

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

最終回

アプリケーションを
パフォーマンス
チューニングしてみよう

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

パフォーマンスは 大事

データベースアプリケーションにパフォーマンスの問題はつきものです。しかし、アプリケーションの処理が遅いとわかっていても実際どこから手をつけていったらよいのか、最初はなかなかわからないものです。

そこで今回のテーマは、

パフォーマンスチューニングをしてみよう

です。

最高のパフォーマンス を実現するには

アプリケーション開発の現場では、アプリケーションが完成に近づくとも必ずパフォーマンスの問題が話題に上ります。それに反して本

屋に行ってもパフォーマンスチューニングをテーマとした本はほとんど置かれていないのが実情です。

これにはいくつか理由が考えられます。まず、どの場面でも適用できるような普遍的な方法がパフォーマンスチューニングにはあまりないこと。ある場面ではパフォーマンスが改善したのに、同じ方法を別の場面で利用してもほとんど効果がないといった話はよく聞きます。「パフォーマンスチューニングは経験だ」などという人もいるくらいです。

経験がものをいう場面が多いことも事実です。しかし、一目見て明らかにパフォーマンスに問題があることがわかるコードもあり、そのようなコードを書かないようにすることがパフォーマンスに優れたアプリケーション作成の第一歩です。

一目見て明らかに問題のあるコードがある一方で、じっくり見ても問題点がなかなかわからないコー

ドもあります。そのようなコードではいろいろ試行錯誤を繰り返し、もっともパフォーマンスのよい方法を探る必要があります。

ここで必要なのは「こうするとパフォーマンスがよくなる」という知識ではなく、「こうするとパフォーマンスがよくなる可能性がある」という知識です。むしろ「こうするとパフォーマンスがよくなる」と思い込むことは試行錯誤の幅を狭め、結局最善のパフォーマンスが得られないこととなります。この段階になると実際に動かしてみなくては判断がつかないと思ったほうがよいでしょう。

テーブルへ高速に アクセスするために

III 行で持つか、列で持つか

毎月の総売上額を格納するテーブルを用意する場合について考えてみましょう。普通に考えると表1のようなテーブルを作ることになると思います。しかし、月別にレコードを作成するのではなく、年度別にレコードを作成する表2のようなテーブル設計も考えられます。果たしてどちらの設計がよりよいのでしょうか。

柔軟性からすると表1のほうが優れています。たとえば、半月単位に集計するようにシステムを変更してほしいという依頼があった場合、表1のような設計であれば対象年度の列をnchar(6)からnchar(7)に変更して2004年10月上旬なら「200410a」という具合に1文字付加するだけでテーブルの変更はおしまいです。このような拡張に備えてはじめてから対象年度の列をnvarchar(10)にしておくという方法もよく採られます。それに対して表2の設計の場合は列を12個追加する必要があり、お世辞にも簡単に変更できるとは言えないでしょう。

ところが、パフォーマンスの点から考えると表2のほうが優れていることが多いようです。情報を別レコードとして持つか、列として並べて持つかという問題はどちらの設計でもパフォーマンスはほとんど変わらないと考える人もいますが、私が以前SQL Server 2000で試してみたときは検索速度に大きな違いが出ました。しかしこれは、データの質や量の影響を受ける可能

表1: 「月別売り上げ」テーブルの定義1

列名	データ型	長さ	備考
対象月度	char	6	200410のように年月だけを保持
売上金額	money		

表2: 「月別売り上げ」テーブルの定義2

列名	データ型	備考
対象年度	int	2004のように年だけを保持
4月売上金額	money	
5月売上金額	money	
6月売上金額	money	
7月売上金額	money	
8月売上金額	money	
9月売上金額	money	
10月売上金額	money	
11月売上金額	money	
12月売上金額	money	
1月売上金額	money	
2月売上金額	money	
3月売上金額	money	

性が大きいので必ず列で持つほうがよいとは言いきれません。レコードが少ない場合はほとんど違いは出ないでしょう。いずれにせよ、そのアプリケーションで想定しているデータを使用してテストをしてみる価値はあります。

III 非正規化とインデックス付きビュー

この連載ではテーブル設計については一切触れてきませんでした。テーブル設計は直接パフォーマンスに影響します。データベースのメンテナンス性の向上などを目的に、テーブル設計するときには正規化されるのが普通ですが、この正規化はパフォーマンスに悪影響を及ぼします。

正規化とは「一事実一箇所」という考えに沿って、他のレコードから取得できる値を保持しないようにすることですが、いろいろなテーブルのいくつかのレコードを集計した値を頻繁に利用することがわかっている場合は、その集計した値を保存する列を用意したほうがよいこともあります。もちろんこうした非正規化はデータベースのメンテナンスに悪い影響を与えます。本来は1レコードを更新すればよい場合にそのレコードの値に依