ISSN 2630-0540
Journal of Technology and Innovation, Vol.1, No.1, pp. 41-59, January –June 2018

doi: ……./jti.2018.5

ASQLAG - Automated SQL Assignment Grading System for Multiple DBMSs

Areerat Trongratsameethong[1] Pakpoom Vichianroj[2]
Department of Computer Science
Faculty of Science, Chiang Mai University
Chiang Mai, Thailand
areerat.t@cmu.ac.th[1] pakpoom.vich@gmail.com[2]
Email: areerat.t@cmu.ac.th, Tel: +66865129600, Fax: +6653892281

Abstract
Manual grading Structure Query Language (SQL) assignment is laborious and time consuming. Particularly, one question may have many SQL statement styles having the same SQL result(s). Moreover, different Database Management Systems (DBMSs) may have different SQL statement syntaxes. This may cause error-prone and students may get feedback too late. Rapid feedback is significant to learning performance. If students and instructors get feedback faster, they will have an opportunity to improve their practice and their teaching, respectively. The researchers have therefore developed an automated SQL assignment grading system using Object-Oriented Design (OOD) technique and Model-View-Controller (MVC) framework. The system consists of two main parts: assignment management and automated SQL grader. Instructors can manage their assignment and student information conveniently anytime and anywhere via internet network. The automated SQL grader is designed to support four DBMSs: MariaDB, MySQL, PostgreSQL, and Microsoft SQL Server. Students can choose one of four supported DBMSs arbitrary to practice their SQL skills and their assignments can be submitted in one of the supported DBMS syntaxes. The automated SQL grader was implemented and tested. The results reveal that grading time compared to manual grading was reduced from many hours to a few seconds per one assignment. Students' scores served as good indicator for students and instructors to improve their learning and teaching, respectively. The system is expected to run in the next semester to collect more comments and requirements from database instructors. The automated SQL grader can be extended to enable instructors to perform partial marking SQL statement as needed in their specific teaching contexts.

Keywords*: Automated SQL Grader, Assignment Management Database, SQL Lines, Object-Oriented Design, Model-View-Controller*

1. Introduction

All business companies use databases to store their business data, for examples, sale transactions, customer data, accounting data, financial data, and the like. Furthermore, most of databases used in business companies are Relational Databases (RDBs). Structure Query Language (SQL) is a language used by all Relational Database Management Systems (RDBMSs) for creating and manipulating data. Students who have excellent SQL skills will have greater chance to earn a higher salary and get a better job.

There are plenty of free online SQL tutorials for students to practice such as W3School which is available at "https://www.w3schools.com/sql" and w3tutors which is available at "www.w3tutors.com/sql" but instructors could not directly control contents in the tutorials. As a result, it is not convenient to gather students' learning results of the whole class and instructors will not know what students understand and what they do not. Moreover, there is no freeware tool that integrates an automated SQL grader with an as-

signment management module. In addition, there are no automated SQL grader tools supporting multiple DBMSs.

The researchers therefore would like to develop Automated SQL Assignment Grading System to handle the problems mentioned above. Instructors can create and manipulate their SQL assignments for students to practice. After students submit their assignment solutions, their solutions are graded by the automated SQL grader and students will know their achievement scores immediately after submission. The automated SQL grader was designed to support four DBMSs: MySQL, MariaDB, PostgreSQL, and Microsoft SQL Server. Students can submit one of four DBMS syntaxes supported in the system. Moreover, student scores are stored in a database for later query by students and instructors. Students can practice more on the SQL topics that they obtain low scores and resubmit their assignments up to the due date. Instructors can also adjust their teaching technique on the SQL topics that most of students in their classes do not perform well and get low average scores.

A. SQL Statement

SQL (Elmasri & Navathe, 2011) is a fourth-generation programming language that is closer to human language. SQL statement is divided into 2 groups: Data Definition Language (DDL) and Data Manipulation Language (DML). The DDL is used for creating a database schema or a database structure. Different RDBMSs may have different DDL syntaxes as shown in an example below:

MySQL/MariaDB:

```
CREATE DATABASE 'CompanyElmasri';
USE 'CompanyElmasri';
CREATE TABLE department (
        dname           varchar(15)     not null,
        dnumber         integer(4)      not null,
        mgr_ssn         char(9)         not null,
        mgr_start_date  date,
        primary key (dnumber),
        unique (dname)
);
```

PostgreSQL:

```
CREATE DATABASE 'CompanyElmasri';
USE 'CompanyElmasri';
CREATE TABLE department (
        dname           varchar(15)     not null,
        dnumber         integer         not null,
        mgr_ssn         char(9)         not null,
        mgr_start_date  date,
        primary key (dnumber),
        unique (dname)
);
```

Microsoft SQL Server:

```
CREATE DATABASE 'CompanyElmasri';
USE [CompanyElmasri];
CREATE TABLE department (
        dname           varchar(15)     not null,
        dnumber         integer         not null,
        mgr_ssn         char(9)         not null,
        mgr_start_date  date,
        primary key (dnumber),unique (dname)
 );
```

The example shown above is DDL statements used for creating a database named 'CompanyElmasri' and a later statement is a command for activating the CompanyElmasri database using 'USE' keyword. The last statement is a command for creating a table named 'department'. Examples of the different DBMS syntaxes are as follows:

1) The USE Command:
   a. MariaDB, MySQL, and PostgreSQL: Specify '"'.
   b. Microsoft SQL Server: Specify '[ ]'.
2) Data type:
   a. MariaDB and MySQL: Size of integer must be specified.
   b. Microsoft SQL Server and PostgreSQL: Size of integer do not need to be specified.

The DML is a language for manipulating data. The DML statements are divided into 4 operations: Insert, Update, Delete, and Select. The Insert, Update, and Delete operations make a database change its state. Almost SQL statements can be written in many styles having the same SQL results as examples shown below:

1) Insert:

```
INSERT INTO department (dname, dnumber, mgr_ssn, mgr_start_date)
VALUES      ("Research", 4, "115533367", "2018-05-01");
is equivalent to
INSERT INTO department
VALUES      ("Research", 4, "115533367", "2018-05-01");
```

2) Update:

```
UPDATE employee, department
      SET dno=9
 WHERE dno=dnumber AND dname="Research";
is equivalent to
UPDATE employee INNER JOIN department ON dno=dnumber
      SET dno=9
 WHERE dname="Research";
```

3) Delete:

```
DELETE e
FROM    employee e INNER JOIN department d ON e.dno=d.dnumber
WHERE  d.dname="Research";
is equivalent to
DELETE FROM employee
WHERE  dno IN (SELECT dnumber
                    FROM department
                  WHERE dname="Research");
```

4) Select:

```
SELECT ssn, fname, lname, dname, mgr_start_date
  FROM employee, department
WHERE dno=dnumber;
is equivalent to
SELECT ssn, fname, lname, dname, mgr_start_date
  FROM employee INNER JOIN department ON dno=dnumber;
```

B. Grading SQL Statement

There are two kinds of grading SQL assignments: manual grading and automated grading. For both grading styles, instructors can evaluate assignment using two techniques: evaluate the SQL statement or evaluate outputs of the SQL statement. The first assessment technique is appropriate for SQL statements having only one style and syntax. The second assessment technique is suitable for SQL statements with a small number of outputs. However, some SQL statements can be written in many styles. Moreover, to practice many DBMSs, we cannot avoid multiple SQL statement styles and syntaxes. To support multiple DBMSs, the second assessment technique is selected to be implemented in this system. *The SQL grader* is designed to compare students' SQL results to test case generated for each SQL statement. As a result, the practice database used for a database class should be in a practical size to reduce the processing time.

2. Research Objectives

There were three research objectives in the study:

1) To develop an assignment management module for instructors to manage assignment information and student information,
2) To design a database to store assignment information and student information, and
3) To develop an automated SQL grader for grading students' SQL assignment automatically.

3. Research Methodology

Developing software program is similar to building a house. Construction engineer uses a house model to visualize a house concept. Later, a set of detailed blueprints is designed to represent much more specific information about the house. Finally, the house is built following the blueprints. For developing software, diagrams and User Interfaces (UIs) are very good tools for a software engineer to visualize a software system concept. A set of diagrams are later designed to represent details of the software system. Finally, the software programs are developed following concepts described in the diagrams and UIs.

In this research work, the system was designed following four fundamental phases of System Development Life Cycle (Dennis, Wixom, & Roth, 2012) or is called SDLC:

1) Planning: This phase focuses on "why we develop this system?" Project plan and break down of work plan are delivered. This phase is not described in this paper.
2) Analysis: This phase focuses on "Who, what, where, and when for this system?" Database instructors were interviewed and requirement definition in forms of business flow diagram, use case diagram, and UIs are delivered. These diagrams describe system main features. The UIs are used to visualize how users interact with application programs. These diagrams and UIs are used to

confirm with users. If users agree with these features and UIs then next phase is started.

3) Design: This phase focuses on "How will this system work?" The main features delivered in the analysis phase are later performed on the system design and followings are delivered: use case specifications, sequence diagrams, class diagram, and database schema.
4) Implementation: This phase focuses mainly on coding and software testing. Test plan and test results are delivered.

A technique used in the analysis and design phases are Object-Oriented Design (OOD). Analysis and design concepts are represented by Unified Modeling Language (UML) diagrams. Notations used in the UML diagrams are described in (Rumbaugh, Jacobson, & Booch, 1999).

A. Analysis Phase

Requirements can be gathered from many techniques such as questionnaire, document analysis, observation, interview, and so on. This system uses interview technique because the system is mainly developed to support database instructors. In addition, the interview technique is suitable for discovering facts and opinions held by users. Mistakes and misunderstandings can be identified and cleared up. The requirements captured from database instructors are:

1) Instructors can manage SQL assignments and solutions. For each question in the assignment, instructors can identify:
    a. Type of SQL operations: Insert, Update, Delete, or Select.
    b. Complication levels: Simple, Intermediate, and Advance.
2) Instructors can print each assignment in Microsoft Word file from the system.
3) Instructors can manage student information in their classes.
4) Instructors can view their student class scores in form of:
    a. Student individual report, and
    b. Student class scores.
5) The system must provide an automated SQL grader and the grader must support multiple DBMSs.
6) Students' scores must be computed and displayed immediately after students have submitted their assignment solutions.
7) Students can submit their assignment solutions many times up to the assignment due date, and their scores of all submitted versions are computed and stored in the database.
8) Students can view their individual scores.

Normally, *a use-case diagram* is delivered from the analysis phase if OOD technique is used. The use-case diagram represents user's interaction without flow of the system. However, the authors use the UI driven technique in this phase that requires the flow of the system. As a result, a business flow diagram represented in UML activity diagram is designed first to represent the system work flow. Three steps of the analysis phase are as follows:

1) Business Flow Diagram Design: Main features are extracted from the requirements and system work flow is analyzed and described in Figure 1(a). There are three groups of actors: instructor, student, and system. Actors are shown in their swim lanes and each lane consists of its own activities.
2) Use Case Diagram Design: The use-case diagram is easy to be generated from a business flow diagram. One activity is mapped to one use-case diagram as displayed in Figure 1(b).

3) UI Design: The UI(s) is/are designed for each use-case represented in an ellipse shape. One use-case may have more than one UIs. In this phase, the UIs are designed in form of a mock-up or a model of a design; program codes are not implemented yet. Examples of UI mock-ups are represented in Figure 2.

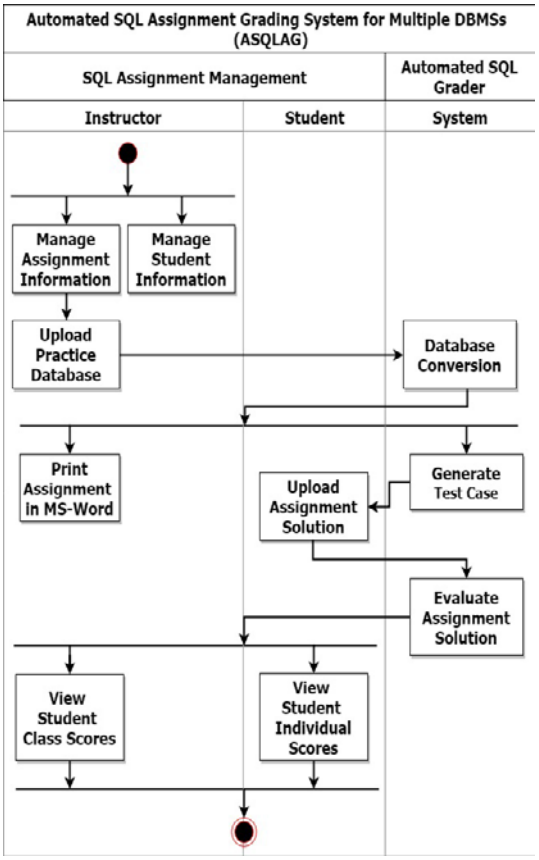
The Business flow diagram and all UI mock-ups are used to confirm with database instructors. It is good to get comments from users early from this phase because program codes will be implemented according to user requirements and their feedbacks. This will save time from modifying program codes at testing each phase or after the software is launched. Furthermore, the users can imagine what software will look like.

B. Design Phase

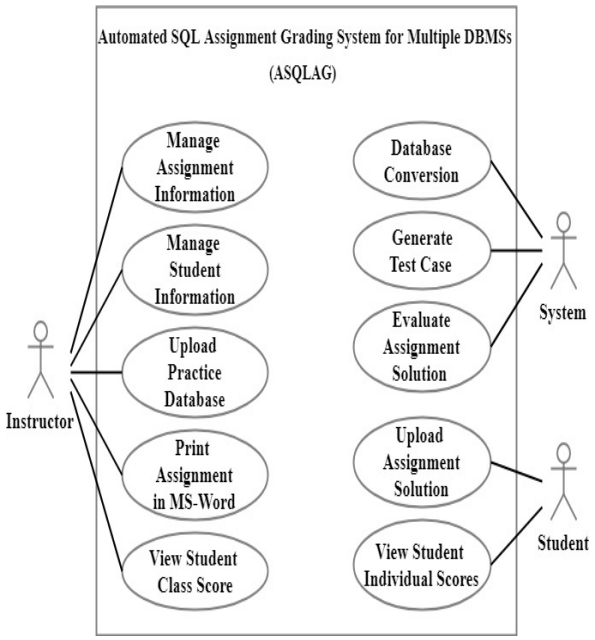
The design phase explains how the system operates with hardware, software, and network infrastructure. This phase delivers UIs, use-case specification, sequence diagram, class diagram, and database schema. The design phase has four steps (Dennis, Wixom, & Roth, 2012):

1) Determine design strategy: This step states whether the system will be developed by our own programmers, outsourced to a software house, or buy an existing software package. This system was developed by our own programmers.
2) Design system architecture: This step describes hardware, software, and network infrastructure that will be used in the system.
3) Design database: This step defines what data will be stored and where they will be store.

Figure 1: Business Flow Diagram and Use Case Diagram of ASQLAG**.**

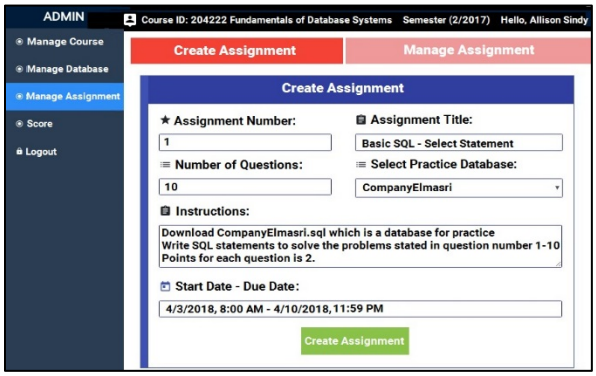


(a) Business Flow Diagram

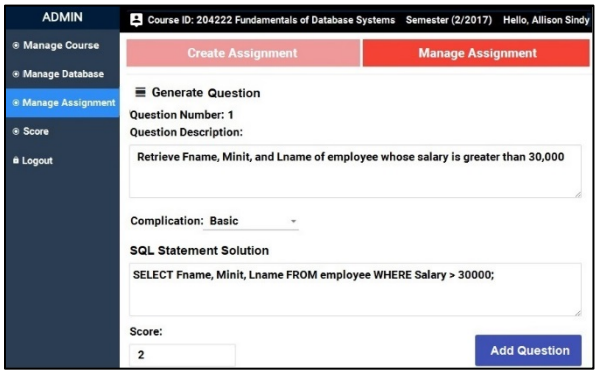


(b) Use Case Diagram

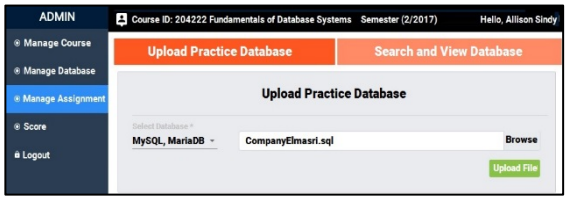
Figure 2: Examples of ASQLAG UI Mock-ups



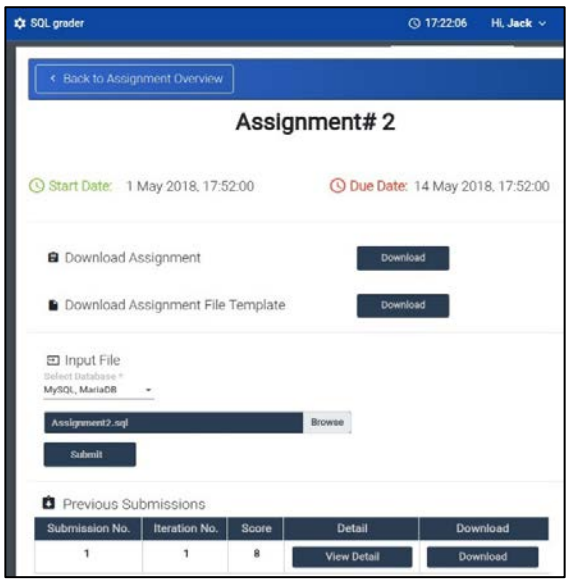
(a) Assignment Management: Create Assignment



(b) Assignment Management: Add Question



(c) Upload Practice Database and Database Conversion



(d) Upload Assignment and Evaluate Assignment

4) Design programs: This step defines the programs that need to be written, and describes what each program will do.

The OOD technique and Model-View-Controller (MVC) framework (Kalelkar, Churi, & Kalelkar, 2014) is used in the program design step. The authors ship the program design to perform before database design because using OOD technique with MVC framework will deliver a class diagram containing model classes. Most of model classes are classes with data to be stored on a database. The model classes then can be used as inputs of the database design step. After the program design step being completed, UIs, use case specifications, sequence diagrams, and a class diagram are delivered. The class diagram delivered from this step consists of three types of classes: model class, view class, and controller class. These classes handle their specific roles. Changing model class will not affect each other and vice versa. The model classes are later sent to perform database design. The roles of these classes are:

1) Model class: this class is for manipulating data.
2) View class: this class is for manipulating UI. It accepts user input data, and displays data sent back from the controller class.
3) Controller class: this class interacts with the view class and the model class. It processes user input data sent from the view class, executes appropriate appli-

cation logic, sends request to retrieve necessary model data, and returns appropriate responses to the view class.

C. System Architecture

The system is developed as a web application for users' convenience. Users can perform their tasks anytime and anywhere via the Internet network. There are two groups of users: instructors and students. They perform their requests via web browser. Instructors can manage their assignment and student information. Students can submit their assignment solutions and the solutions are graded by the automated SQL grader on an application server. Finally, student scores are computed and stored in the Assignment database on a database server. Students and instructors can view their individual scores and student class scores, respectively. The system architecture is displayed in Figure 3. Three-tier architecture is selected to implement the system because the functional processes are developed and maintained independently on separate platforms as follows:

1) Presentation Layer: Users submit requests on client sides thru web browsers and they are later sent to be processed on a web server. Three technologies used for developing client-side scripts on a presentation layer:
    a. Hypertext Markup Language (HTML): HTML is a language used for creating a web page. Data on a web browser are presented in form of HTML tags: text, image, table, link, and so on.
    b. Cascading Style Sheet (CSS): CSS is a language used to describe how HTML elements should be displayed on a web browser including colors, layout, and fonts. It is used to control look and feel of the contents written in the HTML.
    c. AngularJS (Chauhan, 2015): AngularJS is a structural framework for developing dynamic web application. It has a set of ready-to-use modules to build a system as Single Page Application. The SPA (Jadhav, Sawant, & Deshmukh, 2015) loads HTML, JavaScript, and CSS in initial request and will reload only necessary components related to user interaction without reloading the whole web page. As a result, a web page will be loaded and reloaded faster.

2) Application Layer: This layer is at a server side to process user requests sent from web browsers and return results to the clients. The application is run on Ubuntu Operating System (OS), which is a Linux OS. Four technologies are used for developing server-side scripts on an application layer:
    a. Express.js and Node.js (Brown, 2014; Mills, 2018): Express.js is an open source server-side JavaScript framework and hosted within the Node.js runtime environment. The Express.js provides libraries for building web applications and Application Program Interfaces (APIs). The Node.js allows developers to create server-side tools and applications in JavaScript to run on an application server. It is designed to optimize throughput and scalability in web applications.
    b. Knex.js: It is a SQL query builder for JavaScript and Node.js. It is used to build SQL query for PostgreSQL, Microsoft SQL Server, MySQL, and MariaDB. The Knex.js handles SQL statements in json format. For more information of the Knex.js is available at "https://knexjs.org/".

c. SQLines: The SQLines is an open source tool to migrate databases. It is used to perform SQL syntax conversion. Both DDL and DML statements of an original practice database file uploaded by instructors are converted to other three database files to cover SQL syntaxes of four DBMSs: MariaDB, MySQL, PostgreSQL, and Microsoft SQL Server. For more information of the SQLines, it is available at "http://www.sqlines.com".

3) Data Layer: This layer provides data access and data storage which is a relational database. There are two groups of databases in this system:

a. Assignment Database: This database is for storing assignment and student score information. The Assignment database is managed using PostgreSQL RDBMS.

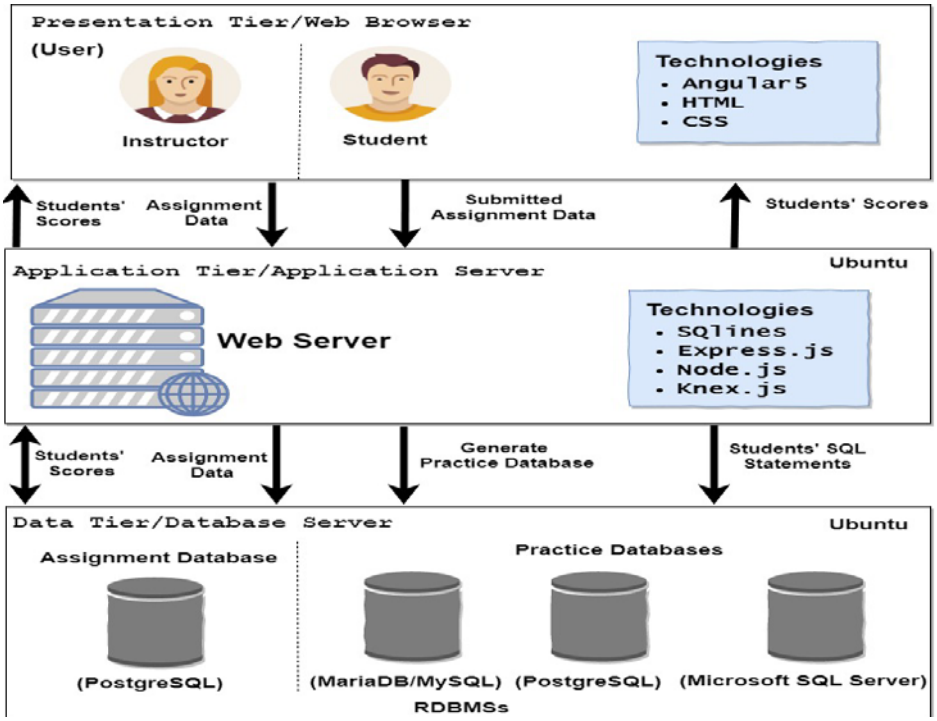
b. Practice Databases: These databases are uploaded by instructors for students to practice their assignments. Students can submit their assignment solutions in one of four SQL statement syntaxes mentioned before. There are only three RDBMSs implemented in this system: MariaDB, PostgreSQL, and Microsoft SQL Server. This is because MariaDB can also support MySQL statement syntax.

D. Program Design

A target of program design using OOD technique with MVC framework is a class diagram containing model, view, and controller classes. The inputs of this phase are use-cases in the use-case diagram together with their UI mock-ups. Each use-case and its UI mock-up(s) are further analyzed. Use-case specifications, sequence diagrams, and a class diagram are delivered from this phase. Using OOD technique, all use-cases can be analyzed parallel if we have many people in a software development team. Each use-case is performed in the following steps:

1) The UI mock-up(s) of each use case delivered from analysis phase is/are adjusted to fit input devices specified on the system architecture.
2) Use case specification is designed for each use case. An important part of use case specification is flow of events. This flow describes what tasks will be performed for each use case. The flow of events normally consists of input, process, and output. The authors use elements and flow of UI as a guide line to design flow of events.

Figure 3: ASQLAG System Architecture



3) Sequence diagram is later designed for each use-case specification. The sequence diagram describes interaction between classes and order of class interaction. The sequence diagram is guided by flow of events specified in the use-case specification. As mentioned above, the MVC framework is used in this phase. The user inputs are handled by view classes. The controller classes handle data processing such as data validation, data computation, and so on. The model classes handle data manipulation. Order of class interaction is designed following the flow of events specified in the use case specification.
4) Classes expressed in the sequence diagram are delivered. Methods of each class are collected from the request messages sent to it.

The steps shown below are an example of the program design for the Assignment Management use-case.

1) Assignment Management UI mock-up(s) displayed in Figure 2(a) and 2(b) are input of this step. It is adjusted to fit a web browser.
2) Use case specification is designed according to elements and flow described on the UI and the use-case specification is delivered as displayed in Table 1. Only the use-case specification of create assignment operation is shown.
3) Sequence diagram is designed following the flow of events specified in the use-case specification as described in the step 3 above. The sequence diagram is delivered as displayed in Figure 4.
4) Classes and their methods are collected from the sequence diagram as displayed in Figure 5.

The rest of use-cases are also performed using the same example steps specified above. After all use-cases being completed, all classes in all sequence diagrams are collected and summarized in the class diagram displayed in Figure 6. The algorithms of the rest use cases are described below:

a. Student Information Management:
   1) Instructor inputs each student information.

2) The student information is validated and only valid data are stored in the Assignment database.
3) The system repeats two steps above until all student information being added.
4) Instructor can upload student information in Microsoft Excel file format to the system.
5) The system also provides functions for update and delete student information.

b. Upload Practice Database and Database Conversion:

1) For each assignment, an instructor uploads a practice database file with SQL file extension in one of supported four DBMS syntaxes.
2) The original database file is saved on the application server and it is imported to the database server. Only the file with error-free is allowed to be saved and imported.
3) The system converts the original database file to the rest DBMS syntaxes using the SQLines. The SQLines converts the original DBMS syntax to target DBMS syntaxes as follows:
   - Between MariaDB or MySQL and Microsoft SQL Server: Identifiers and data types are converted. For more information, please see "http://www.sqlines.com/mysql-to-sql-server#identifiers".
   - Between Microsoft SQL Server and PostgreSQL: data types are converted.For more information, please see "http://www.sqlines.com/sql-server-to-postgresql".
4) The other three database files are generated, saved on an application server, and imported to a database server.

c. Generate Test Case:

1) Instructors specify the assignment that they want to generate test cases or SQL statement results.
2) The system checks what DBMS that the instructor specifies for the assignment.
3) The system reads SQL statement solution of each question.
4) For a SQL statement solution with Insert, Update, or Delete command, the system adds more SQL statements to obtain data that will be modified instead of performing these commands. This criterion is to keep a database in the original database state. The obtained data will be saved as a test case in the next step and the system cancels to change data in a database. The SQL statement to be added for each DBMS is different:
   4.1) PostgreSQL, MariaDB, and MySQL: "RETURNING" is used. For more information, please see "https://www.postgresql.org/docs/9.5/static/dml-returning.html" and "https://mariadb.com/kb/en/library/delete/".
   4.2) Microsoft SQL Server: OUTPUT Clause is used (Laudenschlager, Jenks, Guyer, Twizzes, & Petersen, 2018). More information is available at "http://www.sqlservercentral.com/articles/ T-SQL/156204/".
5) The SQL statement solution is sent to execute on a database server. The SQL results returned from the server are saved as a test case in json format in a text file for later use by the Evaluate Assignment Solution use-

case. An example of json format is available at "https://json.org/example.html".

6) The system performs the rest questions until all test cases of the assignment are generated.

d. Upload Assignment Solution and Evaluate Assignment Solution:

1) Students upload their assignment solutions. Their assignment solutions can be one of four DBMS syntaxes supported by the system.
2) The system reads SQL statement solution of each question submitted by an instructor from the Assignment database and determines what type of SQL operation.
3) If the SQL operations are Insert, Update, and Delete, the SQL statement solution submitted by a student is appended more SQL statements similar to the step described in c(4) above.
4) Each SQL statement solution is sent to execute on a database server. The system gets outputs of SQL statement in json format and the outputs are called student outputs in short.
5) The student outputs are later compared to the test case of each question saved in json file mentioned in c(5). The Select statement with ORDER BY is validated different from the others, both contents and the order of output data are validated.
6) The system computes scores and student scores are saved in the Assignment database. If students submit each assignment more than once, their scores of all assignment versions are also computed and stored in the Assignment database. An example below displays possible scores that students may achieve for each question. If students do not submit their SQL statement solutions, the score for that question is 0 and the score description is "Not Submit".
   - Question: Retrieve Fname, Minit, and Lname of employees whose salary is greater than 30, 000.
   - Question Type: Select operation
   - Complication Level: Basic
   - 1st Student: Scores = 2, Score Description = Correct
     ```
     SELECT Fname, Minit, Lname FROM employee
     WHERE salary > 30000;
     ```
   - 2nd Student: Scores = 0, Score Description = Syntax Error
     ```
     SELECT Fname, Minit, Lname FROM employees
     WHERE salary > 30000;
     ```
     Note: It should be employee (without "s")
   - 3rd Student: Scores = 0.5, Score Description = Logical Error
     ```
     SELECT Fname, Minit, Lname FROM employees
     WHERE salary >= 30000;
     ```
   - Note: It should be "salary > 30000".

e. View Student Individual Scores:

1) Students select assignment number that they want to see their scores.
2) The system retrieves last version of student score information of the selected assignment from the Assignment database.

3) The student score information is shown on screen and students can print their score information as shown in Figure 8(b).

Table 1: Manage Assignment Use-Case Specification

Use case ID	UC01
Use case Name	Manage Assignment
Actor/User	Instructor
Description	Create and manipulate assignment and its questions.
Pre-condition	-
Post-condition	Assignment and question information are stored in the Assignment database.
Flow of events	1. Manage Assignment Screen is loaded. 2. User selects operations to be performed. 2.1. If user chooses create assignment, then the Create Assignment screen is loaded as displayed in Figure 2(a). 2.1.1. User inputs assignment data: assignment number, assignment title, number of questions, practice database name, assignment start date, and assignment due date. 2.1.2. User presses the Create Assignment button, then the assignment data are validated as follows: - Length of assignment title must not exceed 80 characters. - Length of practice database name must not exceed 30 characters. - Assignment number and number of questions must be positive integers. - If all input data are valid the screen for input question data is loaded as shown in Figure 2(b). 2.1.3. User adds question data until all question data of the assignment are added to the Assignment database: - Question number is automatic generated by the system starting from 1. - User inputs other question data: question description, complication level, SQL statement solution, and question score. - User presses the Add Question button. The submitted data are later validated as follows: • Length of question description must not exceed 256 characters. • Question score must be positive integer. • Increasing question number by 1. 2.2 If user chooses manage assignment, then the Manage Assignment screen is loaded.
Alternate Flow	2.1.2.1 If there are any invalid data, the system displays error message and forces user to re-enter assignment data. 2.1.3.1 If there are any invalid data, the system displays error message and forces user to re-enter question data.

Figure 4: Assignment Management Sequence Diagram: Create Assignment

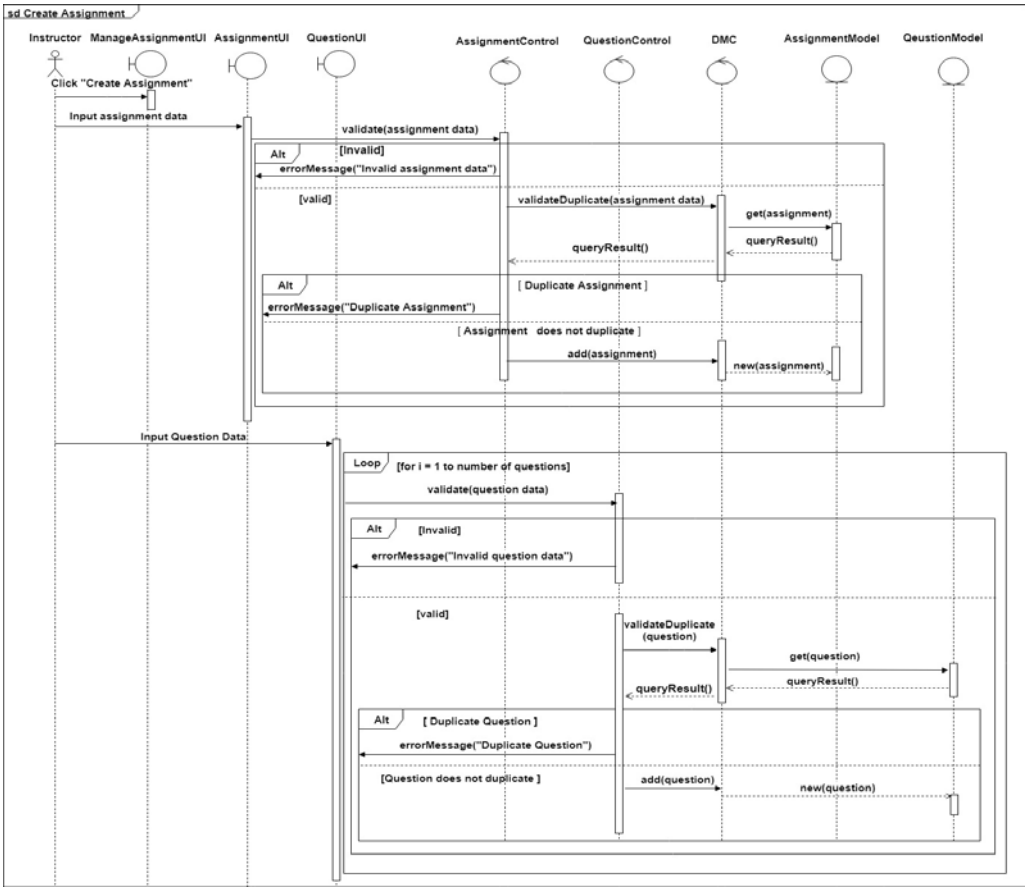
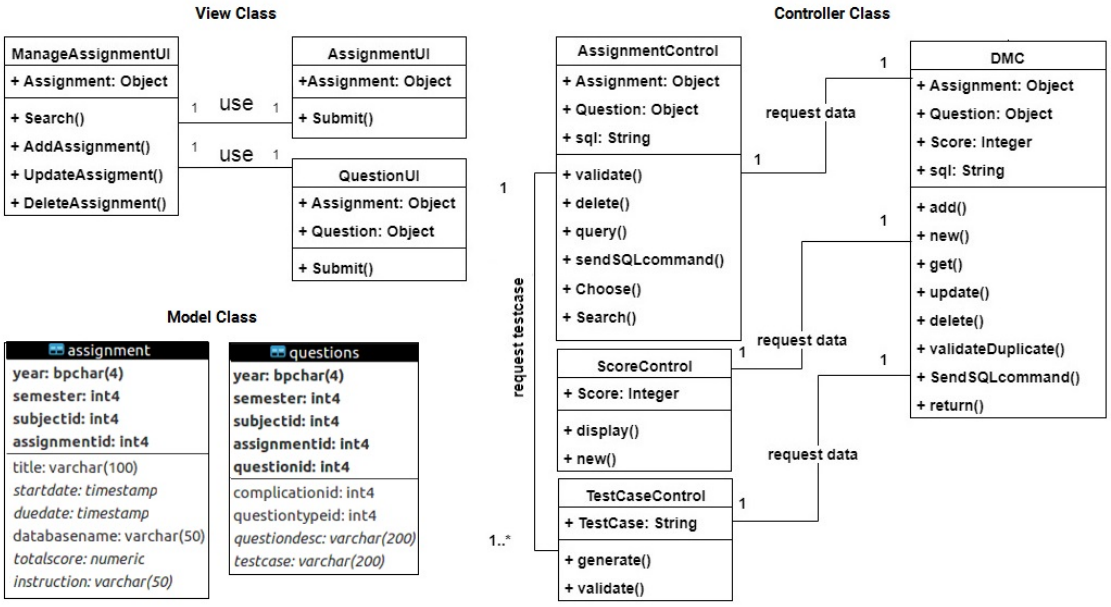


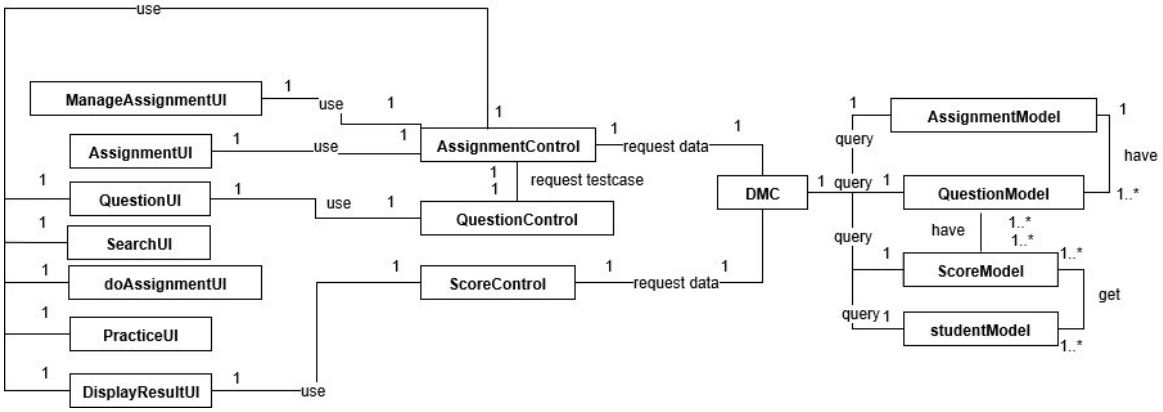
Figure 5: Classes Related to Assignment Management Use Case



f. View Student Class Scores:

1) Instructors select sections and assignments that they want to see their class scores.
2) The system retrieves all score information of the students enrolled in the selected sections and last version scores of selected assignments from the Assignment database.
3) The system counts percent of students' scores group by section, assignment, SQL operation type, and score description as an example shown in Figure 8(c).
4) The student class score information is shown on screen and instructors can print this information as displayed in Figure 8(c).

Figure 6: System Class Diagram



E. Database Design

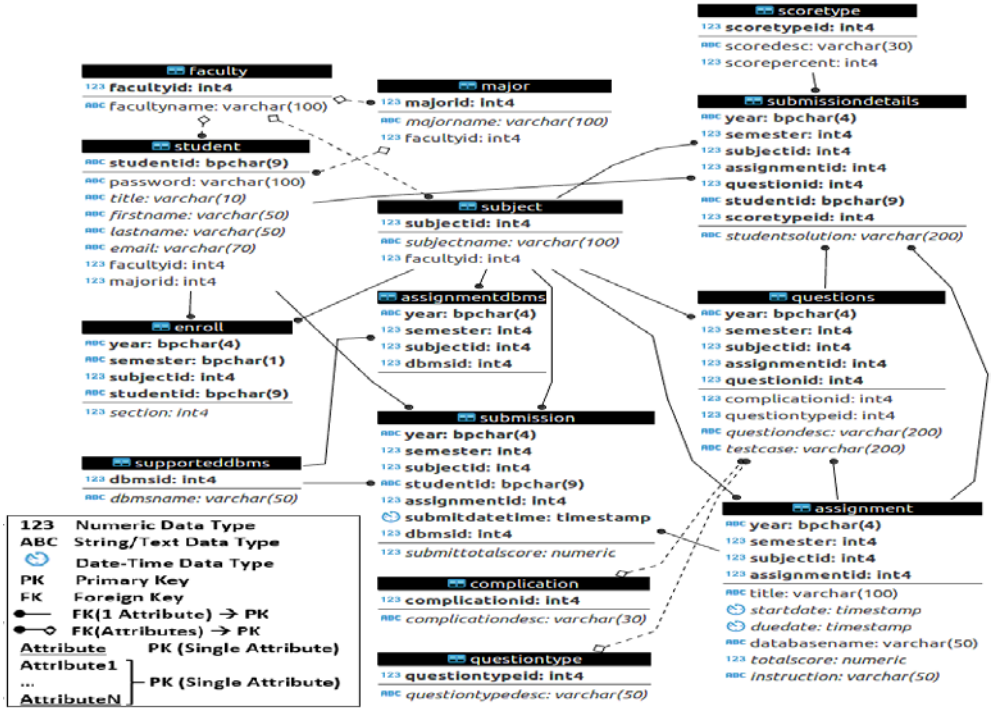
A database named Assignment is designed to store data related to assignment data and student scores. The model classes delivered from program design step are later performed on the detail design. The model classes to be stored on a database are collected and relationships between them are analyzed. The ER-to-relational mapping and normalization of relations (Elmasri, & Navathe, 2011) are later used to map these classes to relations because these data are stored in a relational database. The relations are normalized to the third normal form. The structure of database or called *database schema of the Assignment database* is displayed in Figure 7. The important terms related to assignment information are from the following tables:

1) complicationdesc: This attribute is from the complication table. It describes complication level of each question. The possible values are: basic, intermediate, and advance.
2) questiontypedesc: This attribute is from *the question type table.* It describes question type of each question. The possible values are: insert, update, delete, and select.
3) scoredesc: This attribute is from *the score type table*. It describes type of achievement score for each question. The possible values are: not submit, syntax error, logical error, and correct.
4) scorepercent: This attribute is from *the score type table.* It is a student achievement score of each question in percentage ranging from 0 to 100 percent.

5) Implementation Phase

The system is implemented and the unit test of each program is performed using black box testing. More details of the black box testing are available at "http://softwaretestingfundamentals.com/black-box-testing/". After all programs are passed after the unit test, the integration test, system testing, and acceptance testing are performed, respectively. More details about these tests are available at "http://softwaretestingfundamentals.com/integration-testing/". Later, the programs are deployed on the application server. The details of this phase are not described in this paper.

Figure 7: Assignment Database Schema

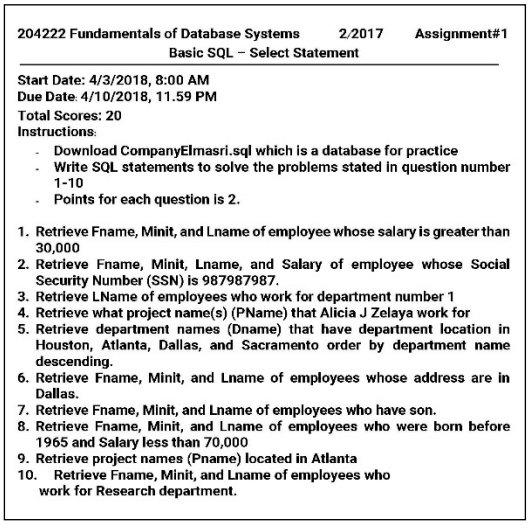


4. Results and Discussion

The system was run for semester 2 of academic year 2017. The results reveal that grading time is reduced from many minutes to a few seconds as an example shown in Figure 8(b). The assignment document in Microsoft Word format is printed conveniently from the system after they have finished entering assignment information as an example displayed in Figure 8(a). Furthermore, students and instructors can view their individual scores and student class scores, respectively as displayed in Figure 8(b) and 8(c).

The system can be applied to grade other subjects such as the subjects using multiple choice answer or short answer. The system is just a bit modified the Assignment database to keep answer in a, b, c, d or 1, 2, 3, 4 and the automated SQL grader is replaced by the program that checks these answers instead.

Figure 8: Example Results of the ASQLAG System



204222 Fundamentals of Database Systems 2/2017 Assignment#1

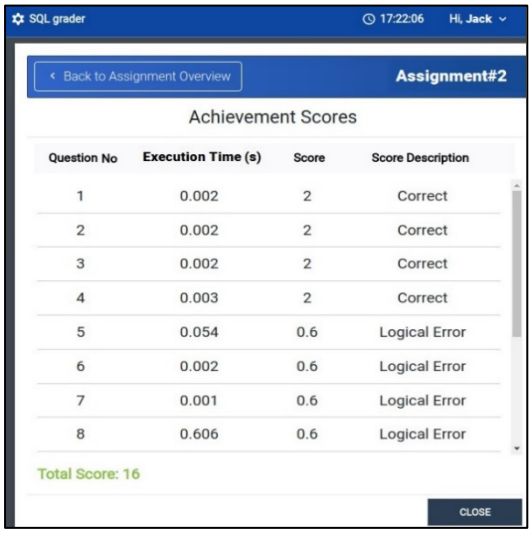
Basic SQL – Select Statement

Start Date: 4/3/2018, 8:00 AM
Due Date: 4/10/2018, 11.59 PM
Total Scores: 20
Instructions:

- Download CompanyElmasri.sql which is a database for practice
- Write SQL statements to solve the problems stated in question number 1-10
- Points for each question is 2.

1. Retrieve Fname, Minit, and Lname of employee whose salary is greater than 30,000
2. Retrieve Fname, Minit, Lname, and Salary of employee whose Social Security Number (SSN) is 987987987.
3. Retrieve LName of employees who work for department number 1
4. Retrieve what project name(s) (PName) that Alicia J Zelaya work for
5. Retrieve department names (Dname) that have department location in Houston, Atlanta, Dallas, and Sacramento order by department name descending.
6. Retrieve Fname, Minit, and Lname of employees whose address are in Dallas.
7. Retrieve Fname, Minit, and Lname of employees who have son.
8. Retrieve Fname, Minit, and Lname of employees who were born before 1965 and Salary less than 70,000
9. Retrieve project names (Pname) located in Atlanta
10. Retrieve Fname, Minit, and Lname of employees who work for Research department.

(a) An example of assignment document in Microsoft Word format.



SQL grader 17:22:06 Hi, Jack

Back to Assignment Overview Assignment#2

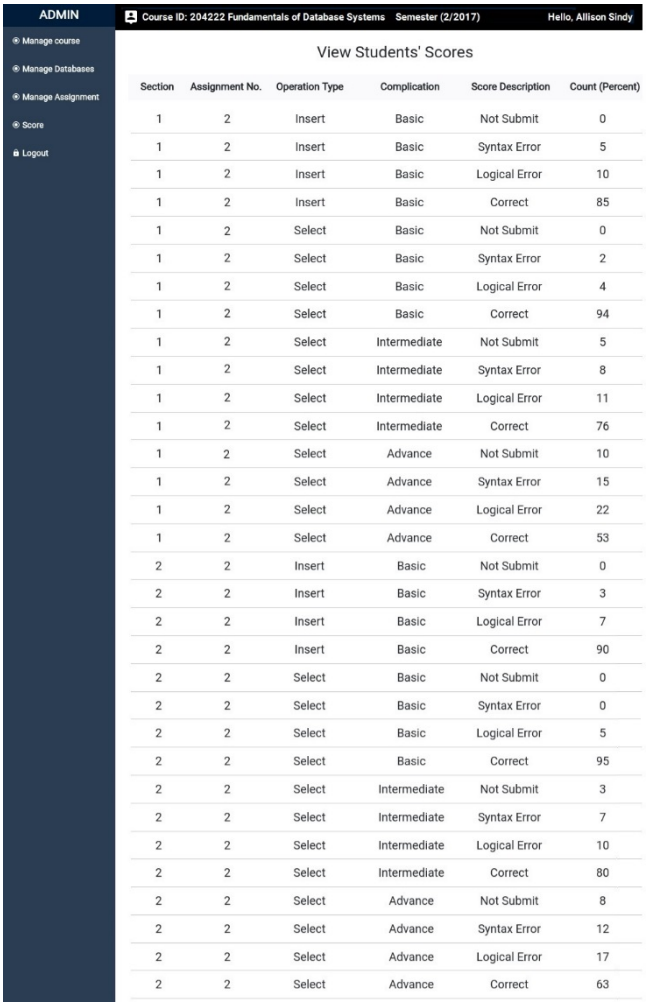
Achievement Scores

Question No	Execution Time (s)	Score	Score Description
1	0.002	2	Correct
2	0.002	2	Correct
3	0.002	2	Correct
4	0.003	2	Correct
5	0.054	0.6	Logical Error
6	0.002	0.6	Logical Error
7	0.001	0.6	Logical Error
8	0.606	0.6	Logical Error

Total Score: 16

CLOSE

(b) An example of Student Individual Scores.



ADMIN

- Manage course
- Manage Databases
- Manage Assignment
- Score
- Logout

Course ID: 204222 Fundamentals of Database Systems Semester (2/2017) Hello, Allison Sindy

View Students' Scores

Section	Assignment No.	Operation Type	Complication	Score Description	Count (Percent)
1	2	Insert	Basic	Not Submit	0
1	2	Insert	Basic	Syntax Error	5
1	2	Insert	Basic	Logical Error	10
1	2	Insert	Basic	Correct	85
1	2	Select	Basic	Not Submit	0
1	2	Select	Basic	Syntax Error	2
1	2	Select	Basic	Logical Error	4
1	2	Select	Basic	Correct	94
1	2	Select	Intermediate	Not Submit	5
1	2	Select	Intermediate	Syntax Error	8
1	2	Select	Intermediate	Logical Error	11
1	2	Select	Intermediate	Correct	76
1	2	Select	Advance	Not Submit	10
1	2	Select	Advance	Syntax Error	15
1	2	Select	Advance	Logical Error	22
1	2	Select	Advance	Correct	53
2	2	Insert	Basic	Not Submit	0
2	2	Insert	Basic	Syntax Error	3
2	2	Insert	Basic	Logical Error	7
2	2	Insert	Basic	Correct	90
2	2	Select	Basic	Not Submit	0
2	2	Select	Basic	Syntax Error	0
2	2	Select	Basic	Logical Error	5
2	2	Select	Basic	Correct	95
2	2	Select	Intermediate	Not Submit	3
2	2	Select	Intermediate	Syntax Error	7
2	2	Select	Intermediate	Logical Error	10
2	2	Select	Intermediate	Correct	80
2	2	Select	Advance	Not Submit	8
2	2	Select	Advance	Syntax Error	12
2	2	Select	Advance	Logical Error	17
2	2	Select	Advance	Correct	63

(c) An example of Student Class Scores.

5. Conclusion and Future Work

The Automated SQL Assignment Grading System was designed and developed to speed up grading time. The system serves as a web application for user convenience, users can perform their tasks anytime and anywhere via internet network. Instructors can manage their assignment and student information, and students can submit their assignment solutions many times up to the due date. Student assignments are graded by the automated SQL grader and scores are computed and displayed immediately after students submit their assignment solutions. If students obtain low scores, they can practice more and re-submit each assignment as many times as they want up to the due date. The scores of all submitted versions are computed and stored on a database for later used by students and instructors. For each assignment, the scores of the whole class are very useful for instructors to improve their teaching performances. The automated SQL grader is designed to evaluate results of SQL statement. Therefore, the database used for practice must be a practical size.

The researchers expected that the system would continue to run in the next semester to gather more comments and requirements from database instructors.

It should be noted that *grading only SQL outputs* is not applicable for the SQL with comparison operators like <, <=, >, and >=. Two SQL statements below may have the same results if there are not any employees who have salary equal to 30,000 in the database. The conditions of SQL statement must be considered.

```
SELECT Fname, Minit, Lname FROM employee WHERE salary > 30000;
SELECT Fname, Minit, Lname FROM employee WHERE salary >= 30000;
```

The system is further designed to cover a practice module. In this module, SQL statement is broken down into sub-statements and each sub-statement is analyzed. The incorrect sub-statement submitted by a student is displayed with suggestions. Such a module shown in the study can be applied to grading SQL assignment in terms of partial marking, as preferred or seen appropriate by some instructors.

6. Acknowledgement

We would like to thank the database instructors of Computer Science Department, Faculty of Science, Chiang Mai University for providing requirements and good comments.

7. The Authors

Areerat Trongratsameethong is a faculty staff member of Computer Science Department, Faculty of Science, Chiang Mai University. She received her Ph.D. in Computer Science from the Mahidol University, Bangkok in 2010. She worked in business company and software house for many years and she gained experiences in software development, system analysis and design, and software requirement analysis before undertaking her doctoral studies. She has been teaching at Chiang Mai University since 2010. Her courses include Fundamentals of Programming, Fundamentals of Database System, Object-Oriented Design, and Ontology Design and Development.

Pakpoom Vichianroj is a software engineer at Ascend Commerce Group Co., Ltd. He received a Bachelor of Science degree in Computer Science from Chiang Mai University, Chiang Mai in 2018.

8. References

Brown, E. (2014). "Introducing Express". *Web Development with Node and Express*. California: O'Reilly, 2-4.

Chauhan, R. 6 Reasons Why Angular JS Should be Used for Development. (Online). https://www.softwebsolutions.com/resources/why-angularjs-should-be-used-for-development.html, May 1, 2018.

Dennis, A., Wixom, H. B., & Roth, M. R. (2012). "The Systems Development Life Cycle". *System Analysis and Design*. 5th Edition. New Jersey: Prentice-Hall, 2-18.

Elmasri, R. & Navathe, B. S. (2011). "SQL". *Fundamentals of Database Systems*. 6th Edition. Massachusetts: Addison-Wesley, 87-139, 285-294, 501-528.

Jadhav A. M., Sawant, R. B., & Deshmukh A. (2015). Single Page Application using AngularJS. *IJCSIT*. *6*(3), 2876-2879.

Kalelkar, M., Churi, P., & Kalelkar, D. (2014). Implementation of Model-View-Controller Architecture Pattern for Business Intelligence Architecture. *International Journal of Computer Applications. 102*(12), 0975 – 8887.

Laudenschlager, D., Jenks, A., Guyer, C., Twizzes, & Petersen T. OUTPUT Clause (Transact-SQL). (Online). https://docs.microsoft.com/en-us/sql/t-sql/queries/output-clause-transact-sql?view=sql-server-2017, May 24, 2018.

Mills, C. Express/Node Introduction.(Online). https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction, June 20, 2018

Rumbaugh J., Jacobson I., & Booch G. (1999). "Appendix B: Notation Summary". *The Unified Modeling Language Reference Manual: UML*. Massachusetts: Addison-Wesley, 519-529.