

Oracle + .NET

3つの言い



EMCジャパン株式会社
ソリューション・コンサルティング部
一志 達也 ICHISHI, Tatsuya
<http://www.emc2.co.jp/>

特別編

その2

Webサービスのトランザクション

Technology Tools

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

Level



Samples

トランザクションも 大事

今回は特別編その1として、Webサービスを用いるシステムにおいて、考慮すべきセキュリティの話題を取り上げました。公衆回線（インターネット）を用いる場合には、特にセキュリティに注意を払うべきだとはいえ、そのために特別な手法を使う必要はないことを理解していただけたのではないのでしょうか。

今回は特別編その2として、Webサービスを用いるシステムにおいて、考慮すべきトランザクションの話題を取り上げようと思います。

トランザクションが 重要なワケ

すでに理解している方も多くは思いますが、あらためてトランザクションとは何か、なぜ必要なのか

について考えてみましょう。

「トランザクション」とカタカナで表現すると、もうひとつピンときませんが、まずは「処理の単位」だと考えてください。この言葉は、概念を意味することもあります。システムの機能を意味することもあります。トランザクションは、システム上で行なわれる処理の概念でもあり、それを実現するための機能でもあるわけです。

では、トランザクションはなぜ重要で、なぜそれを意識しなければならぬのでしょうか。

ひとつの処理を始めてから、終えるまでを管理するための単位、それはシステムごとにさまざまなのが考えられます。たとえば注文処理を始めてから終えるまで、銀行の口座振り込みを始めてから終えるまで、列車や映画館の座席予約から発券までなどです。いずれにしても、そこで発生するデータを記録するためには、いくつかのステップ

を経なくてはなりません。

データの記録先として使われることの多いデータベースの場合であれば、データを記録するためのSQLが複数回発行されます。発行されるSQLのひとつひとつがステップであり、それらがすべて完了しなくては、処理は完結したといえません。万が一、そのステップが途中で中断されてしまったら、データベース上のデータの整合性が保てなくなってしまうでしょう(図1)。

そうならないようにするには、処理の開始地点を記録し、処理の終了地点に来るまでデータを確定しない仕組みが必要です。データベースのトランザクション機能はこれを実現しており、何らかの理由によって処理を中止したい場合には、「ロールバック命令」を発行すると、データの変更を取り消して処理の開始地点の状態へ戻します。オラクルデータベースでは、この仕組みを「Undoセグメント(ロールバックセグメント)」によって実現しています^[注1]。

ユーザーによる取り消しの要求やシステムの不具合、ハードウェアやソフトウェアのトラブルなど、処理を完結できなくなる原因はさまざまです。とはいえデータの整合性を保つためには、どのような原因であろうとも、完了していない(コミットしていない)トランザクションは必ず元に戻さなければなりません。オラクルデータベースでは、主に「REDOログ」を用いてこれを実現しています。

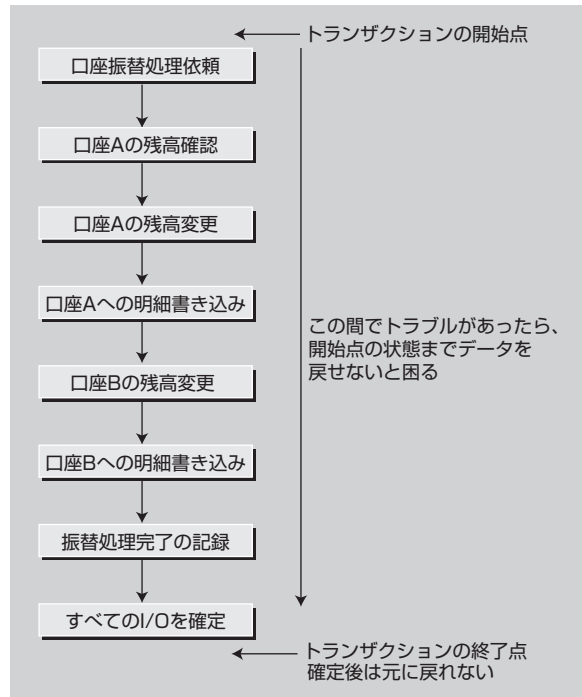
いずれにしても、確定していないデータを残さないこと、これがトランザクションが重要な役割を果たしているという理由です。

排他制御と入力データの記憶

トランザクションの役割や利用目的はデータの整合性を保つことだけなのかというと、決してそうではありません。

注1) ひとつのデータベース内で完結しないようなトランザクションを実行する場合、TPモニタのようなトランザクションを管理する専用の仕組みを利用します。

図1：トランザクションの必要性



次に大切なのは、変更中のデータ(行)を他のユーザーに変更させないこと、つまり排他制御(ロック)の実行です。データベースによって、選択できる排他制御のオプション数や粒度に差はありますが、トランザクションの実行中はロックが継続され、確定(コミット)または取り消し(ロールバック)が行なわれるまで解放されない点は共通しています。データベースを扱うプログラマは、この機能のおかげで自分が書き込みを保留しているデータを、他の人が書き換えてしまう危険性を排除できるわけです。

ところが、インターネット(HTTP)を利用するアプリケーションの場合、トランザクションの利用が制限されてしまいます。なぜなら、HTTPがステートレス、つまり毎回接続を解放するプロトコルだからです。

データベースのトランザクション機能を用いる場合、トランザクションの継続中はデータベースとクライアントの間で、セッションを保っていないとはなりません。ところが、HTTPの場合にはそれが実現できないために、トランザクションもサーバーにリクエストを送信するたびに確定しなければならないのです(図2)。もちろん、