INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 1, January 2026 | ISSN: 2456-4184 | IJNRD.ORG

RETRIEVAL-AUGMENTED ANALYTICS ASSISTANTS FOR SELF-SERVE INSIGHTS IN SQL AND PYTHON

Yashika Shankheshwaria

Washington University of Science and Technology

Virginia United States of America

Abstract- Organizations generate large volumes of data, yet the ability of non-technical stakeholders to extract insights from the data is limited. Traditional analytics workflows rely on the analysts who help translate business questions into SQL queries. They are able to translate the questions into Python code or dashboards. Important to note that this independence limits business growth as it bottlenecks decision-making, and it also restricts the democratization of the data insights. Recent breakthroughs in retrieval-augmented generation (RAG), intelligent agentic systems, and large language models (LLMs) have opened new pathways for self-serve analytics. Important to note that Retrieval-augmented analytics combine the natural-language interfaces with semantic SQL reasoning, a retrieval mechanism, and automated SQL or Python generation to enable users to query organizational data without manual coding.

The paper examines the capabilities, design, and implications of RAG-enhanced analytics assistants. The paper draws insights from scholarly sources. Using a qualitative synthesis and design-science methodology, the paper will integrate insights from the studies of develop a reference architecture and design principles to support retrieval-augmented assistants that are capable of trustworthy self-serve analytics. The findings depict that although RAG-enabled SQL/Python assistants offer benefits like improved accessibility, dynamic retrieval of knowledge, and automation of query reasoning. There are many challenges that remain, and they include semantic ambiguity, hallucinations, and the need for governance, robust semantic layers, and evaluation benchmarks. This research paper proposes a theoretical and practical framework that can be used to deploy retrieval-augmented analytics assistants in real-world environments.

***Keywords*: retrieval-augmented generation, large language models, SQL reasoning, text-to-SQL, semantic SQL, intelligent agents, big data analytics, self-serve BI**

A. INTRODUCTION

The contemporary data landscape is defined by increasing complexity, rapid expansion, and escalating organizational dependence on timely insights. Businesses today collect data from customer platforms, cloud services, IoT devices, and operational systems. They also collect their data from unstructured logs, and this creates an environment that is characterized by high volume, velocity, and variety. Although analytical ecosystems like data lakes, data warehouses, and BI tools have evolved to manage this scale, still a gap between data usability and availability for everyday use by businesses.

Traditionally, the analytics workflows required users to articulate information needs to analysts or the engineers, who then craft SQL scripts manually, create Python notebooks, and BI dashboards. The manual process is time-consuming. It increases the workload for the analytical teams. The process slows down the decision-making process. Organizations are seeking automation and augmentation mechanisms that enable the business to reduce this dependency. This is important in supporting more autonomous data interaction (Kumar et al., 2024).

Large language models (LLMs) have demonstrated strong capabilities in understanding natural language, code generation, and semantic reasoning. Research on text-to-SQL systems has depicted steady improvements in LLMs' ability to translate natural language questions into SQL queries that are executable (Liu et al., 2025; Luo et al., 2025). There are advancements in reinforcement-learning-enhanced SQL and the emergence of intelligent agent frameworks for the generation of SQL, suggesting a maturing direction towards reliable, automated code synthesis.

RAG adds another vital component by enabling an LLM to reference external and domain-specific information like database schemas, query logs, and documentation during inference (Arslan et al., 2024). Important to note that retrieval capabilities are vital in reducing hallucinations, help the system to generate accurate SQL and Python codes, and improve grounding. The cross-domain success of RAG in augmenting model reasoning includes test generation, which depicts its potential (Shin et al., 2024).

Despite these vital breakthroughs, there are critical challenges that persist. They include semantic ambiguity between the database structure and natural language, and a lack of domain context within LLMs. We have difficulty in grounding queries to enterprise-specific schemas. Need for a trustworthy evaluation framework. We have potential errors or hallucinations in the generated Python or SQL.

Therefore, the question guiding the paper is: How can retrieval-augmented analytics assistants be designed to enable reliable self-serve insights in SQL and Python for non-technical users?

The paper will synthesize insights from various sources to construct a unified and academically grounded perspective on how retrieval-augmented assistants can be implemented to operationalize self-serve analytics while ensuring that accuracy, scalability, and governance are not compromised.

B. LITERATURE REVIEW

Relevant scholarly work will be analyzed in this section. The goal is to gather insights about retrieval-augmented analytics assistants.

1. Retrieval-Augmented Generation for Business Intelligence

Arslan et al. (2024) provide comprehensive analyses of retrieval-augmented generation that support the decision-making process. Their review depicts that RAG-enhanced LLMs are critical in improving precision in the extraction of information and the generation of insights by coupling model predictions with the retrieval of external knowledge. By integrating unstructured and structured documents, RAG frameworks are able to address LLMs' limitations by relying on parametric memory. This can be categorized into two advantages: grounding analytical reasoning in the business context and reducing hallucinations in tasks that demand complex reasoning. The findings are directly relevant to analytics assistants. This helps bridge the semantic gap between user queries and the enterprise schemas.

Shin et al. (2024) explore the application of RAG to test-case generation, which is a different domain with similar structural reasoning challenges. Their findings indicate that RAG can improve model performance when a precise contextual alignment is required. Thei reinforces the idea that RAG is suited for the generation of SQL/Python. Here, models need to understand schema, field constraints, and table relationships.

2. AI-Powered Evolution of Big Data Ecosystems

Kumar et al. (2024) explore the evolution of big data systems under the influence of AI technologies. They highlight how AI is transforming data ingestion, analytics workflows, and processing through automation. They depict how adaptive reasoning and enhanced scalability promote better workflows. They emphasize the intelligence agent's emergence as a critical component in bridging human and machine interaction in big data environments. This indicates that modern analytical ecosystems need to meet specific criteria like efficiency in processing large volumes of data. It needs to meet accuracy in analytical outputs, have a human-centric design that supports usability, and be adaptable to changing data environments. These criteria align with the goals of retrieval-augmented analytics assistants. These function at the intersection of automation, usability, and scalable analytics.

3. Text-to-SQL Systems in the Era of LLMs

Text-to-SQL is critical to self-serve analytics. Liu et al. (2025) offer a large-scale survey of these approaches. The authors outline the evolution from early rule-based systems to LLM-driven architectures. They highlight technical challenges like query disambiguation, evaluation inconsistencies, schema linking, and handling complex SQL structures. The authors indicate that LLMs have improved model performance. They are effective when they are combined with schema-aware retrieval components. The authors warn that, despite improved syntactic correctness, there are challenges with semantic accuracy. This means that generated SQL may be valid, but its intent or logic might be incorrect. Luo et al. (2025) also identify persistent obstacles in natural language-to-SQL mapping. They emphasize the need for robust contextual grounding, the limitations of benchmark datasets, the importance of encoding domain knowledge, and why hybrid approaches are vital. These studies confirm that retrieval-based mechanisms like RAG are vital in enhancing SQL generation by supplying LLMs with examples and schema-specific constraints.

4. Intelligent Agent Integration for Text-to-SQL Optimization

Ojuri et al. (2025) explore how intelligent agents improve text-to-SQL conversion. The authors indicate that agent-based systems outperform standalone LLMs. This is possible as they engage in iterative reasoning, engage in error checking, and implement self-correction cycles. These agents can check SQL execution results and re-query the LLM for clarifications. They can validate schema consistency and regenerate optimized queries. Important to note that the iterative agent loop is vital for reliable analytics assistants. They are crucial in environments that require dynamic and multi-step reasoning.

5. Semantic SQL and AI-Powered Databases

Neves et al. (2019) provide critical insights into semantic SQL in databases that are powered by AI. They demonstrate that a system is capable of interpreting natural language queries. The system can do this through semantic parsing and aligning them with relational data structures. The authors underscore the importance of metadata integration, semantic modeling, and declarative query interpretation. The authors depict that semantic SQL remains foundational for retrieval-augmented assistants. This is because the generation of accurate SQL/Python depends on a precise understanding of relationships and the business logic encoded within schemas.

6. Reinforcement-Learning–Enhanced SQL Reasoning Models

RLVR is introduced by Ali et al. (2025). This is a reinforcement-learning-based SQL reasoning model. The authors' systems demonstrate advances in semantic alignment, complex SQL reasoning, and reward-driven optimization. Their research indicates how reinforcement learning can be employed to refine SQL generation. This is achieved by penalizing incorrect logic and rewarding accurate reasoning. The RLVR depicts an important evolution, suggesting that future analytics assistants could employ reinforcement signals to improve output accuracy based on execution feedback.

C. METHODOLOGY

The study employs a qualitative research methodology. The methodology is grounded in integrative review and design-science principles. The approach entails literature identification, with references included to ensure relevance and focused analysis. It entails thematic coding where concepts like intelligent agents, RAG, semantic modeling, SQL reasoning, and big-data ecosystems were coded and they were categorized across sources. These are vital in forming a coherent and theoretically grounded framework.

1.1 Research Design

The research design follows complementary components:

1. Integrative Literature Review

The integrative review method was selected as it allows for the synthesis of studies that employ various methodologies, technological perspectives, and theoretical frameworks. The literature selected includes surveys (Liu et al., 2025; Luo et al., 2025). We have empirical evaluations (Shin et al., 2024; Neves et al., 2019). Literature entails conceptual reviews (Arslan et al., 2024) as well as technical model papers (Ali et al., 2025; Ojuri et al., 2025; Kumar et al., 2024). These sources are vital as they provide a comprehensive foundation for understanding and appreciating retrieval-augmented analytics assistants.

2. Design-Science Methodology

The design-science paradigm is well-suited to research aimed at proposing technical artifacts. In this case, the conceptual architecture and set of design principles are critical for retrieval-augmented analytics assistants. Important to note that the design focuses on identifying relevant organizational problems. It focuses on synthesizing knowledge from prior studies. It helps propose an artifact that can address the identified gaps. It helps guarantee that the artifact is grounded in scientifically validated principles. It is evident that the methodology aligns with the goals of constructing a framework that integrates RAG retrieval and SQL/Python code generation. It integrates semantic knowledge layers, intelligent agent workflows, and helps reinforce learning and reasoning.

1.2 Data Sources and Selection Criteria

To maintain strict academic control and relevance, few sources were used as primary data sources. The sources were selected because they represent leading research across crucial sources that are required for retrieval-augmented analytics assistants. The sources are vital as they reflect a balanced mixture of methodological types like experimental, survey, application, and theoretical research that allow a multifaceted understanding of the problem space.

1.3 Thematic Coding and Data Analysis Procedures

A structured coding process was used once the literature was identified.

1. Initial descriptive coding

Each source was reviewed, and relevant information was extracted. The concepts were related to RAG mechanisms, semantic query interpretation, SQL reasoning, and code generation. We have intelligent agent workflows and big data architectural transformation.

2. Axial coding

Concepts were grouped into analytical categories like semantic modeling requirements, contextual grouping challenges. We have agentic error-correction mechanisms, SQL reasoning complexity, and LLM limitations and hallucination risks.

3. Selective coding

The themes were synthesized into overarching domains that inform the design principles of retrieval-augmented assistants. The analysis had five pillars, and they include retrieval-augmented context integration, agent-based orchestration, semantic modeling, reinforcement learning for continuous improvement, and SQL/Python generative reasoning. This iterative coding structure is vital as it ensures that all included studies contributed to the conceptual model.

3.4 Justification for Methodological Approach

The approach is justified, and this is due to various reasons. The research objective is architectural and integrative. The research is not experimental, and thus the goal is not to test a specific algorithm but to synthesize how various innovations like LLMs, RL reasoning, intelligent agents, and RAG can collectively support self-serve analytics. Another reason is that the domain is rapidly evolving. It is evident that text-to-SQL models change quickly, and this makes qualitative synthesis appropriate compared to empirical replication. Another reason is that RAG-based analytics assistants lead to socio-technical complexity. These blend technological components with organizational needs like governance and usability, which are better addressed through conceptual design. Design-science is vital in bridging practice and theory. This methodology supports creating an actionable framework grounded in rigorous literature.

3.5 Methodological Limitations

There are various limitations of this methodology. The first is a restricted source set. Few references were used, and this means that the information and insights were extracted from the references used during the research. The second is a lack of empirical evaluation. The study does not experimentally test an implemented retrieval-augmented assistant. The study instead constructs a theoretical architecture. There is qualitative bias. As with all qualitative analysis, the interpretation of this research was based on the judgment of the researcher when coding and synthesizing themes. We have a rapidly advancing field. LLMs development, reinforcement learning, and agentic systems may surpass the aspects of the proposed framework. The chosen methodology is appropriate despite these limitations.

D. FINDINGS

1. Core Capabilities of Retrieval-Augmented Analytics Assistants

There are six core capabilities that emerge across the literature. The first is a natural language interface, which is vital in enabling one to interpret the intent of the user. Users are able to interpret the insights, and this means they do not need technical skills. RAG is vital in supporting contextual retrieval. RAG enables assistants to access the metadata, schemas, examples, and documentation. This is vital in enhancing SQL/Python accuracy (Arslan et al., 2024; Shin et al., 2024).

Another finding is automated SQL reasoning. Assistants can generate complex SQL queries by drawing from text-to-SQL models. This involves joins, nested logic, and aggregations. We have Python Code generation. Python generation extends the capabilities of statistical analysis. It extends to data visualization and ML modeling. Another capability is agentic self-correction. The intelligent agent frameworks enhance reliability via iterative refinement. We also have semantic understanding, as semantic SQL and reinforcement learning help align user intent with the database logic.

1.2 Architectural Themes

Synthesis of literature suggests a convergent architecture that entails:

1.2.1 Retrieval Layer

Retrieval mechanisms are vital for bringing context into LLM reasoning. Shin et al. (2024) indicate that schema descriptions, metadata, and example queries must be indexed into a vector store to help provide relevant contextual snippets.

1.2.2 LLM Reasoning and Code Generation Layer

Luo et al. (2025) provide evidence that large models excel in the generation of SQL when supplemented with schema-aware information. Important to note that the layer handles semantic parsing, Python scripting, SQL synthesis, and high-level reasoning.

1.2.3 Semantic Knowledge Layer

Neves et al. (2019) depict that the semantic model bridges the gap between relational logic and natural language. This layer is vital in translating concepts like customer churn or monthly recurring revenue into standard definitions as well as query templates.

1.2.4 **Agent Orchestration Layer**

Ojuri et al. (2025) recommend an agentic pipeline that is capable of checking errors. Capable of iterative regeneration as well as optimization. This layer supervises the LLM as it helps ensure that generated queries are accurate. It helps guarantee its security and aligns with schema constraints.

1.2.5 Reinforcement Learning Feedback Layer

Ali et al. (2025) depicts that reinforcement learning enhances SQL reasoning. RL-driven models are vital in forming the foundation for long-term accuracy improvements. This enables the assistant to learn from execution success or failure.

1.3 Benefits of Retrieval-Augmented Analytics Assistants

There are various advantages that emerge that support the deployment of assistants in an enterprise analytics environment.

1.3.1 Democratization of Analytical Insights

Assistants are vital in reducing dependence as they offer natural language interfaces as well as automated SQL/Python reasoning. Non-technical users have the chance to utilize the system without limits.

1.3.2 Increased Speed of Insight Generation

Automation is vital in analytical workflows. Kumar et al. (2024) depict that automation is a driver as it makes the process better and faster. It promotes agentic correction, and queries can be retrieved within the shortest time possible.

1.3.3 Improved Accuracy

Semantic grounding and reinforcement learning collectively enhance accuracy as well as reduce hallucination risks.

1.3.4 Scalability Across Business Functions

The architecture is vital as it supports scalable deployment. This means that it can scale across multiple departments. It supports consistent reasoning and automated adaptation of new data domains.

1.3.5 Enhanced Analytical Maturity

It supports analytical capabilities by integrating SQL/Python generation, agent-based correction, and semantic layers.

1.4 Challenges

Despite the strong potential, there are various challenges, and they include

1.4.1 Semantic Ambiguity

Luo et al. (2025) depict that text-to-SQL models struggle with ambiguous questions as they lack explicit context.

1.4.2 Logical and Hallucinations

LLMs may produce syntactically valid but semantically incorrect SQL, especially without a well-structured retrieval input.

1.4.2 Integration Complexity

Integrating AI systems into existing big data pipelines demands infrastructural alignment: governance frameworks and semantic modeling.

1.4.3 Limited Evaluation Frameworks

Luo et al. (2025) and Ali et al. (2025) indicate that the lack of robust benchmarks limits the growth of the system, as there are no real systems to reflect real-world complexities.

1.4.4 Dependence on Data Quality

The metadata and schemas must be up-to-date to guarantee better and accurate reasoning. Poor documentation reduces the performance of semantic SQL.

1.5 Synthesis of Findings

The findings depict that retrieval-augmented analytics assistants represent the convergence of multiple research innovations. The success of this depends on the integration of RAG retrieval, semantic modeling, LLM-based SQL/Python, reinforcement learning, and agentic refinement. There are many benefits that exist, and this makes assistants better and effective for many users. Despite the benefits, limits exist, and they include ambiguity challenges and governance.

E. DISCUSSION

1. A Unified Framework for Retrieval-Augmented Assistants

The paper proposes a unified architecture that entails RAG-enabled retrieval grounds. It should include an LLM-driven system that converts queries to code. It entails semantic modeling that ensures alignment with the rules of the business. They need to include agentic oversight that validates and refines outputs. It should embrace reinforcement learning that improves performance over time.

1.2 Implications for Self-Serve Analytics

The assistants have a transformative potential. First, it enables the business users to gain autonomy. It enables the analysts to shift from code generation to strategic analysis. It enables organizations to achieve consistent metric definitions and helps ensure the insights are iterative and immediate.

1.3 Limitations and Risks

The concerns include the trustworthiness of the generated code. We have model errors that lead to incorrect decisions. Another concern is the need for continuous monitoring, and we have a dependence on the quality, as this impacts the accuracy.

1.4 Future Directions

There are a few research gaps noted in the research, and they include evaluation metrics for SQL-Python correctness. We have a longitudinal impact on the analytical team. The gaps included the integration of reinforcement learning into production environments. We also have improved semantic frameworks for LLM reasoning grounding.

CONCLUSION

Retrieval-augmented analytics assistants are vital, and they help promote self-serve business intelligence. They achieve this by integrating RAG frameworks, semantic SQL reasoning, LLM-based text-to-SQL systems, reinforcement learning, and reinforcement loops. Important to note that assistants are vital in democratizing analytics, and they accelerate data-driven decision-making. There are still challenges that need to be addressed, like the accuracy of semantics, reducing hallucinations, implementing

governance, and developing evaluations to help monitor the framework. These need to be addressed to boost trust in these systems. The literature shows that individual components of assistants have matured. They, however, indicate that their integration into a system that is unified is an active research frontier. The assistants offer better opportunity and a path that it is possible to bridge the gap between technical and non-technical users. This is effective as it helps expand organizational data literacy. It offers an opportunity to access reliable insights to support organizational decision-making.

REFERENCES

Ali, A., Baheti, A., Chang, J., Chi, T. C., Cui, B., Drozdov, A., ... & Zhang, Y. (2025). A state-of-the-art sql reasoning model using rlvr. *arXiv preprint arXiv:2509.21459*. https://arxiv.org/abs/2509.21459

Arslan, M., Munawar, S., & Cruz, C. (2024). Business insights using RAG–LLMs: a review and case study. *Journal of Decision Systems*, 1-30. https://www.tandfonline.com/doi/full/10.1080/12460125.2024.2410040

Kumar, Y., Marchena, J., Awlla, A. H., Li, J. J., & Abdalla, H. B. (2024). The AI-powered evolution of big data. *Applied Sciences*, *14*(22), 10176. https://www.mdpi.com/2076-3417/14/22/10176

Liu, X., Shen, S., Li, B., Ma, P., Jiang, R., Zhang, Y., ... & Luo, Y. (2025). A Survey of Text-to-SQL in the Era of LLMs: Where are we, and where are we going?. *IEEE Transactions on Knowledge and Data Engineering*. https://ieeexplore.ieee.org/abstract/document/11095853/

Luo, Y., Li, G., Fan, J., Chai, C., & Tang, N. (2025). Natural language to sql: State of the art and open problems. *Proceedings of the VLDB Endowment*, *18*(12), 5466-5471. https://dl.acm.org/doi/abs/10.14778/3750601.3750696

Neves, J., Bordawekar, R., & Tzortzatos, E. (2019). Demonstrating Semantic SQL Queries over Relational Data using the AI-Powered Database. In *Proceedings of the 1st International Workshop on Applied AI for Database Systems and Applications (AIDB'19)*. https://aidb-workshop.github.io/aidb2019-proceeding/6-neves.pdf

Ojuri, S., Han, T. A., Chiong, R., & Di Stefano, A. (2025). Optimizing text-to-SQL conversion techniques through the integration of intelligent agents and large language models. *Information Processing & Management*, *62*(5), 104136. https://www.sciencedirect.com/science/article/pii/S0306457325000780

Shin, J., Harzevili, N. S., Aleithan, R., Hemmati, H., & Wang, S. (2024). Retrieval-augmented test generation: How far are we?. *arXiv preprint arXiv:2409.12682*. https://arxiv.org/abs/2409.12682