WIP: AI-Powered Knowledge Graph System for Personalized SQL Learning and Adaptive Feedback in Database Systems Education

Weijian Zhou, University of Illinois at Urbana-Champaign

Weijian Zhou is an undergraduate student in Computer Science + Philosophy at the University of Illinois Urbana-Champaign. His research interests include the application of artificial intelligence to computing education, with a focus on improving learning experiences and educational outcomes.

Abdulrahman AlRabah, University of Illinois at Urbana - Champaign

Abdulrahman AlRabah is a Ph.D. student in the Siebel School of Computing and Data Science at the University of Illinois Urbana Champaign, where he also earned his M.S. in Computer Science. He received his B.S. in Mechanical Engineering from California State University Northridge and has industry experience including oil and gas and co founding a food and beverage company. His research focuses on AI for computing education, using customized large language models and learning analytics, including knowledge graph based concept modeling, to deliver better feedback, improve student learning, and support instructors.

Dylan Dunham, University of Illinois at Urbana - Champaign

Dylan Dunham is a Master's student in Computer Science at the University of Illinois Urbana-Champaign, where he previously earned his B.S. in Computer Science. He has industry experience at leading technology companies including Uber and Amazon, working on scalable systems and distributed infrastructure. His research focuses on artificial intelligence for computing education, with an emphasis on leveraging large language models and knowledge graph construction to enable personalized learning and adaptive feedback.

Dr. Abdussalam Alawini, University of Illinois at Urbana - Champaign

Dr. Alawini is a Teaching Associate Professor in the School of Computing and Data Science at the University of Illinois at Urbana-Champaign. He holds a bachelor's degree from the University of Tripoli and master's and doctoral degrees in Computer Science from Portland State University. His research focuses on data management systems and computing education. Dr. Alawini is passionate about building data-driven, AI-based systems for improving teaching and learning.

WIP: AI-Powered Knowledge Graph System for Personalized SQL Learning and Adaptive Feedback in Database Systems Education

Abstract

In database systems education, students often struggle with SQL assignments due to generic autograder feedback, while instructors lack scalable tools to diagnose class-wide misconceptions. We present an AI-powered platform that integrates a fine-tuned LLM with a curriculum-derived Knowledge Graph to (1) classify incorrect SQL submissions into seven semantic error types, (2) map each submission to underlying SQL concepts and prerequisites, (3) generate targeted non-revealing feedback, and (4) aggregate results into instructor analytics to surface error clusters and mastery trends. In a preliminary evaluation on 56 real student submissions, the system achieves 80.35% Hit@2 for error-type labeling and 92.85% Hit@k for concept mapping, providing early evidence of reliable misconception tracking. Future work will conduct an instructor-centered evaluation to assess the system's ability to guide timely interventions in live classroom settings.

1 Introduction

SQL is a core topic in many database systems courses. Students often write queries that run into errors, which can hinder their learning journey. Existing automated feedback systems primarily address syntactic errors or provide generic hints, but few connect semantic error diagnosis to concept-level prerequisites, leaving both students and instructors without actionable, concept-grounded insights. Syntactic errors are often straightforward to diagnose and correct, whereas semantic errors are typically more challenging because a query may execute without failure yet still return incorrect results. It is a challenge for students to pinpoint the underlying misconception in their SQL reasoning [1, 2]. At scale, instructors face a similar challenge in identifying both the specific error types and the underlying concept gaps. This matters because wrong or vague diagnosis leads to wasted attempts, with students repeatedly making random edits without learning the concept they need. At the same time, instructors need a quick way to see class-wide patterns so they can adjust teaching and provide targeted practice [3, 4, 5]. For example, a query missing a `GROUP BY` clause triggers a syntax error, but one that groups by the wrong attribute silently returns incorrect results, diagnosing such semantic mistakes requires more than a compiler. Generative AI has made a promising breakthrough for SQL learning, though the challenge remains how to cater to each student's needs and provide closer insights to instructors [6].

In this work-in-progress (WIP) paper, we present a novel approach that connects error diagnosis with concept-level understanding in a knowledge graph setting. The system uses an LLM-based classifier to categorize student SQL mistakes into seven error types (JOIN, GROUP BY, WHERE, alias errors, misuse of aggregate functions, subquery logic errors, and ambiguous column references) drawn from past research [1]. It also builds a SQL knowledge graph in Neo4j [7] by extracting concept relationships from course materials. Using this graph, the system maps submissions to relevant concepts and their prerequisites, and aggregates results into an instructor dashboard that highlights class-wide learning patterns and common misconception clusters. As an initial validation, we report preliminary hit-rate results from an early experiment, showing the system behaves as expected. We use Hit@k, meaning the percentage of submissions where the correct label is included in the model's top-k predicted labels (e.g Hit@2 checks the top two). This paper addresses three goals: (1) automatically classify semantic SQL errors into typed categories, (2) map errors to prerequisite concepts in a course knowledge graph, and (3) surface class-wide misconception patterns via an instructor dashboard.

2 Related Work

SQL tutoring systems have approached error diagnosis through taxonomies, parse-tree comparison, and rubric-based classification. Ponzini et al. [8, 9] characterize common student SQL mistakes using a detailed taxonomy and propose an interactive tutoring framework that surfaces "tutor actions" (e.g., Explain Error, Locate Error, Provide an Example) through a hybrid diagnosis pipeline that combines the taxonomy with parsing, rules, and metadata, and uses expected solutions and prompts for harder cases. They also generate explanations with a LLM, and deployment logs show substantial use in practice. Kleiner and Heine [10] compare student and reference parse trees to generate similarity-based partial credit and structural hints, while Nayak et al. [11] use rubric-driven classical ML with text and similarity features to classify correctness and generate error-style feedback and student clusters. In contrast, we link semantic-error diagnosis to a course concept prerequisite knowledge graph to activate concept-level.

AlRabah et al. [12] fine-tune a GPT-family model to produce short, instructor-style non-revealing hints for semantic issues, and Alrabah and Alawini [13] extend this into *CodeLens*, a two-stage pipeline that detects semantic issues from contextual signals and produces fine-grained feedback that pinpoints errors in student code. Controlled studies report fewer submission attempts and strong semantic-error detection metrics. Naha et al. [14] introduce *Project 360*, using an LLM to judge query equivalence and semantic distance and to generate explanations, including counterexample tables, for grading and tutoring. Utkan and Öztayşi [15] evaluate GPT-4o for rubric-based exam feedback and find good overall alignment, but also note errors when the model behaves like a strict SQL engine. Unlike these feedback only systems, our approach maps errors to prerequisite-aware concepts.

Yang et al. [16] use GPT-generated concepts to build a Knowledge Graph (KG) for explainable course recommendation, while Canal-Esteve and Gutiérrez [17] propose an LLM-centered text for a KG pipeline to support navigation and personalization. Aytekin et al. [18] use an expert-in-the-loop workflow with graph reasoning, such as transitivity, to recover prerequisite paths efficiently. Alatrash et al. [19] infer prerequisites without labels by combining multiple external signals with a precision-oriented voting scheme. *KnowEdu* similarly extracts concepts from text and mines prerequisites from assessment data at scale Chen et al. [20]. KGs are also used for adaptive learning and curriculum analysis, including ontology and KG-based learning-path recommendation Lin and Wang [21] and syllabus-derived curriculum KGs with symbolic checks for gaps and ordering issues Gacek and Adrian [22]. Surveys and systems further motivate KG+LLM in education while noting challenges such as incomplete KGs and hallucinations Chen et al. [20], Lu et al. [23].

No prior work combines semantic SQL error classification with a prerequisite-aware course KG. Our work bridges this gap by mapping submissions to semantic error types and prerequisite concept nodes, enabling targeted feedback and instructor dashboards.

3 Methods

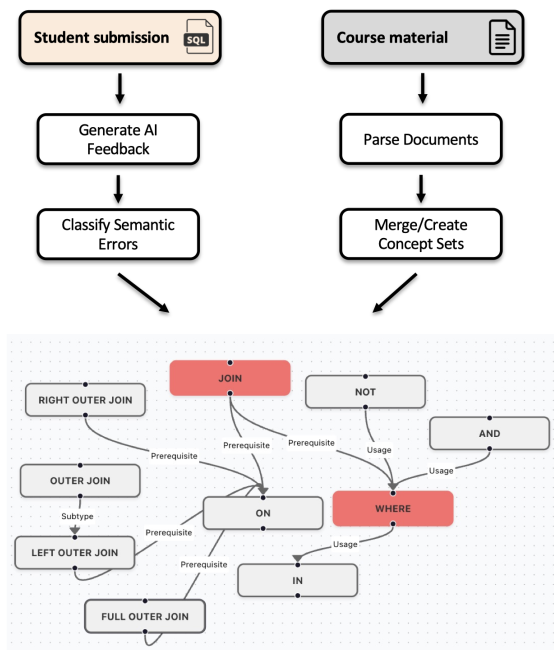


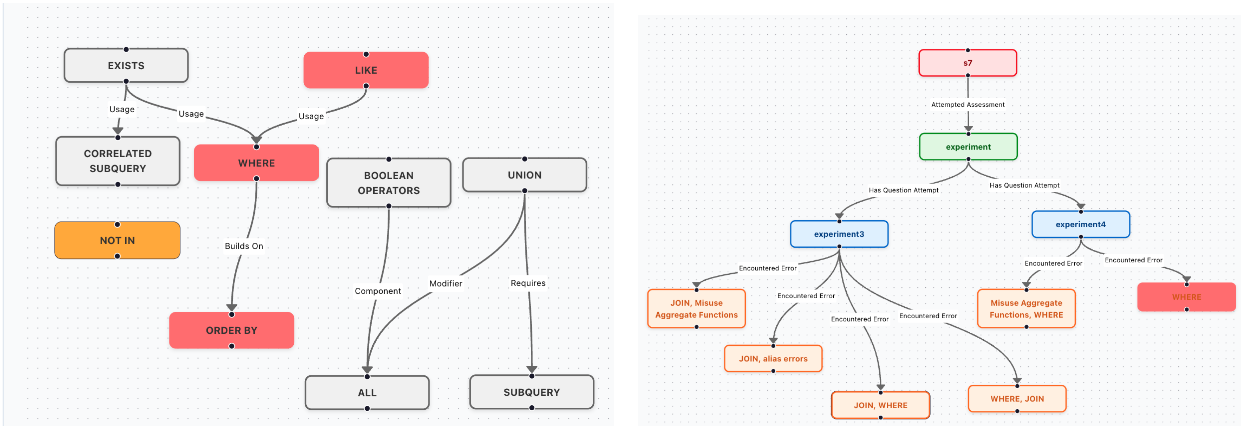
Figure 1: AI-powered pipeline for SQL feedback generation, error classification, and concept mapping via knowledge graph.

In this section, we introduce our system methodologies and how we design the preliminary evaluation.

3.1 System Architecture

As shown in Figure 1, our system is integrated into an online assessment platform where students submit SQL queries and receive real-time feedback. For each submission, the system logs the generated adaptive feedback and classify SQL error types. In addition, the system parses course materials to extract and merge a canonical set of concepts, and constructs a course-level knowledge graph consisting of concepts and their relationships.

3.1.1 Knowledge Graph



(a) Concepts from lecture notes

(b) Student's submission and error classification

Figure 2: Overview of our KG system. (a) We extract and consolidate concepts from instructional materials to construct a course knowledge graph. (b) We log student submissions, generate semantic feedback, classify error types, and build a hierarchical graph of student interactions.

At the core of our system is a course specific knowledge graph representing SQL concepts and their pedagogical relationships. Course materials (lecture notes, textbooks in PDF format) are processed through a three stage pipeline. First, GPT-4o [24] analyzes each document chunk and identifies candidate SQL concepts along with definitions and pairwise relationships. Second, a heuristic filtering layer validates candidates against a curated whitelist of core SQL educational concepts organized by category (e.g., joins, aggregation, schema design) and discards example specific terms such as table names, column names, and numeric values via regex exclusion patterns. Third, validated concept nodes are inserted into Neo4j as **SQLConcept** nodes and linked via directed, typed edges. These edges come from two sources: a set of predefined pedagogical relationships (e.g., `prerequisiteOf`, `buildsOn`, `exampleOf`, `usesWith`) and additional relationships extracted by GPT-4o from the course materials. Difficulty levels are assigned heuristically, as illustrated in Figure 2a. This structured representation enables the system to map student errors to specific concepts and reason about their pedagogical context for more targeted feedback.

The KG also stores student interaction evidence (Figure 2b), where submission records from PostgreSQL are ingested into Neo4j as `StudentError` nodes linked to tagged concepts. The system computes a concept-level progress signal as the mean correct-completion rate across all questions tagged with that concept, visualized on the dashboard using node colors indicating mastery level: neutral, yellow, and red.

3.1.2 Assessment and Feedback Pipeline

Our feedback generation follows a carefully designed two stage process to provide guidance without revealing the solution. When a student's submission fails, GPT-4o first analyzes the student's submission against the reference solution to identify the semantic error type, mapped to a fixed taxonomy of common SQL misconceptions (e.g., JOIN, GROUP BY, WHERE, alias errors, misuse of aggregate functions, subquery logic errors, and ambiguous column references [1]). This raw diagnostic is then passed to a fine tuned gpt-4o-mini model trained on 169 curated examples. Each example consists of a raw GPT-4o diagnostic as input, which often contains solution revealing content such as the correct join type, the expected output, or direct references to the reference solution, paired with a manually rewritten output that preserves the error signal while stripping any solution-revealing details. This ensures the generated feedback directs students toward recognizing their mistake conceptually rather than passively applying a fix. For instance, for a GROUP BY error where a student groups by the wrong attribute, the system would not suggest the correct attribute, but might instead generate: *"Review the columns used in your GROUP BY clause. Does the grouping align with the question's requirement to aggregate results for each customer?"*

3.2 Evaluation Methodology

We validate two core semantic outputs, SQL error-type labeling and concept mapping. Using IRB-approved data, we collected 56 incorrect student submissions from ten students across four high-volume SQL questions. Each submission was manually annotated with a ground-truth error type from our fixed taxonomy and a target concept label.

For error-type labeling, the system outputs up to two predicted error types to account for compound mistakes. We therefore evaluate error labeling using Hit@2, where a submission is counted as correct if the ground-truth error type matches either of the two predicted labels. For concept mapping, the system produces a short list of concept tags derived from questions and knowledge graph, where concepts were extracted from a compilation of half-semester-long course materials. We evaluate concept mapping using Hit@k (with k = X), where a submission is counted as correct if the ground-truth concept appears anywhere in the predicted concept list.

Results

Table 1: Overall preliminary validation results on 56 manually labeled incorrect SQL submissions

Metrics	**Hit Rate**
Error-type labeling (Hit@2)	80.35%
Concept mapping (Hit@k)	92.85%

Table 2: Error-type labeling performance (Hit@2) broken down by ground-truth category.

Ground-truth error type	**Count**	**Hit@2**
WHERE	19	73.68%
SUBQUERY LOGIC ERROR	4	75.00%
MISUSE AGGREGATE FUNCTIONS	13	76.92%
JOIN	9	88.89%
GROUP BY	10	90.00%
ALIASING	1	100.00%

Table 1 summarizes preliminary validation results on the manually labeled benchmark. Overall, the system achieves an error-type labeling Hit@2 of 80.35% $(45/56)$ and a concept mapping Hit@k of 92.85% $(52/56)$. These results suggest that the system's semantic outputs are often consistent with human annotations and can serve as reliable inputs for instructor-facing dashboards that aggregate misconceptions across questions and students.

To better understand where the system performs well or remains challenging, Table 2 reports Hit@2 by ground-truth error type. Performance is strongest for GROUP BY and JOIN, while WHERE and MISUSE AGGREGATE FUNCTIONS show comparatively lower hit rates, consistent with the fact that these mistakes often co-occur with other semantic issues (e.g., missing joins, subquery structure errors) and can therefore be harder to disambiguate from a single incorrect submission.

Discussion

Overall, the system shows promising agreement with human annotations, yet the per category breakdown in Table 2 highlights where the harder classification problems lie. GROUP BY and JOIN errors are labeled reliably, likely because these mistakes tend to manifest as clear and isolated structural problems. WHERE and aggregate function errors, however, are more challenging. Students often confuse WHERE and HAVING for filtering, and they frequently misapply aggregate functions by failing to align them with the correct grouping context. These mistakes reflect a deeper misunderstanding of SQL's declarative evaluation rather than a surface level slip [1, 8]. Moreover, such errors rarely appear in isolation; a single incorrect submission may simultaneously contain a grouping issue, a missing join, and a misplaced filter, making it difficult to assign a single dominant label for both the classifier and human annotators alike [3, 5]. Our findings align with broader trends in database education, where these core concepts consistently prove among the most difficult for students to master [8, 5], highlighting the need for adaptive feedback systems that can handle overlapping misconceptions.

Conclusion and Future Work

This work in progress demonstrates the viability of an AI powered knowledge graph system for delivering personalized, concept grounded feedback in SQL education. Our preliminary results, 80.35% Hit@2 for error type labeling and 92.85% Hit@k for concept mapping, indicate that the system can reliably diagnose common student misconceptions and surface them in an interpretable instructor dashboard. Future work will focus on three directions. First, we plan to embed the system into existing course infrastructure via Learning Management System (LMS) integration, enabling deployment within platforms such as Canvas without requiring changes to instructor workflows. Second, we will expand the evaluation beyond agreement with manual labels toward measuring classroom impact, specifically whether concept grounded feedback improves student performance and reduces recurring error patterns across assignments. Third, we will explore multi label error classification to better handle compound misconceptions, and refine the knowledge graph construction pipeline to improve concept normalization across varied SQL surface forms, moving the system closer to a deployable tool for database education.

References

[1] A. Ahadi, J. Prior, V. Behbood, 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*, 2016, pp. 272–277.

[2] S.-S. Shin, "Effect of semantic distance on learning structured query language: An empirical study," *Frontiers in psychology*, vol. 13, p. 996363, 2022.

[3] 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.

[4] 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*, 2019, pp. 198–203.

[5] R. Alkhabaz, Z. Li, S. Yang, and A. Alawini, "Student's learning challenges with relational, document, and graph query languages," in *Proceedings of the 2nd International Workshop on Data Systems Education: Bridging education practice with education research*, 2023, pp. 30–36.

[6] A. Fayaz, R. Glassey, and A. Baltatzis, "Generating personalized assignments with students in the loop," in *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, 2025, pp. 305–311.

[7] Neo4j, Inc., *The Neo4j Operations Manual (Neo4j 2025.12.1)*, 2025, accessed: 2026-01-19. [Online]. Available: https://neo4j.com/docs/operations-manual/current/

[8] D. Ponzini, A. Livani, G. Guerrini, B. Catania, and M. Coccoli, "Analyzing common student errors in sql query formulation to enhance learning support," in *Proceedings of 33rd Symposium on Advanced Database Systems*, 2025.

[9] D. Ponzini, B. Catania, and G. Guerrini, "Enhancing sql learning through generative ai and student error analysis," in *European Conference on Advances in Databases and Information Systems*. Springer, 2025, pp. 118–128.

[10] C. Kleiner and F. Heine, "Enhancing feedback generation for autograded sql statements to improve student learning," in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, 2024, pp. 248–254.

[11] S. Nayak, R. Agarwal, S. K. Khatri, and M. Mohammadian, "Student outcome assessment on structured query language using rubrics and automated feedback generation." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 3, 2024.

[12] A. AlRabah, S. Yang, and A. Alawini, "Optimizing database query learning: A generative ai approach for semantic error feedback." ASEE Conferences, 2024.

[13] A. Alrabah and A. Alawini, "Codelens: A generative ai framework for automated feedback on sql assignments," in *Proceedings of the 4th International Workshop on Data Systems Education: Bridging Education Practice with Education Research*, 2025, pp. 29–34.

[14] K. Naha, S. Y. Rubaiat, and H. M. Jamil, "Are large language models smart enough for sql tutoring and assessment?" in *2025 IEEE International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2025, pp. 357–359.

[15] K. T. Utkan and B. Öztayşi, "Large language models for automated personalized feedback in sql education: A preliminary study," in *International Conference on Intelligent and Fuzzy Systems*. Springer, 2025, pp. 632–638.

[16] T. Yang, B. Ren, C. Gu, B. Ma, and S. Konomi, "Leveraging gpt for concept generation and knowledge graph construction in educational recommender systems," in *Teaching and Learning in the Generative Artificial Intelligence Age*. Springer, 2026, pp. 75–96.

[17] M. Canal-Esteve and Y. Gutiérrez, “Educational material to knowledge graph conversion: A methodology to enhance digital education,” in *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, 2024, pp. 85–91.

[18] M. C. Aytekin, Y. Saygın *et al.*, “Ace: Ai-assisted construction of educational knowledge graphs with prerequisite relations,” *Journal of Educational Data Mining*, vol. 16, no. 2, pp. 85–114, 2024.

[19] R. Alatrash, M. A. Chatti, N. Wibowo, and Q. U. Ain, “Inferring prerequisite knowledge concepts in educational knowledge graphs: A multi-criteria approach,” *arXiv preprint arXiv:2509.05393*, 2025.

[20] P. Chen, Y. Lu, V. W. Zheng, X. Chen, and B. Yang, “Knowedu: A system to construct knowledge graph for education,” *Ieee Access*, vol. 6, pp. 31 553–31 563, 2018.

[21] L. Lin and F. Wang, “Adaptive learning system based on knowledge graph,” in *Proceedings of the 9th International Conference on Education and Training Technologies*, 2023, pp. 1–7.

[22] P. Gacek and W. T. Adrian, “Automated curriculum analysis using large language models and knowledge graphs,” *Intelligenza Artificiale*, vol. 19, no. 2, pp. 116–126, 2025.

[23] Y. Lu, Y. Pian, P. Chen, Q. Meng, and Y. Cao, “Radarmath: An intelligent tutoring system for math education,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 18, 2021, pp. 16 087–16 090.

[24] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.