Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

The Agentic Data Stack: A Comprehensive Analysis of Generative AI in the Evolution of Enterprise Analytics

Vaibhav Sudhanshu Naik
Independent Researcher, USA

ARTICLE INFO

Received: 13 Jan 2026

Revised: 16 Jan 2026

ABSTRACT

The global data analytics landscape is undergoing a fundamental transformation with the emergence of Generative Artificial Intelligence. Traditional data engineering relied on deterministic pipelines with rigid logic and explicit schema definitions. Any deviation from expected data formats caused immediate system failures. This fragility created substantial maintenance burdens for organizations. Data engineers spent the majority of their time on cleaning tasks rather than analytical activities. Large Language Models and Large Reasoning Models are ending this deterministic era. These technologies introduce Agentic Data Infrastructure, where systems develop a semantic understanding of the data they process. Analytics engines now interpret intent, reason about schema compatibility, and generate remediation logic autonomously. This article synthesizes evidence from recent architectural convergences in the industry. It analyzes the transition from deterministic pipelines to 'Agentic Data Stack,' focusing on three pillars: the standardization of context protocols for code migration, the application of semantic layers for storage optimization, and the unification of analytics through virtualization. The technology addresses critical challenges, including legacy code modernization, Text-to-SQL accuracy, data governance, and synthetic data generation. Organizations implementing these solutions achieve substantial productivity improvements and operational efficiency gains. The transformation shifts analytics from a discipline of syntax to one of semantics.

Keywords: Generative AI, Agentic Data Stack, Large Language Models, Automated Code Migration, Semantic Analytics, Vector Embeddings, Synthetic Data Generation, Agentic AI, MCP.

1. Introduction

The global data analytics landscape is undergoing a fundamental transformation. This shift represents the most significant change since the introduction of relational database management systems decades ago. For many years, data engineering has been primarily deterministic in nature. Pipelines were built using rigid logic and explicit schema definitions. These systems relied on imperative programming languages that followed precise instruction sequences. Any deviation from expected data schemas caused immediate pipeline failures. This fragility created a substantial maintenance burden for organizations, with engineers spending the majority of their time on cleaning tasks rather than analytical activities.

Generative Artificial Intelligence marks the end of this deterministic era. The deployment of Large Language Models (LLMs) and Large Reasoning Models within the data stack introduces a new paradigm: **Agentic Data Stack**. In this architecture, systems develop a semantic understanding of the

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/ Research Article

data they process. Analytics engines no longer merely execute code; they interpret intent, reason about schema compatibility, and generate remediation logic for errors autonomously.

This article examines this transition through a scholarly lens, synthesizing evidence from recent architectural convergences across the industry. Rather than focusing on individual tools, the analysis centers on three fundamental shifts: the standardization of **agentic context protocols** for infrastructure upgrades, the rise of **semantic intelligence layers** that optimize storage based on business logic, and the unification of analytics through **virtualization strategies**.

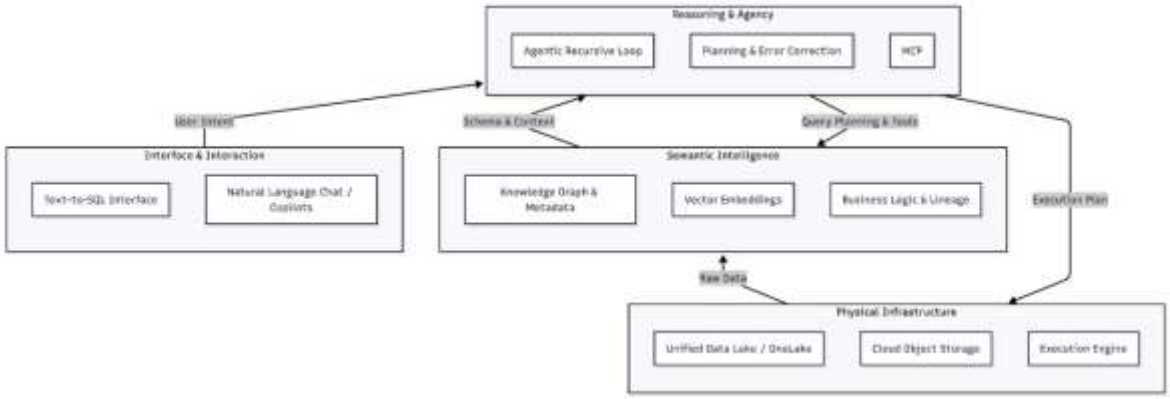

Figure 1: The Layered Architecture of Agentic Data Stack

1.1 The Economic Imperative

The urgency behind adopting GenAI in analytics stems from economic necessity rather than technological novelty. Modern enterprises face overwhelming technical debt. Legacy applications built on aging frameworks like Apache Spark or Java pose severe security risks. Monolithic on-premise Hadoop clusters create operational bottlenecks. Upgrading these systems traditionally requires extensive periods of manual activities. Engineers must perform dependency analysis, code refactoring, and extensive regression testing [2].

Recent findings from Cornell University and Stanford University reveal transformative insights. AI assistance fundamentally reshapes the labor productivity curve [3]. The results indicate that AI improves productivity for all workers. However, it delivers disproportionately positive impacts for novice or less experienced workers. This effectively compresses the skill gap between junior and senior personnel.

In data analytics contexts, GenAI tools can elevate junior analysts to senior engineer capabilities. The technology democratizes access to complex infrastructure tasks. Cluster optimization and code migration become accessible to broader talent pools. However, this transition introduces new complexities. Probabilistic models integrated into deterministic workflows create risks. "Hallucination" and "Model Autophagy Disorder" represent significant concerns where output quality degrades when models train on synthetic data recursively [4].

1.2 Cost Analysis: Inference vs. Engineering Hours

While the deployment of Generative AI introduces new compute costs, a comparative analysis of "Inference Economics" versus "Labor Economics" reveals a stark efficiency gap. The computational cost

of processing a legacy codebase through an LLM—even with multiple recursive calls for debugging—is negligible compared to the burdened cost of senior data engineering hours.

Running a sophisticated migration agent that consumes 100,000 tokens to refactor a complex pipeline may cost between $1 to $5 USD in inference charges. In contrast, a human engineer performing the same tasks—dependency mapping, syntax refactoring, and regression testing—may require days or weeks of effort, costing thousands of dollars. Furthermore, agents provide elasticity; they can parallelize the upgrade of hundreds of jobs simultaneously, a feat impossible for human teams without massive, temporary headcount expansion. The economic argument, therefore, is not merely about productivity but about the arbitrage between the declining cost of token-based intelligence and the rising cost of specialized technical talent.

However, financial guardrails are essential. Without strict 'maximum retry' limits, autonomous debugging loops can spiral, consuming excessive tokens on unsolvable errors. Thus, FinOps for GenAI must evolve from monitoring storage costs to monitoring agentic interaction cycles

2. Theoretical Framework: The Architecture of Cognitive Analytics

Understanding practical applications of GenAI in data analytics requires examining underlying theoretical innovations. These models function beyond simple text prediction. In analytics contexts, they operate as semantic reasoning engines. They manipulate Abstract Syntax Trees and highdimensional vector spaces with precision.

2.1 From Token Prediction to Logic Synthesis

The fundamental capability enabling GenAI to modernize data analytics is the translation of natural language into executable logical plans. When users request system upgrades for Spark jobs, models perform complex translation tasks. They do not simply swap keywords. They must understand code intent at a deeper level.

Consider the migration from Spark version two to version three. The handling of dates and timestamps changed significantly due to the adoption of the Proleptic Gregorian calendar. A simple syntax finder would miss the semantic implications of this change. Advanced LLMs trained on Spark documentation and thousands of GitHub migration commits recognize these patterns. They understand Java. Util. Date is being replaced by Java. Time at the library level.

This capability relies on the model's ability to parse Abstract Syntax Trees. The AST represents the abstract syntactic structure of source code in tree form. Advanced GenAI agents powering modern IDEs and cloud migration tools parse legacy code into ASTs. They identify nodes corresponding to deprecated methods. They then graft in new sub-trees representing modern equivalents [5]. This ensures generated code maintains structural validity rather than mere syntactic approximation.

2.2 The Model Context Protocol: A Standard for Agency

One critical barrier to "agentic" AI has been the lack of standardized interfaces. Models need consistent methods to interact with the tools they manipulate. The introduction of the Model Context Protocol by AWS in SageMaker Unified Studio represents a significant breakthrough [2].

The MCP acts as a universal translator between the reasoning model and execution environment. In the Spark Upgrade Agent context, the Client runs in the user's development environment. The Host houses the agent within SageMaker Unified Studio. The Server provides specific tools, including functions like analyze_dependency_tree, compile_project, and run_unit_test.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

By standardizing these interactions, AWS allows AI models to call tools deterministically. The AI does not guess compilation outputs. It invokes the compile tool via MCP. It receives exact error logs in structured formats. The model then uses this feedback to plan subsequent moves. This architecture transforms LLMs from passive text generators into active system administrators.

2.3 Retrieval-Augmented Generation and Semantic Layers

In data consumption scenarios, particularly Text-to-SQL, the theoretical challenge centers on "schema linking." An LLM cannot query databases it cannot see. Retrieval-Augmented Generation solves this by retrieving relevant table schemas. These schemas are injected into the model's context window.

However, raw schemas often prove insufficient. Column names like T_105_AMT lack meaning for LLMs. The emerging solution involves the implementation of "Semantic Layers" as exemplified by Databricks Lakehouse IQ and Snowflake Cortex. These systems feed LLMs not just schemas but complete metadata. This includes descriptions, data lineage, usage statistics, and historical query patterns. The result creates a "Knowledge Graph" of enterprise data. This allows AI to understand that T_105_AMT refers to "Q3 Adjusted Revenue."

Evidence indicates that providing semantic context represents the single biggest factor in improving Text-to-SQL system accuracy. Success rates improve substantially on real-world benchmarks [6]. Table 1 illustrates the contrasting impacts of Generative AI on workforce productivity and long-term model quality, highlighting the dual nature of AI integration in enterprise analytics environments.

Table 1: Productivity and Quality Dynamics in GenAI Systems

Aspect	Productivity Enhancement	Quality Degradation Risk
Primary Effect	Compresses skill gap between novice and expert workers	Model quality degrades when trained on AI-generated content
Impact on Junior Workers	Dramatic productivity improvements through AI-assisted task completion	Limited understanding of underlying logic reduces learning opportunities
Impact on Senior Workers	Moderate productivity gains with enhanced capabilities	Risk of over-reliance on automated solutions
Long-term Consequences	Democratization of complex analytical capabilities	Autophagy disorder causes precision and recall decline
Mitigation Strategy	Continuous training programs combining AI tools with fundamental skills	Data hygiene protocols separating synthetic from organic training data

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/ Research Article

3. Infrastructure Modernization and Ecosystem Analysis

3.1 Agentic Autonomy in Platform Migration

The transition from "copilot" assistance to "agent" autonomy is most visible in the domain of platform migration. This represents one of the most complex domains of data engineering, particularly for legacy frameworks reaching end-of-life. Migrating between major versions of distributed computing frameworks involves substantial challenges, such as breaking binary compatibility and subtle behavioral changes in null handling that can silently corrupt data.

Modern "Cognitive Refactoring Engines" address this through a standardized four-stage recursive loop:

1. **Dependency Graphing:** Agents scan build files to identify incompatible libraries and propose exact upgrade paths.
2. **Context-Aware Transformation:** Specialized agents employ transformations to understand variable scope and replace deprecated methods with modern equivalents.
3. **Compilation-Fix Loop:** The agent attempts compilation, captures errors via standardized protocols like the Model Context Protocol (MCP), and iteratively applies fixes.
4. **Execution and Validation:** Finally, agents submit test jobs to ephemeral clusters to validate output consistency against reference data.

Table 2: Standardized Cognitive Refactoring Workflow

Phase	Agent Actions	Human Oversight Role
Planning and Dependency Analysis	Scans build files, constructs dependency graphs, identifies incompatible libraries	Reviews and approves comprehensive migration plan
Automated Code Transformation	Employs AST-based transformations, understands variable scope, modifies source code	Monitors transformation progress and validates logic preservation
Compilation-Fix Loop	Attempts compilation, captures errors via MCP, applies fixes iteratively	Intervenes only for complex semantic issues requiring domain knowledge
Execution and Data Quality Validation	Submits test jobs to ephemeral compute, compares outputs against reference data	Validates final output quality and approves production deployment

3.2 Unified Cognitive Context and Virtualization

A major friction point in traditional analytics is the fragmentation of tools. Emerging "Unified Cognitive Architectures" solve this by integrating data engineering, data science, and business intelligence into single SaaS experiences. The core innovation here is the deployment of a "full-stack" copilot that possesses awareness across the entire data lineage.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/ Research Article

For example, strategies employed by platforms like Microsoft Fabric utilize a unified logical data lake (e.g., OneLake) to virtualize storage across different clouds. This enables "Zero-ETL" patterns, where AI engines can analyze data residing in external storage (like AWS S3) without physical movement. This unification allows the AI to trace a metric in a dashboard back to the specific pipeline that created it, significantly reducing environment setup times and maintenance overhead.

3.3 Adaptive Semantic Storage

Beyond code generation, Generative AI is now being applied to the physical optimization of data storage, a concept termed "Data Intelligence". Traditional data warehouses required manual tuning of sort keys, clustering, and vacuum schedules. New adaptive systems utilize LLMs to analyze organizational query history and learn business-specific relationships.

As seen in architectures like Databricks Lakehouse IQ, the system builds a semantic model unique to the enterprise. If users frequently join specific tables, the AI learns these patterns and autonomously optimizes the physical data layout, clustering data to align with query intent. This results in "Predictive Optimization," where the infrastructure tunes itself for price-performance without human intervention, substantially outperforming traditional manual warehousing strategies.

3.4 Multimodal Query Processing

The final pillar of modern cognitive infrastructure is the integration of unstructured data directly into the structured SQL environment. Historically, data warehouses were limited to text and numbers. New "Multimodal Analytics" engines allow users to process images, PDFs, and vector embeddings using standard SQL.

Platforms such as Google Cloud BigQuery have pioneered the integration of large foundational models (like Gemini) directly into the data warehouse. This allows analysts to execute queries such as "Select all customer reviews that mention product defects" or "Find products similar to this image description" without moving data to a separate machine learning environment. By embedding vector search and semantic similarity within the SQL syntax, these tools democratize access to complex AI capabilities for standard data analysts.

3.5 Self-Healing and Predictive Operations

While migration agents address technical debt, a new class of "Operational Agents" is addressing *technical fatigue* through self-healing and predictive capabilities.

- **Self-Healing Pipelines:** In ingestion workflows, agents now act as autonomous guardians against schema drift. Platforms like Microsoft Fabric and Snowflake are moving toward architectures where, upon encountering a file with an unexpected column, the pipeline does not fail. Instead, an agent analyzes the new schema, infers the data type, and autonomously refactors the downstream table definition to accommodate the change.

- **Predictive Optimization:** Traditional data warehousing required manual tuning of sort keys and vacuum schedules. Emerging "Predictive Optimization" agents (exemplified by Databricks Lakehouse IQ) utilize historical query patterns to autonomously execute maintenance tasks. These agents predictively cluster data and remove unused files during low-traffic windows, optimizing priceperformance without human intervention.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/ Research Article

4. Specific Applications and Emerging Frontiers

4.1 Text-to-SQL: Bridging Benchmarks and Reality

The ability to ask questions in plain English and receive SQL query results represents the most visible GenAI application. However, a significant gap exists between academic benchmarks and enterprise reality. Evidence reveals that while models achieve high accuracy on clean academic datasets, their performance drops substantially on real-world enterprise datasets [13].

The complexity penalty is measurable. Findings show a consistent accuracy decline for every additional column in a schema. Real-world tables often contain hundreds of columns. Real-world questions are colloquial and underspecified. A user asks for "sales" without specifying whether they mean booked, billed, or recognized revenue.

Successful implementations utilize multi-step "Chain-of-Thought" approaches. They first use an agent to "link" the schema by identifying relevant tables. They then ask clarifying questions to users. Only after disambiguation do they generate SQL. Snowflake's Cortex Analyst achieves high accuracy by strictly adhering to defined semantic models rather than guessing user intent [6].

4.2 Data Preparation: The Jellyfish Approach

Data cleaning remains notoriously labor-intensive. Recent work titled "Jellyfish: Large Language Models for Data Preprocessing" introduces specialized, instruction-tuned LLMs for data wrangling [14]. Developers fine-tuned smaller local LLMs specifically on data cleaning tasks. These included imputation, error detection, and schema matching.

These specialized models delivered performance competitive with larger models but at a fraction of the cost. They also provided higher data privacy by running locally. This supports the "Shift Left" methodology to data quality. Instead of cleaning data in the warehouse, Edge AI models can clean data at ingestion points. This ensures only high-quality data enters the lake.

4.3 Data Governance and Lineage Extraction

Understanding data lineage is critical for compliance with regulations like GDPR and CCPA. Traditional lineage parsers based on Regex or ANTLR fail when analyzing complex, dynamic SQL scripts. Recent findings on Schema Lineage Composite Evaluation demonstrate that LLMs are superior lineage extractors [5].

LLMs with substantial parameter counts achieved high accuracy in parsing table-level lineage from complex, obfuscated SQL scripts. This allows organizations to build "Policy-Aware" GenAI agents. These agents can analyze user requests and check the lineage of requested data. They determine whether data contains personally identifiable information from specific regions. They cross-reference this with company privacy policies and enforce access controls dynamically. This moves governance from static rules to dynamic, intent-based enforcement.

This shift necessitates a new framework for Non-Human Identity (NHI) management. As agents gain write-access to production schemas, organizations must implement 'Service Principal' auditing to distinguish between changes made by human engineers and those executed by autonomous optimization agents

4.4 Legacy Code Modernization Beyond Spark

The principles used in the AWS Spark agent extend to even older systems. Recent explorations on "Leveraging LLMs for Legacy Code Modernization" examine using GenAI to document and migrate code written in MUMPS and IBM Mainframe Assembly [7], [8]. These systems remain critical in healthcare electronic health records and banking infrastructure.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/ Research Article

The findings indicate that while LLMs struggled to generate perfect Assembly code from scratch, they were highly effective at documenting existing legacy code. They excel at reverse engineering business logic and suggesting modularization strategies. This "Documentation First" methodology represents a critical de-risking strategy for modernizing critical systems in banking and healthcare sectors. Table 3 compares the distinct strategic approaches of major cloud platforms in applying Generative AI to data analytics infrastructure, highlighting their unique value propositions.

Table 3: Platform-Specific Generative AI Strategies in Enterprise Analytics

Platform	Strategic Focus	Key Differentiator
AWS SageMaker Unified Studio	Infrastructure modernization and agentic code transformation	Model Context Protocol enabling autonomous multi-phase workflows
Microsoft Fabric	End-to-end analytics unification with full-stack AI awareness	OneLake virtualization enabling Zero-ETL across multi-cloud environments
Databricks Data Intelligence	Custom model training and semantic intelligence platforms	Organization-specific knowledge graphs learning from query patterns
Google Cloud BigQuery	Multimodal analytics integrating structured and unstructured data	Native vector search and semantic similarity within SQL syntax

4.5 Cognitive Persistence and Reasoning

While early GenAI implementations were stateless, the latest architectural shift involves the integration of Long-Term Episodic Memory. In traditional stateless interactions, an analyst must repeatedly explain context (e.g., "exclude test accounts from revenue") in every session. Emerging "Memoretic Architectures" utilize vector stores to persist user-specific preferences and historical debugging context. This allows the infrastructure to recall that a specific error encountered six months ago was resolved by a particular configuration change, proactively suggesting the same fix when the pattern re-emerges.

Furthermore, Continuous Active Learning is replacing static model deployments. Rather than waiting for quarterly model fine-tuning, modern cognitive layers implement real-time feedback loops. When a human analyst corrects an AI-generated SQL query, the system captures this "diff" and immediately updates its semantic knowledge graph. This ensures that the system does not repeat the same logic error twice, effectively "learning" the enterprise's unique dialect of SQL in real-time.

Finally, the industry is transitioning towards Inference-Time Reasoning (often termed "System 2" thinking). Unlike standard LLMs that predict the next token rapidly, new reasoning models allocate computational time to "think" before generating code. For complex data engineering tasks—such as

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/ Research Article

designing a normalized schema from unstructured logs—these models simulate multiple potential execution paths, evaluate the trade-offs of each, and select the optimal strategy before writing a single line of code.

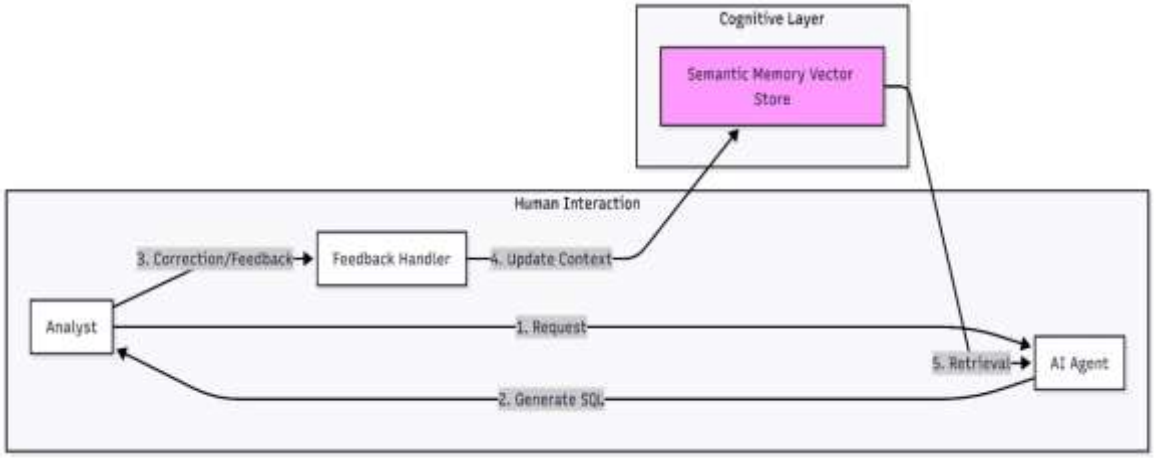

Figure 2: Continuous Learning Loop

5. Synthetic Data and Model Autophagy Risks

5.1 The Power and Promise of Synthetic Data

One of the most potent applications of GenAI in analytics is the generation of synthetic data. This solves two fundamental problems. First, it addresses privacy concerns by using fake data for activities. Second, it addresses scarcity by creating training data for machine learning models.

In industries like healthcare, real data cannot be shared due to privacy regulations. Synthetic data maintains the statistical correlations of original datasets without exposing individuals. Evidence shows that machine learning models trained on high-quality synthetic data can achieve comparable accuracy to those trained on real data. The primary advantage comes from the ability to "oversample" rare edge cases. Fraud detection and rare disease identification benefit significantly from balanced training sets that synthetic data provides.

Industry analysts predict that by the near future, the majority of businesses will use generative AI to create synthetic customer data. This will fundamentally change how organizations handle data collection and model training strategies.

5.2 The Risk of Model Autophagy Disorder

However, significant warnings have emerged regarding Model Autophagy Disorder. As Generative AI floods the web and enterprise data lakes, future models will inevitably train on data generated by past models. This creates a feedback loop where models consume their own outputs [4].

Evidence proves mathematically and empirically that without steady streams of fresh, real-world data, the quality and diversity of generative models will collapse. They drift into a reality of their own making. They amplify biases and artifacts present in synthetic data. For analytics applications, data leaders must implement rigorous "Data Hygiene" practices. They must tag data as "Synthetic" versus "Organic." They must ensure foundational models are periodically grounded with fresh, humangenerated data to prevent this degenerative drift. Table 4 categorizes Text-to-SQL query complexity levels and identifies the corresponding infrastructure requirements for achieving production-level accuracy.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Table 4: Text-to-SQL Query Complexity Taxonomy and Implementation Requirements

Complexity Level	Query Characteristics	Required Infrastructure
Simple Queries	Single table with straightforward selection conditions	Basic schema metadata and column descriptions
Moderate Complexity	Multiple table joins with standard aggregations	Explicit relationship definitions and foreign key documentation
Advanced Queries	Subqueries, window functions, temporal logic	Business rule documentation and fiscal calendar definitions
Expert-Level Queries	Recursive CTEs, complex analytical functions	Comprehensive semantic layer with example query library
Enterprise Production	Cryptic column names, implicit relationships, business context	Full knowledge graph with lineage tracking and validation rules

Conclusion

The evidence presented confirms that Generative AI has moved beyond the hype cycle. It now represents a fundamental component of data analytics infrastructure. The technology is reshaping how organizations handle data engineering, evaluation, and governance. Emerging Cognitive Refactoring Engines and standardized context protocols are effectively solving the technical debt crisis. By automating high-effort, low-value activities of version upgrades and refactoring, they free engineering capacity for innovation. Engineers can focus on building new capabilities rather than maintaining legacy systems. This shift from maintenance to innovation represents a strategic advantage for organizations that successfully implement these technologies.

The success of enterprise Text-to-SQL implementations depends entirely on AI understanding business semantics. The Semantic Layer is no longer optional; it has become a prerequisite for successful AI adoption in analytics. Organizations must invest in building comprehensive metadata repositories that capture not just technical schemas but business context and usage patterns. As AI agents gain abilities to execute code and query data, Policy-Aware architectures become the new standard for data governance. Rigorous validation loops, specifically those utilizing recursive compilefix cycles, are

essential. Organizations must balance automation benefits with appropriate human oversight and control mechanisms. AI does not replace data analysts; it augments them, particularly junior team members. The future analytics team is not smaller; it is faster, more capable, and able to tackle problems of previously insurmountable complexity.

The skill gap between junior and senior analysts has compressed significantly. This enables organizations to scale their analytics capabilities more effectively than traditional hiring and training methods allowed. Generative AI is transforming data analytics from a discipline of syntax to a discipline of semantics. The focus shifts from writing code to defining intent. Organizations that master this transition will achieve decision-making velocity that traditional competitors cannot match. They will leverage agents for infrastructure automation and copilots for evaluation. This combination creates a sustainable competitive advantage in data-driven decision-making. The path forward requires careful attention to emerging risks. Model Autophagy Disorder threatens long-term model quality if organizations do not maintain data hygiene practices. Security and privacy concerns demand robust governance frameworks. However, the potential benefits far outweigh these challenges for organizations willing to invest in proper implementation strategies. The agentic data stack represents the future of enterprise analytics. Organizations must begin their transformation journey now to remain competitive in an increasingly AI-driven business landscape.

References

1. AWS, "Announcing the Apache Spark upgrade agent for Amazon EMR," 2025. [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2025/12/apache-spark-upgrade-agent-amazonemr/
2. AWS, "What is Apache Spark Upgrade Agent for Amazon EMR?" [Online]. Available: https://docs.aws.amazon.com/emr/latest/ReleaseGuide/spark-upgrades.html
3. Erik Brynjolfsson, et al., "Generative AI at Work," NBER, 2023. [Online]. Available: https://www.nber.org/system/files/working_papers/w31161/w31161.pdf
4. Sina Alemohammad et al., "Self-Consuming Generative Models Go MAD," arXiv, 2023. [Online]. Available: https://arxiv.org/pdf/2307.01850
5. Melissa Z. Pan, "Measuring Agents in Production," arXiv, 2025. [Online]. Available: https://arxiv.org/pdf/2512.04123
6. Renee Huang, "Snowflake Cortex Analyst: Evaluating Text-to-SQL Accuracy for Real-World Business Intelligence Scenarios," Snowflake, 2024. [Online]. Available: https://www.snowflake.com/en/engineering-blog/cortex-analyst-text-to-sql-accuracy-bi/
7. Vimal Vyas and Shweta Singh, "Modernizing Java Applications with Amazon Q Developer and Visual Studio Code," AWS, 2025. [Online]. Available: https://aws.amazon.com/blogs/migrationand-modernization/modernizing-java-applications-with-amazon-q-developer-and-visual-studiocode/
8. Venugopalan Vasudevan and Jonathan Vogel, "Three ways Amazon Q Developer agent for code transformation accelerates Java upgrades," AWS DevOps, 2024. [Online]. Available: https://aws.amazon.com/blogs/devops/three-ways-amazon-q-developer-agent-for-codetransformation-accelerates-java-upgrades/
9. Arun Ulagaratchagan, "Introducing Microsoft Fabric: Data analytics for the era of AI," Microsoft Azure, 2023. [Online]. Available: https://azure.microsoft.com/en-us/blog/introducing-microsoftfabric-data-analytics-for-the-era-of-ai/

10. Hitachi, "Hitachi and Microsoft Enter Milestone Agreement to Accelerate Business and Social Innovation with Generative AI." [Online]. Available: https://www.hitachi.com/en-in/press/pressreleases/strategic-ai-partnership-formed/
11. Databricks Staff, "Data Intelligence in Action: 100+ Data and AI Use Cases from Databricks Customers," Databricks, 2025. [Online]. Available: https://www.databricks.com/blog/dataintelligence-action-100-data-and-ai-use-cases-databricks-customers
12. Google Cloud, "Expanding Duet AI, an AI-powered collaborator, across Google Cloud," 2023. [Online]. Available: https://cloud.google.com/blog/products/ai-machine-learning/duet-ai-ingoogle-cloud-preview
13. Hao Wang, et al., "Beyond SELECT: A Comprehensive Taxonomy-Guided Benchmark for RealWorld Text-to-SQL Translation," arXiv, 2025. [Online]. Available: https://arxiv.org/pdf/2511.13590
14. Haochen Zhang, et al., "Jellyfish: A Large Language Model for Data Preprocessing," arXiv, 2024. [Online]. Available: https://arxiv.org/pdf/2312.01678