Chapter 9
Automated Evaluation of SQL Queries: Eval_SQL

Bhumika Shah and Jyoti Pareek

Abstract The assessment of SQL queries is a time-consuming task for the teacher, as each query needs customized feedback. Automation of such a task can prove beneficial for students as well as teachers. Some of the semi-automated evaluation tools for SQL queries are reported in the literature though none of them provides Quantitative as well as Qualitative feedback. All the evaluation tools available for SQL queries provide a binary type of feedback, which results in the query being right or wrong. However, evaluation could be more meaningful if customized self-explanatory feedback is provided to the student stating the level of correctness of the query along with the description of the mistake committed (if any). Authors have developed "An Automated Assessment tool for SQL Queries: Eval_SQL" which provides the marks even for partially correct query (Quantitative) and the feedback on what went wrong in the query (Qualitative). This can improve the student's learning experience in the virtual world. Eval_SQL also helps to reduce teacher workload, allowing them to focus more on learning-centric tasks.

9.1 Introduction

Computer science courses demand extensive laboratory work, and proficiency in practical implementation is an essential skill required for computer science professionals. Implementation of concepts learned is one of the important criteria for any practical subject in computer science courses. However, the assessment of the said implementation is even more important. DBMS is an important subject taught in higher education in computer science. The practical concepts of DBMS are been implemented in Structured Query Language (SQL). Learning SQL syntax is not a difficult task, but converting a requirement given in natural language into an appropriate query is challenging. Students require a lot of practice to master the art of querying the data. The students are often not sure whether the query been written by them is correct or not. SQL has a very logical syntax, and though the SQL

B. Shah (✉) · J. Pareek
Department of Computer Science, Gujarat University, Ahmedabad, Gujarat, India

V. Bhateja et al. (eds.), *Evolution in Computational Intelligence*,
Smart Innovation, Systems and Technologies 267,
https://doi.org/10.1007/978-981-16-6616-2_9

queries look very simple to understand, translating them from simple English statement to a semantic query proves difficult for the students. For each query written, a teacher/instructor needs to provide customized feedback to the students on the correctness of the query. However, the different variants available in SQL queries demand higher practice for each variety of Query Set. For each such variant, a teacher provides the students with a set of practice questions and assignments to master the SQL queries. However, the assessment of such queries becomes a time-consuming task as each and every query is to be provided with the appropriate and customized feedback and at times the teacher is not able to provide timely feedback to each one of them as students are large in number. Hence, there is a need for a proper assessment system that can automate the evaluation of SQL queries resulting in timely feedback to students and a reduced workload of teachers. The automatic assessment tools can help reduce the burden from teachers by allowing them to focus on student-teacher interactions and other learning-centric tasks.

9.2 Background

There have been various systems proposed on Automated Evaluation of SQL Queries. Each of them differs in its functionality. Some of the systems reviewed evaluate the query just by showing a binary type of feedback like correct or incorrect. One of the Systems provides the table structure to the user and lets the student select the attributes and Auto Generates the Query. One such system is able to provide feedback, by taking peer review and eventually getting reviewed by the teacher. However, there is manual intervention; the system is not evaluating the queries (Table 9.1).

This rightly says that there are hardly any systems available that provide Automated Assessment of SQL Queries. The various tools/systems reviewed for SQL Learning and assessment are described in detail as follows:

In 1997, Kearns et al. [1] proposed a system for Learning SQL that displayed a sequence of images to depict query processing stages step by step on how the query result is determined. The authors have emphasized on visual representation to enhance the semantic understanding of students in the area of SQL.

In 2004, Sadiq et al. [2] proposed an Online SQL Learning workbench "SQLator," which is a web-based interactive tool for learning SQL. The author's claim is to have achieved a high rate of success in determining user queries as correct or incorrect. The authors have used heuristic algorithms for comparing the SQL Queries. However, the student is allowed to execute the query before submitting it for evaluation. Moreover, if the learner is unsuccessful in writing the query, they can access the correct solution.

In 2006, de Raadt et al. [3] proposed SQL tutoring and assessment tool "SQLify." They have combined many features of existing systems like the visualization of the database schema, Query Processing, Semantic feedback, and assessment. de Raadt et al. [3] proposed a pedagogical perspective of the said system in **2007**. They compare and review the existing tools concerning database perspective and pedagogical approach. Relational algebra support was added in the enhanced version.

Table 9.1 Comparison of the existing online assessment tools

Features	Tools						
	SQL learning	Assessment of SQL queries available	Feedback on queries available	Partial grading for syntactically correct queries	Partial grading for syntactically incorrect queries2	Laboratory practice available in the same system	Database used
Esql [4]	✓	–	–	–	–	–	NA
SQLator [7]	✓	✓	–	–	–	✓	SQL Server 2000
Sqlify [6]	✓	✓	✓	–	–	–	MySQL
Assessql [5]	✓	✓	–	–	–	✓	PostgreSQL
ACME [8]	✓	✓	–	–	–	✓	Oracle, SQL Server
AutoGen SQL queries [2]	✓	–	–	–	–	–	PostgreSQL
Partial marking system [1]	✓	✓	✓	✓	–	✓	PostgreSQL
Eval_SQL (our proposed system)	✓	✓	✓	✓	✓	✓	GU_DB (our own DB prototype)

In 2006, another system was proposed by Soler et al. [5] named ACME which maintains a repository of the problems submitted by the teacher and a workbook module for students which consists of different exercises. The third module is the correction module, which compares student's solutions with the teacher's solution or with the output and gives the result to the student, whether correct or incorrect. Different pear libraries were integrated to support Oracle and SQL Server.

In 2014, Cruces et al. [6] proposed a system for the Automatic generation of SQL Queries. The system generates the queries automatically based on the selection. The user selects the schema, attributes, and functions that are used. The system generates the query based on the selection.

In 2016, Chandra et al. [7] proposed partial marking for SQL Queries. The system was developed at IIT Bombay and provided an interactive and automated platform for learning and assessment of SQL Queries. The system provides partial grading to the queries. The generated datasets are compared to determine the correctness of the query, and if the query is incorrect, partial assessment criteria are followed based on how close the student query is with the instructor query. However, the queries are considered only after they are syntactically correct.

As Raadt [4] rightly says "Relational query languages are not Turing complete, and because important subsets of these languages allow decidability of query equivalence, tools can be constructed that provide immediate syntactic and semantic feedback".

The two important considerations about syntactic and semantic feedback play a vital role in the development of the assessment system. All the tools reviewed can provide **semantic** feedback, but when it comes to **syntax checking**, the systems need **to rely on the database** they are using and provide with the error message of the database itself. In addition, they **also fail to grade** queries **partially** in case of **syntactic errors**. **Our system** is able to provide exact and **accurate feedback** as it uses our own Database System Prototype in the back-end named **GU_DB** [8].

The authors have developed the complete learning management system in the form of a virtual laboratory for database systems [8]. It consists of our own database management systems prototype **GU_DB**, which allows the users to implement the concepts learned. As discussed in the previous section, our Database prototype (GU_DB) [8] helps us to provide specific hints to the users when the query is syntactically wrong. It is a complete learning management system (LMS) that involves the components like Tutorial of the subject, Procedure, Simulator, Assessment (Theory and Practical), and Feedback.

9.2.1 Automated Assessment of SQL Queries

Assessment is an important criterion for learning. Once the student has learnt the concepts, there is a need for assessment. During the manual assessment, teachers partially evaluate the student's task. The evaluation is not necessarily zero or full marks. Most of the time, students are awarded partial marks based on the logic applied. Even if the syntax is wrong, the teacher may provide some marks looking

at the level of incorrectness. We aim to build an **automated assessment tool which can award partial marks to the students for their task submitted**. Most of the evaluation or auto-grading systems evaluate in the binary whetherthe result is either correct or incorrect. The result is the student does not get to know his/her mistake and do not remain motivated to perform other tasks. Moreover, none of the systems reviewed are able to evaluate the syntactic incorrectness of the Query in terms of the level of incorrectness.

9.3 Proposed Work

Authors have developed a **Framework** for **Automated Partial Marking System for SQL Queries**. The system would award the marks based on the level of correctness of the query. The correctness would be determined as per the rules laid down in the system, which will integrate a proper assessment structure. The marks would be awarded based on the rules defined in the system. Additionally, the system provides rich and constructive feedback to the students on the query submitted.

9.3.1 Objectives of Eval_SQL

Automated assessment systems help the learner(s) by providing the result in terms of marks and feedback instantly. The goal of the proposed system is to make students test their SQL knowledge by applying the queries in the assessment system and getting the result instantly.

The Eval_SQL has been developed to achieve the following objectives:

- Provide **Partial Marking** for SQL Queries.
- Provide **constructive feedback** for student queries to analyse their own mistakes.
- Give a complete learning environment to the student, wherein he/she can learn and immediately apply his/her knowledge and at the same time gets the advantage of validating the knowledge applied by auto evaluation of submitted queries.
- To automate the evaluation to provide immediate feedback/result in order to keep students motivated to keep practicing more.
- Reduce the workload of teacher(s) by automating the task of evaluating a large number of exercises.

9.3.2 Architecture of Eval_SQL

Eval_SQL consists of the following modules.

The diagram in Fig. 9.1 shows the communication between the assessment rule engine and GU_DB to fetch the correctness of the query. The user submits the query

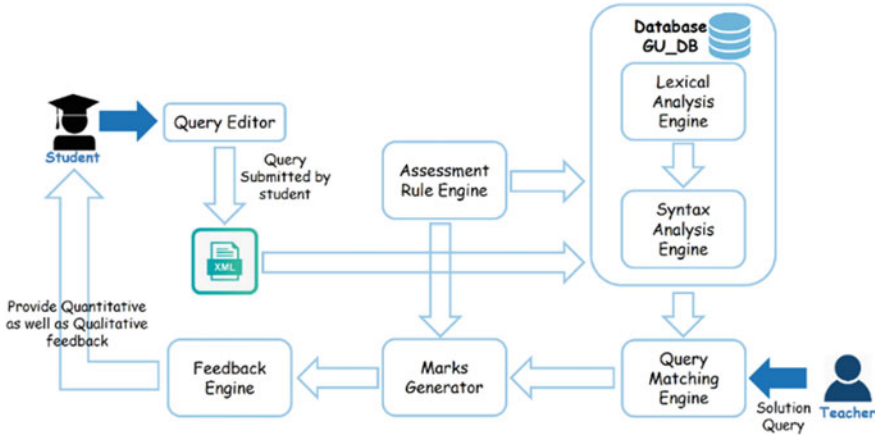

Fig. 9.1 Architecture of Eval_SQL

in the Query editor, which passes the query to the back-end. Eval_SQL integrates GU_DB in the back-end. Hence, the query submitted by the user passes through various phases like Lexical analysis, Syntactical analysis, semantic checking, and finally reaching the Assessment Rule Engine. The diagram highlights how Marks generator and feedback engine display the final result to the user for the query submitted.

9.3.3 Assessment Rule Engine

The Assessment rules are used to generate the final marking for the query, which consists of the following modules.

9.3.3.1 Assessment Module

The Assessment module is one of the core components in the development of an assessment system that provides partial marking for SQL queries along with the feedback.

The image in Fig. 9.2 displays the flow of query being sent for assessment. The query passes through various phases like Lexical, syntactical, and component matching. Any query submitted first needs to communicate through GU_DB, and GU_DB returns the status of the query which is sent for further assessment to Eval_SQL. Eval_SQL then compares the User Query with the Expert query, follows different comparison techniques discussed in the algorithm, and generates the Result. The Result displays the Marks received in the form of percentage and the constructive textual feedback received as displayed in the diagram.

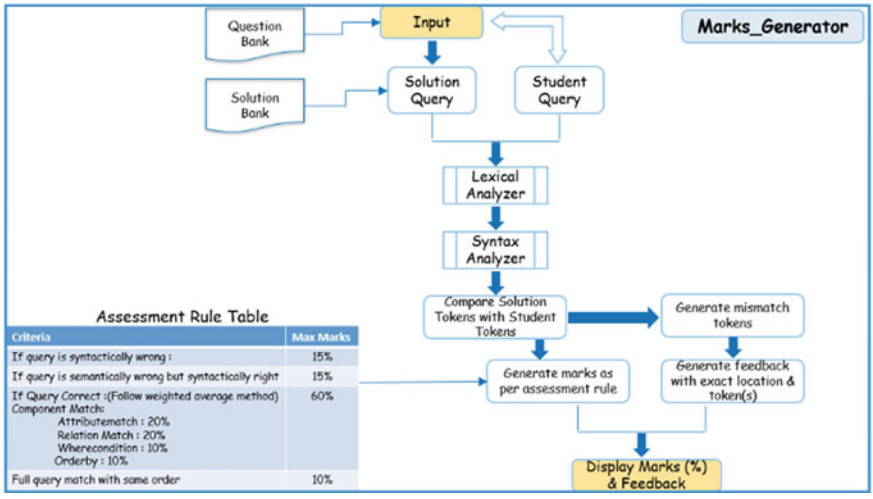

Fig. 9.2 Architecture of Marks_Generator and feedback

The assessment module consists of various modules which are listed as below.

9.3.3.2 Awarding Marks for Different Components of the Query (Marks_Generator)

The system aims at providing partial marks on the query written. However, the partial marks should be awarded only after assessing the validity of the query. The total marks need to be calculated after receiving marks from each of the evaluation stages the query has traversed through and final marks are generated. The Assessment system (Eval_SQL) needs to communicate with the Database (GU_DB) for the validity of the query. GU_DB in turn returns the validity of the query syntactically and semantically. Once the result from GU_DB is fetched, the query is compared with the Solution Query provided by the faculty for the logical analysis. The comparison of a query involves matching the components of the query and having their average. The system is able to provide accurate syntactic error and appropriate feedback to the user as there is GU_DB in the back-end which provides us complete flexibility on the database.

The assessment rules are laid out to have a proper marking scheme for the query. The authors have used the state table for implementation, which informally constructs a finite-state machine. The marking scheme in Marks_Generator is defined as follows.

Marking Scheme

Each component of the query is assigned marks as per the weightage defined for each of them. Maximum marks of the query may vary depending upon the complexity/difficulty level. A Query requiring more components will have more

Table 9.2 Table for marking scheme

Component	Weightage
Keywords (Select, Insert, Update, Delete)	5
Connecting keywords like (values, from, like, group by, where, set, order by, having, etc.)	2
Table name	2
Attribute name	2
Expressions	2

maximum marks whereas a Query requiring less components will have less maximum marks. Later for Normalization, marks obtained are converted in percentage.

The Maximum marks of Solution (Expert) query are calculated first. Since different queries have different components and hence total marks for the query are calculated dynamically. The solution Query is passed to the Marks_Generator for calculation of total marks as described in Table 9.2.

9.3.3.3 Providing Accurate Feedback for the Query (Feedback Engine)

Unlike other systems, our system is able to provide constructive and accurate feedback as we are using our own DBMS Prototype GU_DB. The feedback is provided with an exact component mistake, which helps the learner self-evaluate their mistake. Moreover, all the systems discussed use some or the other database in the back-end like MySQL, PostgreSQL, and so on, hence they need to rely on the said database for the error/message. Our system uses our own database prototype (GU_DB), hence, we have complete flexibility with the system. Our system can fetch the exact mistake including accurate syntactic errors, which the user has made, and accordingly, marks are calculated and feedback is provided.

The distinguishing feature of the Assessment Rule engine is as follows: The system provides two types of feedbacks to the learner**:**

1. Scores in terms of Percentage (Quantitative Feedback).
2. Formative Constructive feedback (Qualitative Feedback).

In the evaluation strategy, providing relevant and timely feedback is an essential criterion. Online assessment systems provide this advantage by giving immediate feedback. There are very few systems in Computer science which provide textual feedbacks, and in Database systems, they are even rare.

Algorithm for Assessment Module

For example, consider the following Queries.

Q.1 List Names and Salary information for employees

Select Name, Salary from Emp; (Expert Query).

In the above query, total marks are calculated as below.

Select [keyword]	5
Name, salary	2 + 2
Emp [table name]	2
From [keyword]	2
Total Marks	13

The total marks calculated for the above query is 13.

This is a simpler query with fewer components

The following is a query with little more complexity.

Q.2. Display names of Clients whose name has the second letter as "A"

Select Name from ClientMaster113 where Name Like "_A%"; (Expert Query).

In the above query, total marks are calculated as below:

Select [keyword]	5	Where [connecting keyword]	2
Name [attribute]	2	Name [attribute]	2
From [keyword]	2	Like [connecting keyword]	2
Clientmaster113 [tablename]	2	"_A%" [expression]	2
		Semicolon	1
Total marks 20			

The total marks calculated for the above query is 20.

The student writes the following query for Q.2

Select Name From Client_Master113 where name like "A%".

This **User Query** will obtain 17 marks (2 marks of expression and 1 mark of semicolon are deducted from total of 20 marks hence user receives 17 marks) as per the expert query and marking scheme. In order to normalize marks, the marks obtained are converted to percentage:

Obtmarksinperc = *obtmarks/maxmarks*100:* e.g. 17/20*100 = 85.

*Obtained maximum marks are computed by the system (e.g. 85%).

Eval_SQL Algorithm

This section describes the algorithm for calculating the marks and generating the feedback for the query written by the user. The student query is first parsed to **GU_DB** to check for its syntactic and semantic correctness and the status from GU_DB is

returned to **Eval_SQL** for component matching, and accordingly, appropriate marks and feedback are recorded.

Input: User Query.

Output: Total Marks received in percentage and Feedback.

Algorithm

1. **Initialization:**
   Establish connection with GU_DB, Parse query and initialise FA
2. **Read the questions from the CSV file (sqlqueries.csv) (goto readfile subroutine)**
   Return with tokenised query, Databasepath, TableMetadata and Tableinfo.Assign MaxMarks for Expert Query [Refer Table1]
   Display qno, question to user
   Accept student Query
3. **Evaluation of student answer**
   Send the student query to GU_DB and return with user answer as Valid or invalid tokens with their exact position from Syntax_validator (For providing feedback to user about exact mistake), Databasepath, TableMetadata and Tableinfo.
4. **Compare User List Tokens with Solution List Tokens**
   Gu_DB returns the syntactical analysis, but to fetch any logical errors, we need to compare it with the Expert query, So, Compare User Token list with Solution Token list,
   Store the Mismatched tokens in "MissingTokens" list
5. **Marks & Feedback Generator**
   Fetch the Token wise marking scheme, assign marks as per their definition and calculate obtained marks. Return with obtained marks and missing tokens and mismatch tokens for providing appropriate feedback obtained marks
6. **Display Result**
   Generate Result in Terms of Feedback displaying the reason for marks deducted and Marks in terms of Percentage.

9.4 Implementation (Eval_SQL)

The following screenshots display the GUI for Automated Assessment of SQL Queries: Eval_SQL (Figs. 9.3 and 9.4).

Once the User clicks on the feedback, the detailed feedback showing the query written by the user and the marks obtained by the user are displayed as follows (Figs. 9.5).

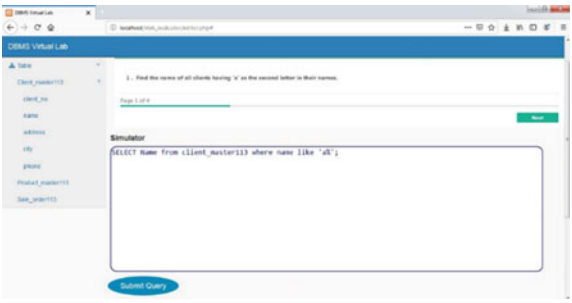

Fig. 9.3 User writes first

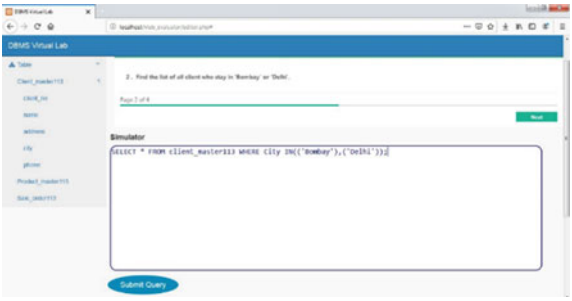

Fig. 9.4 User iterates through next queries

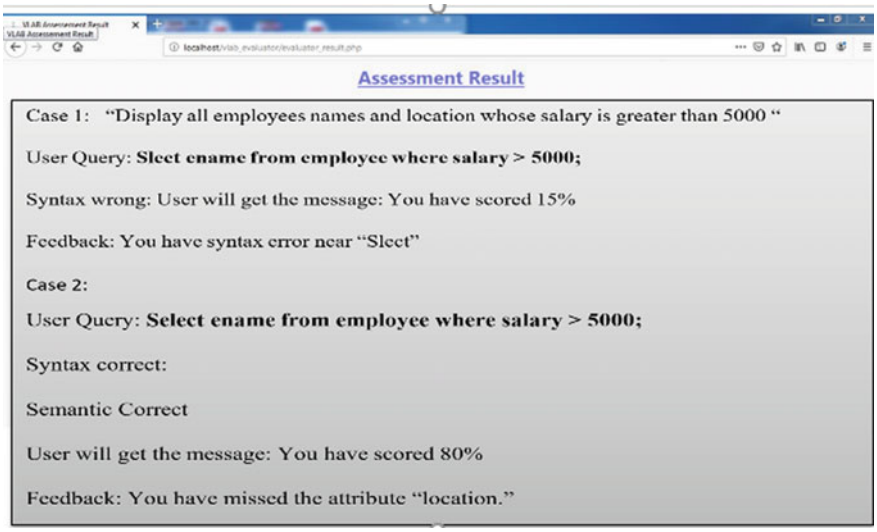

Fig. 9.5 Feedback displayed to the user

The Feedback screenshot displays that the system provides the exact mistake the user has made, which guides him towards learning.

9.5 Results

Our Automated assessment system: EVAL_SQL provides partial marking as well as Qualitative and Quantitative feedback for SQL Queries to the users. It provides a fully automated assessment of SQL queries and has the ability to partially award the marks to the SQL queries written by the user if the query is incorrect. The evaluation system accepts the questions and solutions from the teacher, and Evaluation system is using our own Database management system prototype GU_DB [10]. Hence, the system is able to provide accurate syntactic error, which the user has made, and accordingly, marks are calculated and feedback is provided. The Eval_SQL provides the advantage of teacher-like marking by providing partial marks to partially correct queries, and at the same time reduces the teacher's workload by automating the process of assessment. The pedagogical consideration in the development of the LMS makes it beneficial to all the students with different learning abilities.

Evaluation Process

The effectiveness of the system has been tested with a dataset of **107 SQL Queries**. For evaluation, we have selected semester 2 and semester 3 students of MCA of the computer science department for the study. The same queries, which were given to the Eval_SQL for evaluation, were given to the expert for marking. The marks obtained by Eval_SQL and subject expert are tabulated below.

Results and Discussion

The analysis of SQL queries is given in Table 9.3.

The marks awarded to the students by the system were compared with expert marks manually, and the results were impressive.

The diagram in Fig. 9.6 portrays the strong relationship between system marks and expert marks. The relationship between two variables can be further ascertained with the help of the correlation coefficient.

Correlation Co-efficient

The relationship between two variables is generally considered strong when the value of r is greater than 0.7. The following scattered plot displays the relationship between two variables namely system marks and expert marks (Fig. 9.7).

The authors have compared the results given by the system with the results provided by an expert. The correlation coefficient was computed to ascertain a positive relationship between the two. The correlation coefficient between the marks given by the system and the expert is **0.8764**, which shows that there is a **highly positive** correlation between the results given by the system developed and the results provided by the expert.

Table 9.3 Marks provided by Eval_SQL and subject expert

Query no.	System marks	Faculty marks	Query no.	System marks	Faculty marks	Query no.	System marks	Faculty marks
1	10	10	41	9	8	81	10	10
2	3	6	42	10	10	82	10	10
3	3	0	43	10	10	83	10	10
4	10	10	44	10	10	84	10	8
5	10	10	45	10	10	85	10	10
6	9	7	46	10	10	86	10	10
7	10	9	47	10	10	87	10	10
8	10	10	48	10	10	88	10	9
9	4	2	49	8	7	89	7	10
10	10	10	50	8	7	90	10	10
11	7	10	51	4	1	91	10	10
12	10	10	52	7	5	92	7	10
13	10	10	53	9	10	93	10	10
14	7	10	54	10	10	94	3	4
15	10	10	55	10	8	95	10	10
16	3	4	56	5	1	96	10	10
17	10	10	57	7	9	97	10	10
18	10	10	58	10	10	98	10	9
19	10	10	59	10	10	99	10	10
20	10	9	60	10	10	100	8	8
21	10	7	61	10	10	101	10	10
22	10	10	62	10	10	102	10	10
23	10	10	63	10	10	103	7	5
24	10	10	64	10	10	104	10	10
25	10	8	65	10	8	105	10	10
26	7	8	66	10	10	106	10	10
27	3	2	67	10	10	107	5	4
28	3	0	68	10	10			
29	7	6	69	10	9			
30	7	6	70	4	7			
31	10	10	71	7	10			
32	8	8	72	9	10			
33	10	10	73	10	10			
34	10	10	74	10	8			
35	7	5	75	5	10			

(continued)

Table 9.3 (continued)

Query no.	System marks	Faculty marks	Query no.	System marks	Faculty marks	Query no.	System marks	Faculty marks
36	10	10	76	7	9			
37	10	10	77	10	10			
38	10	10	78	10	10			
39	5	4	79	10	10			
40	10	10	80	10	10			

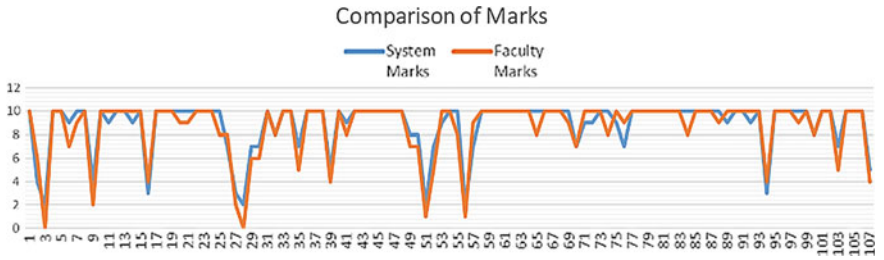

Fig. 9.6 Comparison of system marks and expert marks

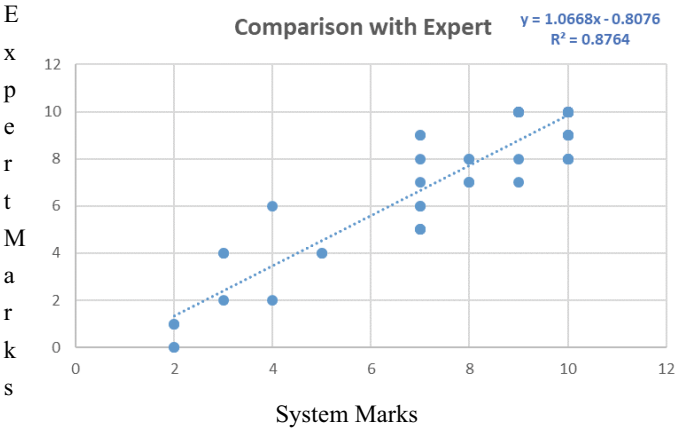

Fig. 9.7 Correlation results

9.6 Conclusion and Future Scope

Automated Evaluation systems are limited in number for SQL Queries. Moreover, there is hardly any system that provides partial marking for the query written along with Qualitative as well as Quantitative feedback for the Queries. The paper presents a partial marking automated system "Eval_SQL," which gives a fully automated experience for the assessment of SQL queries and has the ability to partially award the marks to the SQL queries written by the user. As the evaluation system (Eval_SQL) accepts the questions and solutions from the teacher, the teacher involvement instils confidence in the learner in the evaluation system. Eval_SQL is using the DBMS prototype GU_DB [9] developed by the authors. Hence, the system is able to provide appropriate feedback including accurate syntactic error, which the user has made, and accordingly, marks are calculated and feedback is provided. The Eval_SQL gives the advantage of Teacher like marking by assigning marks to partially correct queries, and at the same time reduces the teacher's workload by automating the process of assessment. In the future, the system will be integrated as part of the Virtual laboratory of Database systems, for automated evaluation of SQL Queries. This will help the students to self-learn and self-assess the progress of the topic learned.

References

1. Kearns, R., Shead, S., Fekete, A.: A Teaching System for SQL, pp. 224–31 (2004)
2. Sadiq, S., Orlowska, M., Sadiq, W., Lin, J.: SQLator: An online SQL learning workbench. ACM SIGCSE Bull. 1–5. http://www.dl.acm.org/citation.cfm?id=1008055
3. de Raadt, M., Dekeyser, S., Lee, T.Y.: Do Students SQLify ? Improving Learning Outcomes with Peer Review and Enhanced Computer Assisted Assessment of Querying Skills, p. 101 (2007)
4. de Raadt, M.: Computer Assisted Assessment of SQL Query Skills, p. 63 (2007)
5. Soler, J., Prados, F., Boada, I., Poch, J.: A web-based tool for teaching and learning SQL. In: International Conference on Information Technology Based Higher Education and Training, ITHET. http://www.acme.udg.es/articles/ithet2006.pdf
6. Cruces, L.: In health informatics, and computer studies. Automatic Generation of SQL Queries Automatic Generation of SQL Queries (2006)
7. Chandra, B., et al.: Partial marking for automated grading of SQL queries. Proc. VLDB Endowm. **9**(13), 1541–1544 (2016)
8. Fuller, U., et al.: Developing a computer science-specific learning taxonomy. In: Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education—ITiCSE-WGR '07, p. 152 (2007). http://www.portal.acm.org/citation.cfm?doid=1345443.1345438

A Publication in Process

9. Bhumika, S., Jyoti, P.: GU_DB: a database management system prototype for academia. Int J Adv Comput Res 11.55(2021):67

Published Doctoral Dissertation or Master's Thesis

10. Shah, B.: An Innovative Framework for Remote Database Experimentation. Department of Computer Science, Gujarat University (2020). http://www.hdl.handle.net/10603/307860sertation