完全掌握 SQL Server プログラミング再入門

DBアプリケーションなんてカンタンだ！

第5回 データの一貫性について考えよう

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

★★½☆☆

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。

¥EMPGRID
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

データベースに対してコマンドを実行するときのみ接続するような「非接続環境」では、データを閲覧しているユーザーは“閲覧中のデータの変更をただちに知る”というようなことは基本的にできません。ただし、Webアプリケーションでは接続を保持し続けることは現実的ではありません。そこで、ADO.NETではこのような環境で使用するための機能も提供しています。

しかし、これらの機能を利用すれば非接続環境であることを意識せずにプログラミングできるかというと答えは「否」です。データの一貫性を保つためにはADO.NETの提供する機能を理解した上で、データの整合性を保つ仕組みを考える必要があります。今回のテーマは、

非接続環境でのデータの一貫性を考える

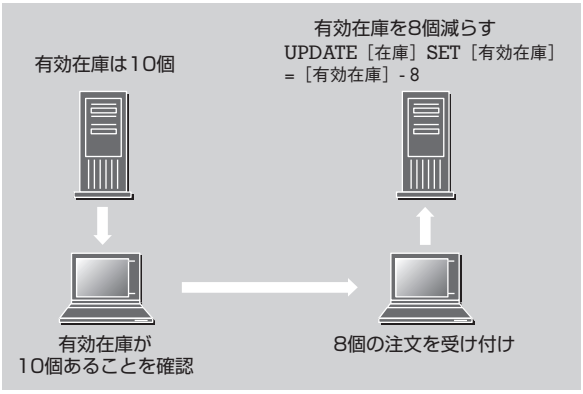
です。

データの一貫性を保つには

一貫性を保つことの必要性

例として受注管理システムを考えてみましょう。このシステムはWebアプリケーションで構成されており、Webブラウザからデータを入力するようなものとします。Webアプリケーションなのでデータベースとのやり取りはブラウザ上からリクエストを発生させたときにしか行なえません。また、Web環境では“ブラウザが閉じられた”ということをWebサーバー側で知ることはできません。つまり、Webサーバーがデータベースとの接続を保持したままにすると接続を解放することができない可能性があるので、データベースとの接続を保持

図1：受注システム



したままにすることはできないのです。

このシステムは在庫の状況を照会し、在庫がある場合は注文を受け付け、その分の有効在庫を減らすというものです（図1）。仮にAとBという2人が別の端末でこのシステムを同時に使用するケースを考えてみましょう。はじめは有効在庫が10個であったとします。

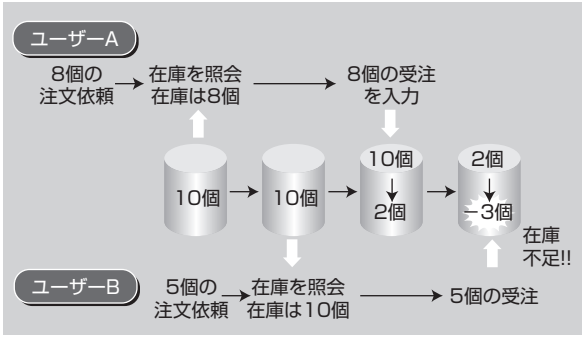
まず、Aさんが8個の注文依頼を受けたので在庫を照会します。Bさんが次に5個の注文依頼を受けたので、Bさんも在庫を照会します。在庫を照会した結果、AさんもBさんも有効在庫が10個あるので注文を受け付けます。Aさんが先に8個の受注を入力したとすると、Bさんが入力するときにはすでに有効在庫は2個になっているのでBさんは注文を受け付けられないはずです。しかし、BさんはAさんの注文が受け付けられたことは知らないので在庫があると思って入力してしまいます。もし、有効在庫の更新時にチェックされない場合は有効在庫がマイナスになってしまうでしょう（図2）。

それでは在庫がマイナスにならないような制約をデータベースに付加すれば問題は解決するのでしょうか？

たとえば「有効在庫が100個以上あるときはどんどん注文を受け付けてもよいが、有効在庫が10個を切ったときは得意先からの注文以外は受け付けないようにしたい」という場合は、この制約では不十分です。人間がその時点での有効在庫数を確認した上で判断しなくてはいけないというケースはたくさんあります。

このようにSELECT文を発行したときとUPDATE文を発行したときとでデータが異なると問題が発生します。

図2：問題が発生するケース



リアルタイムで有効在庫数の変動を受注担当者に対して表示できるシステムならよいのですが、Webアプリケーションの場合はそうもいきません。そこで、このような問題点を解消するための仕組みを考える必要があります。

楽観的ロックと悲観的ロック

この問題はSELECT文とUPDATE文の間に他のユーザーがデータを変更することを許さなければ発生しません。つまり、更新の前段階としてSELECT文を発行した場合、その更新処理が完了するまで他のユーザーが更新できないように制限を加えればよいのです。

先ほどの例ではAさんが受注目的で在庫を照会したときにその行に対するロックをかけ、Bさんが在庫を照会したときには「他のユーザーが受注を入力しようとしているので今その商品の受注の入力はできません」というメッセージを表示するようにします。そうすれば、Aさんが受注入力完了後にBさんが再度、受注目的で在庫を照会したときには在庫はすでに2個になっているため、5個の受注は受け付けられないことがわかります。このように、更新目的でSELECTしたときに更新が完了するまでそのレコードの変更を許さない方法を「悲観的ロック」といいます。

この悲観的ロックは「一貫性の保持」という点では強力ですが、ブラウザを閉じられたことを知ることができないWebアプリケーションでの利用は難しいと言えます。というのは、在庫を照会した直後にブラウザを閉じられたり、ネットワークがつながらなくなった場合にロックを解放することができないからです。このような場合、永遠にその在庫データを変更できなくなってしまう