Letter from the Editor-in-Chief

For decades, data management has been guided by a remarkably durable abstraction: users declare what they want, and the system determines how to obtain it. This idea gave us the relational model, SQL, cost-based optimization, physical data independence, and the large-scale data platforms that now underlie science, commerce, government, and everyday digital life.

The rise of large language models and generative AI does not replace this tradition. It asks us to extend it. In the previous issue, we examined the emergence of agentic data management: systems that plan, reason, invoke tools, and act in pursuit of high-level goals. This issue turns to the data-management foundation required for that agentic future. If agents are to operate reliably over enterprise knowledge, scientific literature, legal records, financial filings, images, tables, and multimodal archives, they cannot rely on prompting alone. They need data systems built for semantic access, evidence, uncertainty, cost control, and trust.

The central challenge is that the data we now want to manage is no longer confined to well-formed tables. It is scattered across documents, spreadsheets, figures, PDFs, web pages, logs, emails, code, audio, video, and heterogeneous data lakes. Much of its meaning is latent. A database may store a document, but not the facts implied by the document. A warehouse may contain extracted fields, but not the long tail of concepts users will ask about tomorrow. A search index may retrieve a passage, but not know whether the passage supports, contradicts, or merely resembles the user's claim.

This changes the nature of query processing. In traditional systems, a predicate is usually symbolic and exact. In the LLM era, a predicate may be semantic: "the filing indicates liquidity risk," "the report recommends future surgery," "the table supports a year-over-year comparison," or "the document contains evidence of noncompliance." Such predicates are not simple filters over stored values. They may require retrieval, extraction, normalization, reasoning, comparison, computation, and evidence selection. They may also be uncertain. The result of a query may be an answer, but it may also need to include confidence, provenance, bounds, examples, and an estimate of what it would cost to improve the answer.

The current state of the field is promising but incomplete. Retrieval-augmented generation has shown that language models can be grounded in external context, but retrieval alone is not enough. Long-context models can read more, but more context is not the same as better evidence. Structure-aware retrieval can respect sections, tables, and reading order, but many questions require reusable structured facts, not just better passages. Agents can search and reason, but without disciplined execution environments they can become slow, expensive, and hard to audit. Graphs, summaries, embeddings, caches, and generated questions all help, but each introduces new tradeoffs in quality, cost, latency, and maintainability.

The next generation of data systems must therefore treat semantic work as a first-class systems problem. We need logical abstractions for semantic operators, but also physical plans for executing them. We need optimizers that reason not only about CPU, memory, and I/O, but also about model choice, token cost, latency, retrieval coverage, extraction quality, approximation error, and provenance. We need indexes not only over strings and vectors, but over document structure, evidence, entities, questions, dependencies, summaries, cached model states, and reusable semantic fields. We need materialized views not only of relational joins, but of extracted facts, validated evidence records, and frequently used concepts.

A key future direction is the separation of semantic intent from model execution. Users and applications should be able to express what they mean at a high level, while the system decides whether to answer from structured data, retrieve evidence, run targeted extraction, invoke a stronger model, use a cheaper proxy, sample the corpus, or return an approximate result with bounds. This is the natural extension of data independence into the generative-AI era. The goal is not to hide uncertainty, but to manage it explicitly.

Another direction is the movement of repeated semantic computation from query time to indexing

time. If many users will ask related questions over the same corpus, then expensive work should not be repeated for every query. Document structure can be parsed once. Tables can be normalized once. Candidate facts can be extracted once. Dependency links can be scored once. Frequently used semantic fields can be materialized and validated. The online system can then become faster, cheaper, and more predictable. This is familiar database thinking, but applied to a new substrate: meaning.

At the same time, not everything can or should be precomputed. Users will continue to ask new, ambiguous, and evolving questions. Real workflows are iterative: users explore, inspect samples, revise definitions, validate outputs, and gradually decide what they truly mean. Data systems must therefore support the full lifecycle of semantic analysis: drafting a query, estimating feasibility, sampling results, refining a rubric, validating evidence, executing at scale, and promoting useful concepts into durable data assets.

Trust will be the defining constraint. In conventional data systems, a result can often be traced to a query over stored facts. In generative systems, a result may pass through retrieval, prompting, extraction, reranking, model judgment, and synthesis. Without provenance, evaluation, and auditability, such systems will be impressive but unsafe. The future database must be not only a system of record, but a system of evidence: it must know where an answer came from, what was read, what was ignored, what remains uncertain, and what would be required to make the answer stronger.

This is the opportunity before our community. The foundations of data management—declarativity, optimization, physical design, approximation, provenance, transactions, evaluation, and data independence—are not obsolete. They are urgently needed. But each must be reimagined for a world in which data is heterogeneous, queries are semantic, execution is model-driven, and correctness is inseparable from evidence and cost.

The future will not be built by language models alone, nor by databases unchanged from the past. It will be built by bringing the discipline of data management to the power and ambiguity of generative AI. I invite you to read this issue as a call to that work: to build systems that do not merely store information, retrieve passages, or generate fluent answers, but that transform raw, messy, multimodal data into trustworthy, cost-aware, and actionable knowledge.

Haixun Wang
EvenUp