

正 誤 表

下記の部分に誤りがありましたので訂正させていただきます。ご迷惑をおかけして大変申し訳ございません。

2009 データベーススペシャリスト「専門知識 + 午後問題」の重点対策（第1版 第2刷）

No.	訂正箇所	誤	正
1	P.161 解説 下から14行目～6行目 下から3行目～1行目	<p>：</p> <p>いずれも、トランザクションの実行によって、データの整合性が崩れることを防止するか、トランザクションの実行結果が正しくなるように保証することを目的としています。READ COMMITTED では、あるデータを更新したトランザクションが COMMIT を発行する前に、他のトランザクションが同じデータを READ することができません。必ず の COMMIT の後に の READ が行われるので、 が先に終了することはありません。したがって、(イ)が誤りとなります。正解は(イ)です。</p> <p>ア：二つの SQL 文は、列(A, B, C)を同時にアクセスしているので、デッドロック発生の原因となります。READ COMMITTED は、データの整合性を維持する目的で使われ、デッドロックを防ぐ機能はありません。</p> <p>ウ：READ COMMITTED では、必ず COMMIT の後に READ を行います。</p> <p>エ：READ COMMITTED では、Non-repeatable read や Phantoms が発生します。したがって、記述のような現象が発生します。</p> <p>解答 イ</p>	<p>：</p> <p>いずれも、トランザクションの実行によって、データの整合性が崩れることを防止するか、トランザクションの実行結果が正しくなるように保証することを目的としています。二つの SQL 文は、一つの表の列(A, B, C)に同時にアクセスしています。 は READ COMMITTED ですから、SELECT ではページ単位で共有ロックを掛けますが、複数のページ(複数の資源)を同時にロックしようとはしません。したがって、デッドロックは発生しないことになり、(ア)が誤っています。</p> <p>イ：READ COMMITTED では、SELECT 文では共有ロックを SELECT 文実行中だけ掛けます。 の実行中に を実行すると、 の SQL 文終了後に、まだ COMMIT 文が発行されない時点で、 の SQL 文のロックが掛かっていないので終了する場合があります。</p> <p>ウ：READ COMMITTED では、必ず COMMIT の後に READ を行います。</p> <p>エ： を連続して実行していても、COMMIT 文は直後にあるので、 が の終了した更新結果を読むのは当然です。</p> <p>解答 ア</p>