Are Large Language Models Smart Enough for SQL Tutoring and Assessment?

Kallol Naha
Department of Computer Science
University of Idaho, USA
naha7197@vandals.uidaho.edu

Sajratul Y Rubaiat
Department of Computer Science
University of Idaho, USA
ruba3062@vandals.uidaho.edu

Hasan M Jamil
Department of Computer Science
University of Idaho, USA
jamil@uidaho.edu

***Abstract*—The rise of Large Language Models (LLMs) as powerful knowledge-processing tools has sparked a wave of innovation in tutoring and assessment systems. Despite their well-documented limitations, LLMs offer unique capabilities that have been effectively harnessed for automated feedback generation and grading in intelligent learning environments. In this paper, we introduce *Project 360*, an experimental intelligent tutoring system designed for teaching SQL. Project 360 leverages the concept of *query equivalence* to assess the accuracy of student queries, using ChatGPT's advanced natural language analysis to measure their semantic distance from a reference query. By integrating LLM-driven evaluation, Project 360 significantly outperforms traditional SQL tutoring and grading systems, offering more precise assessments and context-aware feedback. This study explores the feasibility and limitations of using ChatGPT as the analytical backbone of Project 360, evaluating its reliability for autonomous tutoring and assessment in database education. Our findings provide valuable insights into the evolving role of LLMs in education, highlighting their potential to revolutionize SQL learning while identifying areas for further refinement and improvement.**

***Index Terms*—Large language model, database, SQL, authentic assessment, intelligent tutoring, query equivalence.**

I. Introduction

Tutoring and grading programming assignments have traditionally relied on functional [20] or semantic [7] equivalence testing, assuming that similarities in program structure and control flow imply program equivalence [10]. In this context, the semantic distance $\delta(P, P')$ between two programs, P and P', serves as a measure of error severity. A tutoring system's primary task is to analyze these errors, assess their significance, and generate constructive feedback [19] for both grading [14] and instructional support.

While procedural languages such as C++, Python, and Java pose significant challenges for equivalence testing and feedback generation due to their extensive vocabulary and complex programming constructs, SQL – a declarative query language – seemingly offers fewer obstacles due to its more constrained syntax. However, the reality is quite different. Developing a robust mechanism for testing SQL query equivalence has proven to be a persistent challenge.

Since SQL operates within the same theoretical framework as Relational Algebra (RA) and Datalog, query equivalence testing methods developed for one paradigm can often be adapted to the others with minimal modification. Several studies have explored query containment [13] and equivalence [21], both of which are fundamental to building SQL tutoring and grading systems [17]. However, implementing these techniques remains a complex task. Among contemporary SQL and Relational Algebra [11] tutoring systems, only ViSQL [12] most likely has attempted to incorporate query equivalence testing, leveraging the Cosette automated SQL solver's REST API [6] for grading. Unfortunately, Cosette's inherent language limitations made it impractical for grading even moderately complex SQL queries commonly taught in introductory database courses, and it lacked any mechanism for generating meaningful feedback.

Recent research has explored the potential of modern Large Language Models (LLMs) for feedback generation [18] and grading [2], yet their effectiveness in a structured tutoring and assessment system has not been systematically evaluated. In this paper, we investigate the feasibility of leveraging an LLM, specifically ChatGPT-4o, for SQL query equivalence testing, the cornerstone of an intelligent tutoring and assessment system. Additionally, we assess ChatGPT's capability in generating precise, pedagogically valuable feedback for incorrect queries. We introduce Project 360, an experimental SQL tutoring system, and benchmark its performance against contemporary tutoring solutions, providing insights into the evolving role of LLMs in database education.

II. Use Case: Feedback Generation and Grading of SQL Division Queries

The concept of the division operation in relational query languages such as SQL and Relational Algebra is one of the most difficult ones. Let us assume that an instructor asked the students to construct an SQL query Q_1 below over the database scheme in Fig. 1.

Q_1: list students who took all the database courses.

Naturally, the answer to the query is the student S_1 in Fig. 1(d), which the SQL query Q_r below returns.

```
select s.StuID, s.Name
from Students as s
where (select t.CourseID
    from Takes as t
    where s.StuID = t.StuID)
    contains
    (select c.CourseID
```

	StuID	Name	Age
t_1	S_1	Alice	18
t_2	S_2	Nancy	19
t_3	S_3	Peter	19

(a) Table *Students*.

	StuID	CourseID
t_4	S_1	CS460
t_5	S_1	CS360
t_6	S_3	CS120

(b) Table *Takes*.

	CourseID	Title	Group
t_7	CS360	Intro DB	DB
t_8	CS460	Adv. DB	DB
t_9	CS120	Python	PL

(c) Table *Courses*.

StuID	Name
S_1	Alice

(d) Query result.

Fig. 1. An Example Database to Explain SQL Division Query.

```
        from Courses as c
        where c.Group = ”DB”)
```

Now suppose the student writes the SQL query Q_t below which happens to be functionally and semantically equivalent to query Q_r.

```
    select StuID, Name
    from Students
    where StuID in
        (select StuID
        from ((select StuID, count(*) as Total
                from Takes natural join Courses
                where Group=”DB”
                group by StuID) natural join
                        (select count(*) as Total
                        from Courses
                        where Group=”DB”)))
```

Or, the query $Q_{t'}$ below, which too is equivalent to Q_r.

```
    with (
        select StuID count(*) as Total
        from Takes natural join Courses
        where Group=”DB”
        group by StuID
        ) as StuTotal,
        (
        select count(*) as Total
        from Courses
        where Group=”DB”
        ) as DBTotal
    select StuID, Name
    from Students natural join StuTotal natural join DBTotal
```

Designing a tutoring or assessment system based on query equivalence testing has long been a challenge, as existing approaches, such as those used in Cosette [6], QED [21], and similarity-based query equivalence techniques [15], have proven inadequate for practical implementation. An additional challenge arises when two queries are determined to be non-equivalent: how should meaningful and pedagogically valuable feedback be generated to help students understand the difference between their submitted query (Q_t) and the reference query (Q_r)?

A recent research on NQL [4] and ExplainS [8], two fundamental questions were explored: (1) whether SQL queries can be accurately and efficiently generated, and (2) how useful feedback can be provided when student queries are incorrect. The goal is to automate the generation of the reference query Q_r using ChatGPT, allowing equivalence testing against a student's query Q_t without requiring explicit instructor input. In this approach, the instructor's role is limited to designing the SQL assignment or test, while an intelligent tutoring system, such as Project 360, handles evaluation and feedback generation autonomously.

The Gemini-based ExplainS required an elaborate Abstract Syntax Tree (AST)-driven architecture to detect dissimilarities and edge cases in student queries. In contrast, ChatGPT-4o demonstrated remarkable efficiency in query equivalence testing. Beyond merely identifying whether Q_r and Q_t are equivalent, ChatGPT-4o can explain why they are equivalent, generate example tables illustrating potential discrepancies, and highlight scenarios where they may not produce functionally identical results.

Based on these observations, we hypothesize that ChatGPT-4o can serve as both an automated SQL equivalence testing system and a personalized feedback generator, offering an effective and scalable solution for SQL tutoring and assessment.

III. Evaluation of Project 360

While Project 360 has been implemented and tested with hundreds of SQL queries, its formal evaluation followed two key axes.

1) *Comparison with Existing Systems:* Does Project 360 answer the following key questions in a manner consistent with Gemini, CoPilot, and human experts?
    a) Given two SQL queries, Q_r and Q_t, as presented in Sec. II, can it correctly interpret their meanings?
    b) Regardless of database contents, will these queries produce identical results? In other words, are they logically equivalent?
    c) Can it generate example tables where the two queries yield different outputs?
    d) Are there SQL queries where equivalence cannot be determined? Can it provide an example?
2) *Tutoring and Grading Accuracy:* Does Project 360 accurately tutor and grade query Q_1 (discussed in Sec. II) based on the schema of the student database D (Fig. 1)?

For the four evaluation questions in the first axis, all LLMs tested – ChatGPT-4o, Gemini, and CoPilot – provided nearly identical responses. They unanimously determined that the queries were equivalent, which was correct. However, they exhibited some difficulty with the contains operator, as it is not a standard SQL set operation. Furthermore, since primary key constraints were not included in the prompts, all models inferred that the queries might produce different results in edge cases (e.g., empty databases, duplicate rows). These findings suggest that ChatGPT-4o is well-suited for SQL query equivalence testing, making it a viable engine for Project 360.

For the second axis, we tested Project 360 in both Grading Mode and Tutoring Mode. In both modes, students could load, view, and submit responses for a wide class (select-project-join queries, set queries, sub-queries, correlated sub-queries) of 42 test questions representative of a first-database course. Upon submission, Project 360:

- Generated the reference query (Q_r) using its Text-to-SQL engine.
- Compared the student's query (Q_t) for equivalence
- Assigned a grade (including partial credit, when applicable).
- Generated an explanation detailing why the solution was correct or incorrect.

In this initial edition of Project 360, grading was performed using ChatGPT's native grading capability. However, future iterations will incorporate a custom-designed partial grading mechanism, following methodologies such as [5], to refine the assessment process.

IV. Discussion

The modular design and the architecture of Project 360 open the door to a fully autonomous tutoring and assessment system. This advancement raises several possibilities:

- Automated assignment generation on demand [9].
- Personalized learning paths tailored to student profiles and learning outcomes [16].
- Autonomous reference query generation and comparison.

Since Project 360 can dynamically construct test databases, generate assignments, and assess solutions without human intervention, future research will focus on refining these capabilities and evaluating their impact on student learning outcomes. Project 360 is available for public use at http://dblab.nkn.uidaho.edu/project360/.

V. Conclusion

The primary goal of this paper is to examine the feasibility of LLMs as SQL tutors and assessors. More broadly, it raises an important question: Can LLMs effectively assume the role of intelligent instructors in database education? While LLMs exhibit impressive reasoning abilities, their logical consistency and cognitive limitations remain areas of concern. As a result, we must ask: Should we fully entrust them with tutoring and assessment despite their known limitations and occasional hallucinations?

Our experience suggests that LLMs, particularly ChatGPT-4o, warrant serious consideration as intelligent SQL tutors and assessors. Given the positive results, we encourage the research community to further explore LLM-driven tutoring, a practice that is already gaining traction [1, 3]. While challenges remain, they primarily lie in system design, user experience, and implementation scalability – not in the underlying LLM technology itself. Moving forward, the focus should be on enhancing intelligent tutoring systems through better tooling and pedagogical design, ensuring that LLM-driven education remains effective, adaptive, and student-centered.

Acknowledgment

This research was made possible partially by an Institutional Development Award (IDeA) from the National Institutes of Health grant P20GM103408, National Science Foundation CSSI grant OAC 2410668, and US Department of Energy grant DE-0011014.

References

[1] T. Balart and K. J. Shryock. Work in progress: Empowering engineering education with chatgpt: A dive into the potential and challenges of using AI for tutoring. In *EDUCON*, pages 1–3, 2024.

[2] L. Cagliero, L. Farinetti, J. Fior, and A. I. Manenti. Chatgpt, be my teaching assistant! automatic correction of SQL exercises. In *COMPSAC*, pages 81–87, 2024.

[3] C. Cao. Leveraging large language model and story-based gamification in intelligent tutoring system to scaffold introductory programming courses: A design-based research study. *CoRR*, abs/2302.12834, 2023.

[4] N. Carr, F. Shawon, and H. Jamil. An experiment on leveraging ChatGPT for online teaching and assessment of database students. In *TALE*, pages 1–8, 2023.

[5] B. Chandra, M. Joseph, B. Radhakrishnan, S. Acharya, and S. Sudarshan. Partial marking for automated grading of SQL queries. *Proc. VLDB Endow.*, 9(13):1541–1544, 2016.

[6] S. Chu, D. Li, C. Wang, A. Cheung, and D. Suciu. Demonstration of the cosette automated SQL prover. In *SIGMOD*, pages 1591–1594, 2017.

[7] B. R. Churchill, O. Padon, R. Sharma, and A. Aiken. Semantic program alignment for equivalence checking. In *ACM SIGPLAN PLDI*, pages 1027–1040, 2019.

[8] H. Clark and H. M. Jamil. Mixing up gemini and AST in explains for authentic sql tutoring. In *TALE*, pages 1–8, 2024.

[9] E. Dhanya and K. N. Nikhila. Programming question generation: An automated methodology for generating novel programming assignments with varying difficulty levels. In *AAGPW Workshop (AIED)*, volume 3572 of *CEUR*, pages 27–35, 2023.

[10] S. Ito. Semantical equivalence of the control flow graph and the program dependence graph. *CoRR*, abs/1803.02976, 2018.

[11] H. M. Jamil, K. Naha, and F. R. Shawon. An online tutoring and assessment system for teaching relational algebra in database classes. In *ICWL 2023*, LNCS, pages 62–76, 2023.

[12] M. Karimzadeh and H. M. Jamil. ViSQL: An intelligent online SQL tutoring system. In *ICALT*, pages 212–213, 2022.

[13] M. A. Khamis, P. G. Kolaitis, H. Q. Ngo, and D. Suciu. Bag query containment and information theory. *TODS*, 46(3):12:1–12:39, 2021.

[14] C. Kleiner and F. Heine. Enhancing feedback generation for autograded SQL statements to improve student learning. In *ITiCSE*, 2024.

[15] L. Köberlein, D. Probst, and R. Lenz. Quantifying semantic query similarity for automated linear SQL grading: A graph-based approach. *CoRR*, abs/2403.14441, 2024.

[16] M. Mosbeck, D. Hauer, and A. Jantsch. VELS: VHDL e-learning system for automatic generation and evaluation of per-student randomized assignments. In *IEEE NORCAS: NORCHIP and SoC*, pages 1–7, 2018.

[17] G. Rull, P. A. Bernstein, I. G. dos Santos, Y. Katsis, S. Melnik, and E. Teniente. Query containment in entity SQL. In *SIGMOD*, pages 1169–1172, 2013.

[18] J. Sengewald, M. Wilz, and R. Lackes. Ai-assisted learning feedback: Should gen-ai feedback be restricted to improve learning success? A pilot study in a SQL lecture. In *ECIS*, 2024.

[19] M. Shaka, D. Carraro, and K. N. Brown. Error tracing in programming: A path to personalised feedback. In *BEA*, pages 330–342. ACL, 2024.

[20] Y. Sun, R. Ji, J. Fang, X. Jiang, M. Chen, and Y. Xiong. Proving functional program equivalence via directed lemma synthesis. In *FM*, volume 14933 of *LNCS*, pages 538–557. Springer, 2024.

[21] S. Wang, S. Pan, and A. Cheung. QED: A powerful query equivalence decider for SQL. *Proc. VLDB Endow.*, 17(11):3602–3614, 2024.