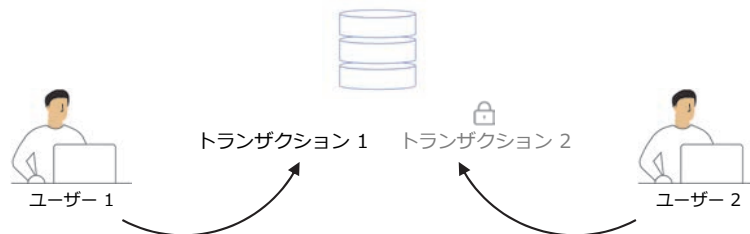


並行性制御

GeneXus[™]

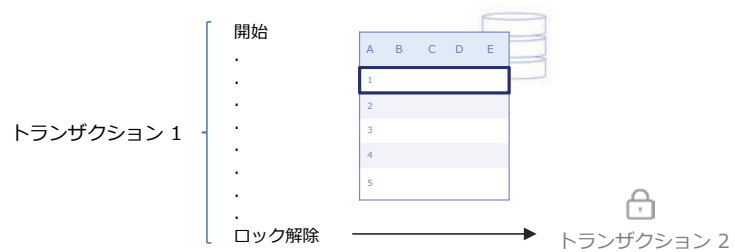
並行性制御



並行性制御について説明するには、複数のユーザーがいる環境で作業するときにデータの不整合を回避するための一連の制御について述べる必要があります。

並行性を制御するには情報をブロックする必要があります。

並行性制御



また、プログラムを並行して実行することによるパフォーマンスの低下を避ける必要があります。

並行性制御

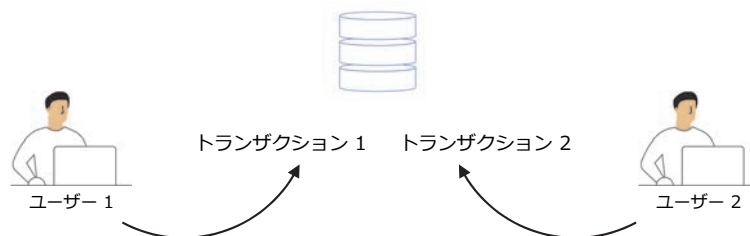
楽観的

トランザクション

ビジネスコンポーネント

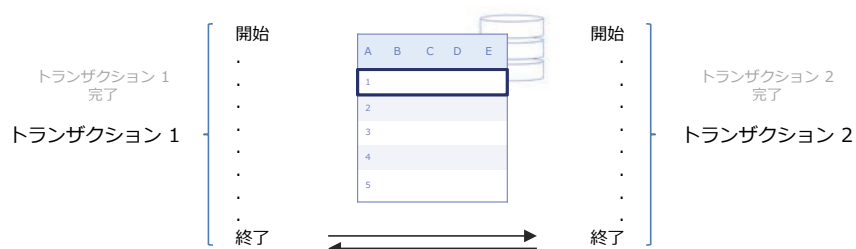
まず、トランザクションおよびビジネスコンポーネントの並行性制御について考えます。次に「楽観的並行性制御」について考えます。

楽観的並行性制御



このメカニズムでは、複数のデータベーストランザクションが相互に影響し合うことなく処理を完了できると想定します。また、関連するデータリソースをブロックせずにこれらのトランザクションを実行できると想定します。

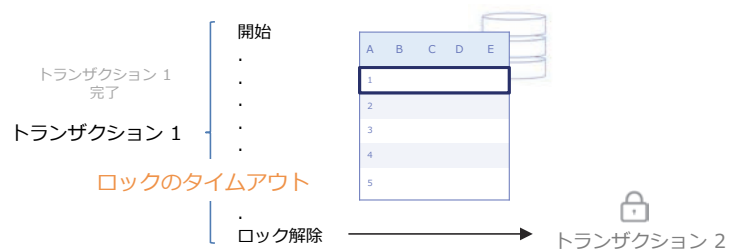
楽観的並行性制御



更新を実行する前に、各データベーストランザクションは、ほかのデータベーストランザクションがデータを変更していないことをチェックします。

これは、複数のユーザーが同じページに同時にアクセスできるという Web のグローバルな特性に関係します。Web ユーザーインターフェースではブロックは現実的ではありません。

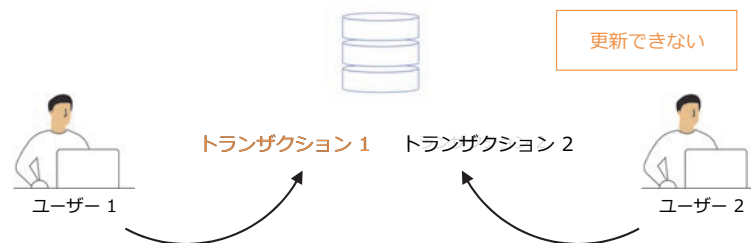
楽観的並行性制御



ユーザーがレコードの編集を開始し、「キャンセル」または「ログアウト」のリンクをクリックせずに離れることがよくあります。

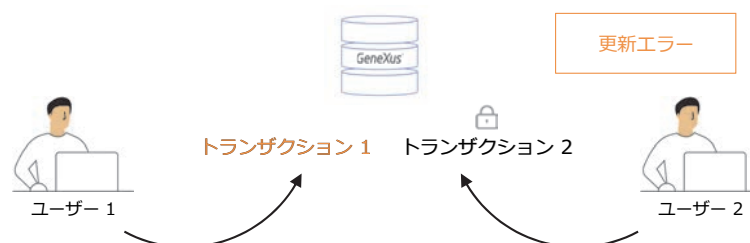
ロックを使用すると、同じレコードを編集しようとするほかのユーザーは、最初のユーザーのロック時間が過ぎるまで待たなければなりません。

楽観的並行性制御



使用されるたびに各レコードをロックする代わりに、システムは単に、2 人のユーザーが実際に同じレコードを同時に更新しようとしたという証拠を探します。このような証拠が見つかった場合は、ユーザーの更新が破棄され、ユーザーに通知されます。

楽観的並行性制御

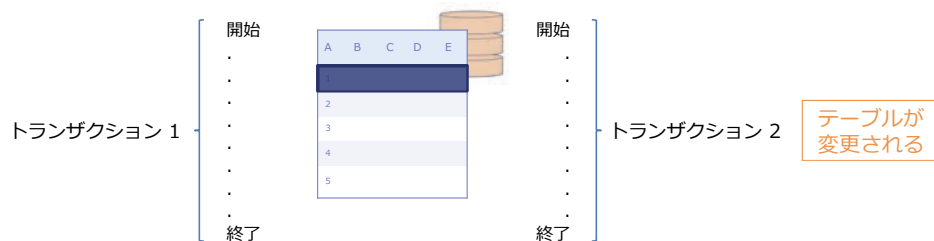


GeneXus では、複数のユーザーが同じレコードを更新しようとした場合、1 人目に更新が許可され、その他のユーザーはエラーを受け取ります (別のユーザーが更新しており、その前の古い情報に新しいデータを保存しようとしたという趣旨)。

そして、このような場合、確定の操作が元に戻されます。

楽観的並行性制御

Old 関数



重要
トランザクション構造内に存在する
項目属性のみが検証されます。

これらの項目属性には適用されません。



楽観的並行性制御は「Old 関数」に基づきます。

レコードが確定されると、各項目属性の古い値がデータベースの現在の値と比較されます。値が一致しない場合は、「The table has been modified」(テーブルが変更されています) のようなエラーが表示されます。これは、値の取得後に別のユーザーがレコードを変更したことを意味します。

次の点に注意してください。

- トランザクション構造に存在する項目属性のみが考慮されます。つまり、その他の項目属性を持つ並行トランザクションがあっても、これらは検証されません。
- このメカニズムは、Image、Video、Audio、LongVarChar などのタイプの項目属性には適用されません。また、トランザクションで宣言されたルールにより更新される拡張テーブルから推論された項目属性にも適用されません。
- レコードはキーによりインスタンス化されるため、キーの一部ではない項目属性にのみ適用されます。

並行性制御

GeneXus オブジェクトの場合

データ読み取り時

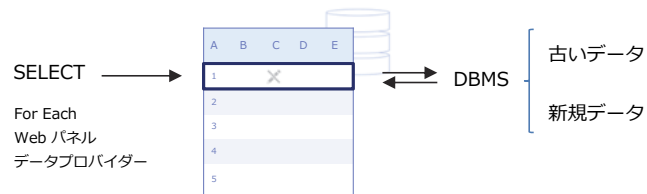
データ読み書き時

ほかの GeneXus オブジェクトでの並行性制御について説明します。

たとえば、データの読み取りのみ、またはデータの読み取りおよび書き込みを行う際に、GeneXus はどのようにロックを処理して並行性を制御するのでしょうか。

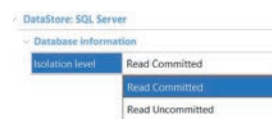
GeneXus オブジェクトでの並行性制御

読み取り専用



GeneXus の場合:
[Isolation level] プロパティ

- [Read Committed]
- [Read Uncommitted]



データの読み取りのみの場合はどうなるでしょうか。

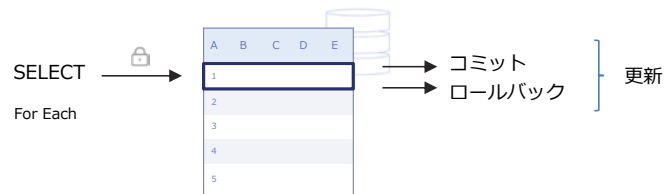
読み取りのみの場合 (For Each コマンド、Web パネルタイプのオブジェクト、データプロバイダーなどを使用する場合)、生成される SELECT でロックは行われません。

このような場合には常に排他的ロックの影響を受けます。たとえば、再編成ではテーブルが排他的に開かれます。この場合、たとえ読み取り専用であっても、ほかのプロセスはそのテーブルを開くことができません。

一方、別の書き込みプログラムにより読み取り対象の情報がロックされている場合、表示される値は DBMS によって異なります。古い値と新しい値のどちらを表示するかは DBMS によって異なります。

GeneXus オブジェクトでの並行性制御

読み取りおよび書き込み



重要

特定の For Each コマンドは最適化され、ロックは行いません。直接更新します。

データが読み取りと書き込みの対象となる場合はどうでしょうか。

データベースの更新を伴う For Each コマンドの場合、For Each の開始時に SELECT がロック付きで実行されてから、更新を実行します。

ほかのロックと同様に、コミットまたはロールバックの実行時にレコードが解放されます。

ただし、特定の For Each コマンドは最適化され、ロックを行わずに直接更新します。

サーバーに送られる条件を含む For Each コマンドおよびその本文内の割り当てのみを最適化できます。