Proposal for a Course Support System for Database Exercises

Rimpei Kunimoto
Graduate School of Information Science and Arts,
Osaka Electro-Communication University
1130-70 Kiyotaki, Shijonawate-shi
Osaka, Japan
mt18a004@oecu.jp

Hiroyuki Hisamatsu
Graduate School of Information Science and Arts,
Osaka Electro-Communication University
1130-70 Kiyotaki, Shijonawate-shi
Osaka, Japan
hisamatu@osakac.ac.jp

ABSTRACT
In this paper, we propose a database exercise support system as a Web application by which a student can learn Structured Query Language easily. By using our system, it is possible to; study an object relational mapper that is an important access method to a database in development of the system. Because our system is a web application, a student can utilize the system without environmental construction. A student can submit assignments on the system, and based on it a teacher can confirm the progress status of a student's seminar problem. The database seminar support system proposed in this paper was utilized for an actual course and was verified by a questionnaire given to the students. From the results of the survey, we show that mitigating the problems related to installing the necessary environment for database learning allows a teacher and the students to focus on the essential contents of database course by using our system.

CCS Concepts
•**Social and professional topics→Computer science education.** *Computing education programs*

Keywords
Databases; SQL; Web-based Learning System; E-learning

1. INTRODUCTION
Databases are used in various information systems such as point of sales systems, smartphones, and web browsers. It is very important for a student in a Computer Science course to learn how databases function. When conducting database exercises and classes at universities, it is necessary to make the database system available to each student's computer and distribute databases to be used for learning. In addition, to learn to use object relational (O/R) mappers that are currently central to operating databases in actual system development, each student's computer must be set up as a complex programming environment.

ICIT 2018, December 29–31, 2018, Hong Kong, Hong Kong

ACM ISBN 978-1-4503-6629-8/18/12...$15.00
https://doi.org/10.1145/3301551.3301594

Under the present circumstances, lesson time is required for the setup and explanation of these environments. This creates a problem as there is less time for practicing the use of database exercises. In addition, the added effort required to set up these environments makes it harder for students to participate in classes, and this stops some students from participating.

Therefore, in this paper, we propose a solution to these problems by using a web application that makes it easier to learn to use databases. In the proposed system, as the operating language for the databases, Structured Query Language (SQL) has been adopted owing to its use in actual databases. Moreover, in the proposed system, learning operations using an O/R mapper are supported. Furthermore, by providing this as a web application, instructors can take advantage of the central management of the databases used for learning on the server-side, allowing them to check student progress on the exercises.

The rest of this paper is structured as follows. In Section 2, we describe the related work on support systems for database exercises. In Section 3, we propose our database exercise support system. Then, we describe the implementation of our system in Section 4. In Section 5, we show results from actual use in lessons. Finally, in Section 6, we summarize this paper, and discuss topics for future research.

2. EXISTING DATABASE EXERCISE SUPPORT SYSTEMS
phpMyAdmin [1] is a web application that can operate a database from a web browser; it is used by many to study databases. phpMyAdmin is aimed at managing database servers via a web browser and makes it possible to execute any SQL statements. However, as this system is not used for learning SQL statements, it lacks the functionality for grading and submitting tasks for educational purposes. In addition, a user cannot operate the database by using the O/R mapper in phpMyAdmin.

There are numerous studies on systems supporting database exercises in classes. For example, in [2], as an expansion of the functionality of the e-learning platform Moodle, the ability to execute SQL statements was implemented. The purpose of this system is to add the execution of SQL statements to the existing e-learning platform. The grading of the SQL statement submitted as an answer is not automated.

In [3], a database study support system was implemented as a Web application, and used in an actual lesson. In this system, an SQL sentence is generated from the natural language that the student inputs, and a database is operated without using an SQL sentence. This system aims at understanding not SQL, but the

concept of a database. However, a database and SQL are closely related now. Hence we think it is necessary to use SQL for the study of a database.

In [4], the study support system aiming to study the SELECT sentence that acquires the data that are in agreement with the conditions from a database among SQL sentences is implemented as a Web application. It is not assumed that an O/R mapper will be used. However, as O/R mappers are prevalent in the current operation of actual databases, even for educational purposes, it is believed that the use of an O/R mapper is necessary.

In [5], the study support system aiming to be implemented in a course at a university is developed as a web application. This system is evaluated by using in an actual course. In this system, a student answers the question of an exercise in the SQL sentence form. In the case of a wrong answer, the execution results of both a wrong answer and a correct answer are shown to the student and the student repeats the correction of the answer, thus approaching the correct answer. However, this system aims at supporting the table design, and the correct SQL sentence is required as an input. This system is not suitable from the viewpoint of mastering SQL sentences. Moreover, this system does not consider the use of an O/R mapper.

In [6], the authors implemented the system as a web application that carries out examinations with answers in the form of SQL sentences, and they evaluated the system by using it in the actual course. In this system, a student writes an SQL sentence for each exercise and the system corrects it by comparing the execution result with that of the correct answer. However, since this system aims at examining, a feature required for its application to the course to display a detailed error message is not provided. Moreover, the system is not provided with the environment for the execution of an O/R mapper.

3. DATABASE EXERCISE SUPPORT SYSTEM

Figure 1 shows an outline of the exercise support system. In our exercise support system, instructors can package and distribute exercises and related databases to students. Furthermore, when students submit answers to exercises, their answers can be checked. As instructors can instantly check the percentage of correct answers from students during lectures, lessons can progress more effectively.

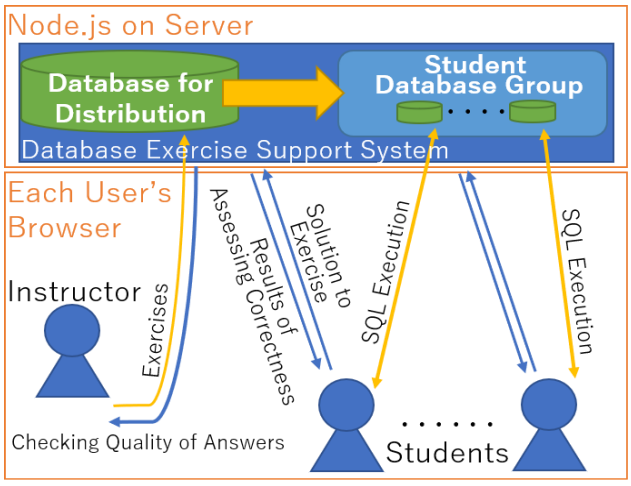

Figure 1. Outline of the database exercise support system

In the proposed system, there are two kinds of accounts: student and instructor. With a student account, it is possible to execute SQL in the exercise database, roll back the database to its initial state, submit exercises, and verify the correctness of the submitted tasks. With an instructor account, in addition to all the functions available through a student account, exercises and exercise databases can be distributed from the instructor's screen and the quality of answers can be checked. Moreover, instructors can create student accounts and manage the system.

In this study, it is assumed that the flow of practice lessons begins with instructors distributing tasks and databases to perform them to every student on the database exercise support system. Subsequently, the database for distribution is replicated and created for each student account. This method is used because, if students used the database that is shared by all students, the execution results of one student would affect the database, and the other students' execution results would be affected. Furthermore, when a student database is corrupted or has some other problem owing to an operation error, it can easily be restored using the original distributed database. Each student logs in to the database exercise support system, selects a task, and codes and runs a program using either SQL statements or an O/R mapper in the editor in the web browser. Corrections are then made to the contents of the code referencing the execution results, the output of the content set in the task is checked, and it is submitted. Instructors then confirm the overall submission status and content of the submissions. They can conduct their classes, emphasizing tasks with a low amount of submissions or correct answers.

Exercise tasks are classified into tasks where SQL statements are required, tasks to obtain execution results of the O/R mapper, and other tasks. For tasks requiring SQL statements, instructors set answers in advance for each exercise. Students' inputs and answers are executed on each of their databases, and by comparing the output results, it is judged whether the submitted inputs are correct. In the case of tasks such as "find the number of records based obtain conditions," the correctness of the answer is judged by comparing whether the results of the student's inputs are an exact match to the model answer set by the instructor.

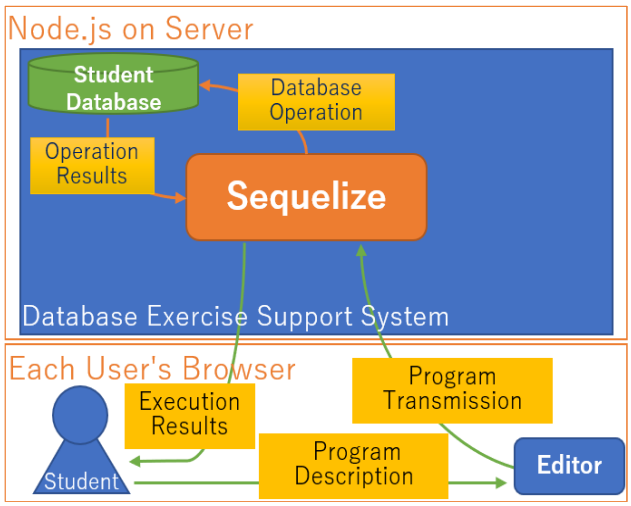

Figure 2. Outline of O/R mapper operation

The exercise support system also supports tasks that require operating a database using an O/R mapper. Figure 2 shows the outline of the part related to the O/R mapper in the exercise

support system. In order to use the O/R mapper, Sequelize [7], an editor that can edit and save JavaScript in a browser, was designed along with a function to check execution results. Sequelize's library for Node.js does not run in a browser but runs on a server. Therefore, we provide a function that allows programs created by students in the editor in a browser to be sent to the server, after which their execution results are acquired. This allows Sequelize to be executed without students needing to load Node.js onto their computers. In addition, when an O/R mapper is used for a problem, as it does not determine the correctness, only the submitted answer is saved in our current system.

4. IMPLEMENTATION OF THE PROPOSED SYSTEM

The proposed system is implemented as a web application. The system is developed with two screens; the first screen of the exercise is for the student to obtain an SQL sentence, and the second to operate a database using an O/R mapper. Figure 3 shows the screen of the exercise used to obtain an SQL sentence. The objective of the exercise is displayed on the top of this screen, as shown in Fig. 3(A). The SQL sentence entered in the input field, shown in Fig. 3(B) can be applied to the database of the exercise by clicking the execution button, and the execution result is displayed as shown in Fig. 3(C). A student can try to execute an SQL sentence freely in our system. A student then submits the execution result, considering it the correct answer to the exercise, with a submission button.

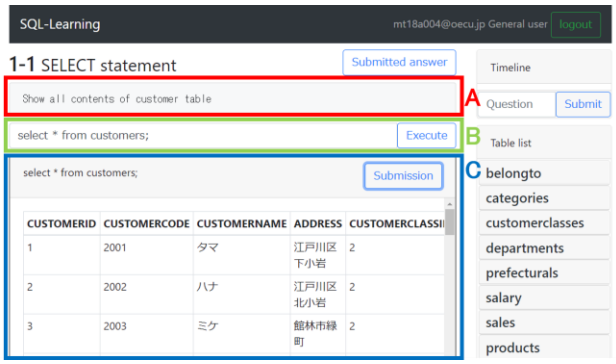

Figure 3. Operation screen: exercise to obtain an SQL sentence

Figure 4 shows the screen after submitting the execution result. Figure 4(A) shows the contents submitted to the server. If it is an exercise that can be corrected automatically, the result is displayed (Fig. 4(B)).

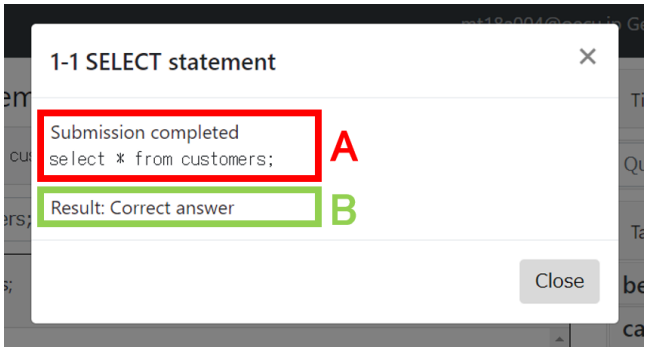

Figure 4. Operation screen: after submitting the execution result

Figure 5 shows the screen of the exercise that a student needs to operate a database using the O/R mapper. The contents of the exercise are displayed in Fig. 5(A). The program of JavaScript developed in the editor, as shown in Fig. 5(B), can be executed on the server-side with the execution button. The standard output and standard error output at the bottom of the screen, as indicated by Fig. 5(C). Sequelize, the O/R mapper, is implemented into the server-side, and a student can execute the program of JavaScript that operates the database using Sequelize. Moreover, a student can submit the execution result, considering it the correct answer of the exercise, with the submission button.

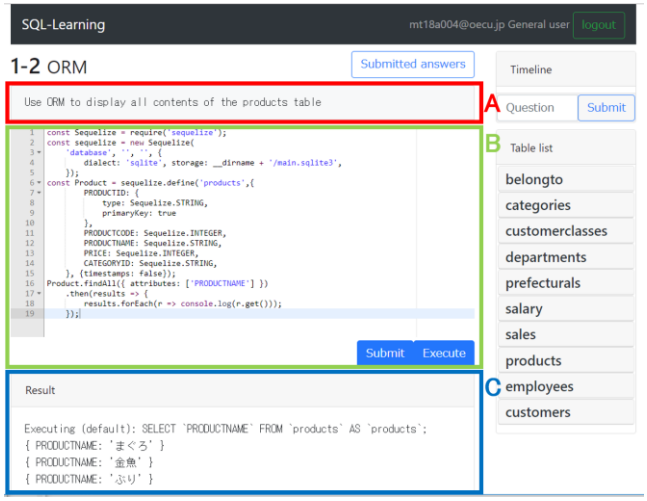

Figure 5. Operation screen: Exercise to operate a database using an O/R mapper

We developed the course support system by using JavaScript in the server-side and TypeScript in the client-side. To be specifically, the application server in server-side is implemented using Node.js [8] and utilizes LoopBack 3 [9] as an API framework, and Passport [10] as an OAuth2.0 authentication library.

We implemented the client-side using Angular 5 [5]. In the exercise to obtain an SQL sentence, in order to prevent the increase in response time even if a large number of students use the system simultaneously, we implemented the feature to apply an SQL sentence to a database in the client-side. When the exercise is displayed for the first time, the database used in the exercise is downloaded from a server, and the database is saved in the browser, i.e. at the client-side, that a student uses. The SQL sentence that the student inputs is applied to the database saved in the browser using sql.js [12]. Therefore, our system can execute the SQL sentence at the client-side without communicating the server-side, and the time taken from inputting an SQL sentence to displaying the execution result is very short. The source code of the proposed system is acquirable from [13].

5. VERIFICATION USING SURVEY

We used our exercise support system in a class and conducted a survey about the experience. The exercise support system was used at the Osaka Electro-Communication University in the databases class from April 10, 2018 to July 24, 2018. Of the 15 classes in the course, the exercise support system was used from the 8th class to the 14th class for a total of seven times. From the 8th class to the 13th class, exercises involving the execution of SQL statements were conducted, and exercises using an O/R mapper were only conducted in the 14th class. There were 76

participants, and the average class attendance per lesson was 64.5 people. On the last day of the course, a survey was conducted, and of the 64 participants in the 15th class, 61 people responded. The five items for which results were collected in the survey are shown in Table 1. Participants were required to write their names on their surveys, and for Questions 1 through 3, multiple answers were allowed. Questions 4 and 5 were evaluated on a five-point scale with the lowest rating being 1 and the highest being 5.

Table 1 Survey Results

Choice	Number of Responses
Q1) Where did you first experience a hurdle in your learning?	
System login and usage	8
Tasks about SQL statements	23
Tasks about an O/R mapper	39
Other	2
Nothing in particular	8
Q2) Where did you encounter problems with tasks inputting SQL?	
SQL statements descriptions	21
SQL execution	14
Grading and submitting	36
Other	3
Nothing in particular	4
Q3) Where did you encounter problems with tasks using an O/R mapper?	
Understanding an O/R mapper concepts	27
JavaScript grammar	25
Sequelize usage	15
Checking for error locations	30
Executing and submitting programs	16
Other	1
Nothing in particular	4
Q4) Did this system make learning SQL easier?	
1	1
2	5
3	30
4	20
5	5
Average	3.3
Q5) Did this system make learning using an O/R mapper easier?	
1	4
2	10
3	31
4	14
5	2
Average	3.0

From the results of Question 1, it is clear that the fewer students found that the exercise support system itself was a hurdle to learning than the tasks themselves. Thus, we believe that the goal of the system to mitigate problems that arise at the outset of learning, from aspects such as setting up the database system, was largely achieved.

From the results of Question 2, when executing SQL statements using the exercise support system, the students felt that there was a problem with the grading and submission functions. As the automatic grading function of the exercise support system scores is based on exact matches in the output, including column names, it is believed that this problem occurs as a result of a failure in judgment when a different column name is used. This is solvable by excluding the column names as grading targets of the exercise support system or clarifying a specified name to be used in the exercise.

From the results of Question 3, when executing the O/R mapper using the exercise support system, the students felt that there was a problem understanding the concepts of the O/R mapper and using JavaScript. This is because the core lectures focused on the concepts of databases and the use of databases with SQL statements, and very little time was devoted to explaining the methods of operating the O/R mapper. Moreover, most of the students attending these lectures were using JavaScript for the first time, and before operating the O/R mapper, they already faced hurdles such as the coding grammar. These problems are solvable by securing additional lecture time for explanations on the concepts of O/R mappers and by changing the programming languages used by students in other lectures.

In the results of Question 4 and Question 5, many students marked 3 or higher on the five-point scale, showing that they did not find it difficult to learn using the exercise support system. The rating marked for learning using the O/R mapper (Q5) was lower than that for learning SQL statements (Q4). From the responses to Question 3, we believe that this is owing to the difficulty in using the JavaScript grammar and locating errors.

Based on the above results, it can be observed that we were able to mitigate problems arising from installing the environment, which is necessary to learn using SQL and O/R mappers, and allowing students and a teacher to focus on the essential contents of the database course by using our system.

6. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a database exercise support system that aids in lessons involving database exercises. Furthermore, we used this system in an actual course, and conducted a survey of the students using a questionnaire. From the results of the survey, we discovered that our goal of mitigating the problems related to installing the necessary environment for learning on each student's computer was largely achieved.

The questionnaire revealed a need to improve the automatic grading function and the accuracy of the exercise support system and this corresponds to the tasks that involve using the O/R mapper. Moreover, we plan to improve the overall operability of the exercise support system in the future.

7. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP17K00144.

8. REFERENCES

[1] "phpmyadmin." available at https://www.phpmyadmin.net/.

[2] K. Harezlak and A. Werner, "Extension of the moodle e-learning platform with database management

mechanisms," in 3rd International Conference on Human System Interaction, pp. 491–495, May 2010 DOI=. http://dx.doi.org/10.1109/HSI.2010.5514525

[3] A. Grillenberger and T. Brinda, "eledSQL: A new web-based learning environment for teaching databases and SQL at secondary school level," in Proceedings of the 7th Workshop in Primary and Secondary Computing Education, WiPSCE '12, pp. 101–104, ACM, 2012 DOI= http://doi.acm.org/10.1145/2481449.2481474.

[4] P. Garner and J. Mariani, "Learning SQL in steps," Journal of Systemics, Cybernetics and Informatics, vol. 13, pp. 19–24, Jan. 2015

[5] I. Bider and D. Rogers, "YASQLT - yet another SQL tutor," in Proceedings of the 4th Symposium on Conceptual Modeling Education, pp. 197–206, Nov. 2016 DOI= http://doi.org/10.1007/978-3-319-47717-6_17.

[6] A. Kleerekoper and A. Schoeld, "SQL tester: an online SQL assessment tool and its impact," in Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, pp. 87–92, July 2018 DOI= http://doi.org/10.1145/3197091.3197124.

[7] "Sequelize." available at http://docs.sequelizejs.com/.

[8] "Node.js." available at https://nodejs.org/.

[9] "Loopback." available at https://loopback.io/.

[10] "Passport." available at http://www.passportjs.org/.

[11] "Angular." available at https://angular.io/.

[12] "sql.js." available at https://github.com/kripken/sql.js/.

[13] "SQL-learning." available at https://github.com/hisa-lab/SQL-Learning-Pub