Experience with Active Learning and Formative Feedback for a SQL Unit

Christine F. Reilly
Computer Science Department
Skidmore College
Saratoga Springs, New York 12866
Email: creilly@skidmore.edu

Abstract—This Full Paper in the Research to Practice Category describes experience with using active learning combined with formative feedback for the SQL unit in both undergraduate and master's-level database courses at a regional public university in the United States. The contributions of this paper are: a detailed description of the implementation of this active learning unit including software configuration and the queries used for the activity; providing hints about the correct solution as a method of formative feedback, and how the example database and SQL problems must be carefully designed in order for this feedback to be useful; and experience with implementing the activity in both a computer lab and a classroom, with class sizes ranging from 20 to 50 students. Additionally, experience with using a virtual machine to provide a computing environment that students can use on their personal computers is discussed as part of this paper. The computer programs and SQL problems that are described in this paper are available for download from the author's website.

I. INTRODUCTION

The structured query language, SQL, is a topic that is commonly covered in both undergraduate and graduate-level database courses. SQL is a declarative language where the programmer provides a query that describes the desired result and the database management system (DBMS) software determines how to execute the query. Learning SQL can be challenging because many learners are accustomed to programming in imperative languages such as C++ or Java, and because it is not a simple task to translate a natural-language problem specification into a SQL query that returns the desired result. As with any programming language, countless hours of practice are required when learning SQL.

This paper describes experience with implementing an active learning approach for the SQL unit in both undergraduate and master's-level database courses in a Computer Science department at a medium to large sized regional public university in the USA that is designated as a minority serving institution. Many of the undergraduate and graduate students who take this course are classified as non-traditional students in the USA because they do not live on campus, they have full-time jobs, and some students are older than traditional college student age. The faculty who developed this active learning unit strongly value the idea of providing equal access to learning opportunities for all of the students in the class. In order to support the value of equal access, the following principles were followed:

- Use free and open source software and tools.
- Provide access to course resources in the department computer lab, and as a virtual machine that students can run on their personal computer.
- Do not expect students to have previous knowledge about computer system administration.

Although the consensus in the STEM education community is that it is best to engage the learner during class using various active learning approaches, in practice many college-level classes continue to use a lecture-based approach where students are passive recipients of information [1]. The author of the current paper has limited time and resources for converting lecture-based classes into active learning classes. This challenge is addressed by converting one unit of a course into an active-learning approach each time the faculty teaches the course instead of restructuring the entire course at once. Another challenge is that students at this university were mostly accustomed to passive learning in the computer science courses. The author recognized the need to attach course credit to the preparation required for in-class active learning in order to motivate students who view the course grade as their final goal.

In addition to being of interest to faculty members who teach SQL, the active learning approach described in this paper could be applied to teaching other topics. This paper describes the use of a virtual machine to provide students with a computing environment that they can use on their personal computers. The challenges with the use of virtual machines and approaches to handling those challenges are useful information for any instructor who is interested in using a virtual machine as an alternative or supplement to the institutional computer lab.

This paper proceeds in Section II by discussing two areas of related work: active learning in computer science and engineering courses, and tools and environments for learning SQL. Section III provides a brief description of the lecture-based approach that we previously used for the SQL unit. Then Section IV provides details about the active learning approach for the SQL unit including the necessary preparations prior to the class meeting and the process followed during the in-class activity, followed by a concluding discussion in Section V.

II. RELATED WORK

This related work section begins with a discussion of active learning, then presents related work that is specific to learning SQL.

A. Active Learning

Active learning is generally defined as an instructional method where students are engaged in the learning process [2] with the goal of including the student as an active participant in the learning process [3]. The active learning approach stands in contrast to the teacher-led lecture where students are passive receivers of information. Many different approaches to active learning are described in the Computer Science Education and Engineering Education literature. Some approaches augment a traditional lecture by including time for small group discussions [2], or spending half of class time on problem solving activities [3]. Other active learning approaches entirely get rid of lecture and use student-led learning [4] or a Problem-Based Learning (PBL) approach where students achieve course outcomes through working on open-ended problems [5]. The active learning approach that is described in the current paper is similar to the team based learning approach where students learn content before class, are given a brief lecture at the beginning of class, and then spend most of class time working on problems in small groups [6].

Formative feedback allows students to improve their understanding of a topic, while summative grading is a measure of how well a student has performed a task. In order for formative feedback to be useful to students, it must be provided in a timely manner, and students should be able to engage in multiple feedback cycles [7]. Two challenges when only formative feedback is provided for class assignments are: motivating students to engage in an assignment that does not have a direct impact on their course grade; and being able to provide quick feedback for as many feedback cycles as a student requires.

The evaluations of active learning approaches do not definitively show that active learning results in better student learning due to the wide variety of approaches that are described [2], the small sample sizes in some studies, and the difficulty of measuring long-term outcomes such as the development of life-long learning skills. The literature on active learning in Computer Science Education and Engineering Education have a general consensus that active learning results in greater student engagement during class than is seen in a teacher-led lecture. The need for the instructor to provide an appropriate amount of scaffolding and guidance to the learning activity is also supported by this literature, where the amount of structure required depends on how experienced the students are with active learning and group work, as well as the level of the class.

B. Tools and Environment

There are a wide variety of computer-based tools available for learning SQL. It is often desirable to provide a preconfigured database environment that students are able to use on their personal computer so that the instructor knows all students are using the same environment. Additionally, the preconfigured environment is simple for all students to use because it requires very minimal setup.

Microsoft Access can provide this preconfigured environment because the database is saved as a single file that can be distributed to all students; however there are a number of disadvantages [8]. Perhaps the biggest disadvantage of Microsoft Access is that it is designed for small databases on personal computers, and is not a production-level DBMS. Another disadvantage is that the software includes a query building tool that allows students to use a graphical user interface to create SQL queries without actually writing SQL.

A number of online tools are available from third-party vendors and free websites. While these tools are easy for students to access, they have many drawbacks from the instructor's perspective [8]. A large drawback is that the third-party website may become unaccessible. Other drawbacks to the third-party systems that were reviewed by Cigas [8] include inadequate customer support, and not being able to create a custom schema or database for the class. Additionally, when a third-party tool provides the same set of queries to many different classes, students are likely to be able to find the solutions to these queries on the internet.

Another mechanism for providing a pre-configured environment is for the instructor to host their own database server and allow students to connect to that server. Many DMBS's have a GUI tool that students can use to connect to a server, run queries, and view query results. The benefit of this approach is that the instructor is able to create a customized database for the course. The drawbacks to this approach include acquiring the resources for hosting a database server that students can access over the internet and ensuring that students are following the proper steps for connecting to the server [8]. A related approach is to utilize a tool that can be integrated with the Learning Management System (LMS) that students are already using at the institution [9]. This tool provides an interface where students can submit SQL queries and receive feedback. One drawback of using a LMS tool is that students do not gain experience interacting with an actual DBMS.

There are many different DBMS's that an instructor may select for use in a class. Relational database systems that are currently in common use include Oracle, Microsoft SQL Server, and MySQL. While all of these systems support the ANSI standard for SQL, they each have some system-specific functionality. Some instructors choose to only use one DBMS during class, while others select two or more of the systems and ask students to compare the different systems. Using multiple DMBS's during a class has been shown to provide positive learning outcomes especially for illustrating what is ANSI standard and what is system specific, but can require more class time for covering the differences between the DBMS's [10].

Relational database systems have been the traditional focus in database systems courses. However, during the past ten years a number of different database approaches have been
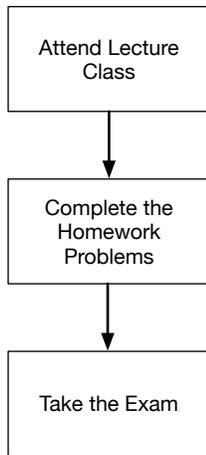
Fig. 1. Learning Activities During the Lecture-Based Approach

Question	List the id and names of sailors who have reserved a blue boat.
Incorrect SQL	SELECT S.sid, S.name FROM Sailor S, Boat B WHERE B.color = blue;
Correct SQL	SELECT S.sid, S.name FROM Sailor S, Reserve R, Boat B WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = blue;

developed and have come into common use. These include those that are called NoSQL (Not only SQL) systems, as well as the distributed storage systems that support the MapReduce programming model. Providing students with a virtual machine that has the systems installed and configured has been shown to be a good approach for teaching multiple types of database systems during a single course [11].

III. PREVIOUS APPROACH

The author of this paper taught three semesters of a master's-level course and one semester of an undergraduate-level course using a lecture format for the SQL unit. Figure 1 shows the sequence of learning activities for this approach. During class the professor used a lecture-style presentation for each SQL operation by discussing the operation and showing a small example to demonstrate the operation. After presenting one or more operations, students were asked to work in small groups on example problems using pen and paper. This lecture and practice approach was repeated throughout the unit. The example problems utilized a small database consisting of a few rows per table. When discussing the correct SQL expressions for each problem the small result data set was also presented and discussed. Neither the professor nor the students used DBMS software during class. The homework assignment for the SQL unit could be completed as a pen and paper assignment without the use DBMS software. Students were encouraged to use DBMS software to test their homework solutions. Given the syntax and logic errors found when grading the homework, it was apparent that many students did not use DBMS software. Assessment of learning consisted of summative grades on the homework and the exam.

Problems observed by the faculty who used the previous approach for the SQL unit included students not understanding the concepts, students not meeting the learning outcomes for the course, and students not engaged during class. Each of these problems is discussed in turn.

When answering students' questions and in grading the homework assignment, the faculty observed that some students had difficulty understanding why the query they wrote was not the correct solution to a problem. The presence of SQL syntax errors in some homework submissions indicated that not all students were taking the extra step of running their homework solutions in a DMBS. The questions some students asked about their homework grade indicated that students had difficulty understanding why their solution was incorrect. Table I shows a common incorrect answer for Question 13 from Table IV, along with a correct answer for this question. The misunderstanding demonstrated in Table I is not explicitly stating the join condition and not using the Reserve table that contains the relationship between sailors and boats. The faculty observed a number of variations on this misunderstanding of how to join tables in SQL, and sometimes had difficulty convincing students of the correct solution.

Another significant issue with the previous approach for the SQL unit was ensuring that students achieve the course learning outcome of gaining experience with the use of DBMS software. The semester-long project was the only time when students were required to use DBMS software. The semester-long project was a group project where some groups delegated parts of the project to different group members. It was possible that some students were completing this course without ever using DBMS software, and therefore not meeting one of the major learning outcomes of this course.

The faculty who taught the course observed student behavior that indicated low levels of student engagement in the lecture approach such as students falling asleep during class and low attendance rates. These problems with the lecture format, combined with the increasing use of active learning in computer science courses, motivated the faculty to develop a new approach for the in-class portion of this SQL unit.

IV. ACTIVE LEARNING DURING CLASS

The hypothesis for the new class format was that replacing lecture with active learning where students spend the class period writing SQL queries and running their queries in a DBMS would enhance learning and keep students more engaged. The approach for this active learning SQL unit was to guide the students through writing a series of queries, where the initial queries demonstrate the use of simple SQL operations and later queries build up to more complex SQL operations.

The design of this active learning unit matches the values held by the faculty who teach it: all students in the class should have equal access to the opportunities to learn. Design decisions that were made in support of this value include using open source and no-cost materials, providing multiple ways for students to use the computing environment outside of class time, and not expecting students to have computer system administration expertise.

Other factors that impacted the design of this active learning unit were related to the context of the university. The students who take this database course have a wide variety of background knowledge and experience. For example, the students in the master's-level course are drawn from two majors: Computer Science, and Information Technology. The students in the Computer Science master's program have an undergraduate background in computer science, while the Information Technology program is designed for students who have an undergraduate degree in a non-computing field. Some of the students have enough expertise to install and configure database management system (DBMS) software on their personal computer, while other students do not have this expertise or do not own a personal computer.

There were also challenges related to motivating students to participate in active learning because many of the students are accustomed to passive learning. Associating a grade with the pre-class preparation and providing formative feedback on assignments and in-class activities were found to increase student engagement in the active learning process.

The pre-class preparations for both the instructor and the students are described in Section IV-A, followed by a description of the implementation of the in-class activity in Section IV-B. The programs and SQL scripts described in this section are available for download from the author's website: http://www.drcreilly.com/

A. Pre-Class Preparation

In order for this active learning approach to be successful, both the instructor and students must prepare prior to the class session. In addition to designing the in-class activity that is discussed in Section IV-B, the instructor must prepare a database server containing a database that is populated with a large amount of data and provide students with access to this database. Additionally, the instructor must assign readings and pre-class problems so that the students are sufficiently prepared for the in-class activity. Table II lists the instructor's pre-class preparations, and a discussion of these preparations follows below.

1) Example Database Schema: The BoatRental schema from the Ramakrishnan and Gehrke textbook [12] was used as the example database. This schema, shown in Table III, was chosen as the example database because it is a very simple schema with three tables that each have few attributes, and it can be used to demonstrate a wide range of SQL operations. The BoatRental schema represents the scenario where a club allows its members, who are called Sailors, to reserve Boats.

TABLE II
INSTRUCTOR'S PRE-CLASS PREPARATIONS

Category	Items to Prepare
Example Database Schema	• Select database schema. • Acquire or generate a large amount of data to insert into the database tables.
Centralized Database Server	• Install a DBMS on a server that students can access. • Create database with example schema and data. • Create a DBMS account for each student. • Grant appropriate DBMS privileges to each student account.
Virtual Machine	• Install a lightweight operating system. • Install DBMS software. • Create database with example schema and data. • Distribute virtual machine to students.
Pre-Class Assignment	• Assign readings and problems one week prior to the class meeting.

TABLE III
BOATRENTAL DATABASE SCHEMA

Table	Attribute List (P: Primary Key; F: Foreign Key)
Sailor	sid (P), name, age, rating
Boat	bid (P), bname, color
Reserve	sid (F to Sailor), bid (P) (F to Boat), date (P)

In order to for students to experience the power of a relational database, a large amount of data was inserted into this database. The definition of large in this case is enough data such that students cannot quickly answer queries by visually inspecting the database. The sample data provided in the textbook is not large enough, so data was generated for the database. The Sailor and Boat tables were populated using an online data generation tool [13]. The Sailor table was populated with 100 rows, and the Boat table with 20 rows. Populating the Reserve table requires that tuples from the Sailor table be matched with tuples from the Boat table in order to preserve the foreign key constraints in the database schema. Therefore, a Java program was created to connect to the database and obtain a list of the primary key sid values from the Sailor table and the primary key bid values from the Boat table. These primary key values are then used when creating rows for the Reserve table. Approximately 1000 rows were inserted into the Reserve table.

2) Database Server: As discussed in Section II-B, there are many choices for what flavor of SQL to use as well as many computer-based tools that have been designed for learning SQL. MySQL was chosen as the database system for this course because it is widely used, is free and open source software, and runs on many different platforms. Others have discussed the benefit of students working with two different database systems so that they can explore how different systems deviate from the ANSI SQL standard [10]. The author of the current paper chose to use only MySQL because of time constraints. As discussed in Section IV-B, ANSI SQL operations that are not implemented by MySQL

were discussed during the in-class activity.

The learning outcomes for these courses include that students gain practical experience with using and administering a DBMS. Additionally, students must use MySQL for the course project. Therefore, during the in-class activity students use the command line interface or MySQL Workbench graphical user interface to connect to a MySQL server and run queries, as opposed to using a tool with a web interface or a SQL tutoring system. Two different options were used for providing students with access to the database server: connecting to a centralized database server, or asking students to download a virtual machine to run a local database server on their personal laptop. A description of each of these access methods follows.

3) Centralized Database Server: The centralized database server was installed on one of the department's computer servers. Due to security concerns, the database server only accepted connections from localhost meaning that students must log into the host machine before connecting to the database server. Students were able to use SSH to connect to the host machine from the on campus network or through a VPN from off campus.

The instructor had administrative privileges on this database server and was able to create database accounts for the students. The MySQL server uses different login credentials than those used for the operating system. To simplify the process of creating the student accounts and of deleting these accounts at the end of the semester, a program was developed to generate SQL scripts for these tasks. The program also prints each username and its corresponding password to a text file that the instructor can reference when assigning a database username and password to each student.

Each of the student database users were granted SELECT privileges on the example databases that are used for the class. This provides read only access so that students can issue queries that read the databases but are not able to make any changes to the example databases. A "sandbox" database was also created for each student. All database privileges were granted to each student on their own sandbox database so that they could use this sandbox database to practice the SQL Data Definition Language statements that alter the database.

4) Virtual Machines: The second approach for providing a practice database to students was to create a Linux virtual machine that included the example databases. The SEED lab computer security exercises [14] provided the inspiration for this use of virtual machines as a learning tool. Providing students with a virtual machine that has the database server already installed and set-up ensures that all students have access to the same platform. An advantage of providing students with this virtual machine was that they were able to connect to the database as a user with read only privileges or as a user with administrator privileges. This provided students with additional opportunities to learn about DBMS administration.

Virtual Box was chosen as the virtual machine software because it is free and open source software that runs on all of the operating systems that the students are likely to have on their personal computers. The instructor created a custom virtual machine using the Lubuntu Linux distribution because it is a lightweight distribution that is similar to the popular Ubuntu distribution. The MySQL database server containing the example databases for class was installed on the virtual machine along with the MySQL Workbench GUI tool. Software that would be used for other units during the class was also installed, such as Java and a development environment. Unnecessary software that is included in a standard Lubuntu installation was uninstalled in order to keep the virtual machine as small as possible.

It was important to keep the virtual machine as lightweight as possible. The file size of the virtual machine that students can download is over one gigabyte and took a long time to download over the campus wireless network. Students could visit the instructor's office to copy the virtual machine file from a flash drive if they had difficulty with the network download. The first iteration of this approach used a standard installation of the Ubuntu Linux distribution. The Ubuntu virtual machine had a larger file size and required more memory resources on the host machine. Some students found that the Ubuntu virtual machine was unusable on their personal computer due to the memory required. There were fewer complaints from students when using the Lubuntu virtual machine. Students were instructed that if they experience slow performance it is best to have the virtual machine as the only program running on their computer, and that they should refrain from swapping between the virtual machine and other programs.

5) Pre-class Assignment: In order to engage in the active learning approach during class, students must learn background knowledge before class. Figure 2 shows the learning activities that students engaged in before class, including the use of formative feedback. Students were required to complete a pre-class assignment in order to become familiar with the database schema and basic SQL concepts. This use of a pre-class assignment allows class time to focus on complex concepts with student centered learning activities. The pre-class assignment consisted of reading the textbook and reviewing lecture slides that demonstrate SQL operations. Students were also required to write SQL expressions that answer Problems 1 through 4 as shown in Table IV. In order to help motivate students to complete the pre-class assignment, they handed in the four SQL expressions at the beginning of class and this pre-class assignment was worth one point toward the assignment component of the course grade. Students were informed that the grade for the pre-class problems would focus on effort rather than on correctness.

Formative feedback was provided by giving students the result set for each of the pre-class queries. This allowed students to check the result of their SQL expression against the correct result. For students who are motivated to participate in this active learning approach, having the correct result set helps them determine whether they have the correct answer for the pre-class problems. In semesters when this formative feedback was given, many more students asked questions about these problems prior to the class meeting showing that providing formative feedback improves the learning process during the
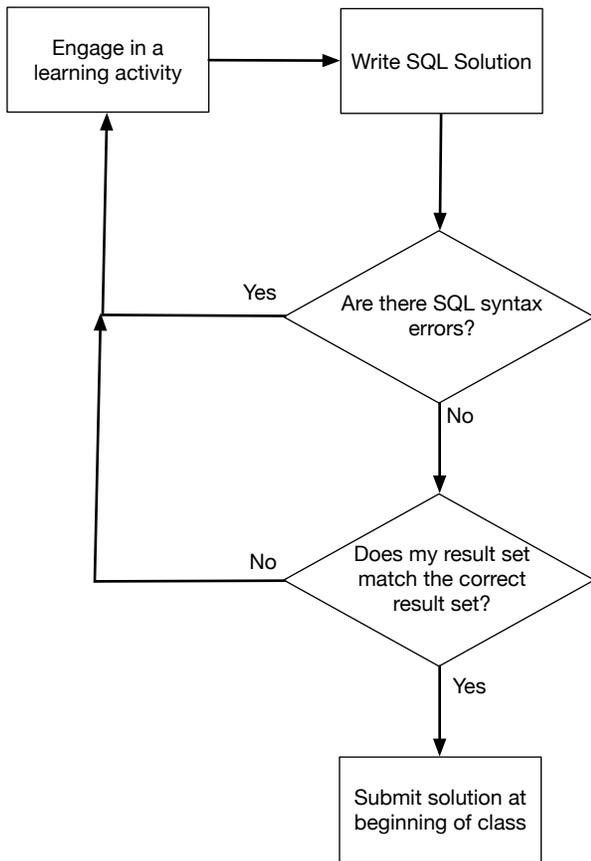
Fig. 2. Use of Formative Feedback During Pre-Class Assignment

pre-class assignment. In contrast, in the first semester of this active learning approach when formative feedback was not provided, some students did very poorly on these pre-class problems and did not have an adequate understanding of basic SQL concepts at the beginning of the class session.

B. In-Class Activity

The in-class activity focused on the students creating SQL expressions and running their expressions on the database to determine if they are correct. Figure 3 shows the learning activities that students engaged in during class, including the use of formative feedback. In order to accommodate students who did not adequately learn the material before class, the instructor gave a short (less than five minute) introduction and demonstration of one or more SQL operations. Students then participated in the active learning activity by completing a few practice problems and running their solutions in the database. Then the professor facilitated a discussion of the solutions to these problems as a whole class. This demonstration, practice, and discuss pattern was repeated throughout the class time in order to cover a variety of SQL operations. This section begins with a description of the series of queries that are used for the SQL lesson, followed by a discussion of ideas for encouraging active learning during class.

1) Queries for the Lesson: A series of twenty-one queries, shown in Table IV, was used to guide students through the
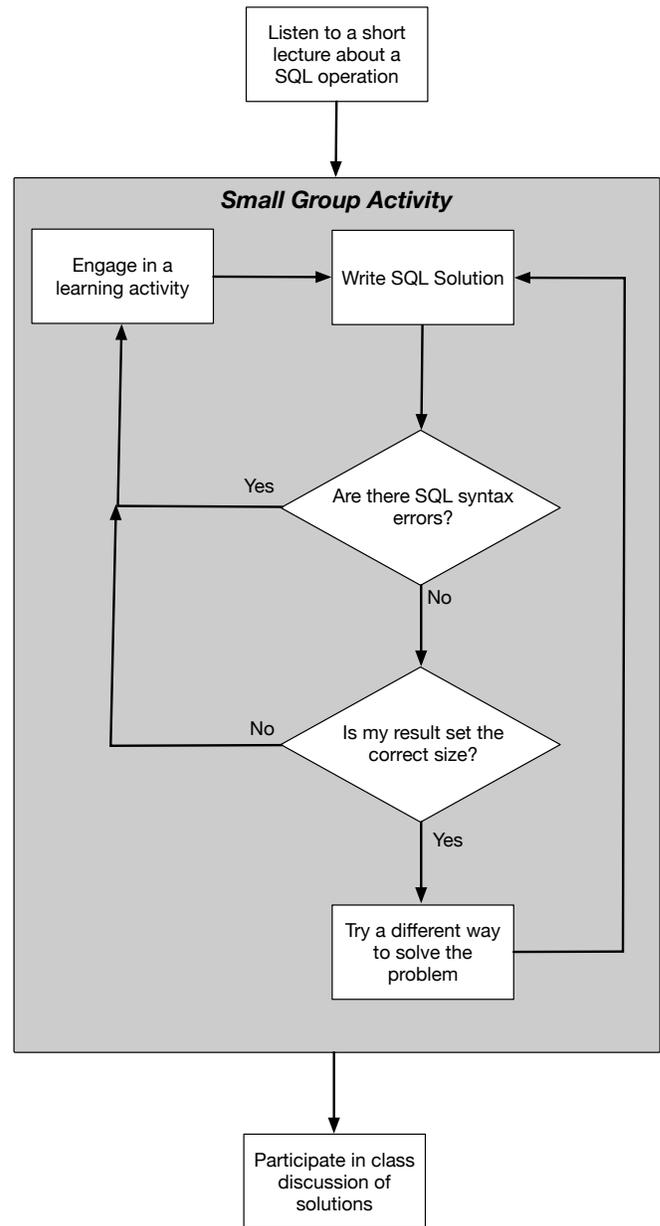

Fig. 3. Use of Formative Feedback During In-Class Activity

process of learning SQL. These queries cover many SQL operations, but not every SQL operation. The overall goal is for students to experience the power of SQL for writing a wide range of queries. Because the course covers many other topics throughout the semester, there is not time for a complete discussion of SQL. The homework and semester project provide additional opportunities for students to learn more about SQL.

Table IV provides the English language problems that were given to students and identifies the SQL skill(s) that each query introduces. Because later queries build on earlier queries, the skill listed for each query is the new skill that is introduced in that query. For a number of these problems, there

TABLE IV
QUERIES FOR THE LESSON

Number	Question	Skill(s) Introduced
1	Find the names and ages of all sailors. Assume there could be multiple sailors with the same name and age.	SELECT DISTINCT
2	List all sailor information for sailors with a rating above 7.	WHERE using a comparison operator
3	List all boat information for boats, sorted alphabetically by color then by ascending bid.	ORDER BY with multiple attributes
4	List the number of sailors who have a rating above 7.	COUNT
5	List all sailor information for sailors with a rating below 4 or over 8.	a) WHERE with complex condition; b) use of NOT BETWEEN
6	List all boat information for boats that are green or blue.	a) WHERE with complex condition; b) use of IN
7	List the ages of sailors whose name begins with the letter 'A', and has at least five characters.	LIKE with wildcard and specific number of characters
8	What is the average age of sailors who have a rating above 7?	AVG as part of SELECT clause
9	List the count of each color of boat.	COUNT with GROUP BY
10	List the color and count of each color of boats where there are at least 5 boats with that color.	use of HAVING with GROUP BY to restrict groups
11	Find the rating level and average age of sailors for each rating level that has more than ten sailors.	use of HAVING with GROUP BY to restrict groups
12	List the id and names of sailors who have reserved boat number 103.	NATURAL JOIN of two tables
13	List the id and names of sailors who have reserved a blue boat.	NATURAL JOIN of three tables
14	List the sid, sname, bid, and date for all sailors and their reservations, including information for sailors who have never made a reservation.	LEFT JOIN
15	List the id and names of sailors who are older than the oldest sailor with a rating of 10.	simple subquery
16	For persons who have sailed two different boats in the same day, find their name and their rating incremented by one.	arithmetic in SELECT clause; and a) correlated subquery using IN; b) correlated subquery using EXISTS; c) JOIN using same table multiple times
17	List the id and names of sailors who have not reserved a purple boat.	simple subquery using IN
18	List the sids of sailors who have reserved both a yellow boat and a green boat. Write two queries: using set operations that are not supported by MySQL; and using operations that are supported by MySQL.	INTERSECT and how express the intersection operation in MySQL
19	List the sids of sailors who have reserved a yellow boat, but not a green boat. Write two queries: one using set operations that are not supported by MySQL, and a second using operations that are supported by MySQL.	EXCEPT and how to express the set difference operation in MySQL
20	List the id and name of sailors who have only reserved blue boats.	correlated subquery using NOT EXISTS
21	List the id and names of sailors who have reserved all purple boats.	expressing set division in MySQL

are many possibilities for a correct answer. When multiple skills could be employed for a correct answer, the "Skill(s) Introduced" column provides a lettered list of the skills. The first eleven queries each operate on a single table and are used to demonstrate the use of the SELECT and WHERE clauses including operations within these clauses, as well as the ORDER BY, GROUP BY, and HAVING clauses. Queries twelve and thirteen introduce the natural join operation. The remaining eight queries introduce more advanced operations: simple subqueries, correlated subqueries, left joins, intersection, set difference, and set division. The answers to some of the queries build upon the answers from earlier queries. This is an attempt to help students recognize how to apply known solutions to solve part of a new problem.

2) Encouraging Active Learning: In order for this active learning experience to work, students must be able to run their queries on the database during class. In some semesters, class was scheduled in a computer lab where students connect to the department's database server. In other semesters, class was held in a lecture room and students were asked to bring laptops with the virtual machine installed. In this case, students who do not bring a laptop are asked to partner with a student who has a laptop.

For the in-class activity, students work in groups with size ranging from two to five. Different group sizes were used in different semesters based on the number of students enrolled in the course and the classroom configuration. When students are working on personal laptops in a lecture room, a group size of two or three students was appropriate. When class was held in a computer lab where there are white boards on all walls of the room, a group size of four or five allowed each group have a designated white board. Students were encouraged to

first brainstorm a solution on the whiteboard or on paper, then try running the solution on the database. Formative feedback was given to students through information about the number of rows in the correct result set, and through feedback from the professor who circulated throughout the groups. When a group developed a correct solution, they were encouraged to explore different solutions for the same query.

During the group work session, the professor circulates around the room and checks with groups on their progress. The professor encourages the group members to work together and utilizes peer interactions for as much instruction as possible. Encouraging peer interactions was especially useful during semesters when there was no teaching assistant. Students are encouraged to break complex problems into smaller problems and reuse solutions from earlier problems as a starting point for the current problem. During this time, the professor also selects groups to present their solution to the class and asks the group to appoint a member as their presenter.

After the group work session, the class comes back together and one or more groups present their solution. The professor ensures that the class discusses different solutions for the same problem and common difficulties with this type of problem. When possible, one of the group's solutions is used to make these points, otherwise the professor provides different solutions for the class to compare. This discussion of multiple correct solutions is used to motivate discussion about factors that impact query performance. During this time, the professor points out cases where different database systems support different SQL operations. After the class meeting, students are provided with a solution file for these queries. The solution file contains the correct SQL expression(s) for each problem along with comments that explain the solution and point out comparisons between multiple correct solutions.

C. Summative Grading

After the class sessions for the SQL unit, a homework assignment and the exam were used for summative grading. For the homework assignment, students are given the number of rows in the correct result set for each problem as formative feedback. This formative feedback encouraged students to test their homework answers using the database software, and to ask questions while they worked on the homework. On the exam, students wrote solutions to the SQL problems using pen and paper because computers were not used for the exam.

V. DISCUSSION AND CONCLUSIONS

This paper provides an example of using an active learning approach for a SQL unit in a database course. This description and the materials available from the author's website can be used by others who are teaching a similar unit, and the way the authors of this paper approached incorporating active learning into an existing course can provide guidance for using active learning in a wide variety of courses.

Providing students with either the full result set or the number of rows in the result set for a query was a simple method of providing formative feedback. This formative feedback

was also provided for the SQL homework assignment in this course. The instructor observed that providing this formative feedback led to more students asking questions while they were working on the assignment, showing that the formative feedback was meeting the goal of encouraging students to develop a greater understanding of the topic.

Using a virtual machine with the database server and example databases pre-installed provides all students with access to the same environment where they can learn SQL, regardless of their ability to go to campus outside of class time and of their prior experience with computer system administration. This paper discussed the need to keep the virtual machine as small as possible in order for the virtual machine to be usable on personal computers with a smaller amount of memory.

The approach taken in this paper is to convert a single unit within a course from lecture to active learning, with the goal of converting the entire course to an active learning approach after a number of semesters. By focusing on a portion of a course, faculty can incorporate active learning into courses in a way that fits into their workload.

REFERENCES

[1] M. Stains, J. Harshman, M. K. Barker, S. V. Chasteen, R. Cole, S. E. DeChenne-Peters, M. K. Eagan, J. M. Esson, J. K. Knight, F. A. Laski, M. Levis-Fitzgerald, C. J. Lee, S. M. Lo, L. M. McDonnell, T. A. McKay, N. Michelotti, A. Musgrove, M. S. Palmer, K. M. Plank, T. M. Rodela, E. R. Sanders, N. G. Schimpf, P. M. Schulte, M. K. Smith, M. Stetzer, B. Van Valkenburgh, E. Vinson, L. K. Weir, P. J. Wendel, L. B. Wheeler, and A. M. Young, "Anatomy of STEM teaching in North American universities," *Science*, vol. 359, no. 6383, pp. 1468–1470, 2018.
[2] M. Prince, "Does active learning work? A review of the research," *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, 2004.
[3] T. A. Budd, "An active learning approach to teaching the data structures course," in *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '06. New York, NY, USA: ACM, 2006, pp. 143–147.
[4] I. C. Moura and N. van Hattum-Janssen, "Teaching a CS introductory course: An active approach," *Comput. Educ.*, vol. 56, no. 2, pp. 475–483, Feb. 2011.
[5] S. B. Fee and A. M. Holland-Minkley, "Teaching computer science through problems, not solutions," *Computer Science Education*, vol. 20, no. 2, pp. 129–144, 2010.
[6] R. M. O'Connell, "Adapting team-based learning for application in the basic circuit theory sequence," *IEEE Transactions on Education*, vol. 58, no. 2, pp. 90–97, 2015.
[7] A. Cain and M. A. Babar, "Reflections on applying constructive alignment with formative feedback for teaching introductory programming and software architecture," in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, May 2016, pp. 336–345.
[8] J. Cigas and B. Kushan, "Experiences with online SQL environments," *J. Comput. Sci. Coll.*, vol. 25, no. 5, pp. 251–257, May 2010.
[9] C. Kleiner, C. Tebbe, and F. Heine, "Automated grading and tutoring of SQL statements to improve student learning," in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '13. New York, NY, USA: ACM, 2013, pp. 161–168.
[10] G. B. Randolph, "The forest and the trees: Using Oracle and SQL Server together to teach ANSI-standard SQL," in *Proceedings of the 4th Conference on Information Technology Curriculum*, ser. CITC4 '03. New York, NY, USA: ACM, 2003, pp. 234–236.
[11] Y. N. Silva, I. Almeida, and M. Queiroz, "SQL: From traditional databases to big data," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16. New York, NY, USA: ACM, 2016, pp. 413–418.

[12] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed. New York, NY, USA: McGraw Hill, 2003.

[13] (2017) generatedata.com. [Online]. Available: http://www.generatedata.com/

[14] W. Du, "SEED: Hands-on lab exercises for computer security education," *IEEE Security and Privacy*, vol. 9, no. 5, pp. 70–73, Sep. 2011.