

親子関係のあるテーブルを 更新するための方法

一気に書き戻すか、レコード単位で書き戻すか?

大澤 文孝
OSAWA, Fumitaka

Technology Tools

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

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¥F01ディレクトリに収録しています。

¥DBCREATE

サンプルDB作成用のSQL

¥PRODUCTWEBSRV

Webサービスプロジェクト

¥PRODUCTWEBAPP

Webアプリケーションプロジェクト

¥PRODUCTWINAPP

Windowsアプリケーションプロジェクト

¥PRODUCTEXCEL

Excelワークブックプロジェクト

※サンプルの利用/実行方法については、F01ディレクトリのReadme.txtを参照してください。

親子関係付きテーブルの 一括書き換え

さて、今度は、親子関係があるテーブルの更新について考えます。

「Webサービスで作るデータベースアプリケーションの基本」で示したように、注文を示すordersテーブルとその明細を示すdetailsテーブルは親子関係になっています。

もちろんこのとき、ここまで説明してきたと同様の方法で、1レコードずつ書き換えるユーザーインターフェイスをとってもかまいません。

しかし注文と明細の関係だと、ひとつひとつ明細をデータベースに逐一書き戻すのではなく、「注文と明細」をまとめて書き戻したいこともあります。

そこで、ここでは、図1に示すユーザーインターフェイスを考えます

図1のユーザーインターフェイスでは、注文一覧(図1の(a))の各行の右に「明細編集」というリンクがあり、それをクリックすると、明細の編集画面(図1の(b))に飛べるようにしてあります。

明細の編集画面では、明細を1レコードずつ編集の都度データベースに書き込むのではなく、一時的に記録しておき、「データベースへ書き込む」ボタンがクリックされたときにだけ、一気に書き戻すようにします。

また、「編集結果を破棄して読み直す」をクリックした場合には、現在の編集結果を破棄して、データベースから再度データを取り直すようにします。

このようなユーザーインターフェイスをとることで、ユーザーは、データベースに書き戻すことなく、明細を一度にまとめて編集することができるようになります。

注文から明細のレコードを呼び出す

誌面の都合上、注文の編集機能(図1の(a))については、商品の編集とほぼ同様なので、ここでは解説しません。実際のコードについては、付録CD-ROMに収録しているサンプルを参照してください。

なお、注文編集画面にある「明細編集」のリンクは、次のように、URLク

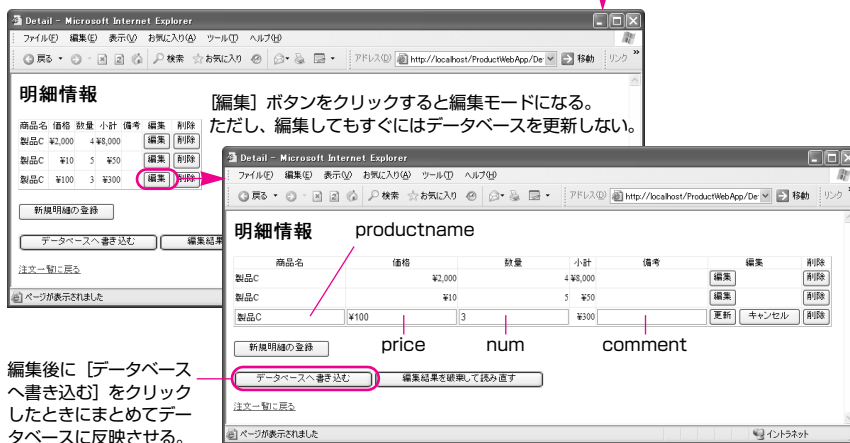
親子関係のあるテーブルを更新するための方法

図1: 注文編集と明細編集

(a) 注文の一覧表示



(b) 明細の一覧表示



編集後に「データベースへ書き込む」をクリックしたときにまとめてデータベースに反映させる。

エリを使って注文IDを埋め込むように構成してあります。

```
<a href="Detail.aspx?id=注文ID">
  明細編集
</a>
```

このとき、明細編集画面 (Detail.aspx) では、次のようにして、編集対象となっている注文IDを得ることができます^[注1]。

```
Dim orderid As Integer
orderid = CInt(Request.Query("id"))
```

注1) このような親子関係のテーブルを表示するユーザーインターフェイスの詳細については、PASSJの「ステップアップ! ADO.NET」(<http://www.sqlpassj.org/bunkakai/web/series/ado/01.html>)などを参照してください。

注文と明細の一括取得と Session オブジェクトへの格納

では、明細編集の画面の処理を詳しく説明していきます。

注文と明細を返す Web サービス

明細編集の処理では、編集対象となっている注文IDをもつ注文の明細をデータベースから取得して、DataSet コントロールにデータバインディングする処理が必要です。

そこでまず、Web サービス側に、注文IDを渡すと、その注文のレコードと、子になっている明細のレコードを DataSet オブジェクトとして返すメソッドを実装しておきます (リスト1)。

リスト1では、引数として注文IDを受け取り、orders テーブルに対する SELECT クエリと、details テーブルに対する SELECT クエリの2つを送信し、その2つの結果を含めた DataSet オブジェクトを返しています。

ちなみに親子関係がある場合には、必ず、親のテーブルから先に読み込みます。もし子から先に読み込むと、子が指す親が DataSet オブジェクト上に一時的に存在しない状況が生じるので、DataSet オブジェクト上でリレーションシップを構成している場合 (38ページの図7を参照) には、例外が発生します。

リスト1では、少々 SELECT クエリが長いのですが、これは、注文にはその明細の合計金額を、明細には該当商品の商品IDだけでなく商品名も、そして単価×数量の小計も含めるようにしているためです。

合計金額の列は「total」、商品名は「productid」、小計は「subtotal」という列名とします。

これらの列は、あらかじめ、図2のように、XMLスキーマの編集画面で追加しておきます。

この例のように、XMLスキーマに列を追加することで、DataSet オブジェクトに保存できる値を増やすことができます。

ここでは計算した値を保存するために列を追加したわけですが、たとえば、アプリケーションで一時的な値を保存