

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

再入門

第 7 回

ストアドプロシージャを
使ってみよう

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。

¥CALLPROCEDURE

Windowsアプリからストアドプロシージャを呼ぶサンプルアプリケーション

¥SQL¥CalculateCategorySales.sql
今回作成したストアドプロシージャのCREATE文

¥SQL¥SalesForYear.sql

「SalesForYear」テーブルのCREATE文

はじめに

ストアドプロシージャと聞くと極端に難しいことのように感じている人もいるのではないのでしょうか。しかし、結論から言うとSQLさえ知っていればストアドプロシージャは決して難しいものではありません。慣れてしまえばいつでも簡単にテーブルを参照できる分、書きやすいとも言えます。そこで今回のテーマは、

ストアドプロシージャを使ってみる
です。

ストアドプロシージャとは?

III. .NETアプリケーションとストアドプロシージャ

今まで作成したアプリケーショ

ンは単純にテーブルの値を照会したり、更新するだけのものでした。しかし、データベースアプリケーションでは値を集計したり、状況に応じて処理手順が異なったり、更新するテーブルが変わることもあると思います。簡単な集計ならSQLの集計関数で事足りるかもしれませんが、通常このような処理はSQLの範囲を超えているので、必要な情報をあらかじめSELECTしておき、アプリケーション側で集計や各種の処理を行ない、それに応じたSQLを発行する必要があります。

Visual Studio .NETを使って作成したアプリケーションは、Visual Basic .NET (以下VB.NET) や Visual C# .NET (以下C#) など記述され、.NET Framework上で動作するわけですが、ストアドプロシージャは「データベース側にこれらのロジックを実装できるようにSQLを拡張したもの」です。つまり、処理の実行に必要なパラメータをスト

アドプロシージャに渡せば処理はすべてデータベース上で動作します。

■ ストアドプロシージャの長所と短所

単純にデータベースに対してSQLコマンドを発行すると、そのSQLは解析され、その後にSQLが実行されます。これに対し、ストアドプロシージャはデータベースに登録したときにコンパイルされ、実行時はそのコンパイル済みのコードを実行するため、実行時にSQL解析の処理は必要ありません。その分パフォーマンス的に有利だと言えます。そもそもSQLは言語的にVB.NETなどと比べて非常に複雑なため、解析に必要な時間がバカになりません。複雑なSQLであればこの違いは非常に大きなものになります。

また、ストアドプロシージャはストアドプロシージャごとにアクセス権を設定することが可能です。これは、テーブルに対してSELECT、INSERT、UPDATE、DELETEといった大雑把なくくりでしかアクセス権を規定できないのと大きな違いです。同じテーブルに対するUPDATE文を発行するストアドプロシージャでも異なるアクセス権を与えることができるのです。

これは、通常のテーブルに対してはあまり意味のないことのように思えるかもしれませんが、システムを管理するような重要なテーブルに対するアクセスを制限する場合に非常に効果があります。現にSQL Serverではシステムを管理するテーブル操作のためのストアドプロシージャを数多く提供していますが、これはテーブルが不用意に操作されてデータベースが動かなくなるのを防ぐためです。

もちろん、ストアドプロシージャに欠点がないわけではありません。

ISOやJISで規定されている標準のSQLには、VB.NETのIf文やWhile文などの条件分岐はありません。つまり、標準のSQLでアプリケーションの処理ロジックを記述するのは不可能です。そこで、SQL Serverでは「Transact-SQL」というSQLの拡張言語を使ってストアドプロシージャを記述します。このTransact-SQLはISOやANSIなどの標準化団体によって標準化された言語ではないの

で、他のデータベースではまったく使用することができません。Oracleでは「PL/SQL」という言語を使ってストアドプロシージャを記述しますが、このOracle上のストアドプロシージャをSQL Server上で動作させることもできません。つまり、ストアドプロシージャにしてしまうとデータベースを容易に変えることができなくなってしまいます^{注1)}。

また、Transact-SQL言語を知らないとプログラミングができません。たとえTransact-SQLを使いこなせるようになったとしても、SQL Serverというプラットフォームでなければその知識を活かすことはできません。ストアドプロシージャに対して二の足を踏む人が多いのはこのためでしょう。

ストアドプロシージャ入門

■ Transact-SQL

新しい言語を学ぶことに抵抗のある人も、「VB.NETだけでもういっぱいばいばいだよ」という人も安心してください。Transact-SQLは非常に簡単な言語の上、Visual Basicに似た言語構造になっています。私は新しい言語に抵抗を感じない人種なのでこの意見は参考にならないのかもしれませんが、まわりにいる初めてTransact-SQLを使うプログラマを見ても言語自体に苦労している人はほとんどいません。むしろVB.NETのほうが言語的には苦労しているようです。

Transact-SQLは標準SQLを拡張した言語なので標準SQLをそのまま使えます。標準SQL同様、大文字と小文字は区別されません。「-」から行の終わりまではコメントとして扱われます。また、標準SQLにはありませんが、「/*」と「*/」で囲まれた部分もコメントです。

改行は行コメント以外の場所では単なる構文要素の区切りの意味しかありません。したがって、文の途中で改行する場合でもVB.NETのように「_」をつける必要は

注1) データの取得にのみ標準のSQLを使用し、具体的な処理をVB.NETで記述しておけば、標準のSQLをサポートするあらゆるデータベースに対応するアプリケーションを作成することができます。