



特集

3

ここがポイント!

Oracle DB

Oracle meets .NET

開発

第4回 トランザクションを実現する  
さまざまな方法

# トランザクションを 極める(後編)

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



「トランザクション編」の後編となる今回は、前回に引き続き.NETアプリケーションからOracleデータベースを利用するための説明を行ないます。

前回は、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、そして.NETアプリケーションからのトランザクション制御について説明しました。今回は、ASP.NETを使用したWebアプリケーションでのトランザクション制御について説明します。

## ASP.NETからの 排他制御における問題点



Webアプリケーションでは、複数ページをまたぐ処理を行なう場合が多くなります。この場合、ページ間でのデータベースとの接続状態は保持しない状態で作業を行ないます。しかしこれ

### Level



### 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

Microsoft Message Queuing

リスト1: 「product」テーブルの作成

```
SQLPLUS scott/tiger
SQL> CREATE TABLE product(
  2  product_id number(4,0) not null,
  3  product_name varchar2(30),
  4  quantity numeric(4,0),
  5  updatenum number(5,0) default 0 not null,
  6  PRIMARY KEY(product_id)
  7 );
```

表が作成されました。

```
SQL> CREATE UNIQUE INDEX PROD_IDX1 ON product(product_id, updatenum);
```

索引が作成されました。

```
SQL> INSERT INTO product(product_id, product_name, quantity)
  2  VALUES(1001,'Salt',10);
```

1行が作成されました。

```
SQL> INSERT INTO product(product_id, product_name, quantity)
  2  VALUES(1002,'Sugar',20);
```

1行が作成されました。

では、前回説明した「FOR UPDATE」での更新ロック、もしくはIsolationレベルを設定したトランザクション制御を行なうことができません。これらのトランザクション制御は同一トランザクション内でのみ有効であるため、Webアプリケーションのようなデータベースとの接続がステートレスなアプリケーションでは、トランザクションの状態を保持することができません。そのため、Webアプリケーションでは、実際データを更新するタイミングで、他のユーザーによるデータ更新が行なわれたか否かを確認後に更新する楽観同時実行制御を使用する場合があります。

前回説明した、DataAdapterとDataSetが持つ“楽観同時実行制御つきデータ更新機能”は、便利である反面、データベースに対して発行されるWHERE条件にSELECT文で取得したすべての列が含まれるため、件数の多いテーブルに対する更新時にはパフォーマンスが悪くなる可能性があります。また、画像情報などを格納するBLOB (Binary Large Objects) 列に対して変更検知ができません。これらの問題を解決するためには、更新回数列をテーブルに追加し、更新検出のための問い合わせを簡略化することによりパフォーマンスを向上させ、BLOB列の更新にも対応させることが可能です。

## ▶テストデータの準備

では、更新回数列を追加し、楽観同時実行制御を実装するための手順を説明しましょう。

## テスト用テーブルの作成

まずテストを実行するためのテーブ

ルとインデックスを作成し、データを作成します。今回はSCOTTユーザーでテストします。SQL\*PLUSからリスト1を実行してください。

## トリガーの作成

次に「product」テーブルに対する更新が行なわれるたびに自動的に「updatenum」列を更新するトリガーを作成します (リスト2)。

## ▶Webアプリケーション作成

### テーブル更新用Webフォームの設定

次にWebフォームで「product」テーブルの内容をDataGridに表示するサンプルアプリケーションを作成します。新

リスト2：トリガーの作成

```
SQL> CREATE OR REPLACE TRIGGER SCOTT.TRG_PROD_UPDATE
2 BEFORE UPDATE
3 ON SCOTT.PRODUCT
4 REFERENCING OLD AS OLD NEW AS NEW
5 FOR EACH ROW
6 BEGIN
7   :new.updatenum := :new.updatenum + 1;
8 END;
9 /
```

図1：Webフォームに必要なコントロールを貼り付け

