

アプリケーションソフトからの利用のされ方の理解を中心としたデータベースの授業実践

渋沢良太

第一工科大学 工学部 情報・AI・データサイエンス学科 (〒890-0052 鹿児島県鹿児島市上之園町 20-2)

A Practical Database Course Focused on Understanding Database Usage in Application Software

Ryota Shibusawa

20-2, Uenosonochō, Kagoshima City, Kagoshima, 899-4332 Japan

Abstract: Many excellent textbooks are available for learning relational databases at universities. However, most of them focus primarily on the relational database itself and deductively explain its theoretical and technical components. The theoretical foundation of relational databases is based on mathematical set theory, which enables the subject to be described systematically. However, many learners of databases do not have a sufficient background in fundamental mathematical theories such as set theory. As a result, they often require a considerable amount of time to understand these concepts, which can ultimately prevent them from developing a deeper understanding of databases. Based on this problem awareness, this study implemented a database course for second-year university students that focused on how databases are used by application software, with the aim of deepening students' understanding of databases through practical usage contexts.

Key words: データベース教育, リレーショナル・データベース, ローコードツール

1. はじめに

大学においてリレーショナルデータベースの学習に使われる良書は多く存在する[1],[2],[3]が, その多くはリレーショナルデータベースそのものに焦点をあてて, その理論, 技術の要素を演繹的に示していくものである。リレーショナルデータベースの理論基盤[4]は数学の集合論であり, それに基づいて体系的に記述することができる。しかし, データベースの学習者には, 集合論等の数学の基礎的な理論が十分でない者も多く, その理解には多くの時間を要してしまい, 結果としてデータベースについての理解を深められない場合が多い。

ある対象を理解しようとする時, 理解の方法として大きく2つの方法がある。一つはその対象の

構成要素を示して, その一つ一つの構成要素を理解していく方法である。もう一方の方法は, その対象と他の対象との関係性を理解する方法, すなわちその対象が他の対象とどのようにインタラクションしているのかを理解する方法である。伝統的なデータベースの教育では, 集合論に基づいて前者のアプローチに基づいてデータベースを構成する要素を示している。近年ではスキーマを圏, インスタンスを関手として, 圏論の記法によってリレーショナルデータベースを記述する方法も提案されている[4]が, データベースについてその内部の関係性を示すことに焦点が当てられている。一方, 本研究ではデータベースとその他の対象との関係性を示すアプローチによってデータベースの理解の支援を行う。

本研究では上記の問題意識のもと、学習者にとって馴染みのある Web アプリケーションからデータベースがどのように使用されるかに焦点を当てて、その理解を通してデータベースの理解を深める授業を、本学の工学部情報・AI・データサイエンス学科の2年生を対象に、2023年度のデータベースⅠ、データベースⅡで実践し、アンケート調査によってその有効性を調査した。

2. 授業の構成

2023年度において、データベースⅠは前期、データベースⅡは後期に開講され、共に情報・AI・データサイエンス学科の2年次の授業として開講された。データベースⅠを履修せずにデータベースⅡを履修した学生は2名のみで、ほとんどの学生はデータベースⅠとデータベースⅡの両方を受講した。データベースⅠ、データベースⅡともに1)データベースの基礎、2)データベースを利用するアプリケーション開発、3)学習者による開発の実践の3つの内容で構成した。どちらの授業においても、最もリレーショナル・データベースを中心とした内容にしておき、NoSQL やブロックチェーンといった内容は少しのみしか授業内容に含めていない。2)データベースを利用するアプリケーション開発では、2022年度以前は、PostgreSQL や MariaDB と Apache, nginx を使用し、Go や PHP といったプログラミング言語を使って実践していたが、プログラミングの言語仕様の学習負担が大きくデータベースの理解に注力できないため、2023年度以降はローコードツールを使用している。

データベースⅠにおいては、Google の AppSheet [6]を使用した。AppSheet は、Google スプレッドシート [7]のファイルをデータベース、スプレッドシート内の各シートをテーブルとしてアプリを構成することができる。リレーショナル・データベースでは、テーブル(リレーション)をデータ保存の基本単位にしていくが、その視覚的表現としてスプレッドシートはわかりやすく、リレーショナル・データベースにデータが保存され

ていく様子をスプレッドシートで確認できるため、データベースの理解にも役立つツールである。AppSheet によるデータベースを使ったアプリの開発手順は次の通りである。まずスプレッドシートを作成し、スプレッドシートのファイル名を設定する。次にアプリに必要な情報を保存する各シートの名前を設定し、各シートをテーブルと見立てて一行目に列名を記述する。その後、作成したスプレッドシートを AppSheet で読み込ませると、スプレッドシートに設定した列名を反映したテーブルが作成され、そのテーブルの中身を表示するモバイルアプリの画面の雛形が作成される。テーブルの型はスプレッドシートでは設定できないため、AppSheet のデータベースの設定で、テーブルの各列の型やキー、デフォルト値等を設定することができる。完全に自動でアプリを作成するのではなく、テーブルの型等をユーザが設定するところも、データベースの学習にとっては重要な要素であると考えられる。

ソフトウェアの多くにとって、視覚情報として見ている情報の見た目の構造と、元のデータ構造は乖離していることが多く、これも学習者にとってのわかりにくさの要因の一つとなっていると考えられる。例えば、Web ブラウザで表示されている Web ページの見た目と、HTML 文書の構造そのものは完全に対応しておらず、CSS 文書によって見た目についての多くは実装されている。また、Word もアウトラインモードで確認できるデータ構造と、印刷レイアウトで表示されている見た目が別であり、学習者は意識しないと見えている情報のデータ構造を把握できていない状態である。Google AppSheet では、リレーショナル・データベースのテーブルのデータ構造を、Google スプレッドシートを用いて表形式で表現することにより、見た目とデータ構造を一致させてデータ構造の理解を促すことができる。

データベースⅡのアプリ開発の実践では、Notion [8]および Notion API [9]を使用した。Notion は様々な情報のノートツールとして普及しており、使用経験のある学生も多いツールであ

り、慣れたツールでデータベースの操作を学ぶことができ、データベースの理解に注力できる利点がある。Notion では、リレーショナルデータベースのテーブルのコンテンツを作成して表形式の見た目でページ内に表示させ、テーブル表示がされていない別のページからテーブルの内容を参照するといったことが可能である。また、Notion API を使うことで、Notion のテーブルを外部のアプリケーションから参照させることができる。

データベース I、データベース II とともに、授業で一方向的に話すだけでなく、学習者が成果物を作成することを通して理解が深まるように、授業の最後には学習者によるデータベースを使ったアプリケーションの制作課題を行わせた。

3. 学生による課題制作の実践

データベース I の授業の課題において学生が制作した成果物の代表的なものを表 1 に、データベース II において学生が制作した成果物の代表例を表 2 に示す。どちらの課題においても、どのようなアプリを開発すれば意義があるのかを学生に考えさせるためにも、開発する内容の具体的なテーマは教員が与えず、学生自らに考えさせ、開発を行わせた。

どちらの授業の成果物も、アルバイト先の業務に役立つアプリや、普段の自分自身の生活を便利にするアプリなど、実用的なアプリを題材にしているものが多かった。中には、家業の業務を支援するアプリを開発する者も見られた。

ほとんどの学生は、データベース I と II で異なるテーマの成果物を作成していた。データベース I と II で類似したテーマで開発する者も見られたが、その場合であっても実現方法が異なるため、異なる方法で同じアプリを開発することでよりアプリの本質が理解できる利点もあると考えられる。

コーディングエージェントを活用してプログラミングで成果物を作る方法もあるが、その場合ソースコードの中身を理解せずに開発してしまうことになり、データベースの学習を目的とする

場合には適さない。また、コーディングエージェントを使わずにプログラミングで開発する場合は、コーディングについての認知負荷が大きく、学生がデータベースの理解に注力しづらい。本研究では、ローコードツールを使うことによって、学生が自身の考えを実現するためにツールをうまく制御して、目的のアプリを開発することに成功していた。

表 1 データベース I の成果物の概要

No.	成果物の内容
1	商品の在庫管理システム
2	商品カタログ
3	タスク管理アプリ
4	家計簿アプリ
5	家業の会社の代車管理アプリ
6	最高気温記録
7	アルバイト先の店舗の清掃チェックアプリ
8	買い物メモ付きカレンダー
9	勤怠管理アプリ
10	大雑把な命令にはしっかり聞いてくれることが多かった。
11	売上、原価、経常利益の管理アプリ
12	読書管理アプリ
13	欲しいもののリスト
14	筋トレ管理アプリ
15	旅行リスト化アプリ
16	取引先企業管理アプリ
17	体調管理アプリ
18	漫画の巻数管理アプリ
19	備品管理アプリ
20	生活管理アプリ

表 2 データベースⅡの成果物の概要

No.	成果物の内容
1	ToDo 管理アプリ
2	アニメ・映画鑑賞記録アプリ
3	部活動の練習メニュー管理
4	読書記録アプリ
5	学習管理アプリ
6	ゲームアシスタント
7	資産管理アプリ
8	アルバイト先のシフト管理アプリ
9	習慣トラッカー
10	買い物リスト
11	家計簿アプリ
12	出演舞台記録
13	ダイエット法紹介アプリ
14	ガントチャート生成アプリ
15	商品価格の変動管理アプリ
16	食品のカロリー表
17	アニメやゲームのキャラクターの情報管理アプリ
18	タスク管理アプリ
19	アルバイト先の勤怠管理アプリ
20	Arknights アシスタント

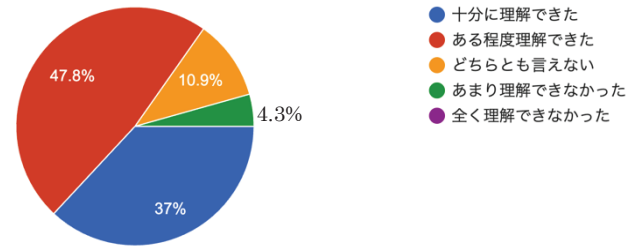


図 2 「DBMS が、どのような機能を持つか理解できましたか？」(I)への回答

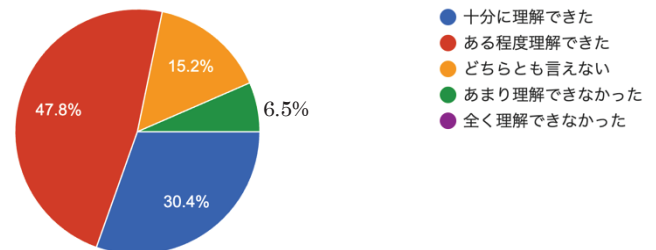


図 3 「SQL の 4 大構文(select, insert, update, delete)を理解できましたか？」への回答

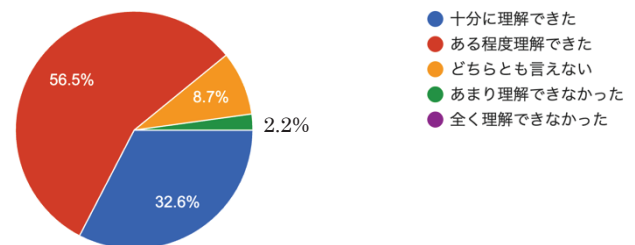


図 4 「データベースを使ったアプリケーションシステムの開発方法を理解できましたか？」(I)への回答

4. データベースⅠの授業評価

データベースⅠの授業(Google AppSheet を使用)の終了時に、学生に行ったアンケートの結果を図 1 から図 5 に示す。データベースⅠのアンケート回答者数は 46 名であった。

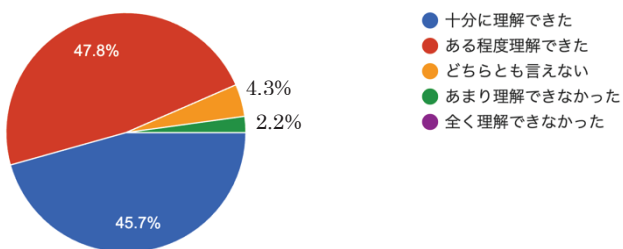


図 1 「データベースが、アプリケーションからどのように使われるか理解できましたか？」(I)への回答

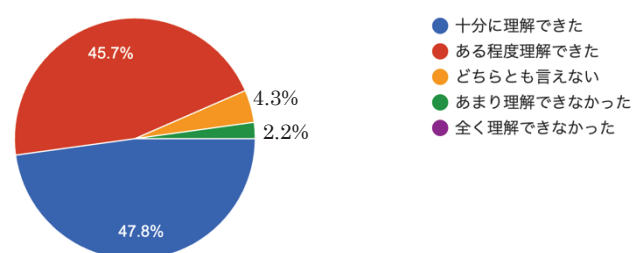


図 5 「データベースの正規化の必要性を理解できましたか？」への回答

図 1 は、本授業でまず目的としている学習項目であったが、ほとんどの学生は一定以上理解ができていた。図 2 から図 5 は、伝統的なデータベース教育での教示方法とも共通する学習項目への回答結果である。本授業ではプログラミングによるデータベース操作はほとんど行わなかったため、プログラミングによるデータベースの操作インタフェースとして機能とする SQL の理解については、他の回答項目と比較してやや結果が悪かったが、全体的に良い理解状況が確認できた。特に図 5 のデータベースの正規化の必要性については、アプリケーション制作の課題で、正規化せずに作成した場合の問題点を実際に経験すること等を通して、正規化しない場合に具体的にどのような問題が起きてしまうのかを考え、理解が促されたと考えられる。

本アンケート調査の結果から、アプリケーションからどのようにデータベースが利用されるのかを中心に学ぶことによって、DBMS の機能、アプリケーション開発方法、データベースの正規化についての理解が深まることが確認できた。

5. データベース II の授業評価

データベース II の授業(Notion を使用)の終了時に、学生に行ったアンケートの結果を図 6 から図 10 に示す。データベース II のアンケート回答者数は、38 名であった。質問の内容はデータベース I と基本的には同じであるが、学習項目がやや異なるため、図 8、図 10 はデータベース I とは異なる質問内容に対する回答となっている。

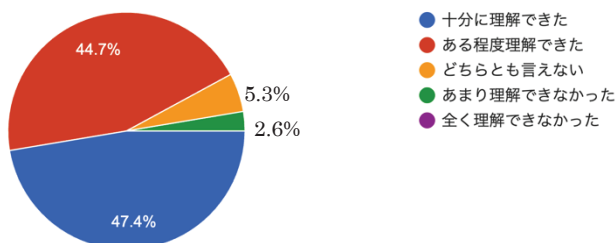


図 6 「データベースが、アプリケーションからどのように使われるか理解できましたか？」(II) への回答

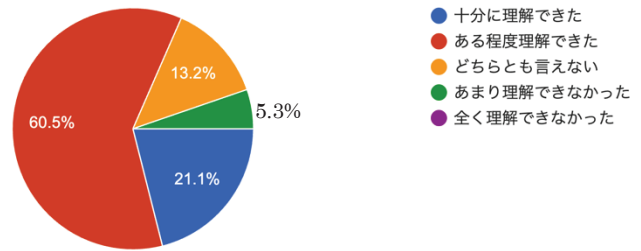


図 7 「DBMS が、どのような機能を持つか理解できましたか？」(II) への回答

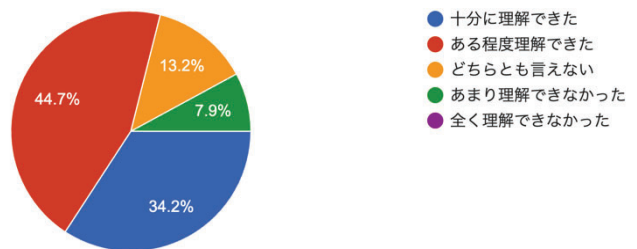


図 8 「複数のテーブルを連携させる方法を理解できましたか？」 への回答

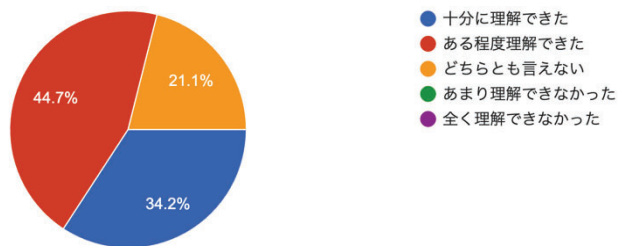


図 9 「データベースを使ったアプリケーションシステムの開発方法を理解できましたか？」(II) への回答

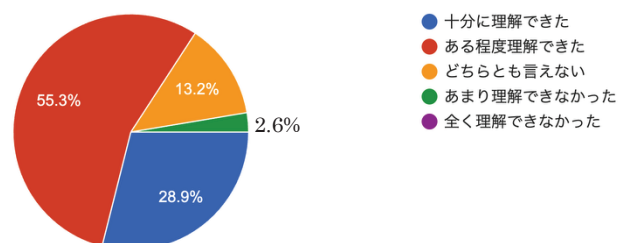


図 10 「業務効率化のためのデータベースの活用方法について理解できましたか？」 への回答

データベースⅡについても、データベースⅠと同様に、一定の水準での理解が確認できた。特に、図9に示した「データベースを使ったアプリケーションシステムの開発を理解できましたか?」という質問にたいしては、「あまり理解できなかった」と回答する学生はいなくなっており、より多くの学生に分かりやすい内容であったと考えられる。

5. おわりに

本研究では、ローコードツールを使って、データベースが Web アプリケーションからどのように利用されるのかを中心に学習することによって、データベースの理解を支援する授業を実践し、アンケート調査によって学生の理解度を調査した。アンケートの結果から、本研究による提案方法によって、学生はプログラミングに注力が不要になる分、データベースの理解に注力を当てられることにより、データベースの理解が一定水準まで進められることが分かった。また、データベースの技術要素を演繹的に示すアプローチではなく、データベースに求められる機能を、データベースを利用する側の立場から考えることで、データベースの理解が促されることが確認できた。

近年では、コーディングエージェントを使って高度なアプリを学習者がプログラミングせずに開発することも可能になったが、アプリケーションシステムの内部の仕組みを理解せずに開発することになりやすい。本研究では、ローコードツールを活用することにより、学習者自身が理解しながら、自身が開発したいことを制御して開発できるようにすることには、データベースの学習者用途として重要であると考えられる。

今後、本研究の結果を踏まえ、データベース教育用の独自のローコードツールを設計、開発して授業の実践、評価を繰り返し、より良いデータベースの教育方法を明らかにしたい。また、リレーショナル・データベース以外に、NoSQL やブロックチェーンについての分かりやすい教材、ツ

ールを開発し、これらの教育方法について研究したい。

参考文献

- [1] 増永良文, “リレーショナルデータベース入門: データモデル・SQL・管理システム・NoSQL”, サイエンス社, 2017 年.
- [2] 増永良文, “データベース入門”, サイエンス社, 2021 年.
- [3] 北川博之, “データベースシステム”, オーム社, 2020 年.
- [4] E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, 1970.
- [5] D.I. Spivak, “Functorial data migration,” Information and Computation 217 (2012) pp. 31-51.
- [6] Google AppSheet, <https://about.appsheet.com/home/>, (2026 年 5 月 7 日参照).
- [7] Google Spreadsheet, <https://docs.google.com/spreadsheets/>, (2026 年 5 月 7 日参照)
- [8] Notion, <https://www.notion.com/>, (2026 年 5 月 7 日参照).
- [9] Notion API, <https://developers.notion.com/guides/get-started/overview>, (2026 年 5 月 7 日参照).