

Item35. トランザクションスループット向上のためにアイソレーションレベルを下げることを考える。

① 【Q】 アイソレーションがない場合、他のトランザクションによって生じる問題を挙げてみよう

P#	問題名	問題内容
P1		
P2	Dirty Read	
P3		
P4		

② トランザクション不要 : (1) サクラメントの住民 DB からプログラマの平均年齢を求める(利用すると非効率だし、利用しない場合に発生する誤差は問題にならない)。アプリケーション開発中。

③ トランザクション必要 : デビッド支払

④ 【Q】 アイソレーションレベルを 4 つ挙げてみよう。

I#	アイソレーションレベル	P1	P2	P3	P4	例
I1						
I2						
I3						
I4						

(注)I4 の場合も ACID の ACD 属性は有効である。

⑤ 【Q】 JDBC でアイソレーションレベルを変えるクラスとメソッドは？そのメソッドはトランザクション開始(begin) の前後のどちらにおくべきか、もしくはどちらでもよいか？

⑥ EJB 仕様によると、エンティティビーンのプロトコルは常にコンテナ管理であるため、ビーン管理でもメソッド中でアイソレーション指定できない。
→ローカルトランザクション最適化機能がない場合は、各アクセス毎に 2 フェーズコミットが実行される。

←コンテナで、NotSupport, Supports, Never を使うと移植性が失われる(!?)。

⑦ アイソレーションレベルはリソースマネージャ API を利用する(EJB は提供していない)。この場合、アイソレーションレベルを変更するメソッドの呼ぶ位置に注意する。

⑧ 単一トランザクションにおけるアイソレーションレベルの変更について仕様中に記述がない。実装によるので注意。

① 【A】

#	問題名	問題内容
1	Lost Update	A,B が同データを同時に参照し、A の更新が B の更新によって上書きされる。
2	Dirty Read	A のコミット前の更新が B から参照されてしまう。
3	Nonrepeatable Read	A が参照し、B が更新し、A が再度読み込んだ時に B の更新のために違う値が取得できてしまう。
4	Phantom read	A が参照し、B が追加し、A が再度読み込んだ時に B の更新のために違う値が取得できてしまう。

③ 【A】

I#	アイソレーションレベル	P1	P2	P3	P4	例
I1	SERIALIZABLE	○	○	○	○	更新
I2	REPEATABLE READ	○	○	○	×	参照
I3	READ COMMITTED	○	○	×	×	バッチ(大量データ参照)
I4	READ UNCOMMITTED	×	×	×	×	1 ページのサマリレポート

④ 【A】

- ・ java.sql.Connection#setTransaction
- ・ コード例は P.214 参照。
- ・ トランザクション開始の前におくべきことに注意
- ・ アイソレーションレベルが Statement レベルで設定されるわけではないことに注意

Item36. ロールバック時に処理の一部を保持するために savepoints を使用する

- ① トランザクションで問題があった場合、全ての処理が失敗となる。
- ② Item28 の旅行計画シナリオにおいて、3 つ目のフライトを予約できなかった時に、2 つ目までの予約を無効にしたい。
- ③ 10,000,000 の預金口座全ての利息を計算するために UPDATE を実行するには数時間かかる。SQL が複雑な場合は、最小限の残高を持つ口座と VIP の口座、2 月 15 日から 4 月 15 日の間に開かれた口座を更新する。9,947,831 行目で思わぬ障害にあった場合、UPDATE は失敗して全ての処理は無効となる。来月再トライする？
- ④ 必要なのは失敗した行に印をつけて、処理を続けることである。
- ⑤ JDBC3.0 仕様には savepoint が盛り込まれた。
- ⑥ コード参照(P.218)。savepoint 時に名前をつけることができ、rollback 時に戻りたいポイントの名前を指定できる。
- ⑦ JDBC3.0 仕様は策定されたばかりなので、ほとんどの DB ドライバはサポートしていない。