

教育実践報告

データベースをオンラインで教える試みについて

— 2024年教員免許証認定講習会「データベース」オンライン実践報告 —

室谷 心

松本大学大学院総合経営研究科, 松本大学総合経営学部総合経営学科

A Trial Online Class for Database Management

MUROYA Shin

Graduate School of Comprehensive Management, Matsumoto University
Department of Comprehensive Management, Faculty of Comprehensive Management, Matsumoto University

要 旨

長野県教育委員会の2024年教員免許証認定講習会「データベース」の講義は、新型コロナウイルス感染症再流行の恐れがまだ無視できないということから、30名を対象としたオンラインでの開催となった。データベースは教職課程科目の『情報システム（実習を含む。）』であり、本認定講習会では、ブラウザで実行可能な環境であるGoogle Colaboratory上でPythonのライブラリーとして利用可能なDBMSであるsqlite3を使って実習を行うことにした。オンラインでのSQLを使った実習の実践を報告する。

キーワード

情報教育 高等学校教科情報 オンライン授業 データベース Python

目 次

- I. 初めに
 - II. Colab上でsqlite3を使ったDBMS実習環境の構築
 - III. SQL命令の学習
 - IV. まとめ
- 謝辞
注
文献

I. 初めに

長野県教育委員会からの依頼で、2024年教員免許証認定講習会をオンラインで行った。依頼はデータベース論1単位の実技を伴う授業で、90分授業7コマを2日間で行うものであった。データベースの実技を想定し当初10人程度の対面授業の計画であったが、情報科の教員が不足しているという現場の事情もあって参加希望者が増え、また新型コロナウイルス感染症再流行の恐れがまだ無視できないことから、30名を対象としたネットでの授業となった。

高等学校学指導要領（平成30年告示）¹⁾ 第7節情報では、「第1情報産業と社会」では指導項目（3）コンピュータとプログラミングで“ウ 情報通信ネットワークの活用 データベースの活用”について扱うこととなっていることに加え、第8はそのまま「データベース」となっており、内容として、表1の項目が挙げられている。昨年の『中央教育審議会初等中等教育分科会教員養成部会教科に関する専門的事項に関するワーキンググループ(技術・情報)(第3回)議事次第』で記述が変更になったが、データベースは従来『情報システム(実習を含む。)]』であり、表1の(3)(4)パートがその主たる実習の対象であった。

本年度のこの認定講習会では、1単位2日間計7コマということで、1日目の午前中の2コマで、(2)の概略と正規化を扱い、午後の2コマと2日目の3コマを、SQLを使ったデータマネージメント(3)の実習にあてた。本稿で報告するのは、このSQLを使った実習の部分である。

データベースの実習を行うには、データベースマネージメントシステム(DBMS)が必要である。高校や大学でのデータベースの授業では、マイクロソフト社のACCESSのようなパソコンのアプリケーションを使って各人のローカルなシステムでDBMS利用の実習を行う場合と、PostgreSQLやMySQLのようなサーバをネットワーク上に立ち上げ、サーバ・クライアント型のシステムで行う場合がある。

本認定講習会の計画に当たって利用システムの検討を行ったが、講習を終えた受講者が教科担当者として情報の授業を行うことを考えると、高等学校の初級レベルのデータベースの授業では、データベースシステムサーバの運用よりもクライアント側で

表1 第8データベース

2 内容

〔指導項目〕

- (1) データベースと私たちの社会
 - ア データベースと社会との関わり
 - イ データベースを支える情報技術
 - ウ データベースの目的と機能
 - エ データベースのデータモデル
- (2) データベース管理システムとデータベースの設計
 - ア データベース管理システムの働き
 - イ データの分析とモデル化
 - ウ データベースの正規化
- (3) データとデータベースの操作
 - ア データの操作
 - イ データベースの定義
 - ウ データベースの操作
- (4) データベースの運用と保守
 - ア データベースの運用管理
 - イ データベースの保守

のデータ処理の実習が中心であり、さらに、認定講習会の受講生にとって高校での授業のためにサーバを自分で立ち上げるのは敷居が非常に高いことが予想される。

松本大学のパソコン室のパソコンにはマイクロソフト社のACCESSがインストールされているが、商業科や工業科以外の高校では、ACCESSをインストールしたパソコン室を使える高校は多くはなく、一人一台端末の時代を受けてBYODとして生徒が持ち込むタブレットパソコンを使った形態での授業が現実的である可能性が高い。さらに、今回の講習会は新型コロナウイルス感染症に対する警戒から全面オンラインで行うことになり、大学のパソコン室の利用ができなくなったために、大学の授業と同じようにACCESSを利用するためには、本認定講習会参加者がオンラインで利用するパソコンに、有料アプリケーションをインストールが必要となることから、ACCESSの利用は難しいと判断した^{注1)}。

本認定講習会では、ブラウザで実行可能な環境であるGoogle Colaboratory（以下Colabと略記する）上でPythonのライブラリーとして利用可能な

DBMSであるsqlite3を使って実習を行うことにした。ColabはGoogle社が提供するPythonプログラミング環境²⁾、メモリーと実行時間の制限の下で無料で使うことができ、すでに高校の現場でもプログラミングの授業に使われている。SQLiteはもともとはC言語のSQLデータベースエンジンライブラリーで³⁾、パブリックドメインとなっているソースコードからPythonのライブラリー moduleとしてsqlite3がつくられている⁴⁾。

II. Colab上でsqlite3を使ったDBMS実習環境の構築

1. 3つのステージ

Colabでsqlite3を使ってデータベースの実習を行う際に特に注意した点は、作業には下記の3つのステージがあることを受講者に意識させる点である。

1. Google Driveのステージ
2. Pythonのステージ
3. sqlite3のステージ

1のGoogle Driveのステージはシステム上でファイル操作を行う段階で、実習をするにあたっての、テーブルデータの準備やアップロード、結果ファイルのダウンロードの段階である。認定講習会参加者は現職の高校教員であり、コロナ禍を乗り切った経験を持つ方たちなので、Googleアカウントの取得とオンラインクラウドとしてのGoogle Driveの利用には慣れていない様子が見られたが、階層構造などのファイル操作に習熟しているようには見えない受講生も見られた。

2のPythonのステージはPythonでmodule sqlite3を動かしたりデータを操作したりする段階で、言語としてのPythonを利用する段階である。本認定講習会の目的はSQLの学修であってPythonの修得ではないので、Pythonの文法に深入りすることのないように注意した。しかし、作業上はPythonのライブラリーのNumpyやPandasを利用すると便利であり、また使わざるを得ない場面もある。

3のsqlite3のステージが本認定講習会の主目的で、データベースを動かすSQL命令を学修する段階である。三連引用符⁵⁾で囲んだ文字列としてSQL

文を定義し、sqlite3のカーソルに代入してSQL命令を実行する。

2. Google Driveのステージ

関係データベースの実習のハイライトは、複数のテーブルを組み合わせたデータ操作にある。テーブルは正規化によって分割されて作られる場合もあるが、本認定講習会では学習時間の関係も考えて正規化の終わった複数のテーブルを予め配布することにした。受講者の実習に使うアカウントは受講者が準備したアカウントであり、すでにいろいろな形で利用している可能性を考慮し、本認定講習会共通の名前の講座用のフォルダーを各自で作成し、ファイルと作業を集中させるカレントフォルダーとした(図1)。

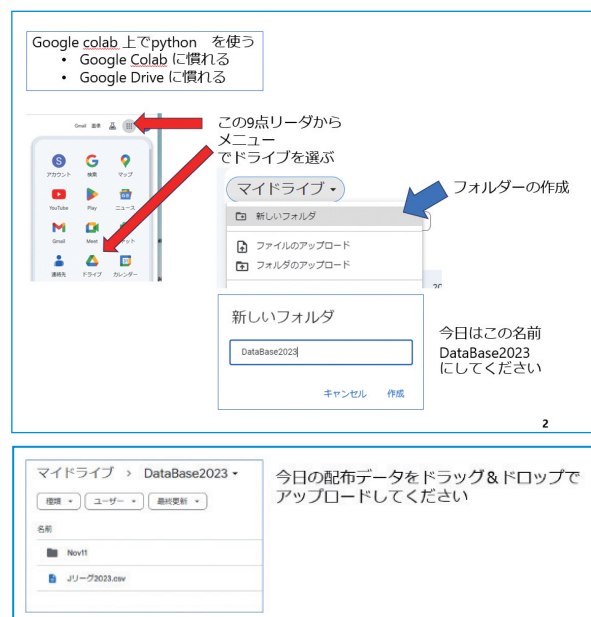


図1. Google Driveの解説スライド

3. Pythonのステージ

Pythonのステージはsqlite3 moduleを動かすプログラミング言語Pythonを使うステージである。言語としてのPythonの習熟が目的ではなく、また、受講生はプログラミング経験の浅い参加者が多いことから、Colab利用のための最低限の作業ルールに焦点化し、プログラミング言語を意識する必要のないように話を進めて行った。キー入力

```
# ドライブをマウントする ファイルを継続的に使いたければ、google drive に置く必要がある。
from google.colab import drive
drive.mount('/content/drive')

# グーグルドライブの確認
!ls /content/drive/MyDrive

!ls /content/drive/MyDrive/DataBase2023 # これから使うフォルダーの確認

# sqlite3 と csv、その他のライブラリーの準備
import sqlite3
import csv
import pandas as pd
import numpy as np
import math
import datetime # 日付を扱うライブラリーの読み込み
```

図2. Driveのマウントとファイルの確認

(Character user interface、以下CUI)での作業の利点を生かしつつ誤入力によるエラーのストレスを軽減するために、サンプルコードや既入力コードの複写&貼り付けを活用して実習を進めることにしたので、Colabで便利な、Ctrl+cで複写、Ctrl+vで貼付けというショートカットと、Shift+enterまたはCtrl+enterというショートカット実行命令の訓練を行った。

Colabのプロンプトでは、!に続けてコマンドを入力し実行することでUNIXの標準的な命令を実行することができる。ファイルやディレクトリー操作のUNIX命令を使えることは、トラブルシューティングに有効であった。また、Pythonでコメント文の書き方(#から行末まで)を覚え、メモを残しておくことは、後からの学修の振り返りには重要である。図2はDriveのマウントから、ライブラリーの読み込みまでのColabでのPythonコードである。途中「!ls フォルダー名」でフォルダー中のファイルリストを書かせアップロードしたファイルの存在とファイル名を確認している。

Google Driveは多国語のファイル名に対応しているため、Google Chromeなどのブラウザ上では実際

のファイル名が正しく表示されない場合がある。また、アップロードの際にファイルが誤って上書きされることを避けて、自動的に番号を付けて別名ファイルにする場合がある。Pythonのプログラム上からファイルを指定する際の正しいファイル名の確認や、アップロード再試行のためにファイルを消去する際には、このステージでの、ls、mv、rm、cdといったUNIXコマンドが非常に有効であった。本認定講習会でUNIXコマンドを教えた訳ではないが、トラブル解決のために、UNIXコマンドを使った対応の指示を出す場面が何度かあった。

マウントの終わったドライブからcsvファイルを読み込む作業が図3である。データベースに読み込む前の作業として、文字化けなく読むことができていることの確認のためにcsvファイルを表示させたり、読み込み作業の準備としてcsvファイルにフィールド名(header)があるかどうかの確認を行ったりしている。この後、headerと整合性的に変数型を指定してcsvファイルをPythonのlistに読み込んで行くことになる⁶⁾。この段階ではPythonでの変数型指定を利用しているが、sqlite3では使える変数型に制限があることには要注意である。

```

# CSV ファイルからデータを読み込む
csvfile = open('/content/drive/MyDrive/DataBase2023/Jリーグ 2023.csv', encoding='CP932')
# Shift-jis なのに CP932 で読み込む
csv_reader = csv.reader(csvfile) # 読み込み

# ただ書かせてみる
print(csv_reader) # 書かせてみると特殊なオブジェクトだという素性を書く

# ただ書かせてみるその2
for row in csv_reader:
    print(row)

# もう一度読み直す 読み直さないと header 命令でエラーが出る
csvfile = open('/content/drive/MyDrive/DataBase2023/Jリーグ 2023.csv', encoding='CP932')
# shift-jis なのに CP932 でうまくいく
csv_reader = csv.reader(csvfile) # 読み込み

header = next(csv_reader) # 先頭は field 名
header # header を書かせる
import numpy as np

```

図3. Driveのマウントとファイルの確認

4. sqlite3のステージ

csv ファイルのデータベースへの読込は pandas のデータフレームを経由して行う (csv -> pdDataFrame -> データベース) のが簡単である。図4でデータベースファイル名を指定してデータベースと接続しカーソルを定義している。これ以降

SQL 命令を定義して接続カーソルの実行関数に代入すれば SQL 命令を実行してテーブルデータの処理を行うことができる。csv を読み込んだ pandas のデータフレームは接続 (conn) を指定した pandas コマンド pdDataFramafile.to_sql でデータベースに読み込むことができる (図5)。

```

# まず、データベースと接続。なければDBを作成する。今日使うファイルは前回作ったNov-11.dbです。
dbpath = '/content/drive/MyDrive/DataBase2023/Nov11/Nov-11.db'
# と使うデータベース名を定義する テーブル名ローマ字版
conn = sqlite3.connect(dbpath) # 今使っている Google Colab とデータベースを接続
#
# 命令を受け取るカーソルを c と名付ける
c = conn.cursor()

```

図4. データベースとの接続とカーソルの定義

```
# pd.DataFrame の sql への読み込み
# pandas.DataFrame.to_sql で sql に読み込まれる
生徒マスタ.to_sql('Seito',conn,if_exists='append',index=None)
```

図5. pd.DataFrameFile 生徒マスタをSeitoテーブルとしてデータベースへ読み込む

Pythonでsql文を定義する際には、sql=""命令文""と三連引用符で囲って命令を書くことによって、複数行に改行して記述することが可能になり、長いSQL命令文を構造的に記述することができる。Google Colab入力プロンプトでは全角スペースがマークアップされるが、三連引用符中では全角スペースのマークアップが無いので、日本語を含めたSQL文の場合には、先頭の三連引用符を外して入力を確認すると全角スペースの誤入力を見つけやすくなる。sqlでは--以降がコメント文となる。

SQL命令の結果がテーブルで出てくる場合には、sqlからpd.DataFrameへの変換を行うと、pd.DataFrameの表示命令やpd.DataFrameからcsvへの変換書き出し命令を利用することが可能である。

テーブルの更新を行った場合には、commit()を

行うことによってデータベースファイルに結果が反映される。接続の解除はclose()命令であるが、close()命令を行わずにColabを閉じてても、とくに問題は起こらなかった。

Ⅲ. SQL命令の学習

1. sqlite3のSQL実装

前章で構築したColabとsqlite3でSQL命令を実行する環境でSQL文の学習を行った。SQLはもともとはStructured Query Languageの略称で問い合わせ言語と呼ばれることもある。SQLは表2のようにデータ定義言語、データ操作言語、データ制御言語からできている。SQLiteはクライアント・サー

表2 SQLの種類とSQLiteで実装

データ定義言語	Data Definition Language	
• CREATE TABLE	テーブルを作る	
• DROP TABLE	テーブルを削除する	
• ALTER TABLE	テーブルを変更する	△
• CREATE VIEW	ビュー表を作成 ビュー操作	△
データ操作言語	Data Manipulation Language	
• SELECT	レコードを検索する	
• INSERT	レコードを追加する	
• DELETE	レコードを削除する	
• UPDATE	レコードを更新する	
データ制御言語	Data Control Language	
• GRANT	アクセス権限を与える	×
• REVOKE	アクセス権限を取り消す	×
×	SQLiteは実装せず	
△	SQLiteは完全には実装せず	

パー型ではなくローカルなシステムで動く組み込み型のライブラリなので、アクセス権限を定義する命令は実装されていない。また、テーブル変更命令とビュー表はreadオンリーであり、消去や挿入には対応していない⁷⁾。

2. テーブル定義言語

テーブルの作成については、あらかじめエクセルなどでcsvファイルを作成しておき“テーブルデータ”としてデータベースに読み込む方法と、SQLのCREATE命令で表を作成し、そこにSQL命令でデータを読み込む方法がある。csv形式を利用する場合、Colabのsqlite3では、前章で行ったようにPythonのpdDataFrame形式を介してデータの取り込みを行うのが便利であり、取り急ぎデータベースの体験を目指した1日目の講習では、こちらの方法でテーブルの作成を行った。

CREATE命令で作った表にデータを読み込む際には、データをタプルごとに挿入することになるが、Colabの利用でPythonのステージがあるので、複数行の読み込みではSQL文の複数回実行にPythonの繰り返し命令を利用することができる。GUI操作で外部データの読み込みが可能なACCESSと比較すると、Python言語の学習による繰り返し命令の利用とわかりやすいが冗長なGUI操作の繰り返しとのトレードオフになる。sqlite3の実行命令cursor.execute(sql)をPythonでの繰り返しコードに入れる方法と、複数行読み込み命令cursor.execute(sql,data)利用の2通りが可能である。

本認定講習会では、2日目は関係データベースらしく複数のテーブルを使った課題を扱うことにし、2日目の初めにCREATE命令を使った表の作成、SQLを使ったデータの読み込み、DERETE命令を使った表の削除の演習を行い、1日目同様のcsvデータの読み込みと合わせて複数の手段を紹介した。細かい注意点であるが、sqlite3ではテーブル名はASCIIコード対応英数文字(いわゆる半角直接入力)で入れておかないと動作が不安定であったが、これはSQLの仕様であった⁸⁾。

3. データ操作言語

SQLの演習では、データ操作言語で特にSELECT文が主となる。1日目午後の講習で一つの表に対する選択、射影、集約関数の演習を行った。2日目の講習のハイライトは複数の表を組合わせたデータ処理を行うSQL文の作成で、図6の5つのテーブルから図7の表を作る作業を行った。

テーマは学校食堂の献立人気調査アンケートの集計で、sqlite3のテーブル名は半角アスキー文字なので、正規化の終わった表をそれぞれ、Seito、Seibetu、Kondate、Ruiseki、Ankeitoという名前のテーブルでデータベースに取り込んだ。Ankeitoテーブルは途中の集計作業の演習に使い、アンケート結果テーブルの内容を累積アンケートテーブルに集約した。最後に指定した日付の結果を抽出し、献立CO、献立名と性別ごとに生徒COを集計(カウント)して女子比率(女子回答/全回答)を求め女子比率の降順に並べた表(図7)を作ることにした。クロス集計に簡単な計算が入っている。

生徒マスタ		累積アンケートテーブル		
生徒CO	整数	実施日	時刻	
生徒名	テキスト	生徒CO	整数	
性別CO	整数	献立CO	整数	
クラスCO	整数			
性別マスタ		アンケート結果テーブル		
性別CO	整数	実施日	時刻	
性別	テキスト	生徒CO	整数	
		献立CO	整数	
献立マスタ				
献立CO	整数			
献立名	テキスト			

図6. 演習に使うテーブル

献立CO	献立名	男子	女子	女子比率
12	ピーマン肉詰め	15	26	0.634146
13	さばフライ	15	16	0.516129
11	ハンバーグ	17	10	0.370370

図7. 6月10日のアンケート集計結果の表

```

sql=""" with
tmp_Danshi as
(SELECT count(ruiseki.生徒 CO) as 男子 , ruiseki. 献立 CO as 献立 CO
from Ruiseki, Seito, Seibetsu
where (実施日 = datetime('2011-06-10')
and Ruiseki.生徒 CO = Seito.生徒 CO
and Seito.性別 CO = Seibetsu.性別 CO
and seibetsu.性別 = '男子')
group by ruiseki. 献立 CO) , -- order by を消してコンマを忘れずに

tmp_Joshi as      -- 2つ目は with いらぬ
(SELECT count(ruiseki.生徒 CO) as 女子 , ruiseki. 献立 CO as 献立 CO
from Ruiseki, Seito, Seibetsu
where (実施日 = datetime('2011-06-10')
and Ruiseki.生徒 CO = Seito.生徒 CO
and Seito.性別 CO = Seibetsu.性別 CO
and seibetsu.性別 = '女子')
group by ruiseki. 献立 CO) -- order by を消す

SELECT Kondate. 献立 CO, 献立名, 男子, 女子,
cast(女子 as real)/cast(女子 + 男子 as real) as 女子比率
from tmp_Danshi, tmp_Joshi, Kondate
where (tmp_Danshi. 献立 CO = Kondate. 献立 CO
and tmp_Joshi. 献立 CO = Kondate. 献立 CO)
order by 女子比率 desc ; """ # SQL 命令

kekka = pd.read_sql_query (sql, conn)
# SQL 命令の実行結果を pandas DataFrame の kekka に代入
kekka #kekka の表示
    
```

図8. 図7の最終結果を表示するSQL命令

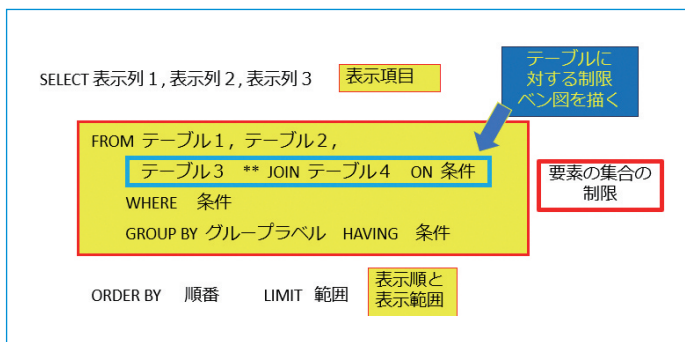


図9. SQL文の基本構造

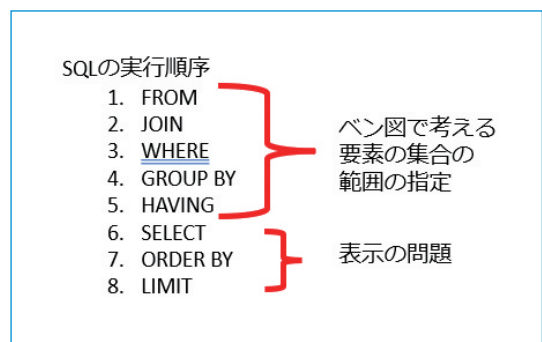


図10. SQL文の実行順序


```

# 献立ごとに男子の人数を数える
sql= """ SELECT count(ruiseki.生徒CO) as 男子 , ruiseki.献立CO as 献立CO
        from Ruiseki, Seito, Seibetsu
        where (実施日 = datetime('2011-06-10')
              and Ruiseki.生徒CO = Seito.生徒CO
              and Seito.性別CO = Seibetsu.性別CO
              and seibetsu.性別 = '男子')
        group by ruiseki.献立CO order by 男子 desc; """ #SQL 命令
kekka = pd.read_sql_query(sql, conn)
# SQL 命令の実行結果をpandas DataFrameのkekka に代入
kekka # kekkaの表示

```

図11. 図8の一時テーブル定義部分を実行するSQL

図8が最終集計結果を与えるSQL命令で、これだけ見ると長い複雑なSQL文に見えるが、本講習会では構造を分解し、手順を踏むことによってこのSQLの理解を目指した。

SQL命令の基本構造は図9のようにになっている。また、文献^{8,9)}によれば、SQL文の実行順序は図10の順になっている。すなわち、SQL命令は、FROMやWHERE、JOINといったテーブルの組み合わせやテーブルからのデータの抽出を先に行い、その後GROUP BYとHAVINGの集約を行い、最後にORDERやLIMITといった表示の指示を実行する仕様になっている。学習においても、このSQLのデータ処理の順番を意識することがポイントとなる。

図8のSQL文は大きく3つの部分(A1、A2、B)からできている。A1とA2は副問い合わせによって一次テーブルtmp-Danshiとtmp-Joshiを作っている部分である。A1とA2は全く同じ構造をしており、A1のtmp-Danshiにおける男子に対する処理を女子に変えたものがA2のtmp-Joshiである。

Bは一時テーブルであるtmp-Danshiとtmp-Joshiのデータに対して、行について献立名との関連付けを行い、列データについて女子比率を求める計算を行って表示データを作り、最後に順番を指定して表示している。図8の先頭でwithで利用している一時テーブルは、SELECT文で作られているので、図

11のようにA1の部分だけ取り出してSQL文として実行することが可能である(図12)。図8のA1では、図11のsqlから末尾のorder by命令が削除されている。

	男子	献立CO
0	17	11
1	15	13
2	15	12

図12. 図11のSQL命令の出力

授業の手順としては、データベースへの表の読み込みの後、

1. 表の表示命令
2. Whereを使った行の選択
3. 集約関数countの練習
4. キーを利用した関連付け

を順に練習して来れば、その次のステップとして図11、図12のDanshi及びその女子版のJoshiの表の作成が可能となる。図11の“男子”の結果から“女子”の結果を作るところをSQL文の「複製+貼り付け+書き換え」で処理することで、CUI操作のメリットを強調することも期待できる。図11の副問合せ

ができてしまえば、図8のSQL文は献立名との関連付けと女子比率の計算が加わっただけである。

以上の手順で授業を行い、2コマ（3時間）程度の時間で、最終結果のSQLの作成までの実技を終了させた。1日目の午後の授業で、単一の表に対する選択、射影、集約関数の練習を行っているので、2日目はもっぱら、外部キーによる関連付けの演習が中心となった。

SQLによる外部キーによるテーブル間での行の関連付けは、ACCESSのGUI操作と比べると直感的なイメージに劣る傾向があるが、今回はWHEREテーブル1.データA=テーブル2.データBという形での結合だったので、作業内容の理解は容易であったようで、学習に問題のある受講生がいるようには見えなかった。むしろ、「複製+貼り付け+書き換え」で同等の別の条件命令を作れるCUI操作の便利さに共感してもらえたように見えた。

IV. まとめ

本稿では、オンラインでのデータベース授業の実践例を報告した。データベースの授業であっても理論の講義については対面授業と比べて大きな違いはないと考えられるが、実技の実習に関してはシステムの準備から始まってオンライン独特の難しさがあり、本稿では実技実習部分のみを報告した。

実践は昨秋に開講した教員免許証認定講習会で、参加者は情報以外の教員免許保持者31人で主分野は商業6名、理科4名、数学3名、工業3名、社会科3名、その他（農業や福祉、美術など）12名であった。プログラミング経験に関するアンケートでの自己申告では、プログラムを日常的に組んでいる方0人、練習問題以外のプログラムを組んだことのある方6人、練習問題はやったことがある方4人、授業に参加した程度の経験の方11人、プログラミング経験があるとはいえない方10人であった。認定講習会本番では、全員が演習の最後（図7）まで到達し、修了課題レポートでは23名が類似のアンケート集計のSQLを完答した。

今回の課題（図11）ではクロス集計を実行するために一次テーブルを利用したが、一時テーブルを使わずにCASE文を使った表現も可能である。また、テーブルの集合演算に関してはOUTER JOIN

を扱っておらず、集約関数に対するHAVINGも扱っていない。1単位2日間という時間制限があるために、取捨選択をせざるを得なかった。これらのSQL命令については、データベース運用の話題（表1の（4）や（1））と合わせて、次回認定講習会の機会があれば扱う予定である。

高校の教科「情報」などによるプログラミング教育の普及に伴い、無料で利用できるPython環境であるColabが広く使われるようになってきており、併せてsqlite3によるデータベースの利用も広がってきている。ビッグデータの統計的取り扱いツールとしてのSQLも情報教育での興味深いテーマである¹⁰。本講座では、Google Driveに作業ファイルを置きColab上でPythonライブラリーであるsqlite3を用いてデータベースの実習を行った。今回の実践の教訓として、学習者の作業のステージを

1. Google Driveのステージ
2. Pythonのステージ
3. sqlite3のステージ

の3つの段階に分け、現在の作業のステージを学習者に意識させることが効果的であることが示唆された。

謝辞

2024年教員免許証認定講習会の機会を与えてくださった長野県教育委員会と講習会に参加してくださった教員の皆様に感謝いたします。

注

^{注1} コロナ禍で遠隔授業となった際には、松本大学では学生各人のパソコンにサイト契約のoffice365をインストールさせ、ACCESSを使ってオンラインでデータベース論の授業を行った。

文献

- 1) 高等学校学習指導要領(平成30年告示), https://www.mext.go.jp/content/20230120-mxt_kyoiku02-100002604_03.pdf (2024.5.30 閲覧)
- 2) Google Colaboratory, <https://colab.research.google.com/?hl=ja> (2024.5.30 閲覧)
- 3) SQLite, <https://www.sqlite.org/index.html> (2024.5.30 閲覧)
- 4) sqlite3, <https://docs.python.org/ja/3/library/sqlite3.html> (2024.5.30 閲覧)
- 5) 三連引用符, <https://www.javadrive.jp/python/string/index3.html#section1> (2024.5.30 閲覧)
- 6) 西潤史朗, 「データサイエンス時代のSQL基本編」『日経ソフトウェア』2021年5月号, pp.044-053 (2021).
- 7) SQLiteに実装されていないSQL, <https://sqlite.org/omitted.htm> (2024.5.30 閲覧)
- 8) ミック, 「達人に学ぶSQL徹底指南書」『翔泳社』, (2018).
- 9) 高橋光, データ分析のためのSQL勉強会～初級編～, <https://github.com/hikarut/SQL-Hands-on/releases/tag/v1.3> (2024.5.30 閲覧)
- 10) 例えば, 本多賢, 「Pythonで学ぶSQL/データベース入門」『日経ソフトウェア』2022年11月号, pp.082-091 (2021).