

SQL Serverで

ど〜んと と どいってみよう!

必ず役立つ
現場のノウハウ

百田 昌馬

HYAKUTA, Shoma

Supported by 松本 美穂

<http://www.sqlquality.com/>

第11回

SQL Server 2005の 読み取り一貫性



読み取り一貫性とは

読み取り一貫性は、排他ロックのかかっているデータを読み取れるようにする、Oracleではお馴染みの機能である。正確には、SELECTステートメン

トを発行した時点またはトランザクションの開始時点でのデータが読み取れることを保証する機能である。図1のように、排他ロックのかかっている更新中のデータ（未コミットのまだ確定していないデータ）は参照させないようにし、更新前のデータ（その時点での正しいデータ）を参照させることで一貫性を保つという動作である。更新中のデータはロールバックされる（取り消される）かもしれないからである。

この機能は、従来のSQL Serverには実装されていなかったが、SQL Server 2005でようやく実装された。それが「READ_COMMITTED_SNAPSHOT」と「スナップショット分離レベル」である。今までOracle経験者は、SQL Serverを利用するときに違和感を感じたり、前回説明した「テーブルスキャンによるロック待ち」というSQL Server独特の動作の理解に苦しむことが多かった。しかし読み取り

一貫性は、これらの違和感を払拭し、Oracle経験者がOracleと同じような感覚でSQL Serverを利用できるようになるのである。そういう意味では非常に画期的な新機能である。

しかし、どんな状況でも読み取り一貫性（READ_COMMITTED_SNAPSHOTまたはスナップショット分離レベル）を利用すべきというわけではない。この機能は、今まで実装されていなかったように、業務アプリケーションを構築する上で必須ではない。また、読み取り一貫性にはオーバーヘッドもある（これについては後述）。したがって、現状のシステムに問題がないのであれば、利用しないほうがよい。



READ_COMMITTED_SNAPSHOT

READ_COMMITTED_SNAPSHOTは、SELECTステートメントを発行

Level

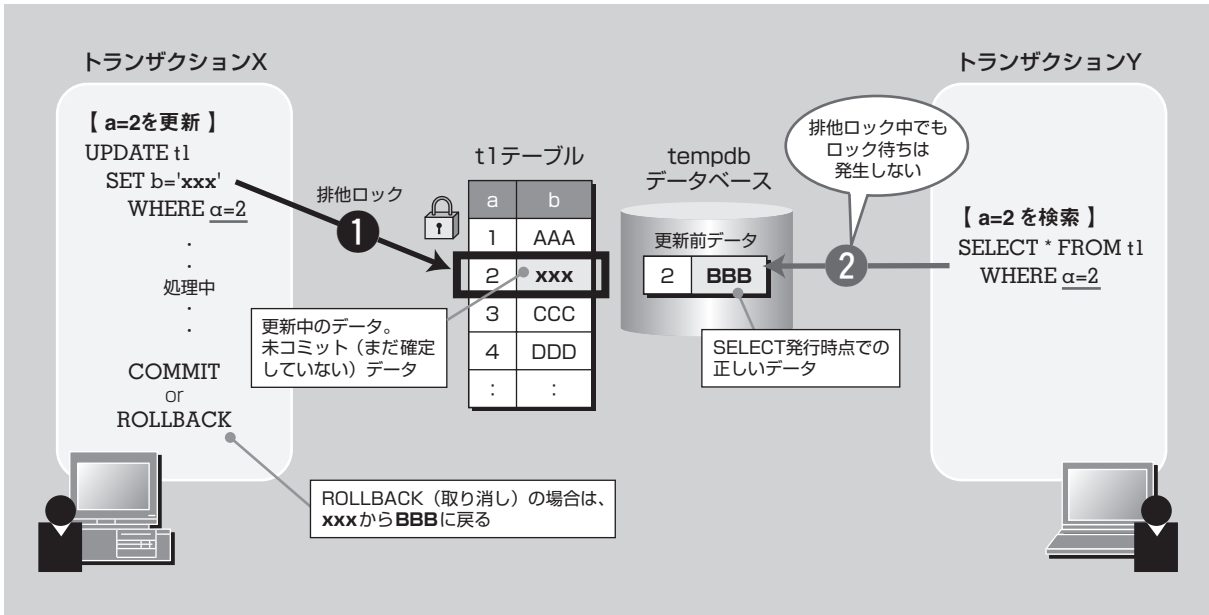
1 2 3 4 5

Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:

↓
SQL Server 2005

図1：読み取り一貫性 (READ_COMMITTED_SNAPSHOTの場合)



した時点でのデータが読み取れることを保証する機能である。これは、Oracleのデフォルトの動作と同じである。図1のように、排他ロックがかかっている更新中のデータ(未コミットのまだ確定していないデータ)は参照させないようにし、更新前のデータ(その時点での正しいデータ)を参照させることで一貫性を保つという動作である。更新前のデータは、tempdbデータベース内に格納される。なお、Oracleでは、更新前データの格納先にはUNDOセグメント(Oracle8i以前はロールバックセグメント)が利用される。

図2：READ_COMMITTED_SNAPSHOTの利用例



*** READ_COMMITTED_SNAPSHOTの使用方法**

SQL Server 2005のデフォルトでは、READ_COMMITTED_SNAPSHOTは利用することができない。次のようにデータベースに対してREAD_COMMITTED_SNAP

SHOTをONに設定することで初めて利用できる (図2)。

```

ALTER DATABASE データベース名
SET READ_COMMITTED_SNAPSHOT ON

```