Uncovering Patterns of SQL Errors in Student Assignments: A Comparative Analysis of Different Assignment Types

Sophia Yang, Zepei Li, Geoffrey L. Herman, Kathryn Cunningham, Abdussalam Alawini
Department of Computer Science
University of Illinois Urbana-Champaign
{sophiay2, zepeili2, glherman, katcun, alawini}@illinois.edu

2023 IEEE Frontiers in Education Conference (FIE) | 979-8-3503-3642-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/FIE58773.2023.10343207

***Abstract*—Structured Query Language (SQL) is an essential skill to acquire for those who interact with databases, such as researchers, developers, and people involved in businesses. However, the challenges that these users face while learning SQL requires further research. In particular, the types of errors that students encounter on various assignment types or under exam conditions are an area that we are interested in to determine an optimal arrangement of coursework materials for improved learning. In this paper, we analyze 156,513 student SQL submissions to homework assignments, collaborative assignments, and exams of the Database Systems course available to 730 upper-level undergraduate and graduate students offered in the Fall 2022 semester at the University of Illinois Urbana-Champaign. We look at the ratio of syntax and semantic errors, and correct submissions for each of these assignment problem types as well as the most frequent syntax error codes. We visualize our data findings and draw recommendations for future coursework arrangements from the comparisons between the assignment types for a more effective acquisition of SQL as a skill. We found that although students most commonly encountered syntax error codes 1064 and 1054 regardless of the assignment type, they made more syntax errors (and fewer semantic errors) on exam problems compared with homework and collaborative assignment problems. We recommend instructors place a higher emphasis on non-timed SQL programming problems, targeted syntax drills during instruction, and syntax support during exams.**

***Index Terms*—SQL; database education; online assessment; syntax; semantics; error**

I. Introduction

The Structural Query Language (SQL) is the de facto standard language for managing and querying relational databases [1]. Therefore, for those who interact with databases, such as researchers, developers, etc., SQL is an essential skill to acquire [2, 3]. With an English-like syntax, this highly structured language is accessible for beginners since it does not require prerequisite programming knowledge; it may therefore be a gateway to lower the barriers of entry into the fields of computing for non-Computer Science students. Despite these factors, many students still experience learning challenges for SQL, leaving further research to be done [2]. In particular, we are interested in determining an optimal arrangement of coursework materials for improved learning based on the types of errors that students encounter on various assignment types or under exam conditions. We base our research upon prior research, which indicates that students may be significantly affected by test anxiety; therefore, their performance on exam problems may be lower, or not an accurate reflection of their knowledge of the tested concepts [4]. The Yerkes-Dodson Law also states that heightened stress levels may enhance performance up to a threshold, beyond which excessive stress may diminish student performance [5]. Furthermore, while complex tasks may benefit from an optimal stress level, simpler tasks maintain elevated performance under higher stress levels. Therefore, students who have not mastered the tested concepts may be affected the most under higher stress testing environments [5]. For this reason, we propose to analyze the SQL homework assignment data alongside SQL queries obtained under exam conditions to compare students' performance and encountered error types to cover both higher and lower stress-level environments. We also examine SQL queries submitted for collaborative learning assignments, because Murphy et al. [6] indicate that students may have significantly higher learning effectiveness in a collaborative environment compared to individual learning and/or assignments.

In order to analyze the types of errors that students encounter on various assignment types or under exam conditions, we analyze students' SQL submissions to homework assignment problems, collaborative assignment problems, and exam problems of the Database Systems course available to upper-level undergraduate and graduate students at the University of Illinois Urbana-Champaign. The Database Systems course is an elective database course in the Computer Science curriculum, providing students with the option to take it for learning purposes or as one of the technical electives required for graduation. To enroll in this course, students must have already completed a data structures course, typically taken during their second year. This prerequisite ensures that students have a foundational programming background, as the course includes programming assignments. The course instructor released all SQL-related assignment problems to students on PrairieLearn [7] - an online learning management system with built-in auto-grader capabilities - so we are presented with the opportunity to analyze the submission traces and errors of students; we look at the ratio of syntax and semantic errors, and correct submissions for each of these assignment problem types as well as the most frequent syntax error codes. Such insights may then assist instructors in determining the assignment types

that may benefit students the most and designing their course curriculum for SQL-related topics. Our research questions include: 1) *What are the error types that students encounter the most?* and 2) *Do the error types that students encounter differ among various assignment types or under exam conditions?*

Our research questions aim to improve the quality of education for learning SQL in courses that utilize auto-grading capabilities. The database course from which we collect our data has an enrollment of 730 students, and students in the course are given multiple SQL in-class group exercises (with 3-4 problems each) and an individual homework assignment containing 15 SQL problems. The first exam (of three) consists of one SQL programming question and a few multiple-choice questions regarding fundamental database concepts. All SQL programming questions are auto-graded, and students are given immediate feedback by the auto-grader after each submission attempt. The feedback includes any syntax error codes (if applicable) or discrepancies between the solution query output table and the student's query output table (semantic error).

Previous research work shows that students' learning experience may significantly be enhanced pending on the instructor's ability to identify the way their students learn [8]. Prior research work also shows that to help students form correct conceptions, we must first understand students' misconceptions [9]. We believe that by analyzing the error types that students encounter while solving an SQL question, we are one step closer to recognizing the student's mental model and misconceptions about database concepts and SQL. We aim to help students receive curriculum assignments that are more inclined to lead to correct conceptions; therefore, we must first grasp the misconceptions and errors that students encounter when learning and completing SQL assignments. We examine students' submissions to SQL-related assignments offered in the Fall 2022 semester. We perform exploratory analyses on the error types and compare our findings between each assignment type to support instructors in designing course assignments more effectively.

II. Related Works

While prior research has extensively examined the errors students make in procedural programming languages like Java [10, 11, 12], C++ [11, 13, 14, 15], and Python [10, 11, 15], less research has been focused on the errors students make in database query languages or declarative languages like SQL [16, 17, 18, 19, 20]. The limited research in this area tends to focus on the SQL problems students tend to struggle with. For example, Taipalus et al. [18] found, based on the analysis of over 33,000 SQL queries submitted by students, that students make a variety of syntax, semantic, and logic errors. Some of the syntax errors Taipalus et al. identified students making, like undefined parameter, data type mismatch, and date time field overflow have been identified in prior work [16]. However, Taipalus et al. [18] also identified syntax errors made by students that have not been documented in prior research, which include "IS where not applicable," "duplicate clause," and "confusing table names with column names". Taipalus and Perälä [19] studied persistent error types that students make when forming SQL queries, and the various SQL query concepts that lead to such errors, while others examined different SQL query types that students found challenging to write [21, 22]. Other researchers have studied methodologies in visualizing and detecting students' learning obstacles and approaches [23, 24, 25], and in visualizing SQL queries for improved understanding [26].

Some of the research on the errors students make when learning SQL has been focused on identifying the most common syntax errors [16, 17]. Understanding the common syntax errors made by students when learning SQL could be important for database instructors who teach SQL. With this in mind, the most common error codes that students ran into were 42601 (for PostgreSQL) and 1064 (for MySQL), and the primary reason for these error codes was due to wrong syntax (this error occurred 21% of the time for [16] and 48% of the time for [17]). Additionally, when writing GROUP BY SELECT statements, 68% of students were unsuccessful due to syntactic errors [16], which is approximately 161,000 SQL SELECT statements. Furthermore, 27% of the time students received a syntax error, they were unable to recover from that error and submitted an incorrect final submission [17]. Ahadi et al. [27] conducts further analysis regarding the most common semantic errors as well, while Brass and Goldberg [28] provide for a categorization of semantic errors.

While prior research has documented how syntax errors are an issue for students when learning SQL, not many studies have been focused on why students have these syntax issues. In order to understand why students make syntax errors, a qualitative approach is required. Miedema et al. [20] took such an approach and conducted a think-aloud study with 21 students and identified four reasons why students make mistakes when solving SQL problems. These four reasons include previous course knowledge interfering with their approach to the SQL problem, generalizing answers to questions when they should not, errors in their mental models, and confusing SQL language with natural language [20].

Despite some research investigating why students face syntax errors [20] and other research identifying the frequency of syntax errors [16, 17, 18], very few actionable recommendations have been made to those who teach SQL. One recommendation is to work towards improving error messages for students learning to program in SQL because when compared to other languages, there is very little that is being done in regards to error messages in database engines [17]. Another recommendation is to work on addressing the misconceptions students have that are caused by the transfer of prior knowledge from mathematics, natural language, and other programming languages [20]. However, this recommendation is not unique to teaching SQL and has been documented as a misconception in other programming languages [29, 30].

Prior research has clearly identified that syntax errors are a common error that students encounter when learning SQL [16, 17, 18]. Additionally, these errors can inhibit their ability

complete homework assignments [17]. However, no work to our knowledge has examined how the errors students encounter differ based on whether students are working on homework assignments or exams. Furthermore, little work has made actionable recommendations to database instructors. Based on prior work, our aim is twofold: one, we aim to investigate if students encounter different error types based on the assignment type or if they are in an exam; two, we aim to provide actionable recommendations to database instructors so they can better help students learn SQL.

III. Methodology & Data Collection

Our data is collected from the Database Systems course available to upper-level undergraduate and graduate students at the University of Illinois Urbana-Champaign. The data was collected from the Fall 2022 semester where student enrollment reached 730 students and instruction was given in person following a flipped-classroom model; pre-recorded lecture videos including a quick knowledge check quiz were assigned to students to review prior to the class meeting time. During the class meeting time, the instructor goes over the knowledge check quiz solutions, solves a few practice problems, and addresses any questions or misunderstandings from the students. Students then utilize the remainder of the class time to work on collaborative group exercises to solidify their understanding of the concepts demonstrated in the pre-recorded lecture videos.

A. Assessment Conditions

To facilitate the development and validation of SQL query writing skills, students were assigned three types of assessments: group activities (GAs), homework assignments, and a midterm exam.

The students were given a set of five short collaborative assignments related to SQL, which covered basic SQL and aggregation concepts in the first three assignments, while the remaining two focused on SQL stored procedure and trigger questions. These assignments were released at the beginning of the lecture and had a deadline of approximately 10 days later at midnight. Students collaborated in groups of 3-4, and most groups were able to complete the majority of the assignments before the end of the class. As these assignments were graded based on effort, with full credit given for demonstrating a reasonable amount of effort, students were not expected to experience significant stress. The grading criteria did not depend on passing the auto-grader test cases.

In addition to the collaborative assignments, students were tasked with an SQL homework assignment consisting of 15 programming questions that tested abstract data operation concepts such as selection, projection, grouping, aggregation, and joining. Abstract data operation concepts are essentially the fundamental operations used in data manipulation, transformation, and analysis across various database systems - selection extracts rows based on the given criteria, projection selects columns for a reduced dimensionality, grouping categorizes data for aggregations and summaries, aggregation combines rows for operations such as summation and average, and joining merges data from multiple sources based on shared columns or keys. The questions were designed with an increasing level of difficulty and complexity, which was reflected by the number of concepts tested per question and the complexity of the criteria used to retrieve the required dataset. This assignment had a two-week deadline, and students were graded based on the accuracy of their submissions using an auto-grader. It was expected that students may feel more pressure during this assignment due to the grading criteria. However, those who were unable to correctly answer a problem could still receive partial credit given their final attempt was manually graded, which could have reduced their stress levels.

The first of three midterm exams evaluated students' SQL proficiency, including an intermediate-difficulty SQL programming question (without correlated-subquery concepts) and four other database-related questions in a multiple-choice checkbox format. A subquery is a nested query used to retrieve specific data within a database query. A correlated subquery, on the other hand, establishes a connection with the main outer query and is executed for each row, allowing for data retrieval based on the values of the outer query. Due to the time limitations imposed, the exam programming question did not include correlated subqueries, as they are considered more advanced concepts and challenging to write. Students were allotted 50 minutes to complete the exam, which was proctored and invigilated in a computer-based testing facility to ensure fairness. To promote flexibility, students could choose their own exam time slots, and the facility's network filtering feature ensured that only exam content was viewable. The combination of test anxiety and time constraints during the midterm exam may have caused students to experience heightened stress levels. Although students were granted unlimited attempts on the programming question, each submission attempt required up to 30 seconds for compilation and grading. This may have discouraged students from testing and altering different parts of their query at random.

B. Description of SQL Problems

Our data is collected from PrairieLearn [7], an online learning management system that includes auto-grading capabilities which provide students with immediate feedback for their submitted query attempt. The auto-grader compares the data outputs between the solution SQL query and the student's SQL query to evaluate the correctness of the student's submitted query. Students must pass all test cases in order to receive full credit for any given SQL problem, and no partial credit is awarded. Students may see the discrepancies between their data outputs and the solution query's data outputs to resolve semantic errors (compiled query with logical errors) or the syntax error code to resolve syntax errors (unsuccessful compilation). On all three types of assignments (collaborative group work, homework, and exam), students can answer the questions in any order until the deadline has passed. Students are given unlimited attempts on the homework and collaborative assignments, and up to 100 attempts on the exam question

(which we suspect that no students would be able to use up given the exam time constraint).

An example of an SQL problem and its instructor solution is shown in Figures 1 and 2.

> *Write an SQL query that returns the ProductName of each product made by the brand 'Samsung' and the number of customers who purchased that product. Only count customers who have purchased more than 1 Samsung product. Order the results in descending order of the number of customers and in descending order of ProductName.*

FIG. 1: SQL Homework Problem Statement Example

```
SELECT Pr1.ProductName, COUNT(C1.CustomerId) as numCustomers
FROM Products Pr1 NATURAL JOIN Purchases Pu1
  NATURAL JOIN Customers C1
WHERE Pr1.BrandName = 'Samsung'
  AND C1.CustomerId IN (
    SELECT C2.CustomerId
    FROM Customers C2 NATURAL JOIN Purchases Pu2
      NATURAL JOIN Products Pr2
    WHERE Pr2.BrandName = 'Samsung'
    GROUP BY C2.CustomerId
    HAVING COUNT(C2.CustomerId) > 1
  )
GROUP BY Pr1.ProductName
ORDER BY numCustomers DESC, Pr1.ProductName DESC;
```

FIG. 2: SQL Homework Solution Example

Our resulting dataset consists of 156,513 SQL files. We followed all of our university's Institutional Review Board (IRB) specified data safety protocols to protect the privacy of the students from whom we collected the SQL submissions data. All identifiers were removed from the submission files, and randomized numbers have been assigned to represent each student.

IV. Results

In this section, we present our findings and insights in the order of our research questions; first, we will showcase the error types that students encountered the most. Next, we look at whether errors differ among various assignment types or under exam conditions. Lastly, we make recommendations to database instructors based on our findings to help design more effective coursework for students' learning.

A. Top Syntax Error Types

Figure 3 shows the top 5 syntax error codes that students encountered while working on the exam, collaborative assignment, and homework problems. For readability, we excluded the other syntax error codes since they only consisted of a very small percentage out of all syntax errors. We observed that error code 1064 was the most frequently encountered syntax error code overall in all three assignment types by a high margin. Error code 1064 is a general syntax error code that indicates an error in matching the query structure to the language grammar. While error code 1054 is commonly classified as a syntax error, it can also be considered a semantic error, as it may arise from a misunderstanding of the database schema or the meaning of a particular column. In some cases, the error may be caused by syntax-related issues such as a misspelled column name or a reference to a non-existent column. Our analysis of student performance shows that 1054 errors were more prevalent on exams than on collaborative or homework assignments, which is consistent with previous research [17]. However, we also note that the error is relatively straightforward to fix by identifying and correcting the source of the issue, whether it is a syntax error or a semantic one. Error codes 1055 and 1140 are both related to aggregation concepts, indicating "Summary Column not Included in Group By" and "Summary Column used without Group By," respectively, and are more frequently seen on exam submissions (since the problem tests for the usage of *GROUP BY* and *HAVING* concepts) compared against other assignment types that have a more relaxed deadline time frame. Error code 1052 signifies that "Column identifier is ambiguous," indicating that the student did not specify the table name that the column identifier is coming from when there are multiple column identifiers with the same alias. Error code 1111 signifies that "Invalid use of aggregating function," and is more commonly seen in collaborative and homework assignments.

From these differing error types, we observe that while all three assignment types had the same top two most encountered syntax error types, the frequency that students encountered the 1064 syntax error code drastically declined on the exam; instead, students more frequently encountered typo-related errors (1054) and aggregation-related syntax errors. In our interpretation, this may indicate that students are better at grasping syntax concepts after completing the collaborative and homework assignments, and are encountering syntax errors that are relatively easier to resolve (with clearer error messages).

Exam Error Codes	%	GA Error Codes	%	HW Error Codes	%
1064 (42000)	32.77	1064 (42000)	44.74	1064 (42000)	38.14
1054 (42S22)	25.21	1054 (42S22)	18.53	1054 (42S22)	18.79
1055 (42000)	13.16	1111 (HY000)	6.95	1055 (42000)	9.27
1140 (42000)	9.63	1052 (23000)	5.62	1140 (42000)	5.56
1052 (23000)	4.61	1146 (42S02)	5.19	1111 (HY000)	4.42

FIG. 3: This represents the top 5 syntax error codes that students encountered while working on exam problems, collaborative assignment problems, and homework problems. The percentages shown are the proportion of total submissions that had the particular syntax error code in the respective category (Exam, GA, or HW).

B. Students Progression on Syntax and Semantic Errors

Aside from the syntax error types that we studied in the previous section, we also studied different error types - syntax errors (unsuccessful compilation) and semantic errors (logical errors where the SQL query successfully compiles but fails to output the correct data table). Together, these two error types will account for all errors that we observed; however, the percentages in Figures 4 and 6 do not add up to 100% because it does not account for correct submissions:

$$\textit{Total Submissions (100\%)} = \textit{Syntax Errors} + \textit{Semantic Errors} + \textit{Correct Submissions}$$

We study the progression of error types that students encounter along their learning journey; first, students interact with SQL queries through collaborative assignments where they work in groups of 3-4 people. Next, the week-long homework assignment is released to the students. A few weeks into the semester, the students then take the exam containing one SQL query programming question and a few other multiple-choice checkbox questions. Since the collaborative assignments and homework problems test for various SQL query concepts (correlated subqueries, group by and aggregation, database updates, triggers, stored procedures, etc.), we will only be looking at the problems that intend to test for similar concepts between the collaborative and homework assignments, alongside the exam question. Although there are various approaches students may take to solve an SQL problem (i.e. using a subquery in their solution when it is not necessarily required), we categorize SQL problems based on their intended tested concepts, focusing on the problem description. This resulted in our analysis with GA 1 (group activity) question 4 (selection, projection, and joining concepts), GA 2 question 1 (selection, projection, and joining concepts), GA 2 question 2 (selection, projection, and joining concepts), and SQL homework question 2 (selection and projection concepts). The exam question includes beginner to intermediate-level usage of selection, projection, joining, aggregation, and grouping, with no advanced correlated subquery concepts.

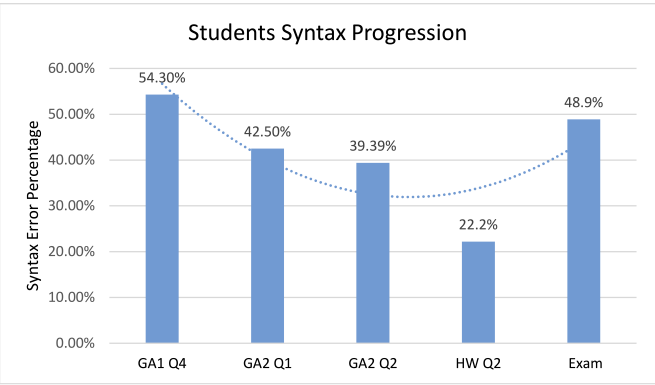

FIG. 4: Since students first worked on collaborative and homework assignments before exam problems, this Figure shows students' progression on the frequency of syntax errors they encountered. As students dedicate more time to learning and practicing SQL through various assignments, the proportion of syntax errors in their total submissions decreases for assignments that have no tight time constraints (homework and collaborative assignments). On the other hand, when students are placed under stressful, timed conditions (i.e. a 50-minute timed exam), the proportion of syntax errors spikes again.

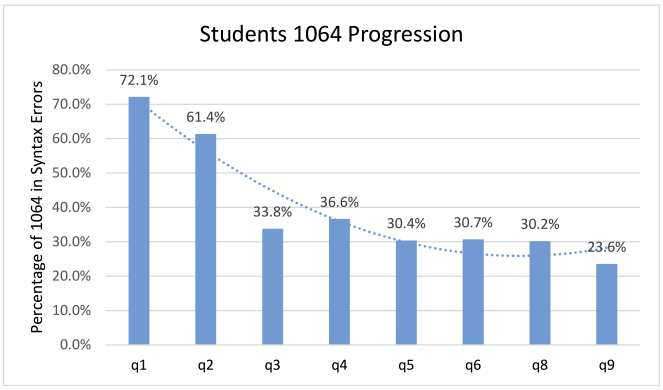

FIG. 5: Error code 1064 is a general syntax error that indicates the query could not be parsed by the query compiler. However, the error message being showcased to the user is generally not regarded as helpful for finding the source of the issue.

Figure 4 showcases the percentage of syntax errors in each of these assignments out of the students' total submissions. We observe that as the time students learn SQL increases, they seem to have encountered fewer syntax errors. This reflects the idea of repairing a flawed mental model, where the flawed areas represent how the syntax error came into being - miscategorization may be a cause. We observe that most syntax errors are either the result of typos or a miscategorization (i.e. using $=$ as an operator instead of the *LIKE* operator).

Since error code 1064 is the universally most frequently encountered syntax error among all three assignment types, we examined this error code more deeply. We observed that students made considerably fewer 1064 syntax errors as they progressed on the homework assignment. In Figure 5, q7 was excluded from our analysis since it assesses students on database update concepts (noticeably different concepts and query formats compared to the earlier questions). We believe that this may indicate increased capabilities in writing SQL queries since there are fewer queries written by students that the compiler cannot parse and determine the type of error (syntax error 1064). An interview study in the future may help to validate this finding.

Figure 6 represents the percentage of semantic errors in each of the assignment questions out of the students' total submissions. We see the opposite trend compared with the syntax errors: as the time students learn SQL increases, they seemed to have encountered more semantic errors. We believe this is a result of the reduction of syntax errors, so students are able to arrive at increased semantic errors. We see a dip in the frequency of semantic errors for exams, because similarly, in Figure 4, we see an increase in syntax errors on exams.

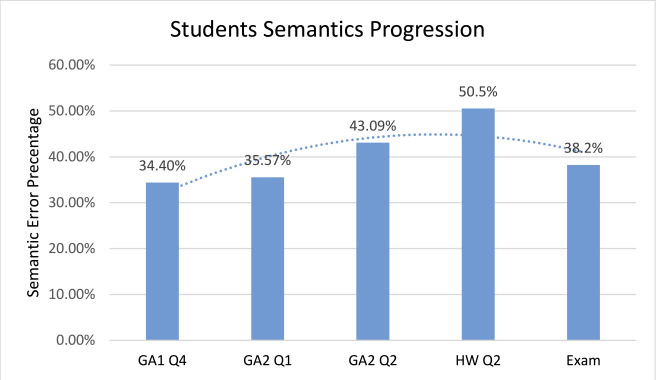

FIG. 6: Among the same questions shown in Figure 4, the proportion of semantic errors in their total submissions increases for assignments that have no tight time constraints (homework and collaborative assignments). On the other hand, when students are placed under stressful, timed conditions (i.e. a 50-minute timed exam), the proportion of semantic errors decreases again.

Students' syntax errors will first be recognized before any semantic errors that take place since the compiler must be able to run the query to indicate a semantic error is present. Therefore, syntax and semantic errors may easily follow an inverse relationship.

C. Discussion

Here, we discuss the implications of our findings. In our earlier section, we observed a spike in the number of syntax errors on exams; we believe this is counterproductive to students' learning due to Trifoni and Shahini [4] idea of test anxiety. We reason that the process of encountering a syntax error will further the cycle of increasing test anxiety due to the following reasons: 1) an increased number of errors on the test (due to existing test anxiety) 2) fear of negative evaluation (when the autograder throws a red syntax error) 3) increased pressure and reduced mental capacity due to time limitations (which is negatively impacted by time lost working on the failed submission and the time necessary for each submissions' compilation), and 4) issues with recalling concepts previously learned (due to test anxiety) [4]. Furthermore, students cannot receive any logical feedback on the correctness of their query; they may only resolve potential semantic errors after all syntax errors have been resolved (due to compilation issues). By increasing test anxiety, students cannot showcase the depth of their understanding of SQL concepts effectively, since their ability to recall SQL concepts may be impacted. Therefore, the validity of the exam may be a concern, since the exam may not capture students' learning fully, as demonstrated in the error data in Figures 4 and 6. The instructor has already taken steps to reduce test anxiety by offering opportunities for students to receive partial credit through manually grading the last submission attempts. This approach may help to alleviate some of the stress and pressure associated with traditional testing methods, but the process of examination and the factor of test anxiety may invite misconceptions to students' existing mental models of SQL query concepts, given the issues with concept recall. The observed escalation in syntax errors correlated to test anxiety could also be linked to the marked contrast in stress levels between the midterm exam and the other SQL assignment types, such as GAs and homework, which had considerably fewer stressors. Due to these negative implications, we make our recommendations to instructors in the next section to mitigate these potential issues.

D. Recommendations for Instructors

Since students consistently displayed progress in lowering the rate at which they encounter syntax errors when working through the collaborative assignment and homework assignment problems, we recommend instructors place a higher emphasis on non-timed assignments with more relaxed deadlines. In particular, we believe that educators should prioritize and implement measures aimed at improving student proficiency in avoiding syntax errors, especially those that are more challenging to address, such as error 1064, which is compounded by its imprecise error message.

These error messages suggest that students struggle with syntax before other issues, similar to what we see in procedural languages [10, 11, 31]; therefore, because syntax skills are a barrier for students, students may benefit from guided syntax drills to foster the necessary skills to avoid such errors. To assist students in diagnosing the root cause of compilation issues associated with error code 1064, instructors should provide guidance on syntax structures that can help clarify common syntax errors. By offering such instruction, students can gain a better understanding of how to identify and address these errors. In addition, since students seem to spend time on syntax errors in timed examples, perhaps it would be appropriate to give learners syntax support during timed exams, so they can allot more time to the semantic components to better showcase their knowledge on exam problems.

V. Limitations and Future Work

Since our study is based on data collected from the University of Illinois Urbana-Champaign, a large public university with a top-ranked Computer Science department, the students and their data may influence the generalizability of our findings. For this purpose, data from other universities or institutions should be collected and analyzed.

We do not know *why* students are making the errors they are making or why some errors might occur more frequently on the homework and group activity assignments versus on the timed exam question. For this reason, we propose to conduct a think-aloud study in the future to better understand the misconceptions students have about SQL.

Upon acquiring individual student submission data for group activities, homework, and exams, outliers from the average submission count came to our attention. Our study centers on examining the overall changes in error percentages between questions, and as such, we made the deliberate choice not to

exclude these outliers, and consequently, the associated students, in order to uphold the integrity and comprehensiveness of our analyses. It is important, however, to recognize that students with higher submission numbers may have underlying reasons driving their actions. Subsequent research endeavors would greatly benefit from conducting a more extensive investigation of these students, enabling a deeper understanding of the factors contributing to their submission patterns.

The SQL problems we chose for syntax and semantics comparisons in Figures 4 and 6, respectively, are based on what we believed to be similar tested concepts by the problem description intention; however, students may use SQL concepts other than the ones intentionally tested to correctly answer the question (which may increase the question's level of difficulty) given the nature of SQL queries. Therefore, our question categorization remains a limitation of our study. Furthermore, because we were unable to select another SQL question from the homework or GA with the same tested concepts and difficulty level as the exam question (which tests for aggregation and grouping in addition), we chose questions that we believed had a similar level of difficulty to make the comparison.

Furthermore, since factors like test anxiety and stress should be studied through either observations or interviews with the students, we propose to conduct a future qualitative study to validate our connections between the assessment environments and our findings. This would allow us to draw stronger associations due to the existence of confounding variables affecting student performance, such as the instructor's ability to develop the students' schema of the topics and skills to handle the cognitive load associated with the exam problems.

Based on our analysis of student SQL queries, we identified several common syntax errors that students encountered, including the "undefined column" error (code 1054) and the "general syntax" error (code 1064). While our study provides valuable insights into the types and frequency of syntax errors that students encounter, further research is needed to develop more effective interventions for addressing these errors. In particular, we plan to conduct a follow-up study that focuses specifically on error code 1064, which encompasses a range of syntax-related issues. Our goal is to identify the most common subtypes of this error and explore strategies for addressing them. We would also like to explore integrating a layer on top of the SQL engine that can provide useful tips for general errors, such as code 1064. This will also help us better define the types of syntax errors in more detail, based on SQL query concepts.

Since students make more syntax errors and fewer semantic errors on exam questions, they may benefit from reviewing syntax concepts with targeted homework drills that lessen the stress factor. However, exam questions help encourage students to study SQL concepts more and pinpoint where they are struggling. For future study, we are interested in studying whether alternating exam or timed SQL programming questions with homework assignment problems will help students better acquire SQL concepts and syntax rules.

VI. Conclusion

In this research work, we presented our findings from 156,513 student SQL submissions collected from the Database Systems course at the University of Illinois Urbana-Champaign. For our research questions - 1) What are the error types that students encounter the most? and 2) Do the error types that students encounter differ among various assignment types or under exam conditions? - we found that syntax error codes 1064 and 1054 were universally the most frequently occurring syntax errors. Then, depending on the environment under which the student completed the SQL problem, the syntax error codes varied; for timed-exam questions, students faced more summary column and group-by-related syntax errors. For homework and collaborative assignment problems with longer deadlines, students faced more varied types of errors resulting from typos and misunderstandings with the usages of the summary columns and Group-By-related concepts.

We also found that as students learn SQL for a longer period of time, they seem to acquire a better grasp of the syntax rules and make fewer syntax errors (and therefore more semantic errors). However, under stressful conditions such as a timed exam, students make many more syntax errors and fewer semantic errors.

We, therefore, recommended that instructors place a higher emphasis on non-timed SQL programming problems, targeted syntax drills during instruction, and syntax support during exams. For non-timed SQL programming problems - homework assignments, in particular, seemed promising for helping students to lower the rate at which they encounter syntax error code 1064.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 2021499.

References

[1] A. DiFranza, "5 reasons sql is the need-to-know skill for data analysts," 2020.

[2] A. Mitrovic, "Learning sql with a computerized tutor," in *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '98. New York, NY, USA: ACM, 1998, p. 307–311.

[3] J. Vigo, "5 programming languages database administrators should learn," November 2020.

[4] A. Trifoni and M. Shahini, "How does exam anxiety affect the performance of university students?" *Mediterranean journal of social sciences*, vol. 2, no. 2, pp. 93–93, 2011.

[5] B. A. T. WELFORD, "Stress and performance," *Ergonomics*, vol. 16, no. 5, pp. 567–580, 1973, pMID: 4772982. [Online]. Available: https://doi.org/10.1080/00140137308924547

[6] J. Murphy, J.-M. Chang, and K. Suaray, "Student performance and attitudes in a collaborative and flipped linear algebra course," *International Journal of*

Mathematical Education in Science and Technology, vol. 47, no. 5, pp. 653–673, 2016. [Online]. Available: https://doi.org/10.1080/0020739X.2015.1102979

[7] M. West, G. L. Herman, and C. B. Zilles, "Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning," 2015.

[8] R. C. Jinkens, "Nontraditional students: Who are they?" *SIGCSE Bull.*, vol. 43, no. 4, pp. 979–987, Dec. 2009.

[9] L. C. Kaczmarczyk, E. R. Petrick, J. P. East, and G. L. Herman, "Identifying student misconceptions of programming," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 107–111. [Online]. Available: https://doi.org/10.1145/1734263.1734299

[10] L. Mannila, M. Peltomäki, and T. Salakoski, "What about a simple language? analyzing the difficulties in learning to program," *Computer science education*, vol. 16, no. 3, pp. 211–227, 2006.

[11] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *Acm sigcse bulletin*, vol. 37, no. 3, pp. 14–18, 2005.

[12] A. E. Fleury, "Programming in java: Student-constructed rules," in *Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, 2000, pp. 197–201.

[13] H.-C. Woon and Y.-T. Bau, "Difficulties in learning c++ and gui programming with qt platform: View of students," in *Proceedings of the 2017 International Conference on E-commerce, E-Business and E-Government*. New York, NY, USA: ACM, 2017, pp. 15–19.

[14] J. Bergin, "Java as a better c++," *ACM SIGPLAN Notices*, vol. 31, no. 11, pp. 21–27, 1996.

[15] N. Alzahrani, F. Vahid, A. Edgcomb, K. Nguyen, and R. Lysecky, "Python versus c++ an analysis of student struggle on small coding exercises in introductory programming courses," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: ACM, 2018, pp. 86–91.

[16] A. Ahadi, V. Behbood, A. Vihavainen, J. Prior, and R. Lister, "Students' syntactic mistakes in writing seven different types of sql queries and its application to predicting students' success," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. New York, NY, USA: ACM, 2016, pp. 401–406.

[17] S. Poulsen, L. Butler, A. Alawini, and G. L. Herman, "Insights from student solutions to sql homework problems," in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM, 2020, pp. 404–410.

[18] T. Taipalus, M. Siponen, and T. Vartiainen, "Errors and complications in sql query formulation," *ACM Transactions on Computing Education (TOCE)*, vol. 18, no. 3, pp. 1–29, 2018.

[19] T. Taipalus and P. Perälä, "What to expect and what to focus on in sql query teaching," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 198–203. [Online]. Available: https://doi.org/10.1145/3287324.3287359

[20] D. Miedema, E. Aivaloglou, and G. Fletcher, "Identifying sql misconceptions of novices: Findings from a think-aloud study," in *Proceedings of the 17th ACM Conference on International Computing Education Research*, ser. ICER 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 355–367. [Online]. Available: https://doi.org/10.1145/3446871.3469759

[21] A. Ahadi, J. Prior, V. Behbood, and R. Lister, "A quantitative study of the relative difficulty for novices of writing seven different types of sql queries," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '15. New York, NY, USA: ACM, 2015, p. 201–206.

[22] A. Migler and A. Dekhtyar, "Mapping the sql learning process in introductory database courses," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 619–625. [Online]. Available: https://doi.org/10.1145/3328778.3366869

[23] S. Yang, Z. Wei, G. L. Herman, and A. Alawini, "Analyzing patterns in student sql solutions via levenshtein edit distance," in *Proceedings of the Eighth ACM Conference on Learning @ Scale*, ser. L@S '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 323–326. [Online]. Available: https://doi.org/10.1145/3430895.3460979

[24] S. Yang, G. L. Herman, and A. Alawini, "Analyzing student sql solutions via hierarchical clustering and sequence alignment scores," in *1st International Workshop on Data Systems Education*, ser. DataEd '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 10–15. [Online]. Available: https://doi.org/10.1145/3531072.3535319

[25] ——, "Mining sql problem solving patterns using advanced sequence processing algorithms," in *Proceedings of the 2nd International Workshop on Data Systems Education: Bridging Education Practice with Education Research*, ser. DataEd '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 37–43. [Online]. Available: https://doi.org/10.1145/3596673.3596973

[26] J. Danaparamita and W. Gatterbauer, "Queryviz: Helping users understand sql queries and their patterns," in *Proceedings of the 14th International Conference on Extending Database Technology*, ser. EDBT/ICDT '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 558–561. [Online]. Available: https://doi.org/10.1145/1951365.1951440

[27] A. Ahadi, V. Behbood, A. Vihavainen, J. Prior, and

R. Lister, "Students' semantic mistakes in writing seven different types of sql queries," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 272–277. [Online]. Available: https://doi.org/10.1145/2899415.2899464

[28] S. Brass and C. Goldberg, "Semantic errors in sql queries: A quite complete list," *Journal of Systems and Software*, vol. 79, no. 5, pp. 630–644, 2006, quality Software. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016412120500124X

[29] M. Clancy, "Misconceptions and attitudes that interfere with learning to program," in *Computer science education research*. Taylor & Francis, 2005, pp. 95–110.

[30] Y. Qian and J. Lehman, "Students' misconceptions and other difficulties in introductory programming: A literature review," *ACM Transactions on Computing Education (TOCE)*, vol. 18, no. 1, pp. 1–24, 2017.

[31] M. C. Linn and M. J. Clancy, "The case for case studies of programming problems," *Communications of the ACM*, vol. 35, no. 3, pp. 121–132, 1992.