2019 IEEE 6th International Conference on Industrial Engineering and Applications

Interactive SQL Learning Tool with Automated Grading using MySQL Sandbox

Tanasab Permpool, Srinual Nalintippayawong and Kanokwan Atchariyachanvanich
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, 10520 Thailand
e-mail: 60606063@kmitl.ac.th, srinual@it.kmitl.ac.th, kanokwan@it.kmitl.ac.th

***Abstract*—This research presents an original tool for automatic grading of SQL statements using MySQL Sandbox[1]. This tool allows students to create and execute the queries in real time. There are many ways to write accurately SQL statement for one question. This is a pain point and burden to the instructors. The existing automated SQL grading algorithms have limitations on supportable statements and understandable error notification. The existing automated SQL grading tool did not support grading on data manipulation language statements i.e. INSERT, UPDATE and DELETE and do not support data definition language commands also; i.e. CREATE, ALTER and DROP. This paper presented an original algorithm which can support these commands. The proposed algorithm can analyze and display the differences between student's answer and correct answer. This algorithm can reduce the instructors' burdens by automatic scoring students' answers and the results directly notify back to both instructors and students. The students can learn and improve themselves faster and effectively.**

Keywords-database; grading system; SQL statement; automated grading; data definition language(DDL) ; data manipulation language(DML)

I. INTRODUCTION

The Online grading system has been developed continuously along with e-learning system. It has been developed to integrate various forms of instruction ranging from regular questions such as fill-in question and multiple-choice question to advanced questions such as programming-based question which require students to solve problems and the online results are displayed instantly. Teaching a Structured Query Language (SQL) in basic database course considers the importance of the online SQL grading system as many educational institutions have adopted it.

Database course requires students to practice SQL commands in two main subgroups i.e. Data Manipulation Language (DML) and Data Definition Language (DDL). For example, SELECT statement in the DML is to retrieve the desired records. CREATE statement in the DDL is to create a database table. However, the existing automated grading systems such as SQLator[2], SQLify[3], SQL-Lab[4] and aSQLg[5] still have limitation because these tools can do grading for only SELECT statements. Thus, they cannot do grading on DDL statement which are CREATE, ALTER and DROP. This limitation involves security issue because it requires external commands to directly execute in the system.

An important objective of teaching SQL is to encourage students to learn SQL statement writing which can affect data in the database table and can change the structure of the database. The visibility makes them understand the SQL statement workflow and processes.

This paper aims to develop an interactive learning tool which operates an original automated grading SQL statement and deploy the tool in the DBLearn [6]. This tool can reduce the effort of instructors and eliminate the necessity in examining individual student's answers. Moreover, it also helps students to solve SQL questions and gain the correct answers faster. In addition, this tool can represent error description to make students understand their mistake which help improving their learning effectiveness.

II. RELATED WORK

A. Existing SQL Grading Systems

The automated SQL grading systems e.g. SQLator[2], SQLify[3], SQL-Lab[4] and aSQLg[5] can do scoring only on the SELECT statement. Only Ke et al.'s work[7] and XDa-TA[8] can do scoring on other DML statements i.e. INSERT, UPDATE and DELETE. However, these systems do not use the SQL statement to be directly executed in the database. They are converted into a SELECT statement before doing grade by using the approach same as the regular SELECT statement, For example

DELETE FROM users WHERE name LIKE "Som%";

can be converted to

SELECT * FROM users WHERE name LIKE "Som%";

The statement is correct when the result set from the converted statement is same as the database changed when using original statement. However, this conversion method is not applicable to complex statement. For example, the retrieving values be stored in variables, the updating values are based on data from other tables as follows.

```
UPDATE table1 INNER JOIN table2
ON table1.id = table2.id
SET table1.name = table2.name;
```

Furthermore, this method works with the grading process only. It cannot be implemented as an interactive tool for students to see data changed immediately.

B. MySQL Sandbox

The existing automated SQL grading systems check the result set of the executed SQL statement. Therefore, the system demands to receive external SQL statements from

978-1-7281-0851-3/19/$31.00 ©2019 IEEE

students. This demanding is a concern about security issues such as SQL-Injection or execution of statement which creates impacts on the database schema and data set. To prevent the issues, a number of available commands is limited to only SELECT which does not affect the database. The solution of these problems is to separate and isolate each student's database by MySQL Sandbox [1] which is developed to solve the problems. There are two basic principles:

- The sandbox has to support concurrent users for a question and each user can execute SQL statements without affecting other users in the same time.
- The students can use multiple databases at the same time without impact to irrelevant databases.

MySQL Sandbox focuses on the interactive SQL learning which can represent the statement process on the database to students in real time. It is the foundation of the development of an automated SQL grading system which has capability of handling DDL statement and DML statements which are declared as the limitation in Section II.A

III. AUTOMATED GRADING ALGORITHM

A. *Automated Grading for DML Statement (INSERT, UPDATE and DELETE)*

An original algorithm for grading DML statements including the INSERT, UPDATE, and DELETE statements is displayed in Figure 1. The algorithm starts with a list of target tables from both instructor and student statements which are extracted from statements of instructor and student respectively. Then, databases storing the same data sets are separated for executing statements of student and instructor. Later on, all records are retrieved from target tables in each database. The result sets from student's database are compared with the result sets from instructor's database to check the correctness. The result sets may be include probably the same records but their sequences are different. For grading DML statement, the algorithm ignores the sequence of the result rows. Therefore, it is necessary to retrieve and compare the sorted rows. The student statement is correct when the result set of the instructor's and student's tables after the execution are the same. In other hand, the student statement is incorrect if the comparison result is difference. The tool notifies the table names with an error occurred. The error notification is represented in TABLE I.

B. *Automated Grading for DDL statement*

This paper describes a new grading algorithm which is developed for DDL statements, as shown in Figure 2. The overall process is to compare the database schema and ignore the stored data. Databases having the same schema are separated for executing statements of student and instructor. The tool executes both instructor and student statements in separated databases then table names in database are listed by using command SHOW TABLES;. The table list of student database is compared with that of instructor database by checking extra tables or absent tables. Each table schema is compared by using command DESCRIBE <table_name>; to get the table structure e.g. column names, data types, data lengths and constraints of each column. Since the DESCRIBE command cannot display a relationship between tables, the command below is used to find the relationship between the tables from the foreign key. Finally, the difference of the foreign key from all tables in instructor and student databases are compared.

TABLE I. ERROR NOTIFICATION FROM GRADING DML STATEMENT

Error Notification	Explanation
<table_name> :Incorrect	The data in the table is incorrect.

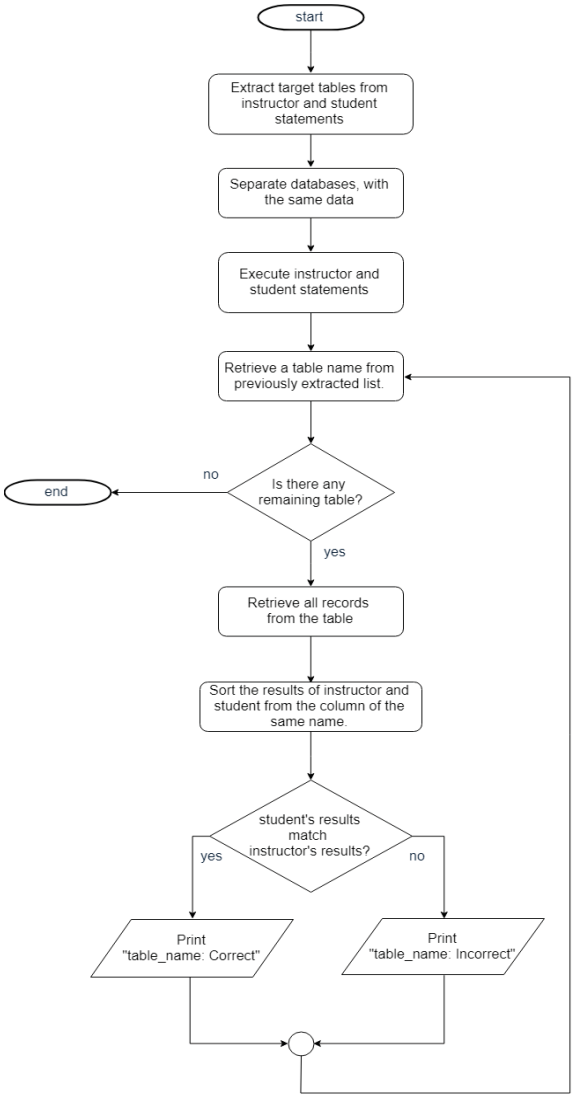

Figure 1. Automated grading algorithm for DML statement (INSERT, UPDATE and DELETE)

SELECT column_name, referenced_column_name, table_name, referenced_table_name
FROM information_schema.key_column_usage
WHERE referenced_column_name **IS NOT NULL ;**

The result set which is generated from above query is converted to a database schema in JSON format as Figure 3. The deep-diff, a plug-in for structural comparison of JSON data [8], is applied in this tool. The deep-diff result from the student's statement is simplified into an easily readable notification to be presented to the student. The errors can be summarized in TABLE II.

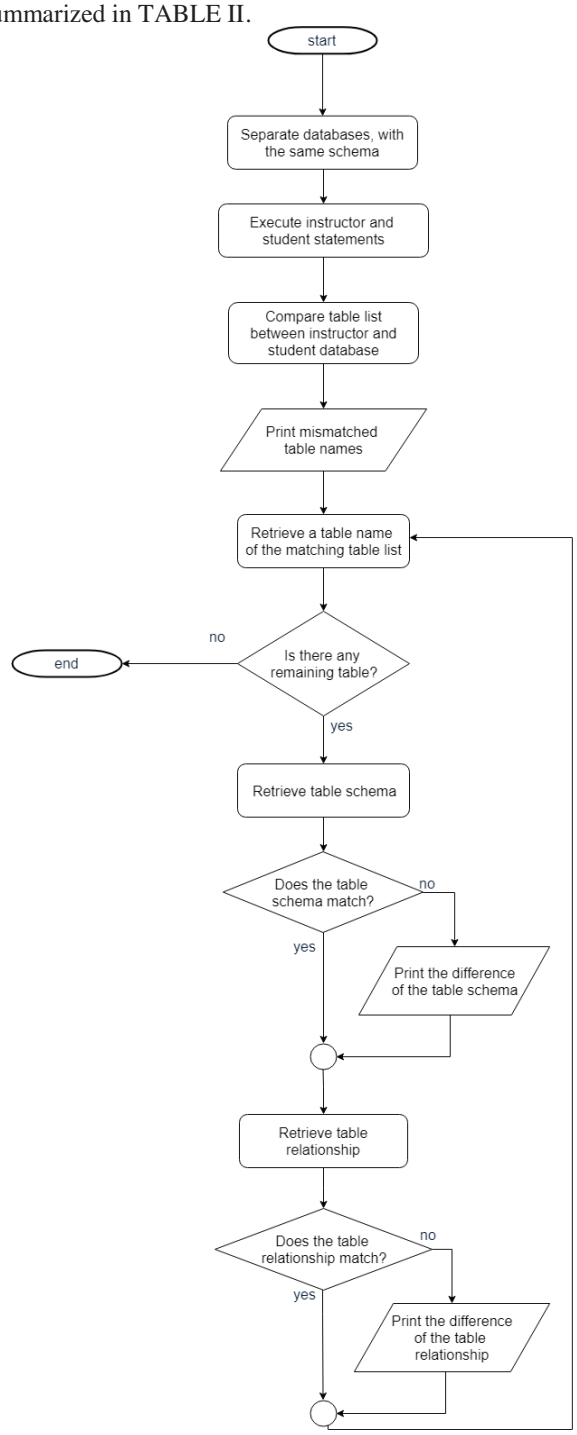

Figure 2. Automated grading algorithm for DDL statement (CREATE, ALTER and DROP)

```
{
  "table": {
    "table_name": {
      "attributes": {
        "column_name": {
          "datatype": "INT",
          "length": 10, // data length or value
          "PK": true, // Primary Key/true or false
          "NN": true, // Not Null/true or false
          "UQ": false, // Unique/true or false
          "AI": true, // Auto Increment/true or false
          "reference": {
            "table":"referenced_table_name",
            "attribute": "referenced_column_name"
          }
        }, ...
      },
    }, ...
  },
}
```

Figure 3. Example of database schema in JSON format

TABLE II. ERROR NOTIFICATION FROM GRADING DDL STATEMENTS

Error Notification	Explanation	Example
Missing <table_name>	Database missing a required table	Missing employees
Missing <table_name>.attributes.<column_name>.	Table missing a required column.	Missing employees.attributes.firstname
Unwanted <table_name>found	Database has an unwanted table.	Unwanted users found
Unwanted <table_name>>.attributes.<column_name>found	Table has an unwanted column.	Unwanted users.attributes.fullname found
<table_name>.attributes.<column_name><should / should not> be foreign key	The attribute of the table should / should not be foreign key	employees.attributes.department should be foreign key
<table_name>.attributes.<column_name>.referenceshould be <correct_table_name>.<correct_column_name>instead of <incorrect_table_name>.<incorrect_column_name>	The foreign key attribute of the table is incorrect.	employees.attributes.department. reference should be departments.id instead of locations.id
<table_name>.attributes.<column_name>.datatype should be <correct_value> instead of <incorrect_value>	Column data type is incorrect.	employees.attributes.firstname.datatype should be varchar instead of char
<table_name>.attributes.<column_name>.length should be <correct_value> instead of <incorrect_value>	Column data length is incorrect.	employees.attributes.firstname.length should be 50 instead of 30
<table_name>.attributes.<column_name><should / should not> be primary key	The attribute of the table should / should not be primary key	employees.attributes.id should be primary key
<table_name>.attributes.<column_name>.nullable should be <correct_value> instead of <incorrect_value>	Check if a column could be null or not.	employees.attributes.id.nullable should be false instead of true
<table_name>.attributes.<column_name><should / should not> unique	The attribute of the table should / should not unique	employees.attributes.ssn should unique
<table_name>.attributes.<column_name>.autoincrement should be <correct_value> instead of <incorrect_value>	Check if a column should be auto increment or not.	employees.attributes.id. autoincrement should be true instead of false

IV. INTERACTIVE LEARNING TOOL

The significant user of learning SQL is a student. The students need to understand impact and consequence when their statements are processed in database. The existing SQL learning systems only support SELECT statement, which can apply only one command of DML [2],[3],[4],[5]. The other DML commands i.e. INSERT UPDATE and DELETE and DDL commands create impact to the database. The students need to see the database schema and data set modification to completely understand their SQL statements.

The interactive learning tool is developed by embedding these DDL and DML grading algorithms in DBLearn [6] using the Atchariyachanvanich et al.'s MySQL Sandbox [1]. The tool is a basis which supports execution of DML and DDL statements and discards the security issues. It also supports simultaneous use from concurrent students. The interactive learning tool screen is shown in Figure 4. The screen includes two sections which are;

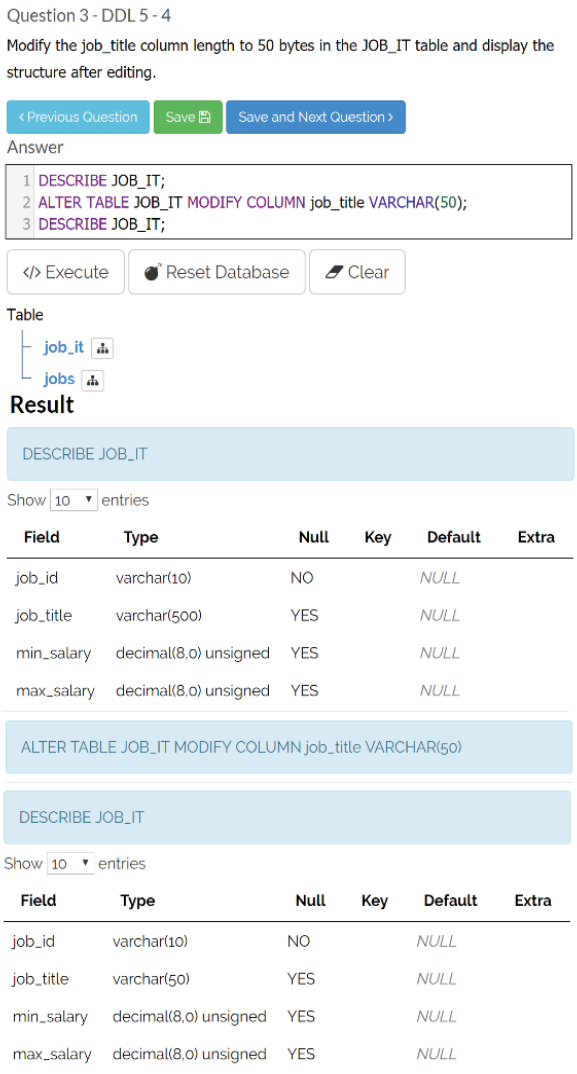

Figure 4. Example of interactive learning tool.

1. The question details and question navigation buttons including the save answer button.

2. The second screen is User interface for MySQL administration. This screen allows students to try out various statements and see result sets including changes manipulated with database. This section displays the tables which are currently existed in the database. The students can click on the name of the table to retrieve all rows in the table or press the button next to the table name to display the schema of the table. The result set of each statement is returned after a student writes a statement and presses the Execute button. The students can rollback the state of the database to the initial state by pressing the Reset Database button.

V. EXPERIMENTAL EVALUATION

The SQL grading result from the automated SQL grading tool is evaluated in accuracy aspect. The questions which use in the SQL examination are chosen then the grading results from the tool are compared against with the results from manual checking by instructor. Both grading methods are strictly calculated without partial scoring. The results from the tool are shown in TABLE III.

TABLE III displays the result from the DBLearn tool which can achieved 100% grading accuracy on DML and DDL statements. However, the tool has a limitation. The tool cannot score default value of the column because of characteristic of MySQL. If the default value of the column is not set then the MySQL may automatically configure it. The example of similar statements having different default values displays in Figure 5. The first statement creates a "title" table when title_id column having no default value but the second statement creates "title" table with title_id column which having "0" as a default value generated by MySQL.

```
CREATE TABLE title(
  title_id int NOT NULL UNIQUE,
  PRIMARY KEY (title_id)
);

CREATE TABLE title(
  title_id int,
  PRIMARY KEY (title_id)
);
```

Figure 5. Example of similar statements having different default values

TABLE III. ACCURACY OF DBLEARN GRADING FOR DML AND DDL STATEMENTS

SQL	*Submitted Answer*	*Manual-check result*		*DBLearn Tool check result*		Accuracy
		Correct	*Incor-rect*	*Correct*	*Incor-rect*	
INSERT	139	100	39	100	39	100%
UPDATE	122	75	47	75	47	100%
DELETE	78	42	36	42	36	100%
CREATE	51	12	39	12	39	100%
ALTER	228	118	110	118	110	100%
DROP	81	81	0	81	0	100%

This paper compares supportable SQL statements of the existing grading system with DBLearn tool as displays in TABLE IV. The DBLearn tool can support all basic SQL statements.

TABLE IV. COMPARISON OF EXISTING SYSTEMS WITH THE DBLEARN

System Name	*Grading Supported SQL Statements*		
	SELECT	***DML (INSERT, UPDATE, DELETE)***	***DDL (CREATE, ALTER, DROP)***
SQLator [2]	/		
SQLify [3]	/		
SQL-Lab [4]	/		
aSQLg[5]	/		
XDa-TA[7]	/	/[a]	
Ke et al.'s work [8]	/	/[a]	
MySQL Sandbox	*x*	*x*	*x*
DBLearn tool	/	/	/

[a] convert DML to SELECT statement

VI. CONCLUSION

An interactive learning tool with ability to grading DML and DDL statements is proposed in this research. The DBLearn is the first tool to grading DDL statements and complex DML statements. This tool has enhanced the range of supportable SQL statements and can provide solution to eliminate the limitations of the existing SQL grading systems. The tool can reduce the effort of instructor and decrease time to grading answers. The students can get faster responses and can speed up the learning ability.

In the future, we plan to improve our grading algorithm by featuring a partial scoring of all basic SQL statements. This ability helps to decrease instructor effort to score student answers when some part of answer is correct.

REFERENCES

[1] K. Atchariyachanvanich, S. Nalintippayawong, and T. Permpool, "Development of a MySQL Sandbox for processing SQL statements: Case of DML and DDL statements," in 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2017, pp. 1-6.

[2] S. Sadiq, M. Orlowska, W. Sadiq, and J. Lin, "SQLator: an online SQL learning workbench," in Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, Leeds, United Kingdom, 2004, pp. 223-227.

[3] M. d. Raadt, S. Dekeyser, and T. Y. Lee, "Do students SQLify? improving learning outcomes with peer review and enhanced computer assisted assessment of querying skills," in Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006, Uppsala, Sweden, 2006, pp. 101-108.

[4] P. Brusilovsky, S. Sosnovsky, M. V. Yudelson, D. H. Lee, V. Zadorozhny, and X. Zhou, "Learning SQL Programming with Interactive Tools: From Integration to Personalization," Trans. Comput. Educ., vol. 9, pp. 1-15, 2010.

[5] 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, Koli, Finland, 2013, pp. 161-168.

[6] S. Nalintippayawong, K. Atchariyachanvanich, and T. Julavanich, "DBLearn: Adaptive e-learning for practical database course — An integrated architecture approach," in 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2017, pp. 109-114.

[7] A. Bhangdiya, B. Chandra, B. Kar, B. Radhakrishnan, K. M. Reddy, S. Shah, et al., "The XDa-TA system for automated grading of SQL query assignments," in Data Engineering (ICDE), 2015 IEEE 31st International Conference on, 2015, pp. 1468-1471.

[8] H. Ke, G. Zhang, and H. Yan, "Automatic Grading System on SQL Programming," in 2009 International Conference on Scalable Computing and Communications; Eighth International Conference on Embedded Computing, 2009, pp. 537-540.