

ブラウザ上での SQL 学習システム

坂爪 拓也[†] 井上 潮[‡]

[†] 東京電機大学大学院 工学研究科 〒120-8551 東京都足立区千住旭町 5 番

E-mail: [†] 13kmc11@ms.dendai.ac.jp, [‡] inoue@c.dendai.ac.jp

あらまし SQL は国際標準のデータベース言語であり、情報システムに携わる技術者には最低限の知識が要求される。本研究では SQL の基本的な操作について、初心者が自習に使える学習システムを提案する。インストラクタが用意した学習問題を学習者が解いていくという形式をとり、学習者が誤った命令文を入力したり、不適切な命令文を実行したりした場合に、分かり易いエラーメッセージの作成やデータベースの自動復元などの丁寧なフィードバックをシステム側から提供することで、初心者の自習という用途に適したシステムを実現する。

キーワード SQL, e-ラーニング, 学習システム, 学習支援

SQL learning system on the browser

Takuya SAKAZUME[†] and Ushio INOUE[‡]

[†] Tokyo Denki University, 5-Senju-Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan

E-mail: [†] 13kmc11@ms.dendai.ac.jp, [‡] inoue@c.dendai.ac.jp

Keyword SQL, e-learning learning system, learning support

1. はじめに

SQL はリレーショナルデータベース管理システム（以後 RDBMS）を操作するための言語で、データベース言語として、国際標準となっているものである。SQL は、現在の情報システムにおいて広く一般に使用されており、システムに携わる技術者にとって、重要な学習内容となっている。

しかし、経験の全くない初心者が SQL を実際に使用できる環境を整えるには労力がかかり、環境によっては SQL 自体の学習とは無関係な問題で時間を浪費してしまう。また、データベースの内容を誤って変更すると、初心者には修正が困難であるため、これも初心者の自習に対する障害となっている。本論文では初心者の自習としての用途を想定した、サーバサイドプログラムによる SQL 文の実行演習支援を行う学習システムを提案する。

SQL の学習において、実際に命令文を実行する環境を整えることは重要視されており、最も基本である表出力の命令文を学習できるシステムや、問題の自動作成、複数の表にまたがる複雑な命令文の実行を練習できるシステムなど多様な研究がなされている。

既存の研究は多機能で、様々な用途に対応できるように作成されているが、それによって失敗時のエラーメッセージなどによるフォローが難しくなり、修正も学習者側任せになってしまうか、インストラクタの介入を要する傾向がある。複数の学習者が同時にシステムを使用する場合、ある学習者のデータ変更によって別の学習者に影響が出る恐れがあるため、学習者ごとにデータベースを用意する必要がある。このとき、個別のデータベースに対して修正や必要なデータの入力が必要になるため、問題の追加や失敗時の修正などでシステム管理者側の負担が増大する。また、自由な命

令文を許可することで想定される失敗も多岐に渡ることになり、エラーメッセージの用意も難しくなる。

このようなシステムの学習課題自体については学習者がある程度独力で解答を探し当てる能力があることを前提としていている場合が多いため、初心者が SQL 文の基本的な操作（データの検索・追加・修正・削除）を初めて習得するために使用できるようなツールの研究は相対的に少ない。

本研究では SQL 文の基本的な操作を一通り実際に実行演習するための初心者をユーザとして想定したシステムを提案する。初心者が SQL を実行演習する場合、幾つかの問題が想定されるが本研究で重要とする問題は以下の 3 つである。

SQL の学習は、まず表出力を行う SELECT 文について学び、それからテーブル行を挿入する INSERT 文や削除を行う DELETE 文などの勉強をするのが一般的である。こうした学習順序に沿って演習を行うとすると、例えば INSERT 文で指定された様に行を挿入する学習問題で、行の内容を間違えてしまった場合、間違えた行の内容やその挿入前のテーブルの内容を把握している上で、訂正には UPDATE 文、元の状態に戻すには DELETE 文の入力が必要になるが、学習順序によってはそれが困難であると予想できる。よって初心者が段階を踏んで学習するのに合わせるように、学習問題での演習を行うには、失敗してもそれをシステム側で元に戻してくれる機能が必要である。

学習問題の要求事項を達成できたかの確認を、初心者である学習者自身が行った場合、見落としや勘違いが予想され、インストラクタが判定する場合は時間の都合から自習するためのシステムとしては不適切になってしまう。このためシステム側で学習問題をクリアできたか、正解判定を行う機能も必要であるといえる。

複数の命令文に対応した SQL の演習システムを作成する場合、基本的にプログラムを介して MySQL などの RDBMS を操作することになる。RDBMS は無効な命令文を入力すると、それに対応したエラーメッセージが返す。しかし、それらは英文であり、また特に初歩的な SQL 文のミスであれば大抵の場合そのメッセージはエラーの原因となったと思われる箇所を指摘するだけであり、初心者がそれを見て、命令文の何が不適切であったのかを判断するのは難しい。また簡単なテーブル名、カラム名の勘違い、スペルミスも指摘されなければ気づかない可能性がある。こうした想定をフォローし、日本語でエラー内容を知らせ、スペルミスなどの SQL の学習に直接関係ない箇所での失敗を即座に指摘するエラーメッセージの作成機能が有用であるといえる。

以上のような前提を置き、本研究では学習問題自体を規定するデータベースと学習問題の実行に必要なテーブルを持つデータベースを、学習者が作業を行うデータベースを別に用意、学習問題ごとに必要なテーブルを学習者のデータベースにコピーする形式をとり、さらに実行可能な命令文や操作可能なテーブルを規定することで、演習の自由度と引き換えに入力される操作、エラー内容をシステム側で予測することで、これらの機能を実装する手法を提案する。

2. 関連研究

兼宗進ら[1]は特定のアプリケーションに限らないデータベース全体の概念と扱いを学ぶシステムの研究をしているが、データベースについて、高校や大学の情報基礎教育で教科書や指導要綱を検討し、その範囲で学習すべき項目として、扱いたい情報のコンピュータで扱えるデータでの表現、入れ物（テーブル）の設計、初期データの登録、データの追加、データの修正・削除、データの検索、検索結果の印刷・ファイル出力等で活用、複数のアプリケーションやサービスから利用を上げている。最後の2つはそれまでの項目を前提にしているので、本研究ではそれ以外の項目を以て「基本的な操作」の定義としている。

石毛亮太ら[2]は本研究と同じように初心者用のアプリケーションを研究しているが、SELECT 文にのみ対応とし、正解として用意された SQL 文と学習者の SQL 文の比較にすることで正解判定を行うようになっている、学習問題ごとに特定の一文の入力に限定することで節ごとの正否を判別し、間違えているときはその節を指摘する機能を実装している。正解判定を SQL 文で行っているため、極めて限定的な学習問題しか対応できない。秋吉賢治ら[3]は CGI 穴埋め問題と InetSQL というツールを用いた命令文実行練習問題を組み合わせた豊富な学習方法を提供するシステムを提案したが、学習者の失敗に対するフィードバックが希薄となっている。

3. 提案手法

3.1 前提条件

学習者が用意された学習問題を選択して実行する形式を想定する。学習問題の実行に提案する各種機能の実装のためには学習問題を実行するためには、プログラムとは別にデータベースを3種類用意する。

- ・学習者毎の作業用スペースとなるユーザデータベース
 - ・学習問題そのものに関するデータを保持する問題データベース
 - ・問題で使用するテーブルを持つ原典データベース
- 問題データベースで学習問題ごとに問題文や許可する命令文種類の他、使用するテーブルの名前を記載し、使用するテーブルそのものは問題データベース内に格納する。システム側で問題データベースの内容を参照し、それに対応して学習問題実行画面を表示、同時に必要なテーブルを原典データベースからユーザデータベースにコピーすることで問題実行の環境を整えるという手法を取る。

原典データベースには学習問題の実行にコピーする操作対象テーブルと、問題文とともに画面に表示する参照用テーブル、正解判定で正解データとして用いる正解テーブルを格納しておく。

問題データベースでは問題文などの表示情報の他に学習問題ごとに操作できるテーブルの名前や、実行を許可する命令文なども記載することになる。これによって操作できるテーブル名や命令文の種類が制限されるが、システム側で学習者の行うべき命令文の種類やその対象となるテーブルを把握することができる。

学習問題実行時に想定されるプログラムと各種データベースの相関性について図1に示す。

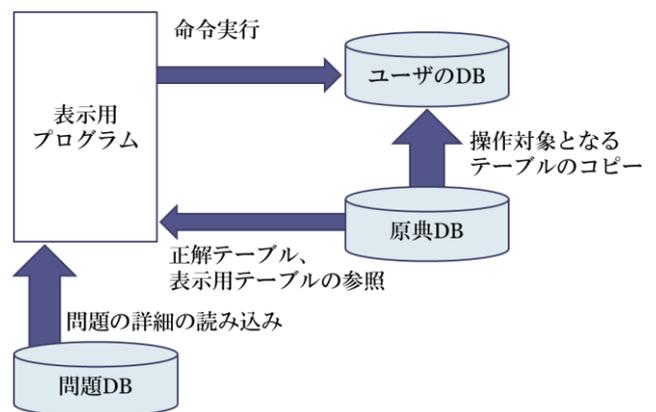


図1. データベース相関図

3.2 リセット機能

学習問題ごとに実行できる命令文を制限しても、システム全体としては基本的な操作を行うための命令文一通りに対応することを前提とした場合、学習者が間

違った操作をテーブルに加える可能性がある。この際に訂正を行うには間違える前のテーブルの状態と、現在のテーブルの状態を把握した上で場合によってはまだ練習していない命令文の実行が必要になり、初心者では困難が予想される。本研究では学習者が任意のタイミングで、その学習問題で使用するテーブルを問題開始時点の内容に戻すリセット機能を提案する。

前提条件で述べた手法を取れば、学習者が自分のユーザーデータベース内のテーブルに操作を加えても、原点データベース内のテーブルは初期状態のまま保持されるため、これを使用して再度のコピーによるリセット機能が実装できる。

実行の流れは。

1. 問題データベースより、その学習問題で使っているテーブル名のリストを取得
2. リストにある名前のテーブルをユーザーデータベースから全て削除
3. 原典データベースよりテーブルを再度コピーする

これによってその学習問題についてのみ、テーブルの初期化を行うことができる。学習問題を解いた後、繰り返し取り組む場合にも使用できる。

3.3 正解判定機能

テーブル内容の変更や望む形の表を出力するなどの学習問題の達成条件のクリアは学習者自身が判断することもできるが、それは勘違いによる誤りが生じる恐れがある。また SQL 文による正解判定を行うとすると、SQL では特定の操作を行うのに複数の命令文が想定されるのが普通であるため対応しきれず、学習者の命令文を制限する必要がある。さらに、SQL 文一回限りについてのみの判定となるため、数度の操作を繰り返してテーブルを特定の形式に変更するなどの問題に対応できない。本研究ではテーブルを参照することによって、学習問題を解いたかをシステム側で判定する正解判定機能を提案する。

変更を加えて正解判定にかけるユーザーデータベース内の判定元テーブルと、想定される操作を経たあとの内容である原典データベース内の正解テーブルを、問題データベースによって指定しておくことで、この機能を実行できる。判定元テーブルと正解テーブルを比較し、一致するかどうかで正解であるか否かを判定することになる。

しかし、学習問題によっては SELECT 文による表出力の内容を達成条件としたり、テーブルの行の順番、カラムのデータ型を考慮したりしなかったりの違いが生じると思われるため、これらの区別も問題データベース内で指定しておき、正解判定はそれによって判定

方法を変えることとする。SELECT 文の表出力については、システム側で出力そのままテーブルとして作成し、テーブルと同じように判定し、カラムの各要素の考慮については別に判定を用意、テーブルの行の順番の考慮についてはテーブルの比較手法を変更することで対応することとする。

- ① 問判定が問題データで指定された名前のテーブルに対して行われるのか、SELECT文によって出力に対して行われるのかを判別する。
- ② ①で判定元テーブルが指定されていた場合、カラムの名前・属性・プライマリキーなどのフラグも判定対象とするか判別する。
- ③ データの行の順番を判定対象とするか判別する。

この判別の結果によって、どのように正解判定を行うかを決定する。正解判定は①で表の出力と指定されていた場合、SELECT 文の出力そのままのテーブルをシステム側で作成し、それを判定に用いるとする。判別の流れと判定で行う動作を図 2 に示す。

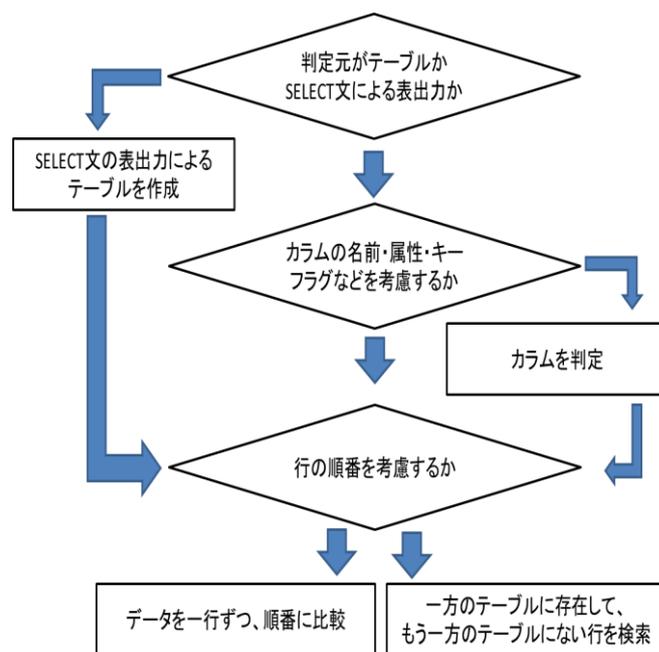


図 2. 正解判定の流れ

インストラクタがその問題で副問合せを使用させたり、REPLACE 文で文字列を置換させるなどの特定の手法を取らせたい場合には操作対象テーブルと命令文を制限した上で、SQL 文自体を判定する正解判定が必要になるため、そのフラグ管理も問題テーブルに規定する。

3.4 エラーメッセージ作成機能

本研究では RDBMS 側のエラーメッセージの日本語への置き換えと、命令文と使用テーブルについて文字

列類似度によるスペルミスの可能性の指摘からなるエラーメッセージの作成機能を提案する。

構文的には間違っていないが、指定されたテーブルやカラムがデータベース内に見つからない命令文を入力された場合の RDBMS のエラーメッセージは定型的なものとなるため、プログラム側でこれを取得、そのまま簡単な日本語に置き換えて表示する。例えば MySQL では操作するテーブルで指定されたカラムが見つからない場合のエラーメッセージ、「Unknown column 'test' in 'field list'」となる、「test」の部分以外は固定であるため、システム側で判別して「カラム test が見つかりません」などと置き換えることもできる。

指定されたテーブルが見つからない場合も同じようにエラーメッセージを日本語に置き換えられるが、その「見つからないテーブル名」を取得し、その学習問題で使用すべきテーブルの名前と比較し、似たものがあればスペルミスの可能性を指摘することができる。SQL 文の文頭の節は命令文によって変化するが、これが適切出ない場合、エラーとなるが、この時文頭の節を取得し、その学習問題で実行を許可された命令文の文頭節と比較し、こちらでもスペルミスの可能性を指摘することができる。

これらの手法は RDBMS 側のエラーメッセージに依存するため、全ての RDBMS に普遍的に用いることはできない。SQLCODE、SQLSTATE などの規定されたコード取得して、該当するメッセージを日本語で表示するという手法であれば幅広い RDBMS に対応できるが、SQLCODE は値の割り当てが RDBMS によって異なる場合があり、SQLSTATE は RDBMS によっては全てに対応している訳でなく MySQL などでは RDBMS 自体で備えたエラーメッセージの方が仔細で間違いが把握しやすいため、こちらに対応させるものとする。

4.実装

これら手法の有効性の検証のため、実際にサーバサイドのシステムを構成し、提案手法を実装した。プログラムの作成には php と SQL を使用する RDMS である MySQL を用いた。

学習者は予めユーザ登録で自分用のユーザデータベースを空の状態で作成してから、システムを使用するとし、問題データベース内の問題データテーブルは以下のような内容のカラムから構成した。

- ・学習問題の名前
- ・その問題で実行を許可する命令文の種類
- ・表示する備考文や表示参照されるテーブル名
- ・その学習問題で使用するテーブル名
- ・正解判定についての各種フラグ管理用データ

また問題データベースには問題を選択する画面の

ためのデータを保持するリストテーブルを作成し、学習問題と説明文をカラムとした。

エラーメッセージ作成の際のテーブル名、文頭節の判別のための文字列比較には、PHP での文字列類似度を判別する `similar_text` 関数を用い、類似度 80% 以上とされた場合にスペルミスの可能性を指摘するように設定した。

実装したシステムでは学習者はユーザ名と使用するリストデータ名を入力し、問題選択画面への移行、問題選択画面で学習問題を選択し、学習問題実行画面へと移行していき、適切と思える SQL 文を入力してそれを解いていくことになる。このとき、システム側は入力された SQL 文に応じて命令を実行したり、エラーメッセージを作成して表示するなどの動作を行う。図 3 にシステム概要図、図 4 に実行の流れ、図 5 に問題実行画面を示す。

こうしたシステムを実装し、幾つかの学習問題を作成、実際に実行することで有効性の検証を行った。

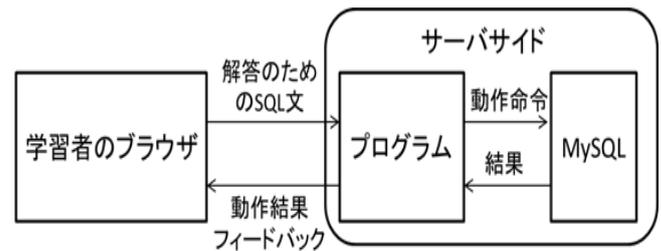


図 3. システム概要図

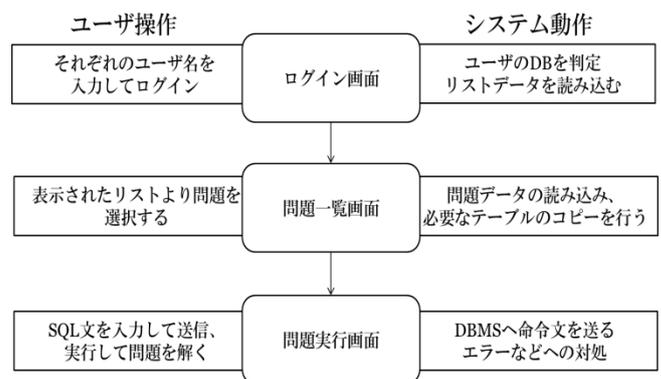


図 4. システム実行の流れ

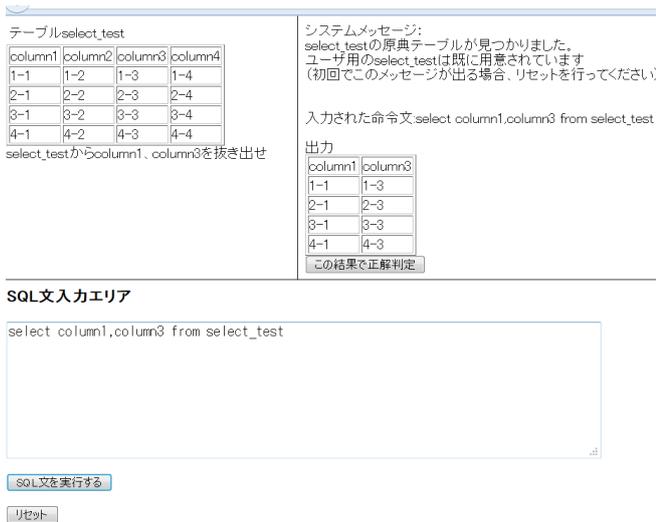


図 5. 問題実行画面

問題実行画面は、図 5、図 6 のように以下の 3 つのフレームで構成される。

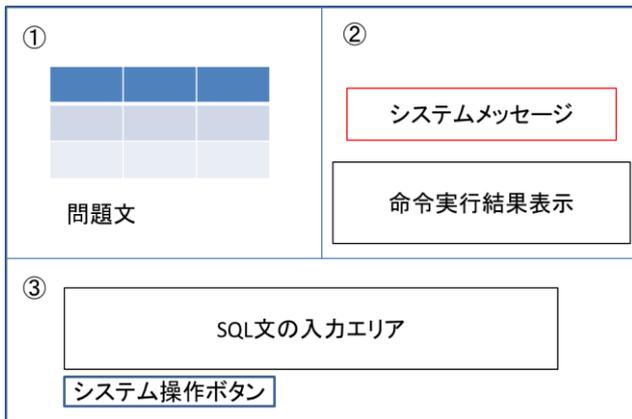


図 6. 学習問題実行画面構成

- ① 問題内容を指示する問題文、必要であればテーブルなどを表示する
- ② 問題に必要なテーブルが正常にコピーできたかなどのシステムメッセージと、ユーザが入力した命令文を受けて結果を告知、SELECT文ならば対応したその出力となる表を表示する
- ③ ユーザが命令文を入力するテキストエリアと、命令文・各種機能を実行するためのボタンを表示する

学習問題の問題文、表示テーブルの有無とその大きさにも依存するが、フレームを調節すれば問題に関する情報を一画面内に収めることができるため、ウィンドウの切り替えやスクロールの必要なく学習を行うことができるようになっている。

エラーメッセージや正解判定結果は②の部分で表示するように設計した。図 7、8 は無効な命令文、特に命令文頭節やテーブル名のスペルミスらしい間違い

に対するメッセージの例、図 9 は CREATE 文などを使って create_test と同じ内容のテーブル user_create を作成する学習問題での正解判定を実行した際の②の部分の例である。

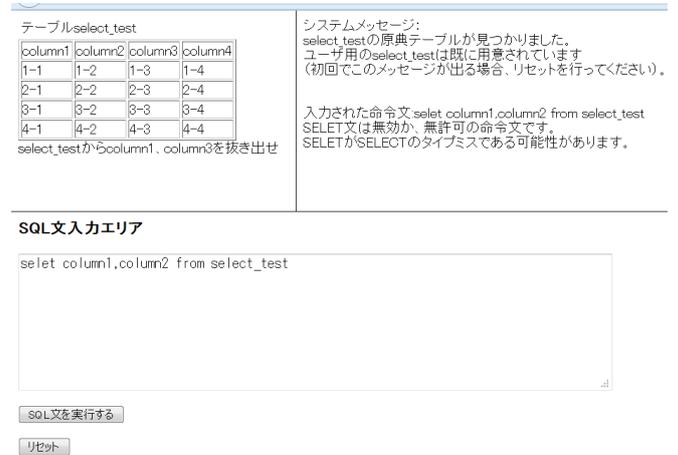


図 7. エラーメッセージ例 1

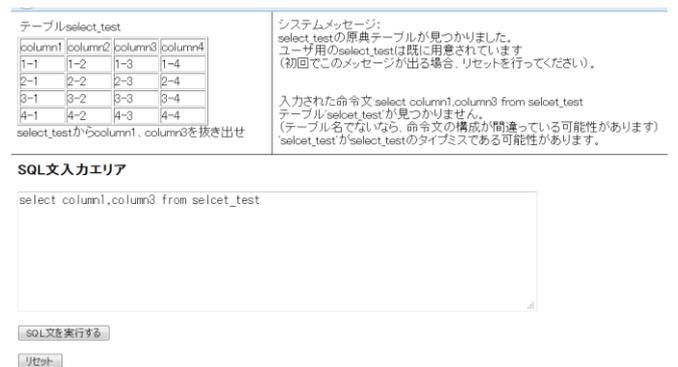


図 8. エラーメッセージ例 2

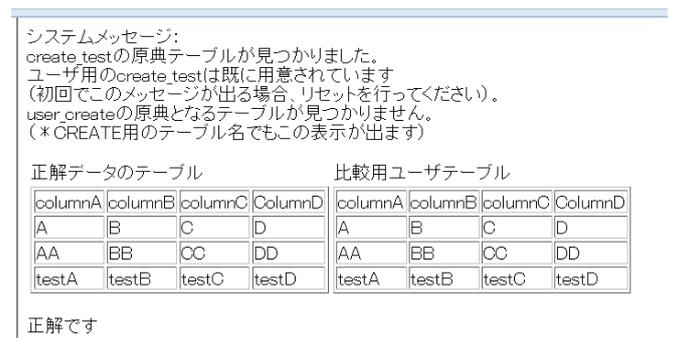


図 9. 正解判定結果例

図 8 の状態では既に問題で作成すべきテーブルを作成しているが、ここでリセットを実行すると図 10 のようにメッセージが表示され問題に必要なテーブルである create_test が再度作成されたことがわかる。さらにその状態で正解判定を行うと図 1 1 のようにメッ

ページが表示され、学習者が CREATE 文で作成した user_create が消去されたことも確認できる。

```
システムメッセージ:  
create_testの原典テーブルが見つかりました。  
create_testの複製テーブルを作成しました。  
user_createの原典となるテーブルが見つかりません。  
(* CREATE用のテーブル名でもこの表示が出ます)
```

図 1 0 . リセット直後のシステムメッセージ例

```
システムメッセージ:  
create_testの原典テーブルが見つかりました。  
ユーザ用のcreate_testは既に用意されています  
(初回でこのメッセージが出る場合、リセットを行ってください)。  
user_createの原典となるテーブルが見つかりません。  
(* CREATE用のテーブル名でもこの表示が出ます)
```

比較用ユーザテーブルが見つかりません。
指定されたテーブルを削除してしまったか、まだ作成できていないと思われる
正解データのテーブル 比較用ユーザテーブル

columnA	columnB	columnC	ColumnD
A	B	C	D
AA	BB	CC	DD
testA	testB	testC	testD

図 1 1 . 指定テーブルがない時の正解判定結果例

5. 検証

以下の SQL の基本的な内容についての簡単な問題を作成し、学習問題として実行、最小の命令で正解になるように解答する他、敢えて多数の命令が必要となるような解答を入力したり、不正解となるような解答も複数パターン入力し、メッセージの妥当性、機能の有用性などを検証した。

- SELECT 文でのテーブルからの特定カラムの抽出
- 副問合わせによる、条件一致行の検索
- 数字カラムを含むテーブルの昇順表出力
- DELETE 文でのテーブルから行の削除
- INSERT 文でのテーブルへの行の挿入
- CREATE 文で指定されたデータ内容でのテーブル作成 (カラムのキーなどは考慮しない)

問題データ、原典データベースの作成・編集は phpMyAdmin を用いて行った。

結果としては、SQL 文自体を正解判定に用いた副問合わせの学習問題以外は一通り破綻なく正解まで実行可能であった。今回は SQL 文による正解判定では全体を正解判定に用いたため、些細な入力形式の違いでも間違いとされと、とても妥当な正解判定にはならなかった大文字小文字などは対応したが、IN と = のどちらを用いるかなど SQL 文全体を把握しての判定は難しく、根本的に手法を変える必要が生じた。

今回の提案内容である各種機能について使用した上での評価を行うと、リセット機能は主観的には便利

であると思えたが、指定されたようにテーブルを作成し、データを挿入していくというやり方次第では操作量が多い問題に取り組むと、直前の命令だけ取り消したいと思える場面があった。ある程度複雑な問題に取り組む、という前提があれば独力の操作で訂正できると思えるが、問題の作成時にはリセット機能が初期化であることを意識し、細かく学習問題を区切る必要性があると感じた。しかし、テーブル内容による正解判定の利点は複数の命令文の操作による問題にも対応できることであるため、学習者にさせたいことを考えて変える必要もある。

正解判定機能については不正解時にも正解とユーザの操作したテーブルを比較できるように表示するが、メッセージ自体は「間違いです」、カラム各要素の判定のどこで不一致があったかなど、簡単なエラーメッセージしか表示してなかった。これでは正解にたどり着くまでにどのような操作が必要か、あるいはどのような操作が余計であったかなどが分からず、学習者が行き詰まってしまう可能性が想定できた。ただ問題が解けたかどうかを確認するだけの機能はさほど有用ではなく、これを有用とするには正解テーブルと学習者が操作した判定元テーブル、学習者が操作を加える前の判定元テーブルを比較して正解まで必要な操作のヒントを出す不正解メッセージを作成するなどの追加機能の必要であると思われた。

エラーメッセージの作成機能については SELECT 文や CREATE 文の文頭節、一定数以上の文字数のテーブル名の文字の入れ替わりなどのスペルミスは指摘できたが、DROP などの文字数の少ない命令文やテーブル名の場合は 1 文字辺りが影響する類似度が大きくなり、スペルミスの可能性を指摘することができなくなった。文字が少ない分、見直した時の訂正も容易だとは思われるが、手法を活かしきろうとするのなら、文字数に応じて判定しきい値を変更するなどの工夫が必要となる。

しかし、これらは SQL について知識を持つ人間の判断であり、実際に初心者ユーザとして使用してもらわなければ妥当な評価であるとは言えない。今後、そうした評価の実行やそれに先だつたより適切な学習問題の作成・実装が必要であると判断される。

6. まとめ

本研究では特に初心者を学習者として想定した際の学習問題実行支援による SQL 学習システムについて、既存のシステムより分かりやすいフィードバックと問題実行のため、データベースの分割と問題データ規定によるリセット、正解判定、エラーメッセージ作成といった機能の実現を提案した。

今回は検証のため、php と MySQL によるサーバサイドのプログラムを作成した結果、特に正解判定機能については、有用性の確立には実装に際して必要な追加機能があると判断できた。副問合わせや特定の命令文を入力させる場合、SQL 文を解析して特定の条件（例えば、副問合わせであれば 2 つの命令文頭節がある）を満たした場合、テーブルによる正解判定を行えるようにするなど、複合的に対処する必要があると分かった。他の機能については現状でも有用であるとは思われるが、更に検証が必要である。

なお、今回、php の変数に MySQL のデータを受け取ると全て string 型として扱われる点などを受け、これらをよく連携させるため、問題テーブルの正解判定におけるフラグ管理周りに変更を加えながら実装を行うことになった、またエラーメッセージの作成も特に初歩的なミスについては SQLSTATE より MySQL としてのエラーメッセージの方が多くのパターンに対応していたのでこれに依存しているが、このことから今回の提案手法は実装する手法・環境によって大いに変更・調整が必要であるとも判断できた。

本研究の本題ではなかったが、管理者視点では問題の管理を問題データベースと原典データベースで行い、学習者はそれぞれのユーザデータベース以外操作できないようにすることで、学習者側の操作に対して管理者が修正を加える必要はなくなり、基本的に指導の必要がない自習用のシステムとして、維持の手間はかなり低くなった。しかし、学習問題を作成するための初期の手間はあり、スムーズに準備を行うには多少の慣れを必要としており、MySQL であれば phpMyAdmin などのツールがあるが、RDBMS によっては学習問題の作成に専用のツールが必要になる可能性もある。

参 考 文 献

[1]兼宗進,長瀧寛之,野辺緑,中野由章『データベース操作の学習が可能なオンライン学習教材の設計と実装』情報処理学会研究報告書 vol.2013-CE-118No.7

[2]石毛亮太,小川辰也佐瀬亘輔,平野正則『Web アプリケーションによる初心者のための SQL 学習支援システム』東京情報大学研究論集 Vol.14 No.1 (2010)

[3]秋吉賢治,横尾徳保,重松保弘『SQL 学習支援システムの設計と開発』九州工業大学学術機関リポジトリ (2005)

[4] Carsten Kleiner, Christopher Tebbe, Felix Heine『Automated Grading and Tutoring of SQL Statements to Improve Student Learning』Proceedings of the 13th Koli Calling International Conference on Computing Education Research

[5]Haifeng Ke,Gaoyan Zhang,Hui Yan『Automatic Grading System on SQL Programming』'09 Proceedings of the 2009 International Conference on Scalable Computing and Communications; Eighth International Conference on Embedded Computing