

SQL データベースのオンライン学習システム の構築

宮田 祐子

目 次

1	はじめに	1
1	1 現状認識	1
2	2 他の教材との比較	2
3	3 比較結果	4
2	本論	4
1	1 製作環境	4
2	2 教材の内容	4
3	3 教材の構成	6
4	4 ファイルの構成	10
5	5 重要な部分の強調	12
6	6 入力した SQL コマンドが間違っているとき	13
7	7 フォームに入力した内容を残す	14
8	8 コピー、ペーストの禁止	15
9	9 MySQL と他の RDBMS の違い	16
10	10 SQL の種類による結果表示	17
3	結果と考察	19
1	1 間違いの指摘	19
2	2 クリアボタンが使えない	20
3	3 データベースの選択	20
4	4 データベースの肥大化	21
5	5 セキュリティ	21
6	6 サポートされていない機能	22
4	まとめ	22

1	ユーザー評価	22
2	今後の改良点	23
3	自己評価	24

1 はじめに

SQL をこれから学習する人のために、SQL をオンラインで学習するためのサイトを作成する。特別な環境を用意しなくても、MySQL コマンドを使うことのできるシステムを、PHP で Web 上に実現する。それを利用して、SQL の学習ができるようにする。

(1) 現状認識

SQL について解説している本やサイトは多い。それらを利用しても SQL を学習することはできる。しかし、文章を読んでいるだけでは、もし SQL を間違えて覚えていても気がつきにくい。実際に SQL コマンドを入力し、その結果を見る事ができたのならば、間違っていた場合にはエラーになり、実行されないのだから、その場で気がつく事ができる。そして各 SQL コマンドの解説がすぐその場に用意されているのならば、その場でどこがどう間違っているのかを確認することができる。そうして SQL コマンドの入力を繰り返しながら、学習していけるシステムがあれば、有用なものになると思う。

実際に SQL を使いながら学習していくのならば、ユーザーが環境を用意すればいい。解説本やサイトにも、環境を用意するにはどうすればいいのか、説明をしているものもある。しかし、学習を始めたばかりの初心者には、自分で環境を用意する手間が、SQL 上達の壁になっていると思う。実際に SQL コマンドを入力し、その結果を確認できるサイトがあれば、自分で環境を用意する手間は必要ない。インターネットに接続することができれば、特別な環境を用意することなく、すぐに学習に専念することができる。

(2) 他の教材との比較

(i) Microsoft Office Access

人文情報学科の学生の場合は、Microsoft Office Access 2003 がインストールされているパソコンを利用できる。Access 2003 には、Northwind.mdb というサンプルデータベースが用意されている。メニューのヘルプ > サンプルデータベース > ノースウィンドサンプルデータベースから呼び出すことが可能である。それを利用して、SQL 学習を進めていくこともできる。

しかし、Access は多機能すぎて、操作方法がわかりづらい。初心者が Northwind.mdb をインポートできたとしても、そこから SQL コマンド入力画面を呼び出す方法も、初見ではわからない。わからなければヘルプなどで検索すればいいが、ここで学習すべきなのは SQL コマンドであって、特定のアプリケーションの操作方法ではない。初心者のための SQL 学習教材に使うには、Access は向いていない。

また、Access では SQL コマンドを入力し、結果を確認した後、今度は違うコマンドを入力しようとする、前のコマンドを一々消去する必要がある。SQL コマンド入力画面に戻るときは、マウスで操作するしかない。SQL コマンドを繰り返し入力しながら学習していく場合には、Access は不向きである。自宅のパソコンではインストールをしていないために、Access を使えない学生も多い。

(ii) TECHSCORE

TECHSCORE(テックスコア)(<http://www.techscore.com/index.html>) というサイトでは、SQL について基礎から応用まで詳しく説明している。表と構文を示し、各 SQL コマンドについて解説をしている。また、実習課題とその解答例も用意されている。しかし、実習課題は各項目ご

とに 1 問である。SQL コマンドを使う環境は、ユーザーが用意しなければならない。

(iii) SAK Streets

SAK Streets というサイトには、SQL プログラミング言語資料 (http://homepage2.nifty.com/sak/w_sak3/doc/sysbrd/sak3sql.htm) がまとめられている。この中には MySQL 編もあり、環境を準備するところから書かれている。しかし、リレーショナルデータベースやトランザクションなどの単語を、ユーザーが既に知っていることを前提に書かれているので、初心者向けのものとはいえない。

(iv) SQL の窓

「SQL の窓」(<http://hp.vector.co.jp/authors/VA003334/>) という SQL 学習用のフリーソフトが存在するが、これは Windows の Internet-Explorer でしか動作しない。ソフトウェアをダウンロードし、RDBMS をインストールしなければならないので、自分の所有しているパソコン以外では学習しづらい。

(v) SQL 書き方ドリル

『すらすらと手が動くようになる SQL 書き方ドリル』(羽生章洋著) は、書き込み式ドリルの形式をとり、各章ごとに、簡単な問題から応用問題まで用意している。付属の CD-ROM には、学習用サンプルデータベースと、PostgreSQL の Windows 版が収録されていて、セットアップ方法が書かれた PDF ファイルも用意されている。しかし、Windows 以外の OS については、説明がない。PostgreSQL 以外にも、Oracle、SQL Server、MySQL にも対応しているが、それらのセットアップ方法についても書かれていない。内容は、トランザクションと、テーブル作成などスキーマに関するところが、各 RDBMS ごとに差があるという理

由から省略されている。

(3) 比較結果

比較の結果、SQL コマンドが使える環境と、教材の役目を、同時に満たしているものはほとんどない。あったとしても、使える環境が制限されている。

Web 上に学習システムを実現すれば、OS やブラウザに囚われることなく使うことができる。またインターネットに接続できる環境ならば学習できるので、場所に制限されることも少なくなる。SQL 入力画面と、解説とが、一体になっているので、自宅以外の場所でも、教材を持ち歩かなくても学習できる。人文情報学科の学生の場合は、授業の空き時間を利用して学習することができる。SQL をこれから学習していく初心者に使ってもらえるような、わかりやすい教材を作成したい。

2 本論

(1) 製作環境

制作には、Linux の一種である、Knoppix を使用する。Knoppix には既に、何もしなくても Apache、PHP、MySQL が使える環境が用意されている。そのため、この Knoppix の LAMP の環境を使用し、作成していく。

(2) 教材の内容

教材の内容を考える。文章で説明するよりも、ユーザーが実際に SQL コマンドを繰り返し入力し、実際に結果を確認することで学習すること

を重視する。そのため、説明の文章が長くなることは避け、できるだけ簡潔にするようにしたい。SQL コマンドの解説と、例文を示した後は、必ずそれをユーザーに試させ、その結果を確認させることにする。練習問題を SQL コマンドごとに数問ずつ用意し、その解答も作成する。

SQL を学習するには、まずデータベースという概念を理解していなければならない。そもそも、データベースを学習したことのない人は、SQL が何かということを知らない。最初に、基本的な単語などを説明する必要がある。

次の内容の SQL コマンドを、ユーザーに教える。

- ・ データベースの作成・削除
- ・ テーブルの作成・削除
- ・ レコードの追加・更新・削除
- ・ レコード検索

ユーザーが、簡単なデータベースを作成できて、そこから必要なデータを自在に取り出すことができるようにしたい。データベースを作成し、テーブルを作成し、レコードを追加していく。作成、追加したデータの変更、削除をできるようにする。

レコード検索の部分では、条件指定に使う演算子などの説明も必要になる。

- ・ 比較演算子
- ・ 論理演算子
- ・ 集合関数
- ・ ワイルドカード

この教材では、データ制御言語を取り扱わない。データ制御言語に含まれるのは、ロールバックと、トランザクションに関するコマンドであ

る。しかし、これらをユーザーにサイト上で試してもらい、結果を実感してもらうのは難しい。

また、MySQL のトランザクション機能には、制限がある。その他にも、結合機能、複製機能に、制限がある。カーソル、トリガ、ストアドプロシージャ、副問い合わせの機能も、使うことができない。(<http://www.atmarkit.co.jp/flinux/rensai/mysql01/mysql01.html>)

使うことができないものは、教材の内容には含めない。サイト上で SQL コマンドを試すことができるという点が、この教材の特徴である。試すことができないのであれば、他の教材で十分である。そのため、これらを説明するページは作成しない。

しかし、MySQL で使うことのできない機能のうち、副問い合わせは、SQL 教材には必ず取り入れられている、重要な機能である。サイト上で実際に試すことができないとしても、そういう方法があるということは知っておく必要がある。これは、MySQL に特化しすぎることを避けるべきだという考えからである。そのため、`sql_select_subquery.html` に、副問い合わせの説明と例文を示しておいた。

また、正規化についても、内容に含めない。効率的なデータベースを設計する上で、必要になる考え方であるが、SQL コマンドを使って学習する内容ではない。初心者向けの教材には、含まれていなくても問題ないと判断した。

(3) 教材の構成

- ・ はじめに：SQL とは
- ・ 列を取り出す
- ・ 複数の列を取り出す

- 列に別名をつける
- 重複の排除
- 演算をする
- 集合関数
- 文字列の連結
- 条件の指定
- あいまい検索
- 条件分岐
- グループ化
- グループ化した結果から抽出
- 並び替え
- 複数のテーブルを扱う前に
- テーブルの結合：where
- テーブルの結合：join
- 外部結合
- select 文を統合する
- データベースの作成
- データベースの削除
- テーブルの作成
- テーブルの削除
- レコードの追加
- レコードの更新
- レコードの削除
- 補足：副問い合わせ

(i) 導入部

最初に、データベースの概念を簡単に説明し、必要な単語を説明した。その後、データベースとテーブルを確認する方法、`show databases` と `show tables` を教える。これらは、データベースを操作する際、必ず必要になる。しかし、SQL の解説本やサイトでは、ものによっては、省かれているものがある。これらを最初に教えるのは、テーブル名を自分で知ることができなければ、ごく簡単な `select` 文を書くこともできないという理由からである。教材の中での解説や、練習問題にあたっては、こういう名前のデータベースがあり、こういう名前のテーブルがある、ということを文章で説明すればいい。しかし、こうした教材を使った学習を離れれば、そうした説明文はない。教材を離れて、ユーザーが SQL コマンドを使う場面を想定したとき、データベース名と、テーブル名を確認する方法をまず知っておくべきだと考えた。

また、この 2 つはこれ以上変化もしない。短く、単純なコマンドなので、ユーザーが間違える可能性も低い。そのため、最初に教えるコマンドとして適していると考えた。

その後、この 2 つを例に、SQL コマンドの基本的な文法を説明している。

(ii) `select` 文

`select` 文を順に教えていく。コマンドごとにページをわけていく。コマンドの説明、構文、使用例を示した後、必ずユーザーに実際に入力してもらおう。

まず、1 つの列を取り出すことを教える。次に、複数の列を取り出す。そして、取り出した列に別名をつける。この後、列の値を使い、演算をする。このとき、演算をした場合は、どんな目的で演算をしたのか、別

名をつけた方がわかりやすい。そのために、別名をつける方法を先に教えておく。

where 句を使った検索を教えていく。like 演算子を使った検索では、ワイルドカードを、使用例を示し、説明した。

結合など、複数のテーブルを扱うところでは、できるだけ使用例は単純な形のものにした。複雑で長いSQLになると、その項目で重要な部分がどこなのかを、理解しづらくなる。そのため、ごく基本的な構文を示すだけにした。

(iii) データベースの作成、削除

ユーザーが簡単なデータベースを作成する。実際にデータベースを作成するときの流れとしては、データベースを作成したのなら、その次にはテーブルの作成がくる。しかし、それより前に、データベースを削除する方法を教える。これは、ユーザーがサイトの目次の順番通りに学習を進めるとは限らないと考えたからである。

目次通りに進むと、データベースの作成を説明した後にユーザーに実際に適当な名前のデータベースを作成してもらう。その後、次のデータベース削除の章で、それを削除してもらうようにしている。データベースの削除の前に、テーブルの作成、レコードの追加などをはさむと、長くなりすぎる。長くなると、学習を途中で中断するユーザーも出てくる。もしも、データベースの削除の項目に進む前に学習を中断することになると、ユーザーが作成したデータベースはそのままサーバーに残ったままになる。残っていても不具合が起きるわけではないが、好ましいとはいえない。

そうした状態になるのを防ぐには、データベースを削除する項目のところで、ユーザーが作成したデータベースを消してもらうようにすれば

いい。

(iv) テーブルの作成、削除

テーブルの作成を教えた後に、テーブルの削除を教える。ここでも、ユーザーが作成したテーブルは、ユーザーに消してもらうようにする。

(v) レコードの追加、更新、削除

レコードの追加、更新、削除を教える。ここでは、追加・更新用のデータベースとは別に、削除用のデータベースを用意した。追加・更新の説明と練習問題に使うためのデータを、レコードの削除で消されては困るからである。

(4) ファイルの構成

作成したファイルは、5種類にわけられる。

(i) HTML ファイル

index.html は、サイトのトップページである。sql_kotae.html は、練習問題の解答を書いている。sql_select_subquery.html は、補足として副問い合わせを説明している。

(ii) CSS ファイル

sql_style.css は、サイト全体のデザインに適用している CSS ファイルである。

(iii) 教材部分の PHP ファイル

SQL コマンドについての説明と、SQL コマンドを入力するフォームをつけたページを作成した。ファイルの種類は、HTML ではなく、PHP にした。これは、SQL コマンド入力フォームに、コマンド実行結果を表示した後、戻ってきたときに入力内容が残る機能を PHP でつけたためである。

(iv) SQL コマンドを処理するための PHP ファイル

SQL コマンドを処理する過程には、2 つの PHP ファイルが使われる。sql_k.php と、sql_include.php の 2 つである。SQL コマンド入力フォームから送られたデータを、sql_k.php が受け取る。さらに、sql_k.php から sql_include.php にデータを渡して、SQL コマンドを処理している。

sql_k.php では、受け取ったデータをそれぞれの変数に入れている。フォームの入力内容は、\$sql に入れられる。その他に、元のページに戻るためのリンクを貼る URL、使用するデータベースの名前、SQL コマンドの種類を受け取っている。SQL コマンドの種類は、select か non_select の 2 種類で、name 属性 kind で渡される。これは、この後 SQL コマンドを処理する過程の条件判断に使われる。

MySQL サーバーに接続するための情報を \$connect に入れ、sql_include.php に渡している。sql_include.php に渡すのは、\$connect、\$sql、\$use_db の 3 つである。

sql_include.php では、sql_k.php からデータを受け取り、SQL コマンドを処理する関数を定義している。sql_select と、sql_non_select の 2 つである。データベースに接続し、バックスラッシュを除去し、クエリーを渡すところまでは、同じ処理になる。フォームに入力された内容に、引用符が含まれていると、その引用符の前に自動でバックスラッシュが挿入される。そのままだとエラーになるので、stripslashes 関数を使い、取り除いている。(sql_include.php6 行目、42 行目)

SQL コマンドが実行されたとき、sql_include.php から sql_k.php に 1 を渡す。(sql_include.php32 行目、46 行目) それ以外のときは、0 を渡している。(sql_include.php34 行目、48 行目)

sql_k.php は、1 か 0 かを if 文で判定する。1 が返ってきたとき、

「SQL が実行されました。」と表示する。0 のときは、\$error_bun が表示される。(sql_k.php39 行目)

sql_k.php からもとのページに戻る際、type 属性”hidden”で、sql_bun と form_name を送り返している。これは、フォームに入力内容を残すための機能に使われる。

(v) データベースファイル

sample_db.sql は、学習用データベースを作成するファイルである。5 つのサンプルデータベースを作成する。

(5) 重要な部分の強調

SQL コマンドの説明の文章では、重要な部分がどこなのか、すぐわかるようにタグで強調した。

SQL の例文を示すときも、SQL コマンド部分をタグで囲み、赤で強調した。それによって、SQL コマンドと、それ以外の、場合によって変わる部分とを、ユーザーが見分けやすいようにした。

ただ、その章で説明している SQL コマンド以外も、SQL コマンド部分は同じように強調している。一目で重要なところをわかるようにするには、その章で説明している SQL コマンドのみを強調したほうが良いのではとも考えた。しかし、SQL コマンドを入力するとき、その章で説明している SQL コマンド以外の部分が間違ってもエラーになる。説明しているコマンド以外も、同じように重要であるという理由から、SQL コマンド部分はすべて同じように強調することにした。

(6) 入力した SQL コマンドが間違っているとき

ユーザーが SQL コマンドを入力したとき、それが間違っていた場合の対処を考えなければならない。入力した SQL コマンドの文法や、指定した列名などに誤りがあり、SQL コマンドが実行されなかった場合と、SQL コマンド自体は実行されるものの、問題の答えとしては相応しくない場合と、2 通りある。

(i) エラーのときの処理

SQL コマンドが実行されなかった場合は SQL サーバーから英文のエラーメッセージが返ってくる。しかし、それをそのまま表示したのでは、ユーザーにとってわかりやすいものとは言えない。まずエラーで実行されなかったことをユーザーに知らせ、そしてどこが間違っていたのかをわからせる必要がある。それを確認した上で、必要な解説を表示し、復習してもらおう。その後、もう一度同じ、もしくは同じような問題を解いてもらおう。つまり、1 つの SQL コマンドについて、問題をいくつか用意しておく。

しかし、どこが間違っているのかを判断するのが難しい。MySQL サーバーからは、間違っているとき、その内容にそってエラーメッセージが返ってくる。試しに `show table;` と入力してみると、`ERROR 1064: You have an error in your SQL syntax. Check the manual that corresponds to your MySQL server version for the right syntax to use near " at line 1` と表示され、実行されない。しかし、このメッセージから読みとれるのは、正しくない文法で書かれた SQL が 1 行目にある、ということぐらいである。正しくは、`show tables;` であり、`s` が足りないということまではわからない。このエラーメッセージ通りに、使っている MySQL のバージョンのマニュアルを調べればわかるのかという

と、そうでもない。初心者がマニュアル(<http://dev.mysql.com/doc/>)から、自分の知りたい情報を探しだし、理解するのは難しい。

PHP でシステムを作るにしても、どこがどう間違っているのかを正確に指摘することは難しい。また、何行目にあるのかを指摘することも、できていない。

SQL コマンドが実行されなかった場合は、エラーメッセージを表示し、文法などを見直すよう、呼びかけるようにした。

(ii) 問題の答えに相応しくないとき

レコード検索などの練習問題において、求めていた結果とは違った結果が表示されていても、エラーが出ないのだから正解だと思い込んでしまうユーザーが出てくる可能性がある。練習問題には解答を用意しているので、それを確認してもらえば、間違いに気付く。しかし、システム側で正解かどうかを判断することはできない。SQL コマンド自体は実行されるものの、問題の答えとしては相応しくない場合は、間違っていることに気付くかどうかはユーザーの注意深さにかかっている。

(7) フォームに入力した内容を残す

入力した SQL コマンドが間違っていた場合、どこが間違っているのかを知るためには、フォームに入力した SQL が消えていては困る。しかし、SQL 実行結果を表示する `sql_k.php` からただ単純にリンクを貼り、それを利用して元のページに戻った場合、フォームに入力した内容は消えてしまう。そのため、`sql_k.php` から元のページに戻るとき、SQL コマンドをもとのファイルに `type="hidden"` で送り返し、それがフォームに表示されるようにした。(`sql_k.php`67 行目)

しかし、ただ送り返しているだけだと、もとのページに複数の入力

フォームがあったとき、その全てのフォームに表示される状態になってしまった。これを防ぐため、各入力フォームに名前をつけておき、それを変数 `form_name` に入れておく。このとき、`form_name` には、`back_url` の、`id` 属性部分を利用している。(`sql_k.php`33 行目) もとのページに戻るとき、`form_name` を `type="hidden"` で送り (`sql_k.php`68 行目)、`if` 文で `form_name` が指定したものと一致したときだけ、フォームに入力した内容が表示されるようにした。

ただ、この方法を使うと、フォームの下に表示されているクリアボタンを押しても、フォームの内容が消えないという問題がある。

(8) コピー、ペーストの禁止

ユーザー自身が SQL コマンドを入力することで、学習していく仕組みなので、コピー、ペースト (貼りつけ) は禁止であることを、サイトのトップページで強調した。長くて複雑な SQL になればなるほど、コピー、ペーストですませたくなるものだが、それを繰り返して入力させることが目的である。禁止と書いてあっても、コピー、ペーストするユーザーは出てくるだろう。

しかし、例文からはコピーできないように、単語間の空白を、` ` に置き換えた。これは、表示されたとき、見た目は空白と変わらないが、空白とは違う文字として扱われる。そのため、これをはさんだ例文を、そのままコピー、ペーストすると、エラーになる。よって、例文からコピーすることはできない。しかし、例文からコピーした場合でも、` ` になっている空白部分のみを入力しなおすことによって、そのまま使えてしまう。

例文などからコピーしてこなくても、練習問題を続けて解いていく

と、似たような解答になりやすい。そうすると、一部分のみを変えれば
その他は前の問題の答えを流用できてしまい、入力しなくてもよくなっ
てしまう。同じ SQL コマンドを繰り返して入力することが重要なのに、
これでは意味がない。問題ごとに別の入力フォームを使用するようにす
れば、解決できるが、それでは 1 ページあたりの内容が長くなりすぎ
る。そうすると、問題に使うテーブルなども見にくくなってしまう。

どこかからコピーし、ペーストするのを、完全に防ぐ方法は無い。こ
れは、ユーザーの自主性にまかせるしかない。

(9) MySQL と他の RDBMS の違い

SQL コマンドの書き方が、各 RDBMS で違ってくる場合がある。そ
ういう場合、ここでは RDBMS に MySQL を使用しているので、ユー
ザーには MySQL でのコマンドの書き方を教える。そうでなければ、実
際にサイト上で SQL コマンドを試してみることができない。しかし、
MySQL でのコマンドのみを教えていたのでは、適切な教材とはいえない。
ユーザーが、他の RDBMS を使用することになったとき、MySQL
での方法しか知らないとしたら困る。そのため、MySQL コマンドのみ
に特化しすぎることは避けたい。SQL コマンドの違う書き方がある場
合は、補足として、説明をすることにした。

具体的には、select 文で文字列の連結を行うとき、|| 演算子を使うも
のと、+ 演算子を使うものと、concat 関数を使うものがある。MySQL
の場合は、|| 演算子か、concat 関数を使う。|| 演算子は、ANSI モード
の場合に使われる。しかし、ANSI モードでないときには、論理和とし
て解釈されるので、concat 関数を使う。ここでは、ANSI モードではな
いので、concat 関数を使う。使用する環境によって違ってくることを、

sql_mojiplus.php の SQL コマンド入力フォームの下に、補足として入れている。

テーブルを作成するときに指定するデータ型も、各 RDBMS によって違いがある。これらを全て説明すると、ユーザーが混乱しかねない。そのため、ここではあえて MySQL で使えるもののみ、説明している。各 RDBMS によって違いがあるため、MySQL 以外のものを使用するときには注意が必要であることを、sql_table_create.php に補足として書いておいた。

(10) SQL の種類による結果表示

SQL コマンド入力フォームは、2 種類にわかれている。これは、入力した SQL コマンドによって、その後の処理が 2 通りに分けられるからである。SQL コマンドを処理する際、SQL サーバーに接続し、データベースを選択して、SQL 発行をする過程は同じである。その後の、SQL サーバーから返ってきた結果を表示する処理が違ってくる。SQL 入力フォームに、select か non_select かを、name 属性 kind で設定しておき、type="hidden" で sql_k.php に送る。それを switch 文で判定し (sql_k.php46 行目) sql_include.php のそれぞれの処理に渡している。

(i) select

select 文のときは、SQL サーバーから返ってきたレコードを一件ずつ、書き出す必要がある。sql_include.php の 20 行目から、while 文を使い、処理している。このループ処理のとき、テーブルタグを同時に書き出している。show 文で、データベースを確認するときと、テーブルを確認するとき、select 文とまったく同じ処理で表示できる。

(ii) non_select

select 文と、show 文以外の SQL コマンドを処理する過程は、まったく同じになる。(sql_include.php 40 行目 ~) kind が non_select のフォームでは、次の内容の SQL コマンドが使える。

- ・ データベースの作成、削除
- ・ テーブルの作成、削除
- ・ レコードの追加、更新、削除

(iii) 各コマンドごとの処理

select と non_select 以外にも、例えばデータベース作成のときには create_db にするなど、SQL コマンドごとにフォームを作成することも考えた。しかし、SQL コマンドの結果確認画面で、実行されたか、エラーになるのか、判定は、入力した SQL コマンドが有効なものであるかどうかで決まる。フォームに入力した SQL コマンドが、そこで入力すべきものとはまったく違っていても、エラーにはならない場合がある。もし SQL コマンドごとにメッセージを用意していたならば、ユーザーは create table 文を使い、テーブルを作成したのに、「データベースが作成されました。」という内容にそぐわないメッセージが返ってくる事態が起こってしまう。これは、データベースを作成したときと、テーブルを作成したときでは、まったく同じ過程で処理できてしまうからである。

そのため、フォームの種類は、同じ関数で処理できるものはまとめることにして、select と non_select の 2 種類のみにした。SQL コマンドが実行された場合のメッセージも、「SQL コマンドが実行されました。」というシンプルな一文にとどめた。(sql_k.php の 49 行目、56 行目)

(iv) フォームの統一

kind が select のフォームに、試しに create database 文を入力してみると、PHP エラーが出る。しかし、エラーが出ていても、SQL コマンド自体は実行されている。その後確認してみると、データベースは作成されていた。これは、このとき出るエラーは、SQL コマンドを実行して、SQL サーバーから返ってきたデータを表示する処理に失敗しているだけだからである。

kind が non_select のフォームに、select 文を入力してみると、エラーにはならず、「SQL が実行されました」と表示される。しかし、肝心なレコード検索結果は表示されない。

この2つの例のように、入力した SQL コマンドは間違っていないのに、エラーになったり、結果表示がされなかったりすると、混乱するユーザーがいるかもしれない。SQL の種類によって、2つに分かれているフォームを1つにまとめることも考えた。しかし、これには、学習を進めていく過程で、もしユーザーが全く違う SQL コマンドを入力した場合きちんと結果表示されないの、ユーザーが脱線するのを防ぐ効果が多少ある。

3 結果と考察

(1) 間違いの指摘

ユーザーが入力した SQL コマンドが間違っていた場合、正確な指摘ができていない。ユーザーアンケートでも、どの単語が入力ミスであったのかを指摘できるよう、改善を望む意見があった。エラーのときの処理でも書いた通り、MySQL サーバーから返ってくるエラーメッセージ

でも、どこがどう間違っているのかを正確には指摘できない。作成したサイトの場合、フォームに入力した SQL コマンドが練習問題の解答とはまったく違うものであったとしても、SQL コマンドが有効であったらエラーにはならない。そのために、間違っていることに気が付かないユーザーも出てくる可能性がある。

(2) クリアボタンが使えない

SQL コマンド実行結果を確認した後、PHP を利用してリンクを貼った戻るボタンを利用すると、入力した SQL コマンドの内容が消えずに、そのまま残るようにした。しかし、この方法でもとのページに戻ると、コマンド入力フォームの下にあるクリアボタンを押しても、入力してある内容が消えない。しかし、もとのページに戻るとき、使用しているブラウザの戻るボタンを使うと、入力した内容もフォームに残っている状態で、クリアボタンも使える状態で学習を進められる。ブラウザの戻るボタンを使うように書いておく方が、PHP でフォームに内容が残るようにするよりも、自然に思える。または、結果表示を別ウィンドウを開いて行うようにすれば、入力した SQL コマンドと結果を並べてみることができる。しかし、結果表示の度に別ウィンドウが開くとすると、それを閉じる手間も生じる。

(3) データベースの選択

使用するデータベースを変更するとき、`use` 文を使う。複数のデータベースを操作するとき、必要になる。しかし、これは内容に含めていない。現在の入力フォームでは、使用するデータベースの選択を、`sql_include.php` の SQL コマンドを処理する関数の中で、`$use_db` とい

う変数で指定している。(sql_include.php4 行目、40 行目)。そのため、use 文をフォームから入力しても、エラーにはならないが、データベースの変更をする意味がない。

use 文は、現在の kind が select と non_select のフォームでは使えない。そのため、use 文を使えるようにするには、別のフォームを作成する必要がある。

(4) データベースの肥大化

作成したシステムが実際に使われたとき、考えられる問題に、データベースの肥大化がある。ユーザーが作成したデータベースやテーブルが削除されないまま残っていくと、データベースは肥大化の一途をたどることになる。高速で、安定した環境での学習システムの提供を考えると、この状態は好ましいとはいえない。確実な対策としては、人の手により解決するしかない。サイト管理者が定期的に、使われていないデータを削除するようにすればいい。

(5) セキュリティ

SQL サーバーに接続するための情報は、sql_k.php の 9 行目に書いてある。ここには、ユーザー名、パスワードが書かれている。しかし、「公開ディレクトリに置くスクリプト中に、データベースへアクセスするためのユーザ名やパスワードを埋め込むのは適切ではありません」(『基礎から学ぶデータベースプログラミング』238 ページ)とあるように、この状態はよくない。セキュリティを考え、変更する必要がある。

(6) サポートされていない機能

MySQL では使うことができない機能については、副問い合わせ以外のものは、内容に含めていない。サイト上で結果を実感することができないという理由から、トランザクションについても説明していない。しかし、トランザクションの考え方は、データベースを学習するのならば、絶対に必要なものであると思う。副問い合わせと同じように、ごく簡単に説明のページを補足として作成しておいてもよかったと思う。

また、現在使うことのできない機能も、MySQL の開発ロードマップ(<http://dev.mysql.com/doc/refman/4.1/ja/roadmap.html>)を見ると、対応する予定があることがわかる。将来、これらの機能が実装されたのなら、それに合わせて、サイト上でも使えるようにすれば、より有用なものになる。

4 まとめ

(1) ユーザー評価

2005 年 11 月 9 日の時点で、作成途中のシステムを、人文情報学科の学生 3 人に使ってもらい、アンケートを行った。全員から、実際に SQL コマンドを入力し、その結果を確認できる点が使いやすいという感想を貰えた。

回答の中に、各コマンドごとだけではなく、内容を大きい章ごとにわけ、適当なところでまとめの練習問題を作成するといいいのでは、という意見があった。確かに、覚えたつもりでいても、時間が経過すると、記憶が曖昧になる。ある程度内容が進んだところで、まとめとして練習問題を用意しておけば、復習できる。

また、演算子などをまとめたページを作成するといいいという意見もあった。確かに、演算子がどこのページで説明されているのか、目次を見てもわからない。演算子だけでなく、SQL コマンドから探すことのできる索引があれば、便利だと思う。

(2) 今後の改良点

(i) 学習途中での保存

このサイトには、データをエクスポートする機能がない。そのため、学習の過程で自分で作成したデータベースや、テーブルなどを別ファイルに保存することができない。データベースの削除、テーブルの削除などの章では、その前のデータベースの作成、テーブルの作成の章でユーザーが作成したものを削除してもらうようにしている。そのため、ユーザーがなんらかの理由で削除の章に進む前で中断した場合、その時点のデータをなんらかの形で保存しておけると便利だと思う。保存しなくても、作成したものはそのままサーバーに残っている可能性が高いが、他のユーザーによって削除される可能性もないとはいえない。

また、データをインポートする機能もない。学習途中でのデータの保存に対応するのなら、それをインポートできなければならない。現在、SQL コマンド入力フォームは一度につき 1 つのコマンドしか処理できない。これを、複数のコマンドを一度に処理できるようにすれば、インポート機能に対応できると思う。

(ii) データベース権限の設定

練習問題などに使うため、サンプルデータベースを、あらかじめ用意している。しかし、テーブルの削除の章などで、ユーザーが誤って他の練習問題に必要なテーブルを消してしまう可能性がある。必要なデータ

を消されたり、変更されては困る。そうした事態が起きないようにするには、事前にデータベースの権限を設定しておけばいい。レコードを検索する場合に使うデータベースとは別に、データになんらかの変更が起これば使うデータベースを用意しておく。そして、それぞれに適切な権限を設定しておくようにしたい。

(3) 自己評価

オンライン学習教材を作成してみて、練習問題などで使うデータに変更が生じる可能性を考慮しなければならない点が難しいと感じた。レコード検索の部分ではデータに変更は生じないが、データを更新、削除する部分では、あらかじめ学習用に準備しておいたデータがそのまま残っている可能性は低い。ユーザーが増えれば、それだけ変更が生じる機会が多くなる。ある程度の変更が起きることを想定して、例文や練習問題を作成したり、使用するデータベースを分けたりしたが、万全とはいえない。

これは、ユーザーがデータを共有するデータベースだから起きる問題である。作成段階で、データベースの特性を十分に理解していなかった。そのために、セキュリティの問題などにも、対応できていない。複数ユーザーに対応するためには、ユーザーごとのデータベース権限の設定もしなければならない。今の状態では、実際に運用できる段階には達していないように思う。

教材としての面では、作ると決めた教材部分は作成したが、用意した練習問題が適切かどうか、わからない。練習問題の数、内容は適当か、そして問題を終えたあとに、ユーザーが理解し、覚えることができているのか、わからない。ユーザーアンケートを行った時点では、練習問題

—SQL データベースのオンライン学習システムの構築—

を作成できていなかったのので、確認できなかった。時間が足りなかったために、練習問題を作成していない部分もある。これでは、繰り返し入力することで SQL コマンドを覚えるというシステムの目的を果たしていない。

実際に多くの人の学習に役立ててもらおうシステムにするには、まだまだ改良が必要であると思う。

注

文献表

紙谷歌寿彦

2005 『すらすらと手が動くようになる SQL 書き方ドリル』技術評論社

羽生章洋

2003 『はじめての人のためのかんたん PHP+MySQL 入門』秀和システム社

堀川久

2003 『基礎から学ぶ Web データベースプログラミング』オーム社