

# データベース学習支援システムの利用状況と実践事例の分析

長瀧 寛之<sup>1,a)</sup> 小林 史弥<sup>2,b)</sup> 兼宗 進<sup>2,c)</sup>

概要：本稿は、著者らが開発したデータベース学習支援システムである sAccess, SQL エディタ, PHP エディタのこれまでの利用状況を分析し、授業実践事例とともにシステムの効果と課題について考察を述べる。本システム群は、データベースとそれを利用する情報システムの仕組みについて体験的な学習活動が容易に実現可能な環境の提供を目指したもので、sAccess は 2014 年から、SQL エディタと PHP エディタは 2016 年から Web 上に公開している。各システムが蓄積したアクセスログ等のデータの分析と、授業実践事例の調査の 2 つの視点から、本システム群の利用状況を確認し、今後の学習支援環境としての展開について考察する。

## A case study of web-based systems for database and information systems learning

HIROYUKI NAGATAKI<sup>1,a)</sup> FUMIYA KOBAYASHI<sup>2,b)</sup> SUSUMU KANEMUNE<sup>2,c)</sup>

### 1. はじめに

情報システムが一般生活と切り離せない存在となっている昨今、例えば高校の共通教科「情報」の指導要領において「情報システムの種類や特徴を理解させるとともに、それらが社会生活に果たす役割と及ぼす影響を理解させる」ことが学習目標の一つとして言及されているように、情報システムの社会的な役割を理解することは、情報教育における重要な学習トピックの一つである。情報システムの理解を進めるにあたっては、基礎となるデータベースの仕組みの理解も必須となるが、これらを体験的に理解しようとした場合、既存の演習支援環境は専門教育における学習を想定しているものが多く、結果として高校の共通教科「情報」や大学の一般情報教育の分野では、データベースの学習について十分に扱われてこなかったのが実情であった。この現状を踏まえ、データベースの体験的な理解が可能な実習支援環境を実現しようと著者らが開発したのが、リレー

ショナルデータベース操作実習支援システム sAccess(サクセス) である。sAccess は本格的に公開した 2014 年以降、様々な高校や大学など利用されてきた。

さらに、SQL によるデータベース操作実習を目的としたツールである SQL エディタ, PHP プログラムの作成と実行を Web ブラウザのみで行える実習用ツール PHP エディタを開発し、これらのシステムを sAccess と連携することで、データベースの仕組みの理解からそれを活用した情報システムの設計までの体験ができる環境を構築した。SQL エディタ, PHP エディタは 2016 年に公開し、公式サイト<sup>\*1</sup> でこれら 3 つのシステムを合わせて紹介することで、広く活用を呼びかけてきた。

本稿では、sAccess, SQL エディタ, PHP エディタ各ツールの利用状況について、各システムが蓄積したアクセスログの分析と、授業実践事例の調査の 2 つの視点から振り返り、本システムの現在の活用状況と今後の問題について議論を行う。

### 2. データベース学習支援システム概要

まず、本稿で対象とするデータベース学習支援システムである、sAccess, SQL エディタ, PHP エディタについて

<sup>\*1</sup> <http://saccess.eplang.jp/>

<sup>1</sup> 岡山大学  
Okayama University

<sup>2</sup> 大阪電気通信大学  
Osaka Electro-Communication University

a) [nagataki@cc.okayama-u.ac.jp](mailto:nagataki@cc.okayama-u.ac.jp)

b) [ht13a036@oecu.jp](mailto:ht13a036@oecu.jp)

c) [kanemune@acm.org](mailto:kanemune@acm.org)

それぞれ概説する．各システムの詳細な仕様については，公式サイト [1]，また著者らがこれまで発表した文献 [2][3] を参照されたい．

## 2.1 sAccess

sAccess は，リレーショナルデータベース (RDB) 操作実習用の環境として構築した学習用ツールである．sAccess は“データベースの概念”や“問題解決へのデータベースの活用”について，データベースの操作を通して体験的に学ぶことに焦点を当て，共通教科「情報」や大学の一般情報教育など専門教育でない情報教育の場面で，リレーショナル・データベースの基本操作を容易に試行できる教材として活用されることを目標とし設計されたものである．

sAccess のデータベース操作は，RDB の基本操作に対応する sAccess 独自のコマンドセットを使用して行う．「選択」「射影」「結合」といった基本的な関係代数演算に対応する日本語のコマンドを用意しており<sup>\*2</sup>，複雑なデータ操作は，これら基本命令を複数並べ，UNIX 系 OS のパイプ処理のように「ある表に対する直前の命令の結果に対して，次の命令を適用する」という逐次処理の形で実現する．入力したコマンドは随時修正や順番の入れ替えも可能であり，またその都度，各命令によるデータ操作結果の遷移が画面に表示される (図 1)．これにより，操作命令とその結果の対応を見比べつつ，試行錯誤的にデータベース操作を行うことが可能となっている．

sAccess は Web アプリケーションとして一般に公開されており，誰でも利用可能である．また操作演習用のサンプルデータベース=「プリセット DB」が予め 4 種類用意されているので，プリセット DB を利用する実習であれば，教師も学習者も特別な事前準備の必要無く，当該 Web サイトにアクセスするだけで利用可能である．

また sAccess では，“クラス管理機能”も用意されている．sAccess におけるクラスとは，利用する学校や教室 (クラス) で行う実習内容に合わせたオリジナルのプリセット DB を登録できるスペースである．クラスを作成すると ID が発行されるので，学習者は教師から指示されたクラス ID を入力すれば，そのクラス専用のプリセット DB を利用した操作実習を行える．またクラスモードでは，データベースにアクセスする際に氏名入力が必須となるため，教師側で学習者の利用状況を把握することも可能となる．

またクラス機能では，特定のプリセット DB に対して“共有”設定にすることで，学習者全員が一つのデータベースを共有して中身を編集できる状態になる．ある利用者のデータ更新や削除操作が全員に影響を与える形になるので，例えば複数人でデータの追加や削除を一斉に行うとどうなるか，またその最中に別の人々がデータにアクセスして

## sAccess: データベース実習支援ツール



図 1 sAccess スクリーンショット



図 2 SQL エディタ スクリーンショット

いと結果がどう変わるか，などデータベース管理で注意すべき問題点を体験する演習も実施できる．

## 2.2 SQL エディタ

SQL エディタは，データベース操作言語である SQL によるデータベース操作実習に利用することを目的としたツールである．SQL 入力欄と操作履歴，クエリ結果の表示というシンプルな画面構成であり (図 2)，コマンドラインでデータベースを操作するのと同じ感覚で，Web 上で SQL クエリ発行によるデータベース操作が可能である．エラーメッセージは SQL 操作の学習を支援することを想定し，日本語での表現に置き換えた上で，修正のヒントと合わせて提示するようになっている．

SQL エディタにも sAccess と同じプリセット DB が予め用意されており，学習者は特別な準備の必要なく，SQL エディタのある Web サイトにアクセスすればすぐ SQL 操作の体験が可能である．また sAccess のクラス機能と連携しており，利用時にクラス ID を指定すると，そのクラスの共有 DB に対して SQL クエリを発行することが可能になる．同じデータベースに対しての操作によって，sAccess 上で体験した関係代数演算と SQL 文の対応関係を比較しながら SQL クエリを練習することを想定している．また SQL の言語仕様上，sAccess での操作と比べて複数レコードの一括変更や一括削除が容易に実現できてしまう<sup>\*3</sup> た

\*2 英語のコマンドも用意されている．

\*3 delete 文実行の際に，うっかり where 句を忘れて 'delete from TABLE' と実行して，全データを削除してしまう，など

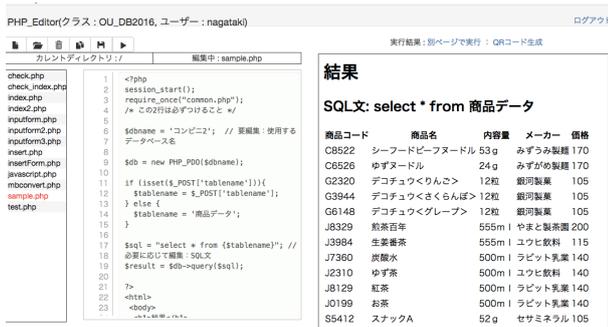


図 3 PHP エディタ スクリーンショット

め、よりデータ更新・削除の影響をわかりやすく体験できる効果もある(授業内容によってはそれだと困るケースもあるとは思われるが)。

なお安全のため、SQL エディタで利用可能な SQL 文は、select, insert, update, delete などテーブル内データの操作を行うものに限定されている。

### 2.3 PHP エディタ

PHP エディタは、PHP 言語による Web アプリケーションの作成実習に利用することを目的に構築したツールである。PHP プログラムの作成から実行、成果物の相互公開まで Web ブラウザ上で完結する仕組みになっており、事前に特別な環境を用意することなく、Web ページや Web アプリケーションの作成実習を体験することが可能である。

画面はエディタ部とプレビュー画面部で構成されており、HTML 文書や PHP プログラムを作成しては随時プレビューするという作業を繰り返しながら作成を行うことを想定している(図 3)。またプレビューした内容はそれぞれ URL が発行されるため、完成した Web アプリケーションの URL を教室内で交換すれば、学習者相互に成果物を閲覧することもできる。なお複数のファイル群を管理する実習形態となることから、PHP エディタの利用の際はユーザ作成が必須となる。ユーザを作成しないデモモードでの利用も可能であるが、その場合作成したファイルは一定時間ごとに削除される。

SQL エディタと同様、PHP エディタも sAccess のクラス機能と連携しており、データベースを活用した Web アプリケーションの作成が可能である。sAccess のデータベースへアクセスするには独自のメソッドを使用する必要があるが、メソッド群は PHP 標準の PDO クラスのものと類似させており、クエリ発行やレコード取り出しなどの手順は、PDO クラスを利用したデータベースアクセスにも応用可能なものになっている。

また教師側で、同一クラスのユーザ全員に共通のファイル群を配布できる機能も用意している。演習用のサンプルファイルを予め用意して配布する形にしておけば、学習者は一からソースコードを作成する代わりに、用意された

表 1 sAccess:月ごとのセッション数

年月	セッション数
2016-04	325
2016-05	2912
2016-06	15466
2016-07	2887
2016-08	1589
2016-09	1384
2016-10	1380
2016-11	2548
2016-12	1594
2017-01	3472

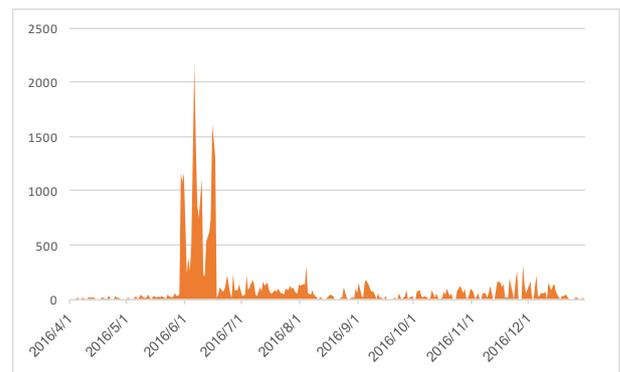


図 4 日別セッション数 (2016 年 4 月 1 日 ~ 2017 年 1 月 31 日)

ファイルを修正する形でプログラムを作成するといった演習スタイルも実施できる他、オリジナルのクラスを利用したプログラミング実習を行うということも可能になる。

## 3. システム利用状況の分析

本章では、各ツールが記録したアクセスログ等のデータをもとに、その利用状況について整理し考察を行う。

### 3.1 sAccess の利用状況

Google アナリティクスを利用してアクセス状況を確認したところ、2016 年度の sAccess の月ごとのセッション数<sup>\*4</sup>は、表 1 の通りであった。また日別のセッション数をグラフに表したものを図 4 に示す。6 月は後述する北海道大学での利用があったことからセッション数が飛び抜けて多いが、それ以外の月は概ね 1500~3000 セッションの範囲内であり、日別で見ても 6 月の期間以外は極端に跳ね上がっていないところはない。4 月のみ極端にセッション数が少ない点については、年度初めの授業ではデータベースを扱うところは少ないことが原因とも考えられるが、正確な理由は不明である。

\*4 Google アナリティクスにおける「セッション」とは、特定期間内における Web サイト内の一連の操作を指す。Web サイト内で 30 分以内に操作を連続している間は同一セッションと判定し、直前の操作から 30 分以上空いた場合や午前 0 時をまたいだ場合は、別セッションとしてカウントされる。詳細は <https://support.google.com/analytics/answer/2731565> 参照

sAccess でのクラス機能の利用について、2017年2月までに作成されたクラスの総数は384であった。ただしテスト用に作成されたと思われるものや、作成後データベース操作を行った記録がほとんどないものも多数含まれていたため、それらを除いた少なからず利用された形跡のあるクラス、具体的には(1)クラス名に「テスト」が含まれておらず、(2)ログインしたユーザ数と同数以上、操作コマンド追加操作が実行されているという基準を満たすクラスは252であり、このうち中学は1校1クラス、高校は10校20クラス、大学は13校233クラスであった(その他一般機関が1クラス、実施機関不明が6クラス)。さらにsAccessでデータ操作を実行したユーザが5人以上確認されたクラスは209で、この条件を満たすクラスは中学1校1クラス、高校3校13クラス、大学8校192クラスまで減少する。(その他一般機関1クラス、不明2クラス)。学校数とクラス数が一致していないのは、同じ学校内で複数クラスが作成されていたケースの他、後述する東京学芸大学や岡山大学の事例のように、データベース作成環境として個人ごとにクラスを作成したケースが存在しているためである。

一方、もともとのsAccessの利点として「ユーザ登録や事前準備なくすぐ利用できる」というものがあり、予め用意されたプリセットDBでデータベース操作演習を行うだけであれば、必ずしもクラスを作成する必要はない。操作履歴を確認したところ、sAccessで「コマンド追加」の操作が行われた総数は、クラスモードでは全クラス合わせて194,525回であったのに対し、ゲストモードは469,501回と大きく上回っており、アクセス元IPアドレスから判明しただけでも、複数の教育機関からゲストモードへのまとまったアクセスがあることが確認されている。ゲストモードで授業を行うケースが多いことがうかがえる。

### 3.1.1 sAccess コマンド利用状況

sAccessで入力されたコマンドについて、その利用傾向を確認した。今回はsAccessのコマンド体系の現状について確認するため、主に入力コマンドの種類と入力エラーに着目して整理することにした。なお入力エラーに着目する目的から、ここで集計対象としたのはコマンド入力欄から入力したコマンドのみとし、入力したコマンドの修正操作については集計に含めていない。

主に入力されたコマンドは表2の通りであった。基本的な関係代数演算である「選択」「結合」「射影」の使用が圧倒的であり、これは本システムが、高校の教科「情報」の教科書にも記載されているこれらの演算を理解するための環境として設計したことからも考えても、妥当な結果である。その後、テーブル「表示」コマンド<sup>\*5</sup>、選択操作の一つである数値比較を行う「比較する(比較)」の後に、「平

<sup>\*5</sup> データベース操作画面では、最初のテーブルを表示する「表示」コマンドが最初から入力されているが、学習者自ら入力していないため今回の集計には含めていない。

表2 sAccess 入力コマンド数(上位15)

コマンド名	入力数
選択	226967
結合	118588
射影	90064
表示	32077
比較する	26143
平均	15597
列指定	12974
整列	9385
数える	6839
カウント	5370
合計	4543
並べる	4369
比較	2627
取り除く	1698
重複削除	1060

表3 sAccess 入力コマンドエラー分類

エラーパターン	個数
フィールド名	2515
値	1152
テーブル名	816
データベース名	32
ローマ字	1991
覚え違い	751
存在しない命令	689
タイプミス	179
区切り記号なし	2426
区切り記号違い	203
正常コマンド	564125

均」「整列(並べる)」「数える(カウント)」「合計」など、集計用のコマンドが多く使われている。「列指定」が比較的多いが、「射影」とは動作が異なる(「射影」は重複分を一つにまとめるが、「列選択」は重複したまま)ことを意識して使い分けていたかどうかは不明である。

また、入力エラーについて、同じコマンド入力ミスが10回以上記録されたもののパターンを分類した結果を表3にまとめた。もっとも典型的なコマンド入力ミスとして、「検索したいキーワードだけ入力して、操作命令が指定されていない」ケースが多く見られた。たとえば、ある値を特定のフィールドに持つレコードを抽出したい場合、sAccessでは「選択 フィールド名 値」と、選択操作であることを明示する必要があるところ、「フィールド名 値」、「テーブル名 値」、あるいは「値」だけ入力しているケースである。表3ではそれぞれ「テーブル名」「フィールド名」「値」が該当する。「データベース名」だけ記入しているケースも若干存在した。「キーワードを入力すると、それが含まれる結果が一覧表示される」という検索エンジンのイメージが強く影響していて、データベースの操作は「データの取り出し」だけではないということ意識するまでにギャップ

があったことが考えられる。もっともこのケースはデータベースの仕組みを学ぶ観点からは、経験が望ましいギャップであると考えられる。

存在しないコマンドを入力したケースには、大きく分けて日本語入力モードへの設定忘れや単純なタイプ・変換ミス(「SENNTAKU」825件,「洗濯」96件), sAccess で用意されていないコマンド名を入力したケース(「フィールド」64件), また関係演算の用語を覚え違いしていると思われるケース(「統合」267件,「投射」68件)などがあった。

その他, コマンド名とオプションの区切り文字を空白でなくカンマ等にしたケース(表3「区切り記号違い」)が該当, 例:「選択, フィールド名, 値」), コマンド名とオプションを空白で区切らずに列挙するケース(表3「区切り記号なし」)が該当, 例:「選択フィールド名値」)が存在していた。特に命令とフィールド名の間に空白を挟まないケースが目立った。そもそも日本語では空白を文字の区切りに使わないため, 空白で区切るという発想自体がコーディングに触れていないと馴染みが薄い文法であり, その点で戸惑うケースが少なくないことが見て取れる。たとえば「価格」フィールドの合計を求めようとして「合計 価格」とすべきところを「合計価格」と入力しても, 日本語として意味が通っているので一見文法間違いに気づきにくい。区切るにしても, 「合計, 価格」のように普段使いなれた読点で区切りたくなることも無理はない。このエラーについては, 授業担当者からも多くの学習者が戸惑っていた事例として報告されていることから, たとえば明らかにコマンド名とオプションの区切りが自動判別できる場合は, 空白を補完する, 読点でも OK とするなど, 何らかの改善方法を検討する必要があると考える。

なおいずれのコマンド入力ミスも, ほとんどのクラスで初日に多く発生しており, また総数を見ても正常なコマンドの数に比べてわずかであることから, 大半の学習者はすぐに sAccess の文法に慣れていったと考えられる。

### 3.2 SQL エディタの利用状況

2016 年度 4 月以降の, sAccess のクラス機能と連携して SQL エディタが利用された状況は, 表 4 の通りであった(伏せ字の学校名は, 後述する授業実践事例で紹介する大学以外の教育機関である)。sAccess と比べると利用学校数, クラス数ともぐっと少なくなる。この原因として, SQL エディタから sAccess のクラス内データベースにアクセスするには, そのデータベースを“共有”状態にしておく必要がある仕様だったことから, 使いにくさがあったことが考えられる。この点は今後システムの改善が必要である。

またクラス機能との連携による SQL エディタの利用記録は大学のみで, 高校の利用はなかった。高校で SQL 文まで実習を行う時間的余裕のあるところは少ないと考えられることから, 想定される結果とも言える。なお, 岡山大

表 4 SQL エディタ:クラスごとのクエリ発行数

学校名	クラス数	クエリ数
大阪電通大	18	22845
岡山大	16	2084
東京学芸大	1	284
S 大学	1	484
K 大学	1	444
G 大学	1	183
A 大学	1	18
テスト用	5	88
ゲストモード	-	15691

表 5 PHP エディタの利用状況

学校名	クラス数	ユーザ数	ファイル数			
			.php	.js	.html	.css
大阪電通大	14	589	82	75	2147	24
岡山大	15	127	275	0	2	0
東京学芸大	3	6	7	0	0	0
K 大学	1	12	68	0	22	33
テスト用	7	4	3	0	1	0

学と大阪電気通信大学ではクラス数がほぼ同等にもかかわらずクエリ数に大きな差があるのは, クラスの使い方が異なることが理由と考えられる。詳細は 4 章で紹介する各大学の実践事例の中で述べる。

一方, ゲストモードでのクエリ数も一定数あり, クラス作成の形跡がないいくつかの教育機関の IP アドレスからまとまったアクセスも確認されたことから, こちらも sAccess と同様, 準備不要で利用できるゲストモードでの授業実施, あるいは自主的に利用したケースが多数であったと考えられる。実際ゲストモードの利用履歴には, 数は少ないながら高校からのアクセスも確認された。

### 3.3 PHP エディタの利用状況

2016 年 4 月以降の PHP エディタの利用状況は, 表 5 に示す通りであった。SQL エディタと比べても利用数が少なくなっている。なおここで“ユーザ数”はクラスごとの人数の合計であるため, 同一人物で複数のクラスにログインした場合は複数回カウントされている。そのため岡山大学の場合は実際の学生数よりはるかに多いユーザ数となっているが, この理由については 4 章にて後述する。

作成されたファイル数を見ると, PHP エディタの利用形態が, HTML と JavaScript に重きを置いた演習を行っているケース(大阪電気通信大学)と, PHP プログラム作成に重きを置いているケース(岡山大学)に分かれていることがわかる。しかし総合的に見て, PHP エディタは活発に使われているとは言い難い。理由としては sAccess でのクラス作成が必須など利用までの敷居の高さや, 実践事例の結果から, エディタ自体のユーザインタフェースに改良の余地があることも考えられる。

## 4. 授業実践事例

2016年度に sAccess, SQL エディタ, PHP エディタを利用して授業実践が行われた事例のうち, 実際に担当教員から情報提供を受けたものについて以下に紹介する.

### 4.1 北海道大学の事例

北海道大学では, 2016年度1学期の開講科目「情報学1」[4]の授業の中で, sAccess を利用した演習が実施された. 具体的な内容は, 第7週の20-30分で sAccess を利用した授業内実習を行い, 学生は授業外で課題を実施(余裕がある学生は追加の加点課題も実施)した上で, 次回の授業中に5-10分のミニテストを実施するというものであった. 対象学生は約2700人であり, 同一機関が同時期に sAccess を利用する人数としては, 本稿作成時点で過去最大であった.

この授業ではオリジナルのデータベース演習教材を作成したのに合わせて, データベースも独自のプリセットDBを用意することになったが, 本科目は大量の少人数クラスで構成されているため, 演習実施にあたって, 同一のプリセットDBを持つクラスを大量に作成する必要があった. 当時の sAccess は Web 上でクラスを1つずつ作成する機能しか有していなかったが, これでは今回のケースには負担が大きすぎるという要望を受け, 複数クラスをまとめて管理できる「教師モード」を用意し, さらに特定の教師ユーザに対しては, 一つのクラスを雛形として, プリセットDBを含めてまるごと複製したクラスを大量生成できる仕組みを実装した. 結果, 授業内実習用クラス, ミニテスト用クラス, スタッフの検証・雛形用クラスも合わせて, sAccess 上に合計192クラスが作成された.

クラスごとに異なる開講時間が設定されているとはいえ, それでも sAccess では前例のない数の同時アクセスが行われた結果, 実習が開始された当初に, 授業中に sAccess がエラーを返すという問題が頻発する現象が発生した. 当初 sAccess の処理によるサーバへの過負荷が原因かと思われたが, 実際は当時 sAccess が稼働していたレンタルサーバの Web サーバが, 短時間の大量アクセスを検知すると自動的に一定時間のアクセス規制を行う仕様だったことが原因であった. そのため, 急遽アクセス制限が発生しない別サーバに北大専用の sAccess を別途用意し, 当年度はそちらで利用してもらう形をとることで, アクセス制限の問題を回避し無事に演習を終了することができた. ただし結果的にアクセスログが分散し, 教師側にとって負担が増してしまった点は否めない.

### 4.2 東京学芸大学の事例

教員養成大学の情報科学の科目群中「データベース」(学部2年生前期)にて, データベース構築の実践を目的として sAccess と SQL エディタを利用した演習が行われた.

sAccess が予め用意しているプリセットDBを用いて, まず関係表と演算についての演習を行った後, 各自がオリジナルのデータベースを作成する演習を行った. オリジナルのデータベース作成は, まず各履修生が考えたテーマのデータベースを設計し, sAccess に登録可能な CSV ファイルとして作成して登録, 設計内容の検証を行った上で, SQL エディタを利用して最終的なデータベース設計を行うという流れであった. 履修人数は18人で, 最終レポートまで提出できたのは15名であった. まず sAccess でテーブル定義を行い確認することで, RDBの基本機能の理解につながり, その上でSQLによるDB操作を行うことで, 殆どの受講生はSQLも理解することができた.

データベース作成の活動が主体であったことで問題になったのは, sAccess では学習者個人のデータをサーバ側に保持しておく機能がない点であった. sAccess 上のみで作業している間は, 操作中のデータベースをファイルとしてダウンロードする機能があるためそちらを作業保存データとすることで対応できたが, SQL エディタから利用する場合に, sAccess 側では個人のデータベースを保持しない仕様のため, そのままでは自分の作成したデータベースに対してSQLクエリを発行する手段がなかった. そこで現状ある機能の範囲での対応として, 各自(グループ単位)に独立した「クラス」を作成して割り当て, それぞれ割り当てられたクラスの管理者になって, そこに自作のデータベースをプリセットDBとして登録するという方法を取ることになった. ただし急遽対応となったため, 受講生がその作業に手間取ってしまい, 結果として成果物の完成度が下がってしまうこととなった点が反省点である.

現状の sAccess が, 学習者が積極的にデータベースを構築することを想定した仕組みになっていないことは, PHP エディタとの連携にあたって問題となる部分であり, データベースや情報システムを体験的に学ぶツールとしては今後の機能改善が急務であると考えられる.

### 4.3 大阪電気通信大学の事例

工学部と医療福祉工学部で開講される情報基礎科目「コンピュータリテラシー2」の中で, 1回目に sAccess, 2回目にSQLエディタを使用するという時間配分で実習が行われた. 内容は2回ともプリセットDBの「コンピニ」「図書館」を利用し, データベースの基本操作である選択・射影・結合操作と集計操作を, sAccess コマンドとSQLそれぞれについて操作を行うものであった. 独自のプリセットDBを準備するわけではなかったが, 学生の活動状況を把握する目的で, sAccess 上でそれぞれのクラスを作成した. テスト用を除き, 実際に授業で使われたクラスは合計で13であった.

授業の実施時期は10月から12月の間で, 実施日時のタイミングはクラスごとに異なっていたが, 8クラス中4ク

ラスは授業日がほぼ重なっていたことから、北海道大で発生した sAccess のアクセス規制の現象を避けるため、うち 2 クラスは 1 回目に SQL エディタ、2 回目に sAccess と順番を入れ替えて授業を実施し、同時アクセス数が分散するようにした。

授業実践時の教師の観察によると、実施の順番にかかわらず sAccess の回は SQL の回に比べて学習のつまずきが少ない様子であり、また当初懸念されたシステム上のトラブルも特になかった。一方教員側からシステムへの要望として、今誰がつまづいているか、どの部分でつまづいているのかを把握しやすくできないか、という意見が出た。また sAccess 操作時によく見られた学生の行動として、想定と異なる操作結果となった時、コマンドを見直してデータベース操作の仕方を修正するのではなく、コマンドを全て削除して一から入力し直す、つまり闇雲に試行錯誤するケースが散見された。「つまずきのポイント」をシステム側としてうまく抽出できる仕組みを提供することが、学生と教師双方にとって重要であると考えられる。

#### 4.4 岡山大学の事例

2 年生以上を対象とした、3 学期 (10 ~ 11 月) 開講の教養教育科目「データベース入門」で、sAccess、SQL エディタ、PHP エディタを利用してデータベース操作、設計から情報システム作成まで体験する授業を実施した。

本科目は大学の時間割の関係で、毎週 60 分 × 2 コマ (連続)、計 8 週 (8 週目は 60 分 1 コマのみ) を、2 ヶ月間で実施する形式であった。最初の 2 週は、sAccess 操作を通してリレーショナル DB の構造と基本操作の概念を学ぶ実習を実施し、3 週目から正規化の説明と練習を行うとともに、SQL エディタを用いた SQL 文の構文演習を、sAccess の命令との対応関係を意識させつつ実施した。第 4 週以降は PHP エディタの利用を開始し、最低限の HTML と PHP の構文に徐々に慣れていきながら、自作のデータベースにアクセスするオリジナルの Web システムを作成するという演習を実施した。なお Web システム作成においては、単独もしくは 2 人グループいずれかで実施してよいこととした。グループごとに個別にクラスを作成して、その中に設計したデータベースを入力するという形を取った。3.2 節において、SQL エディタでクラス数と比較してアクセス数が少ないのは、各クラスのデータベースを扱う人数が 1 人もしくは 2 人という状況であり、全体としての履修人数が少なかったことが理由である。

受講生の大半は、データベース操作はもちろんで、プログラミングの経験もない状態であったため、PHP エディタについては一から作成するのではなく、基本となるサンプルファイルを予め用意して配布し、さらに別途配布した電子資料に、今回の実習で利用する可能性の高い PHP 命令をサンプルコードとして掲載することで、必要なコードをコ

```
$db = new PHP_PDO("コンビニ");  
$db = new PHP_PDO("コンビニ");
```

図 5 PHP エディタ上での double quotation 表示  
(上:正常,下:エラー)

ピーしたりサンプルファイルを編集したりする形で、自分の目標とするインタフェースの Web システムに修正していくようにした。それでも Web システムを構築するには、PHP だけでなく HTML の基本的な構文も理解する必要があり、PHP エディタに触れる期間が実質 1 ヶ月強という短期間では、プログラミングや Web ページの作成経験がない学生には消化不良に陥ってしまったケースも少なくなかった。結局最後まで成果物を完成させたのは 8 グループ 10 人であった。また成果物である Web システムは、ユーザ登録機能まで含む画像共有 SNS を実装したのもあれば、配布したサンプルファイルのアクセス先データベースを自作のものに置き換えるのが精一杯だったというものもあり、個人差が大きく開く結果となった。なお、最後に互いの成果物を相互評価する活動も実施したが、PHP エディタでは、そのクラスに自分のユーザを登録しない限り、クラス内の別ユーザが作成した Web ページを閲覧できない。3.3 節の図 5 にて、岡山大学のユーザ数が実際の履修人数よりはるかに多いのは、成果物を閲覧するためだけに各クラス内にユーザを作成した結果であるが、この点はあまり効率が良い方法とは言えない。

sAccess や SQL エディタの利用については大きな混乱はなかったが、PHP エディタ利用時に、ソースコードに誤りが無いにも関わらず、データベースへアクセスする処理で内部エラーが発生することが何度かあった。後で、その時間帯に別のところから sAccess への一斉アクセスが重なっていたことがわかり、レンタルサーバのアクセス規制が起っていたことが原因であった。

また、PHP エディタはコードの色付けや自動インデントなど作成支援としての機能を搭載していないことから、単純な構文ミスでもその原因を見つけるのに苦労する学生の姿が目立った。さらに PHP 構文のコピー用として用意した配布資料は PowerPoint スライドで作成していたため、記入した double quotation 記号「”」がオートコレクト機能で自動的に別文字「“」に修正されていた上、そのまま Web 上の PHP エディタの編集画面にコピーすると、PHP の文法では「“」記号が double quotation として認められず、それによる文法エラーとして遭遇するケースも目立った。エディタ上の文字が小さいことから画面上は非常に微妙な差異でしかなく (図 5)、エラーの原因を自力で判別できる学生はほぼ皆無であった。本システムが目指す学習環境としては望ましいつまずきではないことから、インタフェースの点で改善が必要である。



図 6 sAccess:各学習者のコマンドリスト表示画面

## 5. 今後の発展に向けて

sAccess が稼働していたレンタルサーバが、大量アクセスに対してアクセス規制を行う仕様になっていたことで、北海道大など大人数がアクセスする授業において sAccess が一時利用不能に陥ったこと、またその後も岡山大学の事例のように、個々の同時利用人数は少なくとも、多地点で同じタイミングで一斉アクセスが発生すると同様のトラブルが発生する事例が増えてきたことは、裏を返せば想定以上に sAccess の利用事例が増えてきていることを示している。なおこの問題について、現在は別のサーバへ sAccess を移設しており、大量アクセス時に突然アクセス制限がかからない環境になっている。ただし今後もサーバの過負荷による制限には注意を払う必要があるため、継続的に内部処理についても高速化する必要がある。

今回各システムのアクセス履歴を分析することで学習者のコマンド入力状況の掘り起こしを図ったが、これらの情報をリアルタイムで有効活用できれば、学習におけるつまづきをうまくサポートできる可能性がある。現時点で、sAccess には操作ログをもとに各学習者のコマンド状況を表示する機能がある(図 6)が、各ユーザのコマンド列を表示しているだけでは、つまづいている学習者を把握する目的には不十分である。授業実践事例の聞き取りの過程でも、複数の担当教員から同様の点を指摘されている。それまでの操作の過程、つまり学習者の思考の過程も把握したうえで、適切なヒントを提示できるようにするのが理想であるが、どのように実現していけばよいか、プログラミング学習におけるソースコードのエラーを自動検出する先行研究 [5], [6] を参考に、本システム群での実装を検討していく必要がある。

また今回の授業実践の結果から、データベース作成演習を実質化するには、sAccess での個人ユーザ管理機能の実装が急務である。しかしそれは同時に、サーバの保存データの肥大化に対してさらに注意を払う必要があることも示しており、定期的に休眠状態のクラスを整理するなど、空き容量不足によるサービス停止に陥らないための実装の工夫を検討する必要がある。

また、sAccess の個人データ管理機能への対応も重要な

検討課題である。sAccess は、もともと「データベースの学習にはほとんど時間をかけられない」授業環境を想定し、ユーザ管理なく利用できる環境を想定して作成されており、実際にゲストモードでの利用が非常に多い状況を見ても、この設計は一定の成功を得ていると考えられる。ただし SQL エディタや PHP エディタと連携して、学習者個人でデータベースを作成する学習環境の実現を考えた場合、「個人ごとにクラスを作成する」ような対応では、学生の活動記録が分散してしまうため、学習支援ツールとしては理想的な状態とは言えない。

しかし個人ごとにデータベース作成環境が作られるということは、個々に大量のデータを含むデータベースの登録によって、サーバの保存データ量の肥大化がより深刻になる可能性が高い。実装にあたっては、登録データベースの管理方法、また定期的に休眠状態のユーザデータを整理するなど、空き容量不足によるサービス停止に陥らないための実装の工夫を検討する必要がある。

## 6. おわりに

本稿では、データベース学習支援システムである sAccess と SQL エディタ、PHP エディタの利用状況についてまとめ、今後の方向性について議論を行った。

今後は、特別な準備なくデータベース操作実習ができるという sAccess の利点は維持しつつ、データベース作成活動にも十分対応できる、情報システムの仕組みを体験的に学べる環境の実現を目指して、各システムの機能を強化することで各システム群の利用拡大をはかる予定である。

謝辞 本研究は JSPS 科研費 16K16319 の助成を受けたものです。

本稿作成にあたって授業実践の情報提供をいただいた、北海道大学 布施泉先生、東京農工大学 並木美太郎先生、大阪電気通信大学 島袋舞子先生に感謝いたします。

## 参考文献

- [1] sAccess, <http://saccess.eplang.jp/>
- [2] 長瀧 寛之, 中野 由章, 野部 緑, 兼宗 進: “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提案”, 情報処理学会論文誌, Vol.55, No.1, pp.2-15 (2014-01)
- [3] 小林史弥, 西川弘恭, 林康平, 島袋舞子, 長瀧寛之, 兼宗進: “情報システム学習を指向した Web ベースのツール開発と授業利用の報告”, 情報処理学会研究報告, Vol.2016-CE-134, No.3, pp.1-8 (2016-03)
- [4] 布施泉, 岡部成玄: “ぺた語義: 北海道大学における全学教育としての情報教育”, 情報処理, Vol.52, No.10, pp.1341-1345 (2011-09)
- [5] 長慎也, 長島和平, 間辺広樹, 兼宗進, 並木美太郎: “初学者向けプログラミング授業における活動ログの評価支援機能”, 情報処理学会研究報告, Vol.2016-CE-138, No.4, pp.1-8 (2017-02)
- [6] 間嶋義喜, 大賀賢志, 竹内和広: “機能に基づくプログラム構造比較に向けた類似度の策定”, 情報処理学会研究報告, Vol.2017-CE-138, No.12, pp.1-6 (2017-02)