

## Letter from the Special Issue Editor

The integration of Large Language Models (LLMs) into data systems marks a fundamental transition toward AI-native environments, where reasoning is elevated to a first-class primitive within the query-processing engine. Rather than serving as external tools, LLMs introduce intrinsic capabilities for autonomous semantic synthesis, multi-step logical inference, and the systematic processing of unstructured data. This shift allows modern data architectures to provide the same analytical rigor for heterogeneous content that was previously restricted to structured data. As autonomous agents and generative workflows redefine computing requirements, the data stack is being re-engineered to facilitate these high-level cognitive tasks at scale. This special issue examines the technical frontiers of LLM-powered data processing, featuring six articles that investigate the evolution from traditional architectures toward a new generation of agentic, reasoning-capable data environments.

Building on the need for agent-friendly systems, Ang et al. introduce the concept of **Agentic Data Environments**. They highlight the critical transition from read-only AI copilots to read-write autonomous agents capable of mutating environment state. To support these agents safely, they propose infrastructure that actively curates task-relevant signals (via Agentic Information Management and Retrieval) while bounding catastrophic risks through advanced mechanisms like state branching and deterministic Data Flow Control (DFC).

Grounding the architectural shifts in the enterprise, Chung et al. present Google Cloud’s transition to an **Agentic Data Cloud**, enabling intelligent and autonomous agents and applications. They describe several building blocks of their scalable AI Agentic Data Cloud ecosystem, designed to support production-scale autonomous applications and agents, including AI functions which are built-in directly into the core processing engines, natural language interfaces to data, and a modern search stack. Crucially, they assert that the trustworthiness of these non-deterministic agents relies heavily on a rigorous semantic foundation powered by universal metadata catalogs and rich semantic models.

The Özsu et al. claim that the traditional approach of treating AI as an auxiliary service layered over isolated databases is no longer sufficient. They frame this transition as a profound **"duality" (LLM4DB and DB4LLM)**, arguing that current data systems are ill-equipped for LLM agents that exhibit variable execution times, non-deterministic behaviors, and ambiguous write operations. To address complex analytical tasks over heterogeneous federated systems, they propose a novel multi-agent architecture where Specialized Data Agents (SDAs) interact natively with individual sources, coordinated by a central orchestrator ensuring strict data provenance.

Patel et al. introduce the **Semantic Operator Model**, implemented in the open-source LOTUS system. By introducing "model-data independence," this framework allows developers to write high-level natural language queries, such as `sem_filter`, `sem_join`, and `sem_topk`, while the underlying system transparently optimizes execution. The system explores cost-reducing proxies and model cascades while providing strict statistical accuracy guarantees relative to a high-quality reference algorithm.

Focusing on the algorithmic execution of the semantic operators, Trummer tackles the high costs of language-based joins with **Optimal Block Nested Loops Implementations for Semantic Joins**. Traditional semantic processing often relies on naive tuple-to-tuple comparisons, resulting in quadratic LLM invocation costs. Trummer proposes a block nested loops approach that batches multiple rows into a single prompt, providing a formal cost model to determine the optimal batch sizes within the strict constraints of limited LLM context windows.

Finally, in the last paper, **The Future Is Bespoke**, Eckmann et al. challenge the decades-old "One Size Fits All" philosophy of general-purpose engines. Instead, they propose utilizing LLM coding agents to automatically synthesize "one-size-fits-one" DBMSs tailored entirely to a specific workload. By shedding the overhead of schema interpretation, generalized data structures, and unused code paths, their synthesized bespoke engines demonstrate staggering performance improvements—achieving up to

an 11.78x speedup over state-of-the-art general-purpose engines like DuckDB.

Together, these six articles present a cohesive vision of the future. Whether it is through semantic operators, specialized hardware-software synthesis, or robust multi-agent orchestration, the boundary between the database and the artificial intelligence that consumes it is rapidly dissolving. We hope this special issue inspires further research into the foundations of AI-native data systems.

We would like to thank all the authors for their valuable contributions. We also thank Haixun Wang for the opportunity to put together this special issue, and Jieming Shi for his help in its publication.

Fatma Özcan  
Systems Research, Google