型の列を登録しています。

リレーショナルデータベースのテーブルと同様に主キーや制約を設けることも可能です。次のコードではColumnsプロパティを利用してDataColumnオブジェクトにアクセスして「社員番号」を主キーに設定し、「氏名」は20文字以内という制約を設けています。

```
employeesTable.PrimaryKey = New DataColumn() _
{employeesTable.Columns("社員番号")}
employeesTable.Columns("氏名").MaxLength = 20
```

その他にもDataColumnオブジェクトには一意制約(Uniqueプロパティ)やNOT NULL制約(AllowDBNullプロパティ)を設定したり、読み取り専用の属性を付加したりすることもできます。主キーの列には自動的にNOT NULL制約と一意制約が設定されます。また、複

数の列の組を主キーにすることも可能です。

## DataRow

リレーショナルデータベースの「レコード」に相当するのがDataRowオブジェクトです。DataRowオブジェクトはコンストラクタが公開されていないので、新しいレコードを作成するときはDataTableオブジェクトのNewRowメソッドを使用します。DataRowオブジェクトを作成し、DataTableオブジェクトに登録するコードを以下に示します。

```
Dim employeeData As DataRow = employeesTable.NewRow()
employeeData("社員番号") = "92022"
employeeData("氏名") = "井上義男"
employeeData("生年月日") = New Date(1958, 4, 21)
employeesTable.Rows.Add(employeeData)
```

NewRowメソッドを利用するとDataTableオブジェクトと同じ列を持つDataRowオブジェクトが作成されます。一意制約やNOT NULL制約はDataTableに登録する段階、つまり上記のコードでは最後の行で制約のチェックがなされます。このとき制約を満たさないDataRowオブジェクトを登録しようとすると例外が生成されます。

このコードを見てもわかるとおり、DataTableオブジェクトに格納されているレコードはRowsプロパティにコレクションとして登録されます。特定のレコードを取得するには主キーを利用するのが便利です。

次の例は先ほどのレコードの生年月日を変更するコードです。

```
Dim employeeData As DataRow = _
employeesTable.Rows.Find("92022")
employeeData("生年月日") = New Date(1958, 5, 21)
```

RowsプロパティのFindメソッドは主キーを使ってレコードを検索するメソッドです。複合キーの場合はキーの配列を引数に渡します。Findメソッドはレコードが見つからない場合「Nothing」を返します。

次の例は社員番号が「92」で始まる社員の生年月日を1日増やすコードです。