Improving Grading of Hand-written Students' SQL Queries in Database Courses

Gontlafetse Mosweunyane
Department of Computer Science
University of Botswana
Gaborone, Botswana
mosweuny@ub.ac.bw

Tallman Z. Nkgau
Department of Computer Science
University of Botswana
Gaborone, Botswana
nkgautz@ub.ac.bw

***Abstract* —This paper discusses the challenges involved in the process of manually grading hand-written SQL queries in student's scripts with a view of coming up with some guidance to improve the process. The challenges stem from experience in teaching and grading SQL Queries in 200 and 400 level Database courses at the University of Botswana. The challenges are presented with a view of detailing the issues involved in SQL query grading that could help formulate a systematic approach to the process. We investigate the structure and execution of SQL queries, specifically that of the SELECT query. We also review assessment methods from literature as a guide to a new approach that we suggest for guiding the grading of students' queries.**

Keywords — SQL, grading, assessments, database

I. INTRODUCTION

Databases, especially relational databases, are ubiquitous in computing systems that form part of our everyday life [1]. SQL is the most common language used to create, update, and query such systems. This makes SQL query formulation a fundamental skill for software developers [2]. Consequently, a database course, including a topic in SQL, is almost always one of the core courses required for students undertaking a computing undergraduate degree qualification [3]. Teaching and assessment of SQL query writing skills is therefore an essential aspect of every course in design and development of databases. Both formative assessments and summative assessments, which generally use paper-and-pencil tests for assessing SQL skills against a standard and for assigning grades, are used [4]. Formative assessments such as laboratory-based exercises use practical problem cases to reinforce understanding through practice, with visual tools which somehow provides a 'guide' that enables students to write, validate, then evaluate the query based on the query's expected results. In this process, continuous reformulation of the query is performed until the expected required results are obtained, thereby correcting the query, and increasing the likelihood of learning the skill by students. Summative assessments, which are usually involve students writing down answers to questions in a constrained environment, do not have this advantage and consequently some students end up writing partially correct or completely incorrect answers. This paper focuses on summative assessment evaluation.

In a written test or examination setup, students are presented with a database schema, with or without relation instances, and a description of a request which they must translate into an SQL statement. Basically, they are presented with a database schema and asked to write SQL queries to satisfy a given information need. Although SQL has been found to be syntactically simple, relatively concise, and highly structured [3], this is not easy for students because they must basically emulate a Database Management System (DBMS), a software system that performs several behind-the-scene operations that are not noticeable by the user, executing an SQL statement [2]. In addition, they have no means of testing the query for correctness. Consequently, students usually come up with different solutions for SQL queries which may be completely wrong, partially correct or correct depending on several factors. Dekeyser [3] summarises problems students have in relation to writing SQL queries. They include difficulty with memorising the database schema, misunderstanding the basic elements of SQL and first order logic and the relational data model in general. Also, students incorrectly perceive query problems as being easy, and they have difficulties with grasping the declarative nature of SQL.

The main aim of students' assessments through query formulation is to assess students' ability to demonstrate understanding and interpretation of a database creation, update, or retrieval request. How this is interpreted and enacted differs among assessors. Grading written SQL queries poses a challenge in most situations. Grading is usually carried out by comparing a student's written answer to a model answer (or several variations of the same) and awarding a mark out of a given ideal/perfect score based on the degree of match between the two. There are many issues to consider during this process such as query structure, syntax of the constructs of the query, query semantics and ultimately what the query might result in if it were to be evaluated. Additionally, if the assessor is assessing a considerable number of scripts, or if many assessors are involved in marking the same set of scripts, then consistency becomes an issue.

Grading is very important as it ultimately determines the individual's query formulation skills [5] and strongly influences a student's learning approach [6]. Challenges and issues related to grading are discussed in this paper with a view of developing a systematic way to help assess SQL queries, and ultimately to develop a system to assist in such.

II. OBJECTIVES

The question that arises then is – how can SQL queries be graded? What rewards and penalties can be applied during the grading process to emphasize important considerations in formulating queries and hence reinforce the learning process? What criteria should be used to eliminate variability and subjectivity in awarding marks?

III. METHODOLOGY

In our experience, grading SQL queries is usually carried out in an ad-hoc manner. We therefore look at our experience in grading students' queries, as well as the approaches for grading that have been followed by other researchers.

ISSN 2756-9160 / November 2020.
International Conference on Advances in Computing and Technology (ICACT–2020) Proceedings

Grading students' written SQL queries has been found to be slow, tedious, and error-prone and hence might impact students' grades [7, 8]. This has resulted in efforts to either automate or semi-automate the process. The basis for allocating marks to a solution differs amongst assessors, resulting in the same query being allocated different marks in different settings. A SQL SELECT query consists of clauses which serve different purposes in the execution of the query and differ in terms of significance.

Based on our experience, the challenges mainly stem from issues in weighing different aspects of the query and consistency during the grading process as discussed below.

a) Syntax

- Spelling – The reserved keywords in the clauses must be spelled correctly. This also applies to the names of relations and columns.
- Order - The clauses must appear in following order: SELECT, FROM, JOIN, WHERE, GROUP BY, HAVING, ORDER BY.
- Case sensitivity and quotes- SQL keywords are case-insensitive and string literals should be enclosed in single quotes.

b) Semantics

- Order - There is an element of reliance/dependence between consecutive clauses such that a wrong step might render subsequent clauses invalid or wrong. For example, if one or all the relations listed after the FROM clause are wrong, this might invalidate subsequent steps.

- Results – supposing the query is executed, what it ultimately results in is important for purposes of assessment. In addition to a correct query with the wrong end-result (columns stated after the SELECT clause), there is possibility of overstating (stating more than is necessary) and understating (stating less than what is needed) in the clauses.

d) Variations – The same query can be written in different ways. This might involve use of sub-queries, use of aliases, or different ordering of components of the query.

e) Dialects – Although minor, there are several different 'dialects' of SQL and an assessor should be aware and be able to embrace these. e.g. some DBMSs require a semicolon at the end of a SQL statement.

f) Query optimization – A query might be correct in the sense that it gives the correct output if it were to be executed. However, the method of getting results might involve longer steps than is necessary.

g) Consistency - Different assessors assessing parts of the same set of scripts, or the same assessor assessing a lot of students' scripts, might result in inconsistency in allocating marks if a systematic way of doing this is not adopted. This problem has been identified before by researchers and efforts to help with this has been presented in literature, for example, Dekeyser et al [3] proposes a tool, SQLify to assist with automatic assessment of SQL queries.

Grading therefore presents a great challenge. Usually the same technique conceptualized when setting up the question and envisioning a solution is probably the same method applied when marking. However, unless this is clearly outlined, captured and used during the grading process, there is possibility of applying a different criterion during the grading process. The grading may also be influenced by other factors such as syntax and semantics of the student's solution, how other students have approached the solution and even how the student answered other questions if grading is approached script-by-script rather than question-by-question.

IV. Results and Discussion

The importance of proficiency in writing SQL queries cannot be over-emphasized. It is therefore essential that the extent of this proficiency, using the important elements recognized as checkpoints, is properly evaluated. It is important that when an SQL query assessment is conceptualized, one bears in mind the points that would be considered as the extent or level of demonstration of ability to write that query.

Criterion-referenced assessments [9] have been recommended by researchers for assessing skill rather than performance, using a pre-set criterion. This contrasts with norm-based assessments, where students' grades are allocated based on the performance of other students in their cohort such that the grades follow a pre-determined distribution. Previous efforts for assessing SQL queries have focused on accurately determining students' SQL formulation skills in a way which is closer to how they will apply them in the real world [2]. Paper-based assessments therefore can therefore be evaluated based on how best the students demonstrate what needs to be done to accomplish the task at hand (join tables, selection criteria).

Curriculum design usually incorporates development of learning outcomes for courses. It is important that these are extended under the different topics that are taught in the course. Specific objectives for a taught topic must be defined clearly. They can later be adopted for assessment.

In our case, for the 200-level introductory database course the high-level learning outcome is stated thus:

At the end of this course, students should be able to:

- *use Structured Query Language*

At course level this is broken down to low-level learning outcomes. For SQL data manipulation using the SELECT statement, for example, the learning outcomes as adopted from [10]:

Students should be able to:

- *Use compound WHERE conditions.*
- *Sort query results using ORDER BY.*
- *Use aggregate functions.*
- *Group data using GROUP BY and HAVING.*
- *Use subqueries.*
- *Join tables together.*
- *Perform set operations (UNION, INTERSECT, EXCEPT).*

The low-level learning outcomes could be used as grading criteria. Several of these could be tested in one question and the marks allocated used to show level of attainment of this outcome. For example, if the question is marked out of 3, marks could be 0 – completely wrong, 1- Poor, 2 - Average,

ISSN 2756-9160 / November 2020.
International Conference on Advances in Computing and Technology (ICACT–2020) Proceedings

3 – good. Similarly, if marked out of 4 marks could indicate levels such as Poor, satisfactory, good and excellent. This could be applied to either the whole question or grading of a specific criterion, the total then being a summation of marks attained in all criteria tested in that question. For example, given the following question:

Consider the following relational schema:
STUDENT (stud-id, stud-name, age, address)
COURSE (c-code, c-name)
TAKING (stud-id, c-code, grade)
Write the query below in SQL.
Get the names of students who are doing both CSI471 and CSI481.

The main features that could be examined in this query are:

- Use of compound WHERE statement/set operation (INTERSECT)
- Joining of tables
- Listing the names of students

The solution can be presented in different ways, for example:

(a)
```
SELECT stud-name
FROM STUDENT, TAKING
WHERE STUDENT.stud-id = TAKING.stud-id AND c-code = 'CSI471'
                    INTERSECT
SELECT stud-name
FROM STUDENT, TAKING
WHERE STUDENT.stud-id = TAKING.stud-id AND c-code = 'CSI481';
```

(b)
```
SELECT stud-name
FROM STUDENT, TAKING
  WHERE STUDENT.stud-id = TAKING.stud-id AND c-code IN
                ('CSI471', 'CSI481');
```

(c)
```
SELECT stud-name
FROM STUDENT S
INNER JOIN TAKING T ON S.stud-id = T.stud-id
WHERE c-code IN ('CSI471', 'CSI481');
```

Using the learning outcomes to assess the solution instead of trying to assess the different clauses could help remove bias and therefore achieve consistency in marking. The focus of the assessor moves away from the granular elements of the query to the extent to which the student has attained the skill such that they can practice it later on (with the help of tools for minor mistakes). This also eliminates bias regarding the way in which the solution is written (variations like longer query vs shorter query, ordering of clauses etc).

In a trial run of this method, the example query from section IV was re-graded from past examination papers. It was previously awarded 4 marks in one setting and 5 marks in another. Using our recommended approach, the query will only need to be awarded a maximum of 3 marks for the 3 features recognized above. Regrading 30 past examination scripts with this method moved the average from 1.88/4 (47%) to 1.75/3 (58%). This means that the older, ad hoc approach generally disadvantaged students by awarding lower marks. Allocating more marks seemed to deviate attention from assessing the main features of the query, and hence seemed to 'cloud' one's judgement when grading.

The main drawback of using this approach is the need for prior planning and careful recognition of the features examinable in queries. Also, some elements of the query consist of sub-elements, which in some cases can be easier to grade if allocated marks individually rather than being given a single mark as a group/feature. The method also did not solve the problem of including unnecessary information (attributes, tables, conditions), which in some cases affects query results. The focus was only on demonstration of knowledge of the features tested. While this method does not offer a complete solution, it offers a step in the right direction towards improving grading and detecting SQL skill proficiency of students.

V. CONCLUSION

This paper presented the challenges involved in grading SQL queries manually. The challenges result from the various issues involved when presenting a query including syntax, semantics, order of the constructs and even different variations of the solution or dialects. It is important, therefore, that a focus for grading is developed such that the assessor can concentrate on the aspects that signify skill attainment in query writing. Criterion referenced assessment has been recommended in literature for skill assessment. We suggest development of clear and detailed learning outcomes in curriculum development for database courses and SQL topics such that these could be utilized as criteria for grading. Future work envisioned for this line of research is development and evaluation of this method in comparison to ad hoc methods of grading.

REFERENCES

[1] *9 - XML and Web Databases.* The Morgan Kaufmann Series in Data Management Systems, 2011: p. 161 - 187.

[2] Prior, J. *Online Assessment of SQL Query Formulation Skills*. in *ACE*. 2003.

[3] Dekeyser, S., M.d. Raadt, and T.Y. Lee, *Computer assisted assessment of SQL query skills*, in *Proceedings of the eighteenth conference on Australasian database - Volume 63*. 2007, Australian Computer Society, Inc.: Ballarat, Victoria, Australia. p. 53–62.

[4] Hammerman, E., *Formative assessment strategies for enhanced learning in science, K-8*. Formative assessment strategies for enhanced learning in science, K-8. 2009, Thousand Oaks, CA, US: Corwin Press. xiii, 129-xiii, 129.

[5] Prior, J.C. and R. Lister, *The backwash effect on SQL skills grading*, in *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*. 2004, Association for Computing Machinery: Leeds, United Kingdom. p. 32–36.

[6] Ramsden, P., *Learning to Teach in Higher Education*. 2003: RoutledgeFalmer.

[7] Chandra, B., et al., *Automated Grading of SQL Queries.* 2019 IEEE 35th International Conference on Data Engineering (ICDE), 2019: p. 1630-1633.

[8] Stanger, N. *Semi-automated assessment of SQL schemas via database unit testing*. 2018.

[9] Burton, K., *Designing criterion-referenced assessment.* Journal of Learning Design, 2006. 1.

[10] Hoffer, J.A., H. Topi, and V. Ramesh, *Essentials of Database Management*. 2013: Prentice Hall Press.