



# Development of Grader System for Data Manipulation Language

Author: Mr. Teeranai Charoenworarak

Advisor: Asst. Prof. Dr. Areerat Trongratsameethong

## ABSTRACT

Traditionally, grading of SQL and NoSQL queries relies on manual grading, a time-consuming and error-prone process. To address these challenges, this research introduces an automated grading system for SQL and NoSQL queries. The system employs tokenization and parsing to meticulously analyze query structures, functions, and conditions, then compares student submissions with the solutions provided by the instructor.

A standout feature is its partial scoring capability, which awards points for each correctly written segment of a query. This approach not only enhances fairness but also encourages deeper conceptual understanding. In the SQL module, the system validates correctness of commands such as SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY. For the NoSQL module, it supports functions like find and aggregate and evaluates operators including \$eq, \$gt, and \$lt. This comprehensive method allows for flexible grading while offering immediate, detailed feedback on errors.

In addition, the system includes robust functionalities for assignment creation, and performance reporting. Instructors can efficiently create and manage assignments, define grading criteria, and review student progress. Meanwhile, students benefit from real time feedback on their submissions, enabling them to quickly identify and correct mistakes. Developed with Django for the backend and React for the frontend and leveraging PostgreSQL and MongoDB for structured and unstructured database.

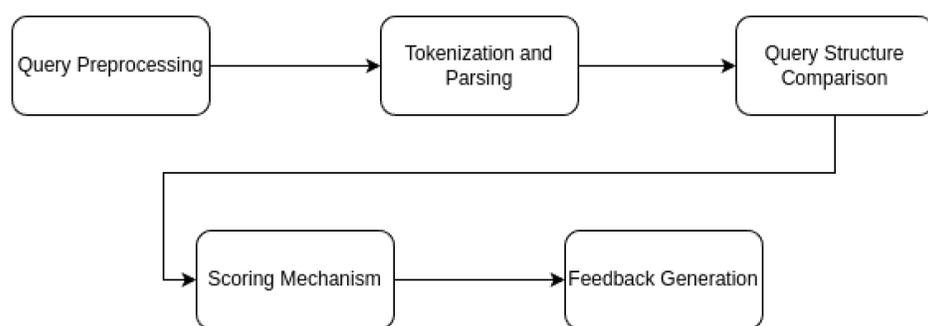
Test results show that the system significantly reduces instructors' workload, accelerates feedback delivery, and effectively aids students in understanding and rectifying errors. As a result, it enhances database education quality whether in relational or non-relational systems, improving learning experiences in the digital era.

## INTRODUCTION

Grading SQL and NoSQL queries is a crucial but time-consuming task in database education. Traditional grading methods rely on manual assessment, which is prone to human errors, inconsistencies, and delays in providing feedback to students. As database queries can be written in multiple syntactically correct ways while achieving the same result, evaluating their correctness is a challenging process that requires both syntactic and semantic analysis.

To address these challenges, this research introduces an automated grading system for SQL and NoSQL queries. It then compares student responses to instructor-provided solutions, identifying errors and assigning scores based on predefined grading criteria. Unlike conventional grading systems that adopt an all-or-nothing approach, this system implements partial scoring, awarding points for correctly formulated query components. This approach enhances fairness and encourages students to refine their database skills.

## METHODOLOGY



## TECHNOLOGY



## REFERENCE

[1] M. Messer, N. C. C. Brown, M. Kölling, and M. Shi, "Automated grading and feedback tools for programming education: A systematic review," arXiv preprint arXiv:2306.11722, 2023.

[2] Kirsti M. Ala-Mutka, "A survey of automated assessment approaches for programming assignments," Computer Science Education, vol. 15, no. 2, pp. 83–102, 2005. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/08993400500150747#d1e184>.

## RESULT

Overall, the program performs as expected within the predefined environment.

Test Case	Data base Type	Sample Query	Student Query	Max Score	Expected Score	Actual Score
TC_1	SQL	SELECT * FROM students	SELECT * FROM students	100	100	100
TC_2	SQL	SELECT name, age FROM students	SELECT name, age FROM students	100	100	100
TC_3	NO SQL	db.students.find({ age: { \$gt: 20 } }, { name: 1, _id: 0 })	db.students.find({ age: { \$gt: 20 } }, { name: 1, _id: 0 })	100	100	100
TC_4	NO SQL	db.students.find({ name: 1, age: 1, _id: 0 })	db.students.find({ name: 0, age: 0, _id: 0 })	100	33.3	33.3

The program successfully evaluates SQL and NoSQL queries, producing accurate results within the predefined environment. The test cases demonstrate that the system correctly processes queries, matches expected outcomes, and provides consistent scoring. Future improvements could focus on expanding query support, enhancing error handling, and optimizing performance. Despite these limitations, the system proves to be a valuable tool for automated query assessment and grading.

## CONCLUSION

The development of an automated grading system for SQL and NoSQL queries has proven to be highly effective in streamlining the assessment process. The implementation of partial scoring allows students to receive credit for correctly written segments of their queries, promoting a deeper understanding of database concepts. This feature provides a more nuanced and fair grading approach compared to traditional binary grading methods, where an entire query is marked incorrect due to minor mistakes.