

SQL Serverで

ど〜んと と 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9回

分離レベル (Isolation Level) を極める



トランザクションの分離とACID

トランザクションは、ACID (Atomicity/Consistency/Isolation/Durability) 属性を備えている必要がある、という説明を聞いたことのある方は多い

だろう。しかし、そういった説明をしている記事や書籍の中で、ACIDの「I」(Isolation)をきちんと説明したものは意外と少ない。それゆえ、Isolationの意味を誤解していたり、正しく理解できていなかったりする方もいるだろう。そこで、今回はIsolationについて詳しく説明していく。

ンザクションは、図1の左側のように複数のトランザクションが並行して(パラレルで)実行されている。しかし、同時に実行されると、読み取ったデータがほかのトランザクションによって同時に更新されたりして、データの矛盾が発生する可能性がある。このように複数のトランザクションが同時実行され、交じり合っている状態は、ACIDの「Isolation」は満たされていない。Isolationは「分離」や

Level

1 2 3 4 5

Technology Tools

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

*Isolationとは

通常、業務システムにおけるトラ

図1：並列実行と直列実行

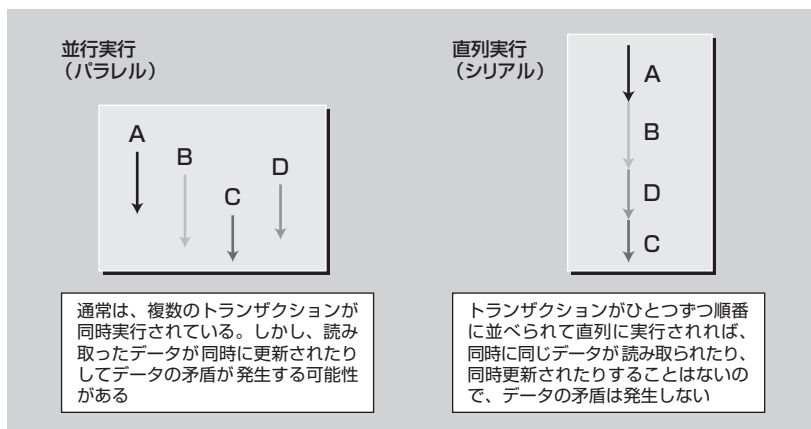


図2：ANSI SQL-92で定義された一貫性水準

分離レベル	起こり得るデータの矛盾		
	ダーティリード (不正読み取り)	反復読み取り 不可	ファントム 読み取り
Read UnCommitted	発生する	発生する	発生する
Read Committed (デフォルト)	発生しない	発生する	発生する
Repeatable Read	発生しない	発生しない	発生する
Serializable	発生しない	発生しない	発生しない

Isolationは満たされないデータの矛盾あり

完全なIsolationの実現データの矛盾なし

Serialize (直列化) がable (可能な) レベルという意味。ACIDのIsolationを満たしたレベル

「独立」「隔離」という意味である。

トランザクションがIsolation (分離) された状態とは、図1の右側のように、トランザクションがひとつずつ順番に並べられ、直列 (Serial: シリアル) 実行された状態である。ひとつずつ実行されるのであれば、同時に同じデータが読み取られたり、同時更新されたりすることはないので、データの矛盾は発生しないのである。ACIDのIsolationは、まさにこの状態を指している。実際には複数のトランザクションが並行実行されるが、トランザクションがひとつずつ実行されたときと同じように、データの矛盾が発生しない状態に保たれていることが、ACIDのIsolation (トランザクションの分離) が満たされている状態なのである。

***Isolation Levelとは**

SQLの業界標準である「ANSI SQL-92」規格では、Isolationを満たしているかどうかを「一貫性水準」として、図2の4つの分離レベル (Isolation Level) に分類している。一貫性が保たれるかどうか (データに矛盾が発生する可能性があるかどうか) でレベルが分かれ、複数のトランザクションが同時実行された場合に起こり得るデータの矛盾を3種類 (ダーティリード、反復読み取り不可、ファントム読み取り) 挙げている。どの矛盾も発生しないのが「Serializable」というレベルであり、これがACIDのIsolationを満たしているレベルである。図1の右側のようなSerialize (直列化) がable (可能な) レベ

ルという意味である。

分離レベルの具体的な実装は、データベース製品によって異なるが、SQL ServerとOracle、DB2のデフォルトの分離レベルは、「Read Committed」である。このレベルでは、反復読み取り不可とファントム読み取りという矛盾が発生する可能性があるが、以降で詳しく説明していく。



**Read Committed
~デフォルトの分離レベル**

SQL Serverにおける分離レベル (Isolation Level) の実装は、ロックの動作が大きく関係している。デフォルト (Read Committed) では、ロックは次のように動作する。

- 共有ロック：読み取りが完了するまで保持
- 排他ロック：トランザクションが完了するまで保持

共有ロックは、データの読み取り時 (SELECTステートメントなど) に使われるロックであり、排他ロックは、更新系のステートメント (UPDATE/DELETE/INSERT) によるデータの更新時に使われるロックである。どちらのロックが使われるかは、ステートメントが実行されたときにSQL Serverが自動的に判断する。また、共有ロックと排他ロックは、表1の関係があり、共有ロック