

問3      トランザクション管理（データベース）

(H23 春-FE 午後問 3)

【解答】

〔設問1〕    aーイ, bーエ (a, b は順不同)

〔設問2〕    ウ

〔設問3〕    cーイ, dーエ

〔設問4〕    イ

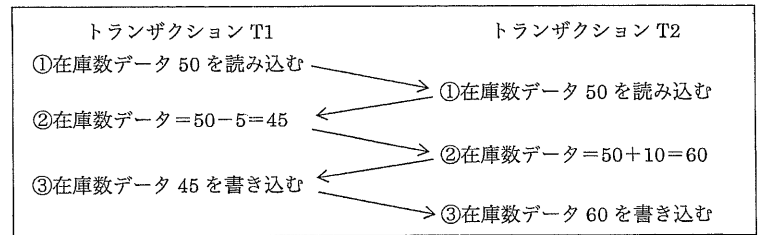
【解説】

トランザクション管理の排他制御に関する問題である。排他制御とは、複数のトランザクションが同時に同じデータを更新することで起きる不具合（二重更新）を制御する機能である。トランザクションは更新処理に入る前に対象データをロックして、他のトランザクションがこのデータにアクセスできなくする。この機能によって、複数のトランザクションは、順番に一つずつデータを更新するよう制御される。

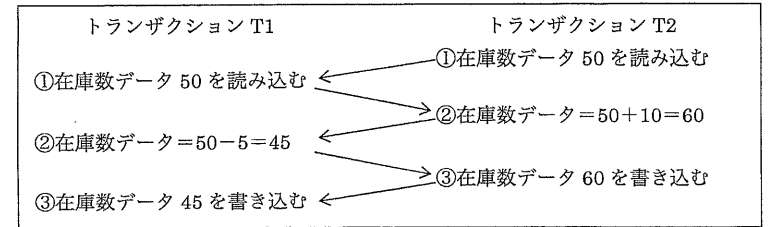
設問1と設問2は、基本的な問題である。設問3も排他制御についてきちんと理解していれば解けるであろう。設問4は、トランザクションと対象データの数が多いが、デッドロックが発生する仕組みを理解していれば、難しい問題ではないと思われる。全体的な難易度はやや低いといえるだろう。なお、設問1の中でACID特性のI (Isolation)のことを“独立性”と記述しているが、本試験が準拠するとされるSQL標準のJIS X 3005では“隔離性”という用語を使用している。

〔設問1〕

排他制御を行わない場合に起きる不具合（二重更新）の例である。次の処理順序でトランザクションT1とトランザクションT2が実行されると、実行後の在庫数は60になってしまう。

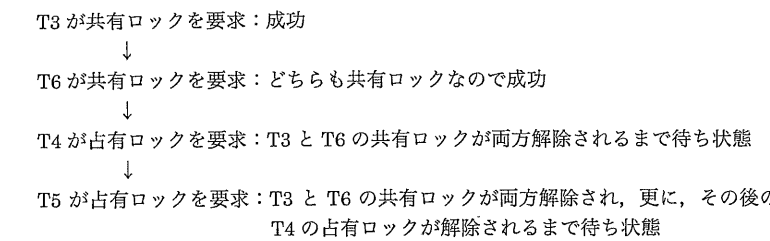


また、次の処理順序でトランザクションT1とトランザクションT2が実行されると、実行後の在庫数は45になってしまう。したがって、(イ)と(エ)が正解である。



〔設問2〕

データを更新しない複数のトランザクションが、同時に同じデータを照会しても、不具合は発生しない。そのため、共有ロックされたデータに対する共有ロックの要求は成功し、待ち状態にならない。図2において、白絵の具の在庫数データに対するロックの状態は、次のようになる。



したがって、(ウ)が正解である。

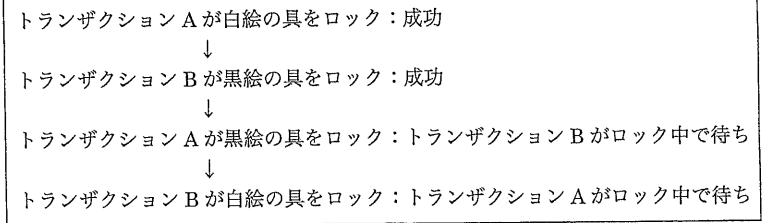
〔設問3〕

データをロックしている時間を最も短くするためには、データを読み込む直前にロックし、書き込んだ直後にロックを解除すればよい。データを読み込んだ後にロックすると、読み込みが終わってロックする前のタイミングに、他のトランザクションがデータを更新する可能性がある。その結果、読み込んだデータの値と、実際のデータの値が異なる状態になってしまう。また、書き込む前にロックを解除すると、書き込む前のデータを、他のトランザクションが読み込んでしまう可能性がある。その結果、他のトランザクションが読み込んだデータの値と、書き込んだ後の実際のデータの値が異なる状態になってしまう。

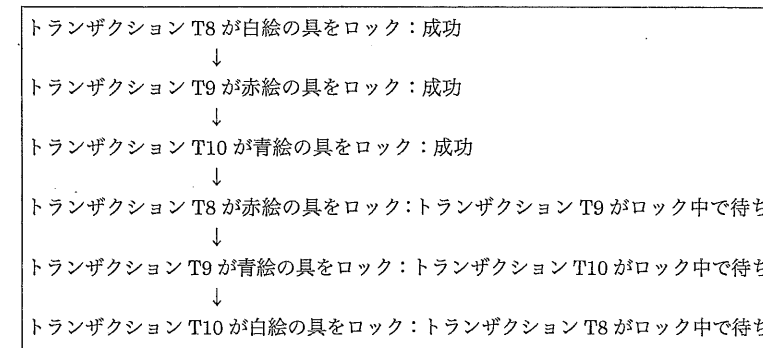
したがって、“赤絵の具の在庫数データをロックする。”のは、“③赤絵の具の在庫数データを読み込む。”直前の(イ)βになる。また、“白絵の具の在庫数データのロックを解除する。”のは、“⑥白絵の具の在庫数データを書き込む。”直後の(エ)δになる。

〔設問4〕

デッドロックとは、複数のトランザクションがお互いにロックの解除を待ち、処理が停止してしまう状態のことである。一般的に、トランザクションが二つで対象データも二つの例で考えることが多い。ここでも、白絵の具、黒絵の具の順にロックするトランザクションAと、黒絵の具、白絵の具の順にロックするトランザクションBの例で考えると、次の場合、デッドロックが発生する。



このように、二つの対象データ（白絵の具と黒絵の具）に対し、二つのトランザクションが両方ともロックする場合、デッドロックが発生する。二つのトランザクションが共通してロックする対象データが一つであれば、デッドロックは発生しない。この観点で選択肢を調べてみると、複数のトランザクションが共通してロックする対象データは、(ア)では赤絵の具だけ、(ウ)と(エ)では青絵の具だけなのに対し、(イ)では白絵の具、赤絵の具、青絵の具の三つになる。したがって、(イ)が正解となる。(イ)では、次の場合、デッドロックが発生する。



なお、デッドロックが発生した場合、いずれかのトランザクションを終了させ、ロックを解除させる。終了させたトランザクションは、ロールバック処理でトランザクション開始前の状態に戻す。また、複数のロックをかける順番を決めておけば、デッドロック発生を回避できる。(イ)の例では、ロックをかける順番を白、赤、青と決めておけば、デッドロックは発生しない。