

データベースのトランザクション同時実行を体感できる学習環境の提案

松本 圭太郎¹ 岡田 信一郎²

概要: IT 技術の進歩により、幅広い分野で膨大なデータが扱われる中で、その情報を扱うためのデータベース技術を学習することは重要である。筆者らはデータベースに関する学習項目の一つであるトランザクションを学習するための「トランザクション実習システム」を開発し、本学の授業の中で運用してきた。このシステムでは、簡単な操作でトランザクションの動きを確認することができる。しかし、学習者は複数のトランザクションを手作業でステップ実行する必要があるため、トランザクションの同時実行における異状、すなわち単一のトランザクションの視点では、処理中のデータがいつの間にか他のトランザクションによって変更されているという状態を感覚的にとらえることが難しいのではないかと筆者らは考えた。そこで、実際に同時実行される複数のトランザクションの中で、注目したトランザクションが、自動的に実行される他のトランザクションの影響を受ける様子を体感する学習環境を提案する。本年度は学習環境の設計および試作を行い、次年度より従来のトランザクション実習システムにこの学習環境を追加し、授業での運用を通じて評価を行う。

1. はじめに

IT 技術の進歩により、幅広い分野で膨大なデータが扱われる中で、その情報を扱うためのデータベース技術を学習することの重要性は高いといえる。本研究室では、データベースに関する学習項目の一つであるトランザクションを学習するための「トランザクション実習システム」[1]を開発し、本学情報工学科の「データベース論」の講義を受講している学生を対象に課題として提示し、運用と評価を行ってきた。データベース教育に関する関連研究では、「情報システムにおけるデータベースの仕組みを学ぶ共通教科「情報」の授業の開発と評価」[2]があるが、筆者らのシステムは情報工学を専門とする大学生を対象としている。また、プロセスの同時実行制御を学修するための手法に関する関連研究では、「情報工学コースのための Linux シェルによるコンカレント・プログラミング入門」[3]があるが、筆者らはデータベースに関する学習項目の一つであるトランザクションの並列処理に注目して、GUI を介してアルゴリズムを理解させることを目的としている。トランザクション実習システムの導入以前は、トランザクションに関

する実習を行う際、学習者は実用データベースマネジメントシステムへ接続し、SQL などの命令を直接入力する必要があった。本システムでは、ボタン操作によってトランザクションの基礎概念の理解を手軽に行える。しかし、学習者は複数のトランザクションをステップ実行する必要があるため、トランザクションの同時実行における異状、すなわち単一のトランザクションの視点では、処理中のデータがいつの間にか他のトランザクションによって変更されているという状態を感覚的にとらえることが難しいのではないかと筆者らは考えた。そこで、実際に同時実行される複数のトランザクションの中で、注目したトランザクションが、自動的に実行される他のトランザクションの影響を受ける様子を体感する学習環境を提案する。今回は、トランザクション実習システムを改良する新たな機能として試作を行った。

2. トランザクション実習システム

2.1 システム概要

トランザクション実習システムは、データベース初学者を対象とし、簡単な操作でトランザクションの基礎概念の理解を手軽に行えるシステムである。データベースに対する命令や操作を覚えることが目的ではないため、命令の直接入力ではなく、ボタン操作によって実習を行う。本システムは、JavaFX アプリケーションとして作成されており、

¹ 茨城大学大学院理工学研究科
Graduate School of Science and Engineering, Ibaraki University

² 茨城大学工学部
Faculty of Engineering, Ibaraki University



図 1 トランザクション実習システムの画面例



図 2 ロールバックの実習画面

jar ファイルを実行するだけでシステムを扱うことができる。アプリケーションの実行画面例を図 1 に示す。また、本システムは java で記述されたオープンソースの関係データベース管理システムである HSQLDB[4] を用いてデータ操作、トランザクション処理を行っている。学習者がデータベースを操作する命令が表示されたボタンを押下した後、システムはそのボタンに対応する SQL を HSQLDB が実行する。その後、操作の結果を本システムが画面に出力する。

データベースにおけるトランザクションには様々な機能、仕組みがあるが、本学情報工学科で開講されている「データベース論」の授業において練習問題などとして取り扱っていた内容であることから、トランザクション実習システムでは「ロールバック」、「アイソレーションレベル」、「デッドロック」の三つの課題を扱っている。課題はタブを選択すると切り替わり、それぞれのタブに対応した課題を実習出来る。

2.2 ロールバック

この課題では、「ロールバック」について実習ができる。ロールバックとは、データ更新が失敗したときに、データをトランザクション開始直前の状態に戻してから、トランザクションを終了する機能のことである。図 2 にロールバックの実習画面を示す。

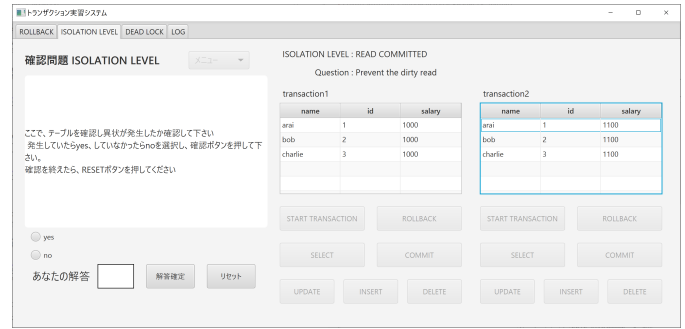


図 3 アイソレーションレベルの実習画面

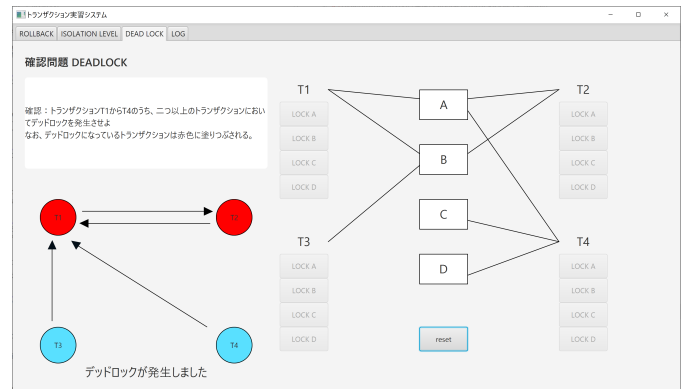


図 4 デッドロックの実習画面

2.3 アイソレーションレベル

この課題では、「アイソレーションレベル」について実習ができる。アイソレーションレベルとは、トランザクションの同時実行制御において、データが他のトランザクションから与えられる影響の程度を示すものである。図 3 にアイソレーションレベルの実習画面を示す。

2.4 デッドロック

この課題では、「デッドロック」について実習ができる。デッドロックとは、複数のプロセスが他プロセスの処理待ちとなり、膠着状態になることをいう。本システムのデッドロックとは、各トランザクションがテーブルに対してそれぞれロックをかけ、お互いにロックの解除を待ちトランザクションの処理を進行できなくなる状態を指す。図 4 にデッドロックの実習画面を示す。

3. トランザクションの同時実行を体感できる学習環境の提案

3.1 既存システムの課題

本システムの実習項目「アイソレーションレベル」では、学習者は複数のトランザクションを手作業でステップ実行するため、トランザクションの同時実行において及ぼす影響を各トランザクションごとに個別にとらえ、理解することが難しいのではないかと筆者らは考えた。以下に動作例を挙げて実習手順を説明する。

表 1 アイソレーションレベルと異状の対応

	DIRTY READ	NONREPEATABLE READ	PHANTOM READ
READ UNCOMMITTED	発生する	発生する	発生する
READ COMMITTED	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

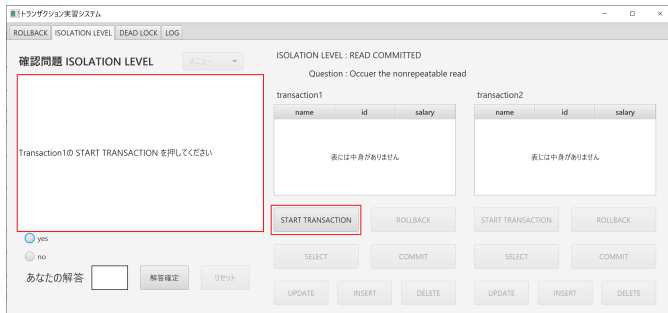


図 5 アイソレーションレベルの実習操作 (1)

アイソレーションレベルの実習開始時の画面を図 5 に示す。「Transaction1」と「Transaction2」の二つのトランザクションを用いて一つのテーブルに同時にアクセスする。各トランザクションの操作ボタンと問い合わせ結果を表示するテーブル、学習者に指示を出すナビゲートエリア、理解度を確認するための問題を出すためのボタンで画面が構成されている。

アイソレーションレベルと発生し得る異状の対応表を 1 に示す。アイソレーションレベルの実習トピックは計 6 個あり、アイソレーションレベル「READ COMMITTED」と「SERIALIZABLE」において、異状「DIRTY READ」、「NONREPEATABLE READ」、「PHANTOM READ」が引き起こされる操作において、それぞれ異状の発生有無を確認し、学習者はその結果を解答する。ここでは、アイソレーションレベル「READ COMMITTED」で、異状「NONREPEATABLE READ」が発生するトピックを例として手順を解説する。

- (1) ナビゲートエリアに示すように、Transaction1 の「START TRANSACTION」ボタンを押下してトランザクションを開始する (図 5)。
- (2) Transaction1 で「SELECT」ボタンを押下する (図 6)。
- (3) テーブル情報が確認できた後、Transaction2 の「START TRANSACTION」ボタンを押下してトランザクションを開始する (図 7)。
- (4) Transaction2 を開始した後、「UPDATE」ボタンを押下してデータを更新する (図 8)。
- (5) Transaction2 の更新と問い合わせ結果を確認した後、「COMMIT」ボタンを押下して処理を確定させる (図 9)。
- (6) Transaction1 で「SELECT」ボタンを押下して問い合わせ結果を表示する (図 10)。
- (7) Transaction1 の問い合わせ結果が操作 2 で行った結果と異なっていることがわかる (図 11)。この結果は、

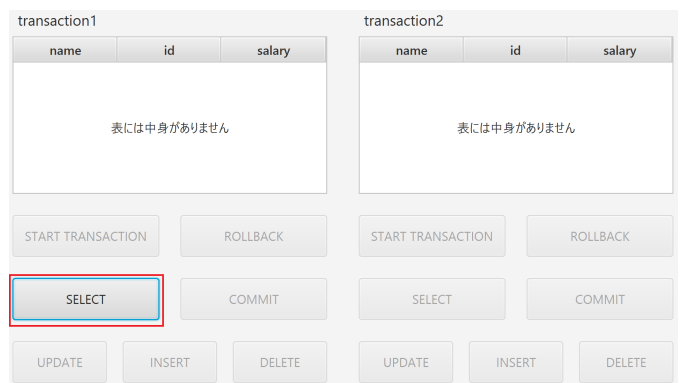


図 6 アイソレーションレベルの実習操作 (2)

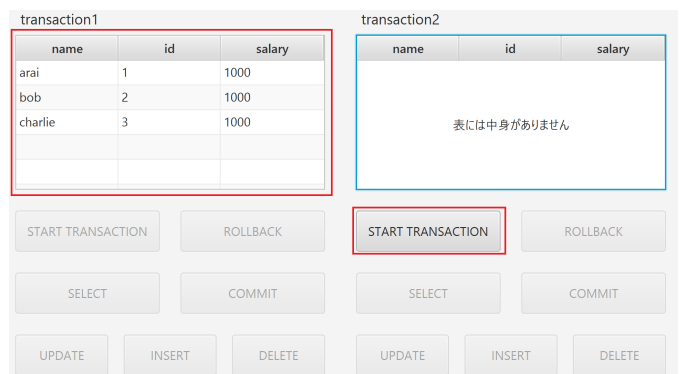


図 7 アイソレーションレベルの実習操作 (3)

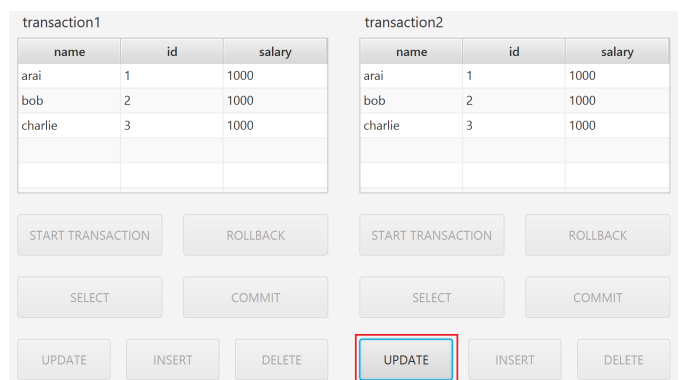


図 8 アイソレーションレベルの実習操作 (4)

Transaction1 が複数回問い合わせを行った結果が異なってしまう異状「NONREPEATABLE READ」に該当する。この画面出力を見て、学習者は異状が発生したことを確認し、問題に解答する。

3.2 目的

本システムの実習は、学習者が行う操作はナビゲートエリアによって指示され、異状を発生させるために不要な操作ボタンは無効化される。学習者は操作に間違いを発生させることなく異状の発生手順を確認できるが、学習者が前提として異状の原因や結果を理解していない場合、Transaction1 において処理中のデータがいつの間にか Transaction2 によって変更されているという状態を感覚的

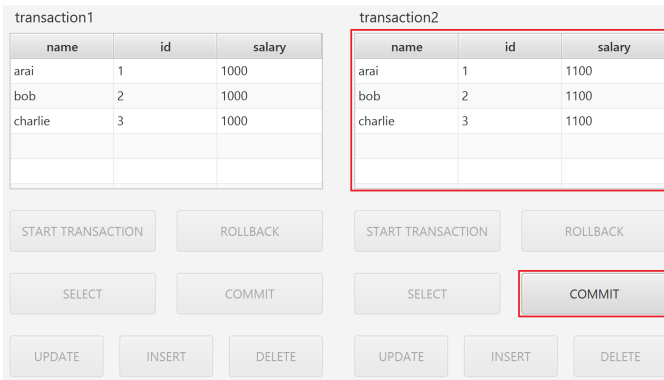


図 9 アイソレーションレベルの実習操作 (5)

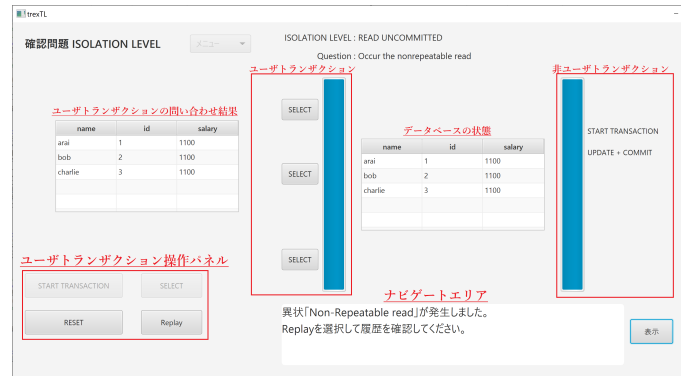


図 12 画面例

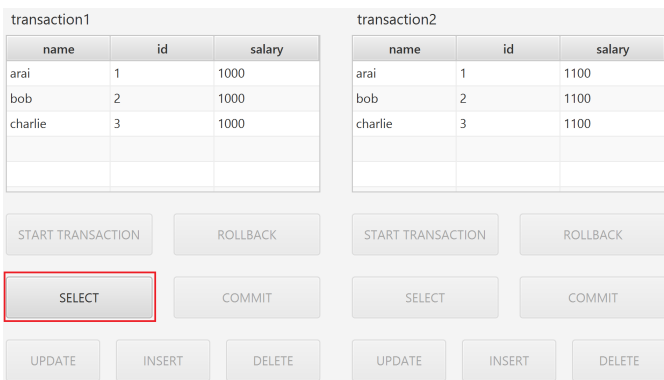


図 10 アイソレーションレベルの実習操作 (6)

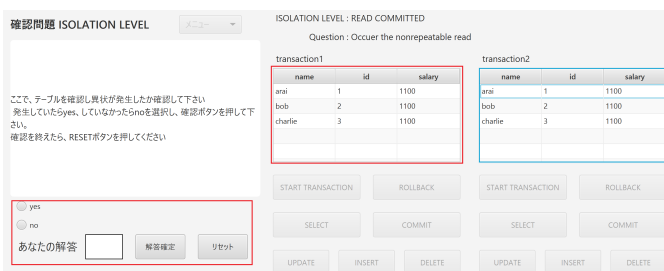


図 11 アイソレーションレベルの実習操作 (7)

にとらえることが難しいのではないかと筆者らは考えた。そこで、実際に同時実行されるトランザクションの中で、学習者が注目したトランザクションが自動的に実行される他のトランザクションの影響を受ける様子を体感できる学習環境を提案し、試作を行うこととした。

3.3 開発方針

同時実行される複数のトランザクションの中で、学習者が注目するトランザクションを「ユーザーランザクション」、自動的に実行されるトランザクションを「非ユーザーランザクション」として扱う。ユーザーランザクションは学習者が操作を行い、非ユーザーランザクションはシステムによって自動的かつ無作為に処理が実行されるように学習環境を設計した。また、各トランザクションの操作ごとにステップ実行していた処理を、JavaFX の AnimationTimer

クラスを利用することで一定時間内のタイムライン上で扱うよう設計した。学習者は、一定時間の経過中にユーザーランザクションの操作を実行し、それに応じて非ユーザーランザクションは同時に自動的に実行される。それぞれのトランザクションによって異状の発生する操作が行われた場合に異状が検出され、学習者は画面で確認できる。学習者が非ユーザーランザクションによっていつの間にか処理中のデータが変更される状態を体感できるように、非ユーザーランザクションの操作は初期状態では認識できないよう設計した。まず、ユーザーランザクションの処理中に異状が発生する様子を学習者が確認する。次の段階では非ユーザーランザクションの操作履歴を確認可能な状態にし、学習者自身によるユーザーランザクションの操作履歴と照合してどのような操作が異状の過程と結果となるのかを理解させることを目的とした。

1 より、全ての異状が発生する様子、発生しない様子を実習するために、扱うアイソレーションレベルは「READ UNCOMMITTED」と「SERIALIZABLE」の二種類とした。

4. 試作

4.1 GUI

本提案で設計した学習環境の画面例を図 12 に示す。ユーザーランザクション、非ユーザーランザクションのタイムラインを視覚的に把握できるようにするため、JavaFX の標準コントロールである ProgressBar を利用して時間の経過を表現した。画面上のそれぞれのトランザクションを示すプログレスバーの側部には各トランザクションの経過時間時点での操作履歴が表示される。画面左部のテーブルにはユーザーランザクションの問い合わせ結果が表示される。画面中央部のテーブルでは、コミット前のデータを含め時間経過中は常に問い合わせ結果を取得してデータベースの状態を表示する。

4.2 実習手順

アイソレーションレベル「READ UNCOMMITTED」、異状「NONREPEATABLE READ」のトピックにおける

実習画面を図 13 に示す。初期状態では、非ユーザトランザクションの操作履歴とデータベースの状態はブラインドで非表示となっている。ナビゲートエリアの指示に従い、「START TRANSACTION」ボタンを押下すると、図 14 に示すようにプログレスバーのアニメーションが開始する。プログレスバーは 10 秒で最大となり、この時をユーザトランザクションの終了とする。時間の経過中、ユーザトランザクションの操作ボタン「SELECT」が押下可能になり、ユーザトランザクションによる問い合わせを 1 秒ごとに任意のタイミングで 10 回まで実行することができる。問い合わせの結果は画面左部のテーブルに表示され、ユーザトランザクションの問い合わせ結果のみに注目して処理中のデータが変更される状態を確認できる。非ユーザトランザクションの操作は無作為なタイミングで開始され、選択したトピックの異状が発生し得る操作を自動的に実行する。

図 15 に示すようにプログレスバーが最大となった時、ユーザトランザクションと非ユーザトランザクション操作が異状を発生させるような組合せであった場合は、ナビゲートエリアで異状の発生を通知するメッセージが表示される。この時点で、学習者は注目していたトランザクションにおいて異状が発生したことを確認できたこととして、異状の発生過程と原因を理解する状態に移行する。

異状発生後にブラインドを非表示にした画面を図 16 に示す。学習者は、この画面でユーザトランザクションで行った問い合わせ履歴をプログレスバーの時間に対応する「SELECT」ボタンを押下して確認することができる。また、「Replay」ボタンが有効化し、再生機能を利用することができる。この機能では、非ユーザトランザクションの操作履歴を視認できる状態で、行った操作とデータの遷移を確認することができる。これにより、異状の発生過程で行われた操作の内容とその影響を時間軸に沿って理解することができる。

また、アイソレーションレベル「SERIALIZABLE」、異状「NONREPEATABLE READ」の学習画面例を図 17 に示す。アイソレーションレベルと異状の対応(図 1)の通り、異状は発生しない。非ユーザトランザクションの操作履歴から、非ユーザトランザクションの操作がブロックされ実行されていないことを確認できる。

5. まとめと今後の課題

本研究では、実際に同時実行されるトランザクションの中で、学習者が注目したトランザクションが自動的に実行される他のトランザクションの影響を受ける様子を体感できる学習環境を提案し、試作を行った。今後の課題として、現在はシステムの指示通りにボタン操作をするだけでも実習を終えることができってしまうため、学習者が理解していることを確認できないことが挙げられる。実習を終えた後に、学習者の理解度を確認する問題などを提示すること

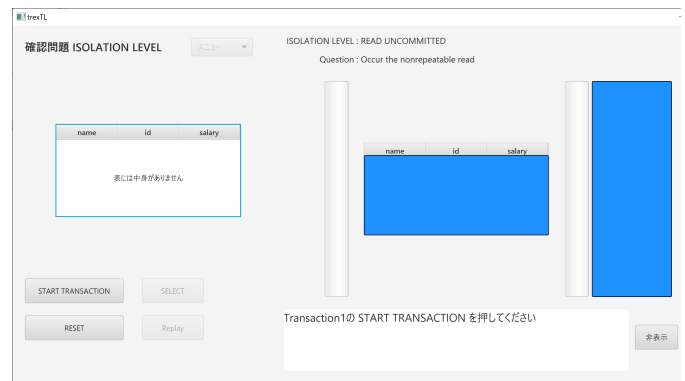


図 13 初期画面

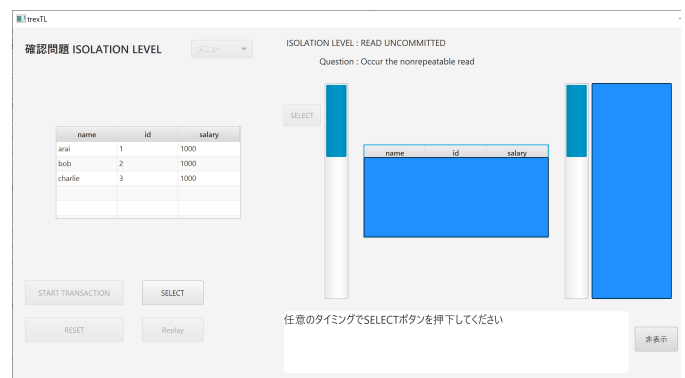


図 14 トランザクション経過中の画面

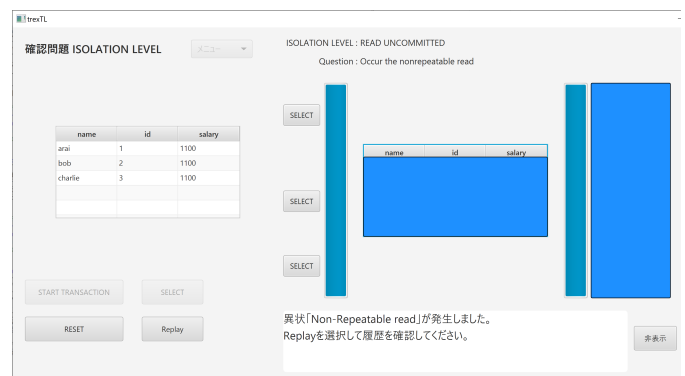


図 15 トランザクション終了後の画面

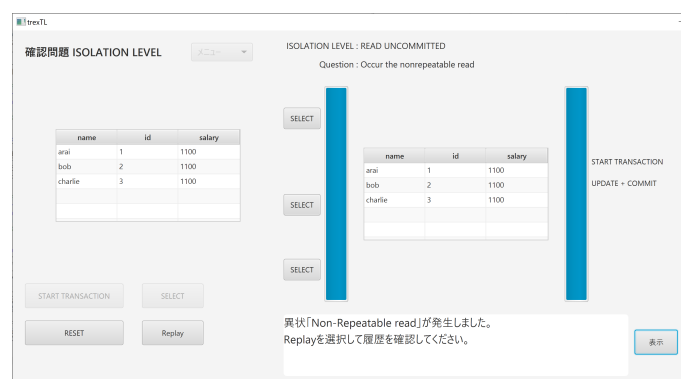


図 16 ブラインドを非表示にした画面

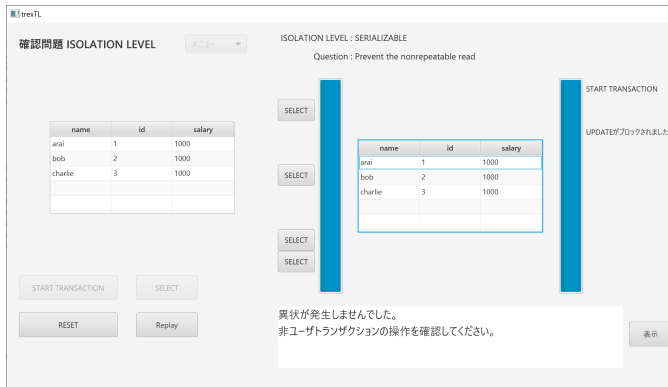


図 17 SERIALIZABLE の学習画面例

で、トランザクションに関する知識の修得を能動的にさせる必要がある。また、本学習環境によって学習者の理解度が向上するのかを確認するため、2022 年度の本学情報工学科の授業で運用し、評価する必要がある。

参考文献

- [1] 新井輝, 岡田信一郎, “データベース学習のためのトランザクション実習システムの開発と運用評価,” 情報処理学会研究報告, Vol.2021-CE-158, No.1, 2021 年 2 月.
- [2] 白井詩沙香, 長瀧寛之, 竹中一平, 武本康宏, 田邊則彦, 兼宗進, “情報システムにおけるデータベースの仕組みを学ぶ共通教科「情報」の授業の開発と評価,” 情報処理学会論文誌 教育とコンピュータ, Vol.5, No.3, pp.23-34, (Oct. 2019).
- [3] 佐藤信, “情報工学コースのための Linux シェルによるコンカレント・プログラミング入門,” 情報処理学会研究報告, Vol.2021-CE-161, No.5, 2021 年 10 月.
- [4] HSQLDB 入手先 (<http://hsqldb.org>) (2022 年 2 月閲覧).