

データベースの演習授業を支援する Web アプリケーションの開発と 実授業への適用

2019/03/15
大阪電気通信大学
國本倫平・久松潤之

1

データベース演習

- データベースの概念や、操作方法を学習
 - 正規化
 - SQL言語
 - O/R マッパー
- 授業の形式
 - 前半: 教員が文法、用法を解説
 - 後半: 学生が実際にデータベースを操作し、学習
 - 教員の用意した課題に取り組む
 - これまでは課題の解答を紙やe-learningシステムで回収

2

授業の問題点: 教員の観点

- 学習用データベースを学生ごとに用意する必要がある
 - SQLサーバーを用意する場合
 - アカウントを作成、配布、データベースのコピー
 - 接続方法の解説
 - SQLiteファイルで配布する場合
 - 操作ソフトのインストールを解説
- 環境の導入に、授業時間を消費
- 教員が学生の学習状況を把握しにくい
 - 何らかの方法で解答を回収する必要がある
 - e-learningシステム
 - 解答用紙

3

授業の問題点: 学生の観点

- あらかじめ授業で用いる環境の導入が必要
 - SQL エディタ、O/R マッパーの実行環境
 - PCの操作が苦手な学生の場合、導入作業がハードルに
 - 環境変数の設定
 - ターミナル操作
- 提出した課題の正誤がすぐわからない
 - e-learningシステムであれば、教員の採点まで
 - 解答用紙の場合、次回の講義まで

4

研究目的

- データベースの演習授業を支援
Webアプリケーションの提案
 - 演習用データベース、演習課題の管理
 - 採点・集計を自動化
- 演習課題に取り組むまでの障害を削減
- 実際の講義に導入し、評価

5

データベース演習支援システム

- データベース、課題をWebアプリケーションで管理
 - データベースを各学生のアカウントへ一括して配布
 - 課題進捗状況の確認
- データベースを操作するWebインターフェース
 - SQL文、O/R マッパーの実行が可能
 - 使い慣れたブラウザが利用可能
 - ソフトウェアの導入が不要
- 採点・集計の自動化

6

データベース演習支援システムを利用した演習授業

1. 教員が、データベース・課題を提案システムに登録
2. 学生は、各課題画面でSQL文やO/Rマッパーを用いたプログラムを記述、実行結果を確認
 - 実行結果をもとに記述内容を修正し実行を繰り返す
3. 学生は、課題内容を満たす実行結果を得られる解答を作成し提出
 - 自動採点对応の課題の場合、提出完了時に採点結果を確認可能
4. 教員は学生の解答状況・提出内容を確認し、提出率や正答率が低い課題を、重点的に授業

7

7

データベース演習支援システム：アカウント管理

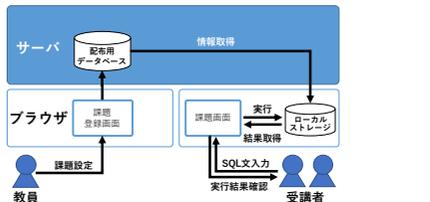
- Googleアカウントによりログイン可能
 - パスワードの配布が不要
 - 事前にメールアドレスが登録されたアカウントのみ利用可能
- 学生用アカウント
 - 各課題の演習の実行、解答の提出が可能
- 教員用アカウント
 - 演習課題の登録
 - 提出状況、解答状況の確認
 - アカウントの新規追加

8

8

データベース演習支援システム：SQL文を実行する課題

- 課題内容に沿った結果が得られる SQL 文を求める課題
 - 設定されたデータベースに対し、SQL文を何度でも実行可能
 - 条件を満たす出力が得られた SQL 文を提出



9

9

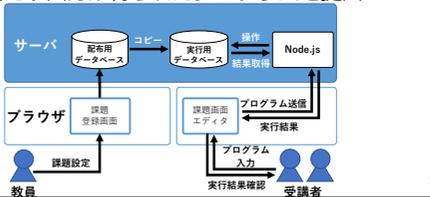
SQL文を実行する課題：操作画面

CUSTOMERID	CUSTOMERCODE	CUSTOMERNAME	ADDRESS	CUSTOMERCLASS
1	2001	タマ	江戸川区 下小岩	2
2	2002	ハナ	江戸川区 北小岩	2
3	2003	ミケ	葛林市 緑町	2

10

データベース演習支援システム：O/Rマッパーを実行する課題

- 課題内容に沿った結果が得られる O/R マッパーを利用したプログラムを記述する課題
 - Node.js で動作するO/R マッパー、Sequelize を利用可能
 - サーバー側に実行環境が存在するため、環境導入が不要
- 条件を満たす出力が得られたプログラムを提出



11

11

O/Rマッパーを実行する課題：操作画面

```

1: const Sequelize = require('sequelize');
2: const sequelize = new Sequelize('mysql://root:root@localhost:3306/testdb');
3: sequelize.authenticate().then(() => {
4:   console.log('Database connected successfully');
5: }).catch(err => {
6:   console.log('Error connecting to database:', err);
7: });
8:
9: const Product = sequelize.define('products', {
10:   name: { type: Sequelize.STRING, allowNull: false },
11:   price: { type: Sequelize.INTEGER, allowNull: false },
12:   category: { type: Sequelize.STRING, allowNull: true },
13:   createdAt: { type: Sequelize.DATE, allowNull: true },
14:   updatedAt: { type: Sequelize.DATE, allowNull: true },
15: });
16: sequelize.sync({ force: true });
17:
18: // Fetch all products
19: Product.findAll().then(products => {
20:   console.log('Products:');
21:   products.forEach(product => {
22:     console.log(`Product ID: ${product.id}, Name: ${product.name}, Price: ${product.price}`);
23:   });
24: });
25:
26: // Close the connection
27: sequelize.close();
    
```

Result

```

[Encoding: (default): SELECT PRODUCTNAME FROM 'products' AS 'products';
]
[ PRODUCTNAME: 'まぐろ' ]
[ PRODUCTNAME: '金魚' ]
[ PRODUCTNAME: 'さけ' ]
    
```

12

データベース演習支援システム: 課題の提出

- SQL 文を提出する課題では、即座に正答が確認可能
 - 模範解答と送信された解答の完全一致
 - 模範解答の実行結果と、送信された解答の実行結果を比較
- O/R マッパーを利用したプログラムを提出する課題は、自動採点に非対応なので、提出内容の保存のみ実行



13

アンケート調査による検証

- データベース演習支援システムを実際の授業に導入
 - 大阪電気通信大学の授業「データベース演習」
 - 授業の受講生 76名 平均出席人数 64.5人
- 全15回の授業中、第8回から第14回の計7回が演習
 - 第8回から第13回は、SQL 文を実行する演習
 - 第14回のみ、O/R マッパーを利用する演習
- 授業最終日に受講生へのアンケート調査を実施
 - 記名式、複数回答可の全7項目からなるアンケート
 - 出席者 64名中、61名が解答

14

アンケート調査:学習を進める上で 最初にハードルを感じた箇所

- 演習支援システム自体が学習のハードルになると考えた学生は少ない
- 課題に取り掛かるまでの障害を減らすという目的は達成できている

学習を進める上で最初にハードルを感じた箇所	
システムのログイン方法など、利用方法	8
SQL 文を直接利用する課題	23
ORM を利用する課題	39
その他	2
特に無し	8

15

アンケート調査: 学習時に問題となった箇所

- 学内ネットワーク内のみで利用可能という点に問題を感じた学生が多い
- 外部ネットワークに公開することで解決可能

学習時に問題となった箇所	
アクセスできる場所が大学内に限られる点	43
課題内容の把握	13
テーブルの内容の把握	13
チャットによる質問機能	6
デザイン	6
その他	4
特に無し	9

16

アンケート調査:SQL を入力する 課題で問題となった箇所

- 自動採点機能に問題がある事が分かる
- SQL 文に別名が設定されていると不正解になるなど、採点が厳密すぎたことが原因
 - 採点範囲の調整や、問題をより厳密にするなどで対処可能

SQL を入力する課題で問題となった箇所	
SQL 文の記述	21
SQL 文を直接利用する課題	14
採点や提出	36
その他	3
特に無し	4

17

アンケート調査:OR マッパーを実行 する課題で問題となった箇所

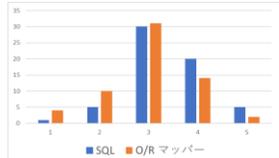
- JavaScript の扱いに問題がある事が分かる
- 初めて JavaScript を利用する学生が多かったことが原因
- 他の講義で利用している言語に変更する、解説時間を増やすことで対応できる

ORM を実行する課題で問題となった箇所	
ORM の概念に対する理解	27
JavaScript の文法	25
Sequelize の利用方法	15
エラー箇所の確認方法	30
プログラムの実行・提出	16
その他	1
特に無し	6

18

アンケート調査:学習しやすさ

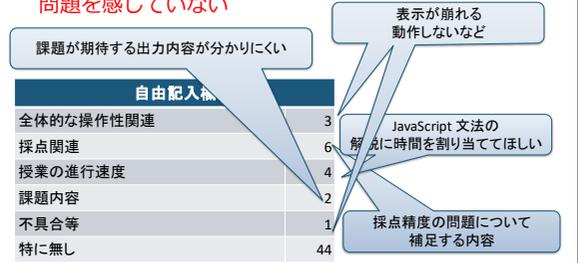
- 各形式の課題の学習しやすさを5段階で評価する項目
- 殆どの学生が、3以上を解答しているため、システムに大きな問題点を感じた学生は少ない
- O/R マッパーを利用する課題は、SQL 文を解答する課題と比較し、評価が低い
 - 文法やエラー箇所の確認が難しかった点が影響していると考えられる



19

アンケート調査:自由記入欄

- 殆どの学生がこれまでの質問項目外の箇所で問題を感じていない



20

アンケート調査: 自由記入欄で指摘された問題点

- 表示が崩れる・動作しない
 - 対応していないブラウザでのアクセスに起因するもの
 - 提案システムは、Chromium系ブラウザでの動作が前提
 - Internet Explorerでは動作しない
- 課題が期待する出力内容が分かりにくい
 - 模範解答の出力結果を表示することで解決できる
 - 求める回答は、SQL 文や O/R マッパーを利用したプログラム
 - 出力結果を学生に公開しても問題はない

21

21

まとめ

- データベース演習支援システムの提案
 - Webシステムであるため、環境の導入が不要
 - 課題の採点結果を即時に確認
 - 学生の進捗状況を容易に把握
 - 環境の導入を省き演習を行う時間を増やすことに成功
- 実際の授業に導入し、アンケート調査により評価
 - 課題を始めるまでにあるハードルを減らす目的は達成

22

22

今後の課題

- 演習課題を提案システム向けに調整
 - O/R マッパーで利用するJavaScriptの説明を追加するなど
- 自動採点機能の強化
 - SQL 文の実行して採点する課題における採点対象範囲の調整
 - O/Rマッパーを用いた課題への対応

23

23