特集3 ここがポイント! Oracle DB 開発

Oracle meets .NET

第3回

データの整合性を保障するために

トランザクションを極める(前編)

大田 浩
OTA, Hiroshi
日本オラクル株式会社
Oracle Direct テクニカルサービス部

はじめに

今月と来月にわたって「トランザクション編」として、.NETアプリケーションからOracleデータベースのトランザクション制御を適切に処理するための手法を解説します。

Level

1 2 3 4 5

Technology Tools

- ✓ Visual Basic
- ✓ Visual C#
- ○ Visual C++
- ○ SQL Server
- ○ Oracle
- ○ Access
- ○ ASP.NET
- ✓ Other:
  - ↓
  - Visual Studio .NET 2003
  - Oracle 9i、10g
  - Oracle Developer Tools for Visual Studio .NET

ご存知のとおり、データベースはもとより業務システムが行なう処理は、トランザクションの連続で成り立っています。それゆえ、トランザクションの理解は、DBAに限らず業務システムの構築／運用に携わるエンジニアにとって必須事項です。

今回は、トランザクションとは何かといった基礎知識から、Oracleデータベース上でのトランザクションについて説明し、最後に.NETアプリケーションからOracleデータベースのトランザクション制御方法について説明します。

トランザクションとACID

トランザクションとは、データの追加／更新／削除といった処理のひとまとまりを指します。また、トランザクションは“ACID特性”を持たなければなりません。ACID特性とは表1に示した4つの特性の頭文字からつけられた名前です。

これら4つの特性に基づいたトランザクションを「ACIDトランザクション」と呼びます。

では、トランザクションとACIDについて、実際の業務に照らし合わせながら説明してゆきましょう。

表1：ACID特性

ACID特性	説明
Atomicity（原子性）	トランザクションの変更は、すべてが完全に実行されるか、まったく実行されないかのどちらか。つまり、トランザクションを構成する一連の処理がこれ以上分割できないことを示す
Consistency（一貫性）	トランザクションが正しいプログラムの場合は、前の有効な状態から新しい有効な状態に変化する
Isolation（隔離性）	トランザクションが同時並行して実行されても、終了するまで他の処理とお互いに干渉しない
Durability（永続性）	トランザクションがコミットした結果は必ず永続化（保存）され、前後の結果は障害によって失われない

銀行業務で考える

トランザクションの典型的な例として銀行でのお金の移動があります。A銀行の口座からB銀行の口座にお金を振り込むケースを考えてみます。この場合、

①A銀行の口座からお金の引き出し
②B銀行の口座へお金の振り込み

という業務が生じます。

①と②の処理をひとつのトランザクションとみなすことができます。

この処理を完全に実施する際には表1で示した4つの特性が必要不可欠です。

各特性ごとに細かく見てみましょう。

・Atomicity（原子性）

お金の引き出し作業と振り込み作業は、どちらか片方が失敗するともう片方も失敗しなければなりません。振り込んだはずのお金が実は振り込まれていなかったら大変です。そのためこの作業は、引き出しも振り込みも両方とも成功したか、あるいは両方とも失敗したかの2通りの結果しかありえません。

・Consistency（一貫性）

お金の引き出し作業によって引き出された金額と、お金の振り込み作業によって振り込まれた金額は、当然のことながら同じでなければなりません。100万円引き出してそれを振り込んだはずなのに、実際は1万円しか振り込まれていなかったら問題です。

・Isolation（隔離性）

お金の引き出し作業と振り込み作業を行なっている途中に、別の作業（たとえば公共料金の支払いとかカードの自動引き落としとか）が影響をおよぼしてはいけません。知らない間にお金が別の口座に入っていたら困ります。

・Durability（永続性）

お金の引き出し作業と振り込み作業が正常であれ異常であれ完了したら、その結果が障害などによって失われてしまうことがあってはなりません。お金の振り込みが成功しているはずなのに、それがいつの間にか無効になっていたら最悪です。

トランザクションと分離レベル

複数のユーザーが同時に同じデータにアクセスするようなトランザクションを実行する場合、データベースでは、データの不整合が発生しないようにトランザクション分離レベル（Isolation Level）という機能が実装されています。トランザクション分離レベルは、4つのレベルがANSIにて規定されています（表2）。

Oracleのデフォルトのトランザクション分離レベルは「Read Committed」です。Oracleと他のRDBMSでは、トランザクション制御の実装の仕方が異なるため、同じRead Committedであっても、トランザクションの振る舞いが異なります。

最も大きな違いは、更新中のデータに対して検索を行なった場合です。Oracle以外のRDBMSでは、検索時に共有ロックをかけます。しかし、読み取るデータが更新中の場合は、すでにそのデータに排他ロックがかかっているため、共有ロックをかけることができません。つまり、多くのRDBMSでは更新中のデータに読み手はアクセスできず、検索処理は待たされます。一方、Oracleでは、マルチバージョン同時実行制御により、UNDO表領域に保持された更新前の情報を読み込むため、読み取るデータが更新中であっても検索処理が待たされることはありません（図1）。

表2：トランザクション分離レベル

ANSI／ISO SQL92	Oracle	説明
Uncommitted Read	-	他のトランザクションで変更中の未確定のデータにアクセスできる（ダーティリード）
Read Committed	Read Committed	データにアクセスする場合、確定したデータだけが参照可能。Oracleデフォルトのトランザクション制御
Repeatable Read	-	問い合わせで使用されるデータをロックし、他のトランザクションから変更できないようにする
Serializable	Serializable	問い合わせで使用されるデータ全体をロックし、更新はもちろんのこと、データ挿入によるファントムリード[注*]も防ぐ
-	Read Only	そのトランザクションを開始した時点でコミット済の変更のみが参照され、更新は不可

注*）ファントムリードとは、トランザクションが複数行のある集合を返す検索条件で問い合わせを再実行したとき、別のトランザクションがコミットしてしまったために、同じ検索条件で問い合わせを実行しても異なる結果を得てしまう現象のことをいいます。