iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81046

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

A Review of Interactive and Step-Wise SQL Learning Platforms for Enhancing Learner Engagement and Query Comprehension

Anita Patil[1], Vinayakgottur[2], Shreyas S[3], Vinayakgottur[4], Sharanabasava[5], Shivaraj J[6]
[1]*Professor, Ballari Institute of Technology and Management, Ballari, India*
[2, 3, 4, 5, 6]*Department of CSE, Ballari Institute of Technology and Management, Ballari, India*

Abstract: ***SQL remains a foundational skill in modern software development and data management, yet many learners struggle to understand complex queries when presented as complete solutions. Traditional SQL learning tools fail to convey the incremental thought process behind query construction, leading to confusion and learner dropout. Recent advances in interactive web technologies, intelligent tutoring systems, and gamification have opened new pathways for improving SQL pedagogy. The integration of step-wise scaffolding, immediate feedback mechanisms, and adaptive difficulty adjustment provides learners with a structured and confidence-building experience. This review analyses recent developments in SQL learning platforms, interactive editors, gamified approaches, and AI-driven assessment tools, highlighting their advantages and limitations, and emphasizes that combining step-wise query decomposition with real-time feedback offers a promising solution for effective and engaging SQL education.***

Keywords: SQL learning, step-wise query building, interactive editor, gamification, intelligent tutoring, feedback systems, adaptive learning, database education.

I. INTRODUCTION

SQL (Structured Query Language) is a foundational skill in modern software development, data engineering, and data analysis. From managing relational databases to powering business intelligence dashboards, SQL proficiency is consistently ranked among the most in-demand technical competencies in the technology industry. Despite its importance, SQL remains a challenging subject for many learners — particularly beginners who struggle to bridge the gap between conceptual database theory and practical query construction [12],[18].

The core difficulty of learning SQL lies not in the syntax of individual clauses but in understanding how multiple clauses combine to express a precise retrieval or transformation of relational data. Learners must develop both syntactic fluency and an ability to reason about data transformations — understanding, for example, why a WHERE clause must precede GROUP BY, or how an INNER JOIN changes the cardinality of a result set compared to a LEFT JOIN. When errors arise in complex queries, learners often cannot identify the specific clause where their reasoning failed, leading to frustration and high dropout rates [8],[19].

Traditional SQL learning approaches — including textbooks, lecture-based instruction, and video tutorials — typically present queries in their final, completed form. While these resources provide theoretical explanations, they fail to model the incremental thought process experienced developers use when constructing queries. A learner watching a video tutorial sees the final query written from scratch in minutes, with no insight into the exploratory, iterative process of drafting, testing, and refining sub-queries that a professional developer actually employs [3],[8].

The rise of web-based interactive tools has partially addressed these limitations. Platforms allowing learners to write queries and observe results instantly offer improved engagement and faster feedback cycles. However, most such tools provide little structural guidance or pedagogical scaffolding — learners are still required to manage their own learning path, which is particularly challenging for novices who lack the metacognitive skills to recognize gaps in their own understanding [10],[14].

Recent research has explored several promising directions: block-based visual query builders that lower the initial syntax barrier [1]; gamified SQL platforms leveraging motivational mechanics such as points, badges, and narrative contexts [4],[5],[20]; intelligent tutoring systems that model individual learner knowledge and adapt difficulty [6],[12]; automated assessment tools supporting partial credit and formative feedback [3],[11],[15]; and step-wise query decomposition approaches that break complex problems into smaller validated stages [10],[13].

©IJRASET: All Rights are Reserved

This paper reviews the major approaches in SQL learning platform design, highlights their features, limitations, and research gaps, and discusses how a step-wise, feedback-driven SQL learning platform such as Query-path can address the shortcomings of existing systems by combining incremental query construction with immediate intermediate validation and targeted conceptual support.

II. FEATURES AND LIMITATIONS OF EXISTING SYSTEMS

Existing SQL learning systems span a broad range, from passive instructional resources to sophisticated intelligent interactive platforms. Each category has been developed in response to specific educational challenges and offers distinct pedagogical affordances alongside notable constraints. Understanding both capabilities and limitations is essential for identifying what a next-generation SQL learning platform must provide. These are summarized in Tables I and II [1],[3],[4],[6],[10],[11],[14],[15].

TABLE I. Features of Existing SQL Learning Systems

System Type	Features
Textbooks & Videos	Broad explanations and worked examples of SQL queries
Static Online Judges	Automated query evaluation against expected outputs
Interactive SQL Editors	Write and run SQL in-browser with live result feedback
Block-Based SQL Tools	Visual drag-and-drop query building for beginners [1]
Gamified Platforms	Points, badges, and missions to motivate learners [4],[5]
Intelligent Tutoring (ITS)	Personalized hints and adaptive problem selection [6],[10]
Step-Wise Platforms	Query decomposed into validated sub-tasks [10],[12]
AI-Driven Assessment	Automated grading with partial credit and feedback [11],[15]

TABLE II. LIMITATIONS OF EXISTING SQL LEARNING SYSTEMS

System Type	Limitations
Textbooks & Videos	Static; no interactivity or real-time feedback
Static Online Judges	Evaluate final output only; no step validation [3],[11]
Interactive SQL Editors	No guided scaffolding; learners navigate independently
Block-Based SQL Tools	Not scalable to advanced SQL topics [1]
Gamified Platforms	Game mechanics may overshadow conceptual depth [4]
Intelligent Tutoring (ITS)	High development complexity; costly to maintain [6]
Step-Wise Platforms	Require extensive content design for each problem
AI-Driven Assessment	Risk of incorrect feedback; depends on training data [11],[15]

As shown in the tables, traditional resources offer broad conceptual coverage but lack interactivity. Modern platforms using gamification, ITS, and automated grading improve engagement and feedback, but often do not address the core challenge of step-wise query comprehension. Despite these beneficial features, existing systems remain limited in practical scenarios: most platforms evaluate only final query outputs with no visibility into intermediate reasoning; gamified platforms prioritize engagement over depth; ITS require extensive domain engineering; and automated graders apply binary judgments without supporting partial understanding. Learner confidence, comprehension, and completion rates are adversely affected, indicating the need for more structured scaffolded solutions.

III. REVIEW OF INTERACTIVE AND INTELLIGENT SQL LEARNING TECHNIQUES

To understand and improve SQL learning systems, it is important to examine the full landscape of interactive, adaptive, and scaffolded approaches. With the growing emphasis on learner-centered education and the availability of powerful web technologies and AI tools, researchers have proposed diverse solutions aiming to improve comprehension, reduce errors, and maintain learner motivation. We review the major techniques as follows.

A. Block-Based and Visual SQL Learning Tools

Block-based programming environments have been adapted for SQL instruction to reduce the cognitive load of syntax learning. Learners assemble query components visually by dragging and connecting labeled blocks representing clauses such as SELECT, FROM, WHERE, and JOIN, enabling novices to focus entirely on query logic rather than syntactic correctness, lowering the initial barrier to database programming [1].

SQLatch [1] integrates block-based query building with a collaborative learning interface, allowing students to construct SQL queries interactively and receive immediate visual feedback. Research on SQLatch showed improved initial engagement and a measurable reduction in syntax-related errors among first-year learners compared to text-based SQL environments. The collaborative aspect also encouraged peer learning, allowing students to discuss and compare query structures in real time.

Visual tools that pair query construction with output visualization [14] further support comprehension by allowing learners to trace precisely how each SQL clause affects the result set at every stage. Zhou et al. [14] demonstrated that linking syntax to semantics through step-level output previews helped learners understand the role of individual clauses more concretely than static instruction alone. However, as query complexity increases, block-based representations become cumbersome, limiting their scalability to intermediate and advanced learners.

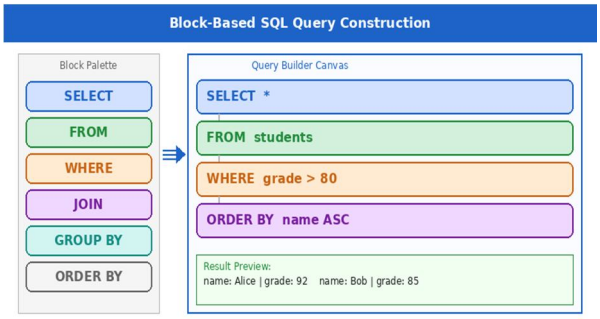


Fig. 1. Block-based SQL query construction interface with drag-and-drop clause assembly [1].

B. Gamified SQL Learning Platforms

Gamification has gained significant traction in SQL education as a strategy to address the well-documented motivation and engagement challenges in database courses. Gamified SQL platforms incorporate mechanics such as points, badges, leaderboards, timed challenges, narrative storylines, and progressive achievement systems to sustain learner interest and encourage consistent practice over extended learning periods [4],[5].

Balla et al. [4] demonstrated through a controlled study that game-based SQL assessments significantly improved student engagement and task completion rates compared to traditional homework. Learners in the gamified condition reported higher intrinsic motivation and were more likely to reattempt failed queries. Kiraly et al. [5] combined gamified tutorials with a flipped classroom model, showing that learners who engaged with game-based exercises before lectures outperformed peers in assessments and reported substantially higher motivation scores.

The SQL Murder Mystery [20] is a widely-used narrative-driven SQL learning tool where learners solve a fictional crime by writing progressively complex queries across multiple database tables, blending entertainment with genuine skill development. Despite these benefits, most gamified platforms target introductory SQL and do not adequately address advanced concepts. The coupling between game rewards and deep conceptual mastery is also rarely empirically verified [9].

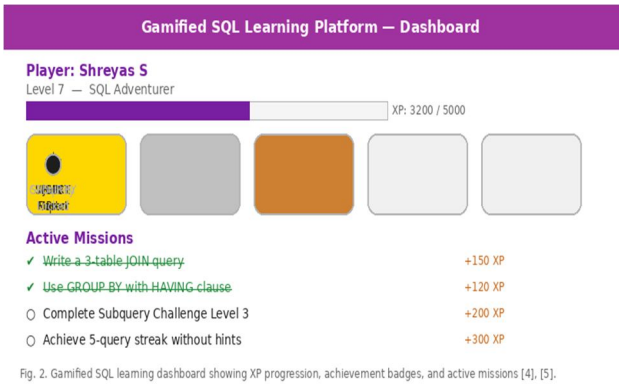


Fig. 2. Gamified SQL learning dashboard showing XP progression, achievement badges, and active missions [4],[5].

C. Automated Assessment and Partial Grading Systems

Automated SQL assessment has evolved significantly beyond simple output matching. Modern systems apply structural analysis of query components, semantic equivalence checking, partial grading of incomplete submissions, and natural language feedback generation [3],[11],[15].

Martin-Martin et al. [3] conducted a large-scale study of LearnSQL finding that immediate automated feedback significantly improved both submission accuracy and learner confidence compared to courses relying on delayed instructor feedback. Learners in the automated feedback condition submitted more attempts per problem, suggesting that low-friction feedback loops encourage iterative problem-solving behavior rather than single-shot submission strategies.

Wanjiru et al. [15] extended grading with dynamic partial credit that awards marks based on the correctness of individual query clauses — such as a correctly specified WHERE condition even when the overall query fails — providing more granular and fair assessment. Motiwala et al. [11] developed Auto-Assess, combining automated grading with detailed performance analytics dashboards enabling instructors to identify common learner errors at scale. Obionwu et al. [7] proposed retrospective evaluation of student SQL submission patterns to improve course design over time.

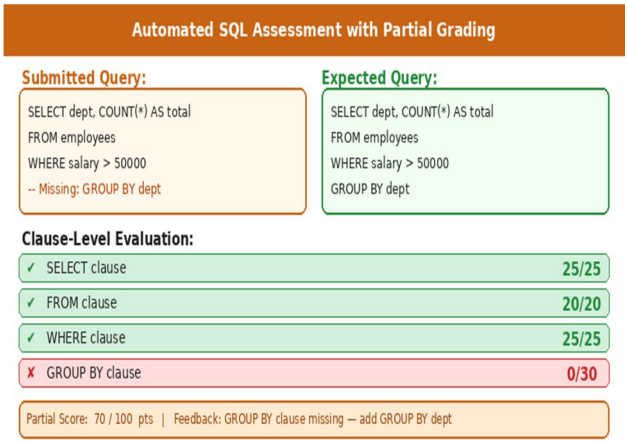


Fig. 3. Automated SQL assessment showing clause-level partial grading and targeted feedback [3],[11],[15].

D. Intelligent Tutoring and Adaptive SQL Learning Systems

Intelligent Tutoring Systems (ITS) for SQL use computational models of learner knowledge, common misconceptions, and domain structure to provide personalized guidance, adaptive problem selection, and targeted hints. These systems represent the most sophisticated approach to SQL instruction, capable of diagnosing specific conceptual gaps and directing learners toward the most relevant exercises to address those gaps [6],[12],[19].

Mitrovic [19] pioneered one of the earliest ITS systems for SQL using constraint-based modeling, demonstrating that formally specifying correctness constraints for each SQL concept could produce highly specific and actionable feedback beyond simple right/wrong judgments. Piyayodilokchai et al. [12] applied a structured learning cycle framework — comprising exploration, concept introduction, application, and reflection phases — showing learners guided through this cycle significantly outperformed those using unstructured practice on measures of query understanding and retention.

Yathongchai [6] proposed an SQL Learning Object Ontology that formally encodes relationships between SQL concepts and prerequisite dependencies, enabling precise recommendation of learning objects matched to identified knowledge gaps. Brusilovsky et al. [16] built an open integrated exploratorium for database courses combining ITS-style adaptive guidance with exploratory SQL tools. More recently, Mullah and Jayachandran [9] explored Generative AI integration in SQL learning environments, opening possibilities for dynamic hint generation and personalized content creation. While highly effective, ITS approaches require substantial domain modeling effort and are costly to maintain.

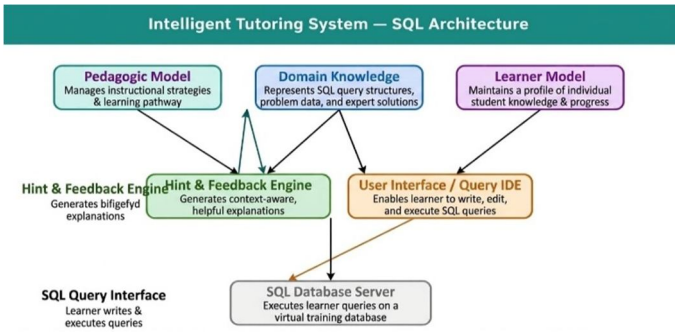


Fig. 4. Architecture of an SQL Intelligent Tutoring System with learner modelling and adaptive feedback [6],[12].

E. Step-Wise and Scaffolded SQL Learning Approaches

Step-wise scaffolding in SQL learning refers to the decomposition of complex query problems into a sequence of smaller, intermediate sub-tasks, each of which must be validated before the learner proceeds to the next stage. This approach directly addresses the core pedagogical challenge — helping learners develop not just knowledge of the correct final query, but a clear mental model of the incremental reasoning process required to construct it [10],[13],[18].

Cagliero et al. [8] applied a data-driven approach to SQL practice design, analyzing historical student submission patterns to identify common confusion points and optimize the ordering of practice exercises. Their findings demonstrated that algorithmically sequenced practice informed by real learner data produced better outcomes than instructor-curated problem sets. Bhuse et al. [10] developed SQLearn, a browser-based adaptive SQL learning environment where learners receive contextual hints and step-level guidance based on their progress through multi-stage problems.

Wojtowicz et al. [13] demonstrated through a controlled experiment that intensive practice with incrementally staged test problems produced significantly higher SQL proficiency gains than equivalent time spent on standard self-study. Wardani et al. [2] designed self-study exercise problems emphasizing explicit problem decomposition and immediate answer verification as core learning mechanics. Siepermann et al. [18] found that structured progression through sub-problems improved both conceptual understanding and long-term retention compared to open-ended practice environments.

These findings strongly support the step-wise learning paradigm as the most effective approach for building deep SQL query comprehension, and directly motivate the design of platforms like Query-path, which systematically applies sub-task decomposition, intermediate result validation, and targeted conceptual hints across a complete SQL curriculum.

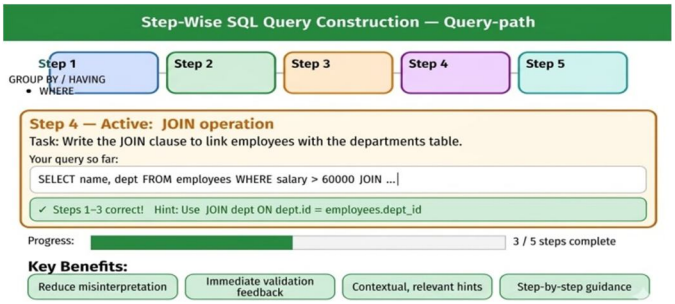


Fig. 5. Step-wise SQL query construction in Query-path with intermediate validation and contextual hints [10],[13],[18].

IV. HIGHLIGHTS OF REVIEWED TECHNOLOGIES

The following table summarizes the major technologies reviewed in this paper, describing their core capabilities and key trade-offs that motivate further research and the development of integrated platforms such as Query-path.

TABLE III. Highlights of Reviewed Technologies

Technology	Description
Interactive Editors & Block-Based Tools	Enable browser-based SQL execution and visual query construction. Reduce syntax barriers but lack scaffolding [1],[14].
Gamified SQL Platforms	Game mechanics drive engagement. Effective introductory level; limited for advanced SQL [4],[5],[20].
Intelligent Tutoring & Adaptive Systems	Personalized paths, adaptive difficulty, targeted hints. Improve outcomes but require significant domain modeling [6],[10],[12].
Automated Assessment & Partial Grading	Beyond binary correctness — partial credit and formative feedback. Scalable but risk inaccurate AI feedback [3],[11],[15].
Step-Wise Scaffolded Construction	Decomposes queries into validated sub-tasks. Significantly improves comprehension and confidence [12],[13],[18].

V. CHALLENGES, RESEARCH GAPS, AND FUTURE DIRECTIONS

Interactive and intelligent SQL learning systems have significantly improved the accessibility and effectiveness of database education. However, several persistent challenges remain that limit their practical impact across diverse learning contexts, user populations, and SQL complexity levels.

One of the most significant challenges is the absence of step-wise visibility into the query construction process. The overwhelming majority of existing platforms evaluate only the final submitted query, providing no mechanism for learners to validate intermediate reasoning. This is particularly damaging for learners working with JOIN operations, subqueries, and aggregation pipelines, where errors in early clauses propagate invisibly through subsequent stages [12],[18].

Engagement and motivation present persistent challenges for learners progressing beyond introductory SQL. Gamified platforms that succeed at introductory levels typically lack the content depth required for intermediate and advanced skill development. The motivational effects of extrinsic game mechanics tend to diminish over time without meaningful progression of challenge complexity [4],[5]. Automated assessment systems face the challenge of balancing evaluation thoroughness with feedback quality — binary grading fails to reward partial understanding, while partial grading requires sophisticated semantic analysis difficult to generalize [3],[15].

Despite these advancements, several important research gaps remain. Most systems focus on a single learning modality rather than integrating complementary approaches into a unified, curriculum-aligned platform. There is limited longitudinal research measuring how different teaching strategies affect SQL skill retention over time. Many models also struggle to generalize across different SQL dialects such as MySQL, PostgreSQL, and SQLite, limiting portability across teaching contexts. Future research should prioritize step-aware SQL learning systems capable of validating intermediate query stages, integrating adaptive difficulty and AI-driven personalized hint generation into accessible editor-based environments.

VI. CONCLUSIONS

SQL learning requires not only syntactic knowledge but also the ability to reason incrementally through the construction of complex, multi-clause queries. This paper reviewed the evolution of SQL learning systems, highlighting limitations of traditional approaches and significant advances achieved through block-based tools, gamified platforms, automated assessment systems, intelligent tutoring systems, and step-wise scaffolded learning environments.

Interactive SQL editors and visual tools lower the initial barrier to entry and support exploratory learning for beginners. Gamified platforms improve motivation and sustained engagement at introductory levels through narrative and reward mechanisms. Intelligent tutoring systems provide personalized learning paths, targeted feedback, and prerequisite-aware problem sequencing that improve learning outcomes beyond what static tools can achieve [6],[12]. Automated grading systems with partial credit capabilities enable scalable, consistent evaluation that benefits both learners and instructors in large database courses [3],[11],[15].

Among all reviewed approaches, step-wise and scaffolded query construction methods have demonstrated the most direct and consistent improvement in learner query comprehension, error diagnosis capability, and task completion rates. By decomposing complex queries into validated sub-tasks and providing immediate intermediate feedback, these methods address the fundamental gap that all other approaches leave unresolved: the inability to observe and correct reasoning errors before they compound across query stages [10],[13],[18].

Although the individual technologies reviewed each demonstrate promising results, challenges such as the absence of intermediate-step validation, limited scalability of gamification to advanced SQL topics, the high development cost of full ITS implementations, and insufficient integration of multiple complementary modalities into unified platforms persist across the field. Addressing these challenges requires a new generation of SQL learning platforms that combine the accessibility of editor-based tools with the pedagogical structure of step-wise scaffolding and the personalization capabilities of intelligent adaptive systems.

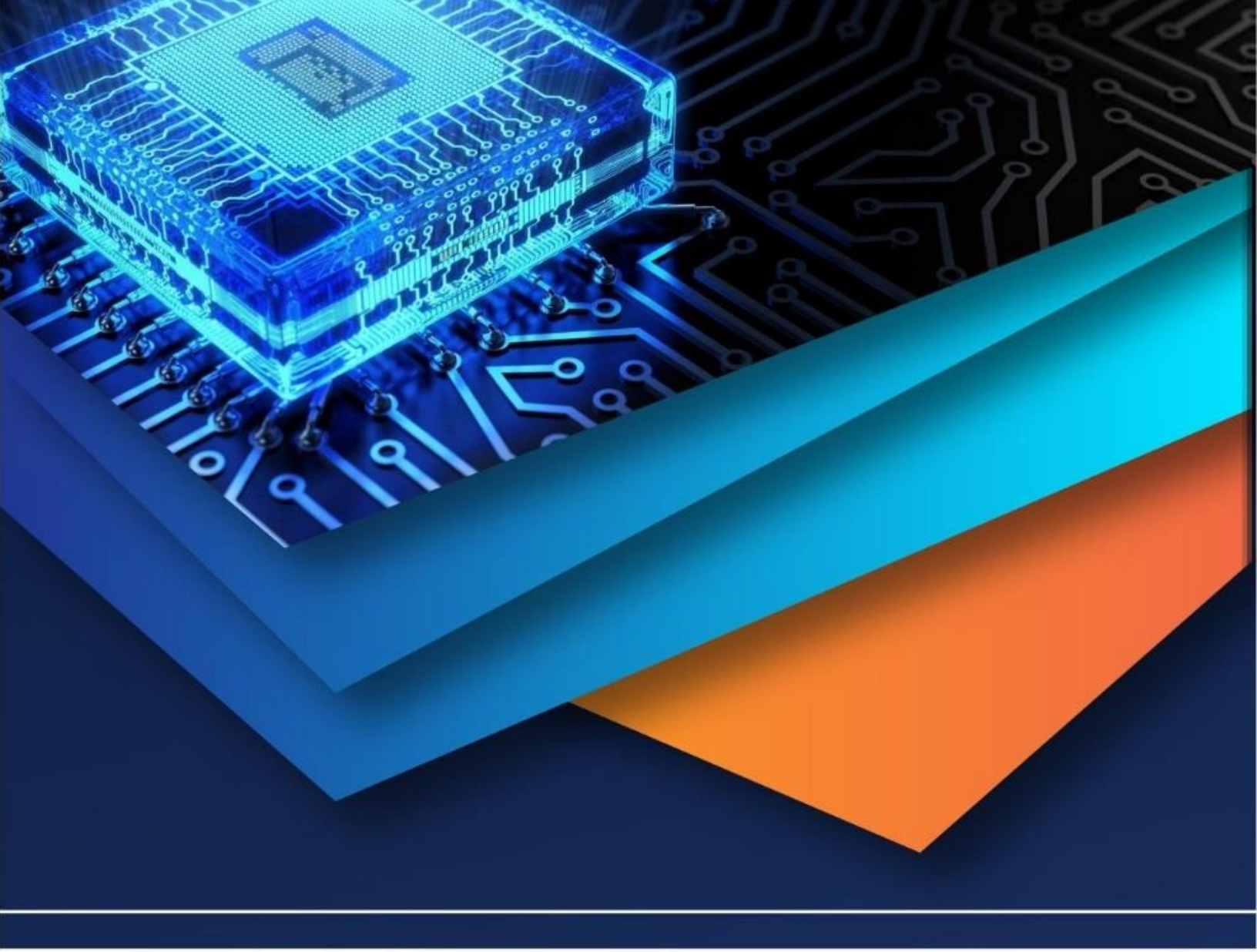
Overall, the combination of interactive query editing, step-wise decomposition, intermediate result validation, and AI-driven hint generation — as embodied in the Query-path platform — presents a compelling, educationally grounded, and practically scalable solution for modern SQL instruction, offering significant potential to reduce learner confusion, build genuine query comprehension from first principles, and improve completion rates in both formal database education and self-directed professional learning contexts.

REFERENCES

[1] M. Selvesakis, G. Harizanis, and I. Kazanidis, "SQLatch: Enhancing SQL Learning through Interactive Block-Based Programming," International Conference on Educational Technology and Online Learning, pp. 268–278, May 2024.
[2] M. G. Wardani, N. Funabiki, and W. C. W. S. Samarasinghe, "Self-study Exercise Problems for SQL-Python Programming," Engineering Letters, vol. 33, no. 1, pp. 24–35, Mar. 2025.
[3] E. Martin-Martin, M. Montenegro, A. Riesco, R. Rubio, and F. Saenz-Perez, "LearnSQL: Impact of an Automatic Judge in Database Learning," ACM Transactions on Computing Education, vol. 26, no. 1, pp. 1–37, Nov. 2025.
[4] T. Balla, S. Kiraly, and R. Kiraly, "Enhancing SQL Programming Assessments through Educational Games: A Gamified Approach," Discover Education, vol. 4, no. 365, 2025.
[5] S. Kiraly, T. Balla, D. Kiraly, and G. Vaughan, "Enhancing SQL Learning: Gamified Tutorials and Flipped Classroom Synergy," Social Sciences & Humanities Open, vol. 12, 101762, 2025.
[6] W. Yathongchai, "SQL Learning Object Ontology for an Intelligent Tutoring System," Int. J. e-Education, e-Business, e-Management and e-Learning, Jan. 2013.
[7] C. V. Obionwu, K. Oji Kalu, P. Blockhaus, D. Broneske, and G. Saake, "A Strategy for Retrospective Evaluation of Students SQL Learning Engagements," Jul. 2023, pp. 1–7.
[8] L. Cagliero, L. De Russis, L. Farinetti, and T. Montanaro, "Improving the Effectiveness of SQL Learning Practice: A Data-Driven Approach," Jul. 2018, vol. 1, pp. 980–989.
[9] Mullah and S. Jayachandran, "AI-Driven Innovation in Education 4.0: Generative AI Tools, Learning Analytics, and Gamified SQL Learning," Int. J. Environmental Sciences, vol. 11, no. 9s, pp. 1062–1066, Jun. 2025.
[10] P. Bhuse, J. Jain, A. Shaju, V. John, A. Joshi, and R. Rajendran, "SQLearn: A Browser Based Adaptive SQL Learning Environment," Springer, 2021, pp. 139–152.
[11] M. A. Motiwala, A. Joshi, and R. Chapaneri, "Auto-Assess: Automated Evaluation Platform for Enhanced SQL Learning," Jan. 2025, pp. 1–8.
[12] H. Piyayodilokchai et al., "Promoting Students' Understanding of SQL in a Database Management Course: A Learning Cycle Approach," Int. J. Learning, vol. 17, no. 11, pp. 325–338, 2011.
[13] Wojtowicz, A. Stachowiak, and E. Pankowska, "Train Hard, Score High: The Impact of Practice Tests on SQL Learning," Jun. 2025, pp. 35–40.
[14] Y. Zhou, D. Towey, and M. Pike, "FROM Syntax to Semantics: An OER-Powered SQL Learning and Visualisation Tool," Jul. 2024, pp. 1522–1523.
[15] B. Wanjiru, P. Van Bommel, and D. Hiemstra, "Dynamic and Partial Grading of SQL Queries," J. Engineering Research and Sciences, vol. 3, no. 8, pp. 1–14, Aug. 2024.
[16] P. Brusilovsky et al., "An Open Integrated Exploratorium for Database Courses," Jun. 2008, pp. 22–26.

[17] E. Hosam, H. Hosny, W. Ashraf, and A. S. Kaseb, "SQL Injection Detection Using Machine Learning Techniques," Nov. 2021, pp. 15–20.
[18] M. Siepermann, R. Lackes, and C. Borgermann, "E-Learning Supported SQL Teaching," Apr. 2014.
[19] A. Mitrovic, "Learning SQL with a Computerized Tutor," Mar. 1998, vol. 30, no. 1, pp. 307–311.
[20] L. Canale and L. Farinetti, "SQL Murder Mystery: A Serious Game to Learn Querying Databases," Jun. 2022, vol. 1, pp. 129–138.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)