Semantic Data Systems: From Data Management to Data Understanding

Juliana Freire
New York University
New York, NY
juliana.freire@nyu.edu

ABSTRACT

For decades, data management systems have relied on syntactic, structural, and statistical representations of data, leaving semantic understanding largely to human users. This division has become increasingly problematic as users analyze data disconnected from their original collection context, and yet must determine whether datasets are relevant, compatible, and trustworthy. Large language models (LLMs) introduce a new computational capability: the ability to reason about the meaning of data and leverage accumulated background knowledge. I argue that semantic understanding should be viewed as a new systems primitive for data management, enabling a new class of *semantic data systems* in which understanding becomes a first-class computational capability that complements syntactic and statistical representations of data.

Drawing on our recent work spanning dataset discovery, semantic type discovery, schema matching, data harmonization, and agentic workflows, I show that LLMs alone do not solve data management problems: the central challenge is no longer enabling machines to understand data, but designing systems that can represent, maintain, evaluate, and exploit that understanding. These systems should externalize semantic knowledge into reusable artifacts, combine semantic reasoning with algorithms through structured workflows, subject decisions to human validation, and expose specialized capabilities as composable primitives that agents can orchestrate. This perspective suggests a research agenda for semantic data systems that extends beyond data discovery and integration toward a broader transformation of data management.

PVLDB Reference Format:
Juliana Freire. Semantic Data Systems: From Data Management to Data Understanding. PVLDB, 19(12): 4973 - 4980, 2026.
doi:10.14778/3827998.3838712

1 INTRODUCTION

Data management systems have long excelled at manipulating data. They efficiently process queries, integrate heterogeneous sources, discover relevant datasets, and support increasingly sophisticated analytical workflows. Over the past decades, the database community has developed elegant algorithms and scalable systems that operate on syntactic, structural, and statistical representations of data, including schemas, metadata, indexes, summaries, and learned representations.

Yet these systems have historically relied on a fundamental division of labor: systems manipulate data, while humans provide semantic understanding. People interpret the meaning of attributes, recognize that different values refer to the same entity, determine whether two datasets are compatible, assess the relevance of discovered data, and ultimately decide whether analytical results are trustworthy. Semantic understanding, the ability to reason about the meaning of data and relate it to background knowledge, has largely remained outside the computational system.

This division became increasingly problematic as the scale and diversity of data grew. The emergence of open data repositories, scientific data commons, enterprise data lakes, and large-scale observational datasets created unprecedented opportunities for data reuse. At the same time, it fundamentally changed how data are consumed. Data are now routinely analyzed by people who were not involved in their collection and who may have limited knowledge of the assumptions, conventions, and context underlying the data. As a consequence, finding relevant datasets, integrating heterogeneous sources, and drawing reliable conclusions increasingly depend on semantic understanding rather than computational efficiency alone. The bottleneck has shifted from processing data to understanding it.

Large language models (LLMs) fundamentally change this landscape. Beyond their impressive generative capabilities, they introduce something largely absent from data management systems: a computational mechanism capable of reasoning about semantics while leveraging broad accumulated knowledge. For the first time, this reasoning can be performed dynamically, without pre-specified schemas or ontologies, *making semantic understanding a first-class computational capability rather than remaining exclusively in the minds of human experts.*

However, semantic understanding alone does not solve data management problems. Like previous attempts to externalize knowledge through ontologies and knowledge graphs [16, 28], LLMs provide a way to represent and reason about meaning. Unlike these earlier approaches, they can dynamically construct semantic knowledge for new datasets and domains. Yet, without algorithms that make reasoning scalable and robust, without systems that transform semantic reasoning into reusable artifacts, and without humans who validate and refine the results, LLMs alone are insufficient for reliable data discovery, integration, and analytics.

The *central argument of this paper is that semantic understanding should be viewed as a new systems primitive for data management.* My research has explored this perspective across a sequence of problems, including dataset and semantic type discovery, schema

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3838712

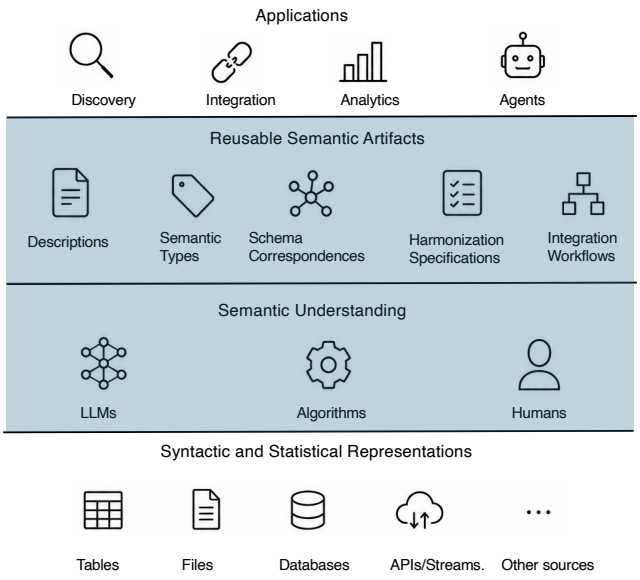

Figure 1: Semantic Data Systems. Semantic understanding becomes a first-class systems primitive that produces reusable representations that algorithms, humans, and agents can all build upon.

matching, data harmonization, and agentic workflows. The community demonstrated that LLMs could be applied to data management problems (see e.g., [25]). We then asked what assumptions remained and redesigned the problem around semantic understanding. While we addressed different problems, our approaches follow the same design philosophy: LLMs provide semantic understanding and background knowledge; algorithms and systems make it effective, scalable, and robust; humans provide oversight and judgment; and together they externalize semantic knowledge into reusable artifacts that improve data discovery, integration, and analytics. As illustrated in Figure 1, this primitive opens the opportunity to redesign data management systems by adding a semantic layer (highlighted in blue), making it possible to push semantic understanding into the computational system. I refer to systems designed around this new capability as *semantic data systems*: systems in which semantic understanding becomes a first-class computational capability that complements syntactic and statistical representations of data.

The remainder of this paper develops this perspective through a series of examples drawn from our research that progressively build semantic understanding at different levels of abstraction: understanding datasets, understanding attributes, understanding correspondences among data, and ultimately constructing agentic workflows that orchestrate semantic reasoning, algorithmic methods, and human expertise. Rather than providing a comprehensive survey of recent work, we discuss these examples to illustrate our perspective on how semantic understanding can be incorporated into the design of future data management systems.

2 UNDERSTANDING DATASETS

Dataset discovery is a prerequisite for data integration and analytics. Before data can be integrated or analyzed, users must determine whether relevant datasets exist and whether they are suitable for the task at hand. This requires understanding what a dataset contains, how it was collected, its spatial and temporal coverage, and how it relates to other datasets.

Most existing dataset search systems inherited their design from document search engines. They index textual metadata, such as dataset titles, descriptions, and tags, and retrieve datasets by matching keyword or natural language queries against these descriptions [2, 4, 5, 32]. While this approach is effective when metadata are complete and informative, it is often inadequate for structured data. Dataset descriptions are frequently incomplete, inconsistent, or simply absent, and many important characteristics of datasets, including their schema, value distributions, and relationships to other datasets, are not captured in textual metadata [18, 39]. This makes a significant fraction of datasets effectively invisible through search over repositories and data lakes.

Our first step toward addressing this limitation was to reconsider the representation of datasets used for search. Rather than relying solely on metadata, Auctus [3] constructs richer dataset representations by combining metadata with automatically generated profiles that summarize important dataset characteristics, including spatial and temporal coverage, value distributions, and relationships to other datasets. These representations support queries that cannot be answered through metadata alone, such as finding datasets that are joinable or unionable with a given dataset [10, 26, 27, 40], or identifying datasets covering a particular geographic region or time period [3].

Although richer profiling substantially improves dataset discovery, it does not fully address the semantic gap. Profiles summarize structural properties of datasets but provide only limited information about their meaning. Users and search systems still rely heavily on textual descriptions to assess whether a dataset is relevant, and many datasets lack descriptions that accurately reflect their contents.

This observation motivated AutoDDG [39]. Instead of asking users to document datasets manually, AutoDDG combines algorithmic profiling with LLM-based semantic reasoning to automatically generate textual dataset descriptions. The profiling stage extracts structural characteristics that would be difficult for an LLM to infer directly from raw tables or row samples. At the same time, LLMs serve two roles: uncovering the underlying semantics of the dataset and synthesizing these signals into coherent descriptions. These automatically generated descriptions significantly improve retrieval effectiveness, particularly for datasets with sparse or low-quality metadata.

This progression illustrates a recurring theme that will appear throughout this paper. Semantic understanding does not replace algorithmic methods. Profiling remains essential for characterizing datasets efficiently and systematically. LLMs contribute the ability to interpret these profiles and relate them to broader semantic concepts. Together, they produce richer dataset representations that improve discovery.

A key consequence of this approach is that these descriptions become *reusable semantic artifacts* across both human and machine reasoning. Once generated, they can support multiple downstream tasks without having to reinterpret the underlying data. They can be used both by human users searching for relevant datasets and by AI systems that reason about available data sources. Recent deep-research agents [1, 13] primarily operate over textual documents because structured datasets rarely provide comparable semantic descriptions. By automatically generating such descriptions, datasets become first-class information resources that can be indexed, retrieved, and incorporated into agentic workflows alongside scientific publications.

> **Design Principle.** *Reusable Semantic Artifacts*
> Algorithmic methods characterize datasets efficiently, but semantic understanding is needed to interpret them. By combining profiling with LLM-based reasoning, systems can construct reusable semantic artifacts—in this case, dataset descriptions—that improve discovery and can be consumed by both humans and AI agents.

3 UNDERSTANDING ATTRIBUTE SEMANTICS

Understanding the semantics of individual attributes is a prerequisite for many data management tasks, including dataset discovery, schema matching, join discovery, and data integration. While dataset descriptions characterize datasets as a whole, semantic types capture the concepts represented by individual attributes. For example, an attribute named Location may contain New York City boroughs, public schools, or hospitals. Distinguishing among these possibilities is essential for determining whether datasets are related and how they should be integrated.

Type Annotation. Column Type Annotation (CTA) has addressed this problem by assigning semantic types to attributes. Early approaches relied on manually curated ontologies and knowledge bases, while more recent methods use supervised learning or pretrained language models to classify attributes into a predefined vocabulary of semantic types [6, 15, 33, 38]. These methods have improved annotation quality, but they share an important limitation: they assume that the relevant semantic types are known a priori. This assumption is often violated in large, heterogeneous dataset collections, particularly in specialized domains where many concepts are domain-specific and absent from existing ontologies.

Large language models provide a natural mechanism for reasoning about attribute semantics without requiring task-specific training. Following the emergence of LLMs, several approaches explored their use for Column Type Annotation (CTA) by prompting models to classify attributes into predefined semantic types [17, 19]. These methods demonstrated that LLMs could substantially improve annotation quality while avoiding the need for task-specific training data. Our work on ArcheType [11] took a systems perspective. Rather than treating CTA as a prompting task, it models CTA as a classification problem and develops a multi-stage workflow that combines algorithmic methods with LLM reasoning. The system first selects representative examples from an attribute, constructs token-efficient prompts, queries the LLM, and finally maps the generated output back to the predefined semantic vocabulary when necessary. By combining algorithmic components with LLM inference, ArcheType achieves both robustness and cost efficiency while establishing a new state of the art for CTA.

ArcheType illustrates a design principle that recurs throughout this paper: LLMs are most effective when embedded within algorithmic workflows, rather than invoked as standalone solutions. This is a pattern we return to below and revisit again in Sections 4 and 5.

However, ArcheType still relies on users to provide the candidate semantic types from which the model selects. While this assumption is reasonable for well-defined benchmarks, it becomes increasingly difficult to satisfy in large dataset collections where neither the complete set of semantic types nor the appropriate level of abstraction is known in advance. This observation motivated a shift from type annotation to type discovery.

Type Discovery. Rather than asking an LLM to assign one of a predefined set of labels, StraTyper automatically discovers semantic types that are specific to the target dataset collection [20]. The system combines clustering algorithms, controlled LLM prompting, and iterative propagation to construct a consistent vocabulary of semantic types while minimizing redundant or inconsistent labels. It further supports attributes containing values from multiple semantic categories, a common occurrence in real-world datasets.

Similar to dataset understanding (Section 2), the contribution of the LLM here extends beyond producing an immediate answer. The discovered semantic types become reusable semantic artifacts that support multiple downstream tasks. They improve join discovery and schema matching by providing explicit semantic representations that complement value- and schema-based similarity. More broadly, they provide a semantic vocabulary tailored to a dataset collection that can be reused by both human users and AI systems to reason about the data. Rather than relying exclusively on manually constructed ontologies or knowledge graphs, semantic vocabularies can be constructed dynamically, adapted as collections evolve, and refined through continued interaction between algorithms, LLMs, and human experts.

> **Design Principle.** *Structured Workflows Combine Algorithms and Semantic Reasoning*
> LLMs provide semantic reasoning, but robust systems require carefully designed workflows that determine what information is presented to the model, constrain the reasoning task, interpret the model's outputs, and integrate them with algorithmic methods.

This principle recurs in the remainder of the paper. ArcheType and StraTyper provide the first examples, while MosaicJoin (Section 4.1) and Magneto (Section 4.2) demonstrate that it also applies to semantic correspondence tasks.

4 UNDERSTANDING CORRESPONDENCES

Data discovery and integration ultimately depend on identifying semantic relationships among data. Whether the goal is augmenting a machine learning model, integrating heterogeneous datasets, or harmonizing data with a reference model, systems must determine

when different representations correspond to the same underlying concept. These correspondences occur at multiple levels, including values, attributes, schemas, and complete data models.

Traditional approaches infer correspondences using syntactic similarity, statistical properties, manually engineered similarity functions, or learned embeddings [12]. These methods have enabled substantial progress over the past decades, but they remain limited when semantic equivalence cannot be inferred directly from the observed data. For example, two values may refer to the same entity despite having entirely different representations, or two schema attributes may describe the same concept while sharing neither names nor value distributions.

4.1 Value Correspondences

Many data discovery tasks rely on identifying value correspondences across datasets. Given a query attribute, join discovery aims to identify attributes in a data repository whose values can be joined with it. Such correspondences are particularly important for data augmentation, where additional datasets are used to enrich training data for machine learning models. In our earlier work on join-correlation queries [29, 31], we observed that joinability alone is not sufficient for effective data augmentation: many datasets are joinable, but only a small subset contributes information that improves predictive models. This led to methods that jointly reason about joinability and predictive utility. However, these methods, like most existing join discovery systems, assume equi-joins and therefore fail whenever semantically equivalent values are represented differently.

Semantic join discovery reframes the problem to address this limitation directly: instead of reasoning about exact value equality, the goal is to determine whether two attributes contain values that refer to the same entities despite differences in their surface representations. The availability of modern embedding models, including those derived from large language models, enables systems to reason about semantic similarity among values rather than relying on exact string equality. Existing methods, however, force a trade-off between accuracy and scalability. Value-level approaches compare embeddings for individual values, preserving fine-grained semantic evidence but requiring expensive pairwise comparisons [7]. Column-level approaches represent an entire attribute with a single embedding, enabling efficient retrieval but often failing to capture the value-level correspondences that determine whether a join is actually possible [8, 14].

MosaicJoin [9] addresses this problem by combining semantic representations with algorithmic techniques for scalable search. Instead of comparing all value embeddings, MosaicJoin constructs compact semantic sketches that preserve the semantic coverage of each attribute while dramatically reducing the cost of comparison. At query time, these sketches are combined with an efficient similarity measure and query subsampling to estimate joinability with provable accuracy guarantees. The result is a training-free system that retains the accuracy of value-level semantic matching while approaching the efficiency of column-level retrieval.

As in the previous examples, semantic understanding alone is insufficient. The semantic representations produced by modern embedding models make it possible to reason about value equivalence, but algorithms remain essential to organize this information into compact, searchable representations. By combining semantic reasoning with sketching and approximation algorithms, MosaicJoin demonstrates how systems can make semantic understanding both accurate and scalable.

4.2 Schema Correspondences

The same challenge arises at a higher level of abstraction. Instead of determining whether two values refer to the same entity, schema matching seeks to identify correspondences between attributes that represent the same concept. These correspondences are a prerequisite for data integration and harmonization, particularly in scientific domains where analyses often require combining datasets collected by different groups using different conventions.

4.2.1 Automatic Schema Matching. Schema matching has been studied extensively, and recent work has shown that both small language models and large language models improve the ability to identify semantic matches. Nevertheless, biomedical data exposed an important limitation of existing approaches [22]. Domain-specific terminology, heterogeneous attribute names, and ambiguous value representations often require knowledge that goes beyond the information contained in individual columns. At the same time, directly applying LLMs to all possible attribute pairs is computationally expensive and does not scale to realistic integration tasks.

Magneto addresses this problem through a two-stage workflow that combines small language models and LLMs [21]. A small language model first retrieves a limited set of candidate matches for each attribute. These candidates are then examined by an LLM, which uses its semantic reasoning capabilities to rerank them. This architecture substantially reduces the number of expensive LLM calls while preserving high matching accuracy. Magneto also uses LLMs to automatically generate training data for fine-tuning the retrieval model, eliminating the need for manually curated training sets. Experiments on a challenging biomedical benchmark and on standard schema matching benchmarks show that this combination of algorithmic retrieval and LLM-based reranking achieves both high accuracy and good generalization across domains.

Like ArcheType, Magneto treats semantic reasoning as one stage of a larger workflow rather than as a standalone solution. Algorithmic methods identify a small set of plausible matches, allowing the LLM to focus its reasoning where it is most valuable. The result is a system that is both more accurate and substantially more efficient than approaches that rely exclusively on traditional matching methods or direct LLM inference.

4.2.2 Human Validation. Automatic schema matching systems produce candidate correspondences rather than definitive answers. This is particularly true in domains such as biomedicine, where subtle semantic differences can make multiple candidate matches appear plausible even to domain experts. Incorrect correspondences may propagate throughout downstream integration and analysis pipelines, making human validation an essential component of data harmonization.

BDIViz addresses this challenge: it is an interactive visual analytics system designed to support the validation of schema correspondences [35, 36]. Rather than replacing human judgment, *the system*

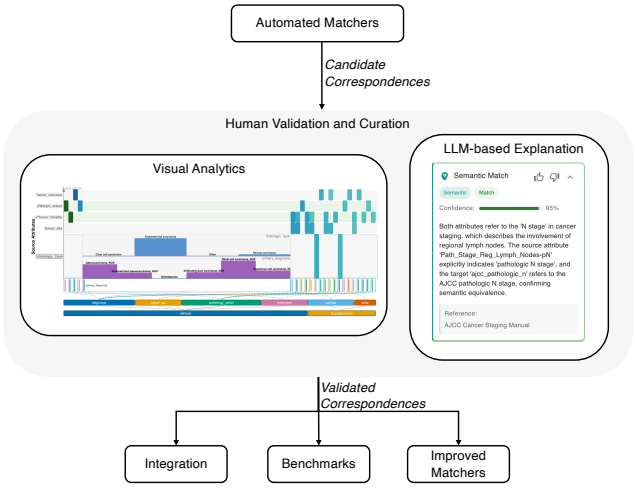

Figure 2: BDIViz combines interactive visualization with LLM-generated explanations to support efficient human validation of schema correspondences. Users inspect candidate matches, compare evidence, and validate ambiguous cases to transform candidate correspondences into validated knowledge. Validated correspondences are semantic artifacts that different tasks can use.

helps users make informed decisions by combining complementary sources of evidence. It summarizes candidate matches in an interactive heatmap that scales to large schemas, provides coordinated views that compare attribute names, values, and distributions, and uses LLMs to generate explanations and provide domain-specific context for ambiguous matches. By combining visualization with semantic explanations, BDIViz enables users to validate correspondences more efficiently and accurately than traditional curation workflows.

Beyond supporting data curators, BDIViz also supports the development of automated matching methods [35]. The system captures user validation decisions as reusable ground truth, making it possible to benchmark different schema matching algorithms, diagnose their failure modes, and rapidly evaluate new matching strategies. In this way, human validation not only improves individual integration tasks but also contributes to the development of better automated methods over time. An additional benefit is that *validated correspondences become reusable artifacts.* They support not only the current integration task, but also the systematic evaluation and improvement of automated matching methods.

BDIViz also illustrates a broader lesson for AI-assisted data management. While conversational interfaces provide a flexible mechanism for interacting with LLMs, they are not always the most effective interface for complex data management tasks. Schema matching requires users to compare many candidate correspondences, inspect multiple sources of evidence, and identify patterns across large schemas. Interactive visual representations summarize this information in ways that are difficult to achieve through conversation alone. In BDIViz, semantic explanations complement these visual representations rather than replacing them, allowing users to efficiently combine algorithmic evidence, semantic reasoning, and domain expertise.

> **Design Principle.** *Humans Remain in the Loop.*
> Semantic reasoning reduces, but does not eliminate, uncertainty. Effective data management systems should support efficient human validation, capture expert decisions, and reuse them to improve future workflows.

5 FROM CORRESPONDENCES TO WORKFLOWS

The examples discussed so far illustrate that effective data harmonization requires a diverse collection of semantic capabilities. Schema matching, value matching, transformation, validation, and explanation all require different methods, and no single algorithm performs best across all scenarios. Rather than embedding these capabilities within monolithic systems, they should be exposed as reusable primitives that can be composed into task-specific workflows.

5.1 Developing Primitives

To address this challenge, we developed BDI-Kit, an open-source library of composable harmonization primitives [23, 24]. Rather than providing a monolithic integration solution, BDI-Kit exposes a diverse collection of schema matching, value matching, transformation, and assessment methods that can be combined into custom harmonization workflows. The toolkit supports both algorithmic and LLM-based methods, allowing users to select the most appropriate technique for each task rather than relying on a single approach.

BDI-Kit separates semantic capabilities from workflow orchestration. It exposes harmonization primitives through multiple interfaces, including a Python API for developers and an MCP server[1] that allows AI assistants and agents to invoke the same primitives. This separation enables different interaction modalities while preserving a common set of reusable semantic operations.

5.2 Agentic Workflow Orchestration

While reusable primitives provide the building blocks for data harmonization, constructing effective workflows still requires deciding which primitives to invoke, how to compose them, and when human intervention is needed. These decisions depend on the characteristics of the data, the target schema, and the user's objectives. Manually assembling these workflows is challenging, but LLM agents have proven to be effective at planning and executing multi-step, complex tasks [37].

Unlike traditional workflow systems, which execute predefined sequences of operations, LLM agents can reason about the workflow itself. They can select appropriate primitives, determine the order in which they should be applied, and adapt the workflow based on the evaluation of intermediate results and user feedback.

Harmonia explores this paradigm for data harmonization [30]. Rather than replacing existing algorithms, it uses LLMs as workflow orchestrators that coordinate schema matching, value matching,

[1] https://modelcontextprotocol.io

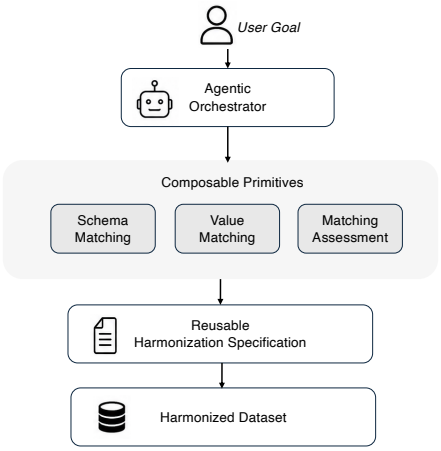

Figure 3: Agentic workflow orchestration for data harmonization. An agent translates a user goal into a harmonization workflow by selecting and coordinating reusable schema matching, value matching, and assessment primitives. The resulting harmonization specification is a reusable semantic artifact that documents the integration process and can be applied to harmonize additional datasets.

transformation, and assessment primitives provided by BDI-Kit [24]. When existing primitives are insufficient, the agent can also generate custom code to perform specialized transformations.

A key capability of Harmonia is that it reasons not only about which operations to execute, but also about the quality of their outputs. After invoking a primitive, the agent examines the returned results, assesses whether they are consistent with its semantic understanding and domain knowledge, and decides whether additional actions are needed. These actions may include invoking complementary primitives, correcting the output, or requesting clarification from the user. In this way, the *agent acts as an additional layer of semantic validation over existing algorithms rather than merely executing them.* The resulting workflows combine algorithmic methods, semantic reasoning, and human expertise.

Our preliminary evaluation suggests that this additional reasoning can substantially improve harmonization workflows [30]. In several instances, the agent identified incorrect schema and value matches produced by the underlying primitives, selected alternative candidates, and recommended corrected mappings to the user. These examples show that agents can contribute not only by automating workflow construction, but also by improving the overall quality of the resulting workflows.

Harmonia illustrates a different role for semantic reasoning than the examples discussed earlier. In previous sections, LLMs were used to understand datasets, attributes, and correspondences. Here, the LLM reasons about the harmonization process itself.

BDI-Kit was designed so that different AI agents can orchestrate its primitives. Harmonia was our first prototype. More recently, we have exposed BDI-Kit through MCP, allowing general-purpose agents such as Claude to invoke the same primitives. This separation between semantic capabilities and orchestration makes it possible to experiment with different agents while preserving a common set of harmonization primitives. As agent capabilities continue to improve, this orchestration layer is likely to evolve rapidly, while the underlying semantic primitives remain reusable across different agentic systems.

> **Design Principle.** *Agents Orchestrate Semantic Workflows*
> Agents contribute at a higher level of abstraction than individual semantic tasks. Rather than directly performing schema or value matching, they orchestrate specialized primitives, evaluate intermediate results, identify potential errors, recommend corrections, and engage users when additional context is required. This separation between semantic capabilities and workflow reasoning enables flexible and adaptive data integration systems.

6 OPEN PROBLEMS AND FUTURE RESEARCH DIRECTIONS

The examples presented in this paper illustrate how semantic understanding can improve individual data management tasks. Realizing the broader vision of semantic data systems requires advances in several directions. Below, we highlight a few particularly important challenges.

Evolving Semantic Artifacts. The semantic artifacts discussed throughout this paper – dataset descriptions, semantic types, correspondences, and harmonization specifications – are currently treated largely as static objects. In practice, they should evolve as new datasets become available, new evidence is observed, and users refine previous decisions. Developing methods for continuously maintaining and updating semantic artifacts while preserving consistency and provenance remains an open problem.

Semantic Memory. Current systems typically reason about each task in isolation. Future semantic data systems should accumulate knowledge across tasks, remembering previous integration decisions, user preferences, discovered semantic types, and harmonization specifications [34]. Such semantic memory would improve consistency, reduce repeated effort, and allow systems to continuously improve over time. Realizing this vision requires new representations for semantic knowledge. Today, semantic artifacts take many forms, including natural-language descriptions, semantic types, correspondences, and JSON harmonization specifications. Future systems will require richer, interconnected representations that can be queried, updated, reasoned over, and shared across tasks. Developing data models for semantic artifacts may prove to be as important for semantic data systems as the relational model was for traditional data management.

Human-AI Collaboration. Although LLMs substantially improve semantic reasoning, many decisions still require external context and domain expertise. A central challenge is determining when systems should act autonomously, when they should seek clarification, and how they should present evidence to users. This requires interaction models that go well beyond conversational interfaces and effectively combine visualization, explanation, and direct manipulation.

Autonomous Workflow Construction. Current agents can orchestrate existing primitives, evaluate intermediate results, and recommend corrections. Future systems should reason about entire data management workflows, automatically selecting algorithms, adapting execution strategies, and optimizing workflows under multiple objectives, including accuracy, computational cost, and human effort. Achieving this goal will require advances in both agent reasoning and data management algorithms.

Trustworthy Semantic Reasoning. As semantic understanding becomes part of the data management system, its outputs become increasingly consequential. Systems should be able to estimate uncertainty, explain their decisions, detect inconsistencies, and identify situations where semantic reasoning is likely to fail. Trustworthiness should become a first-class design objective rather than an afterthought.

Evaluation. Evaluating semantic data systems is substantially more challenging than evaluating traditional data management algorithms. Many semantic tasks admit multiple valid answers, and their outputs must often be assessed along multiple dimensions. For example, in AutoDDG, dataset descriptions were evaluated not only for correctness, but also for completeness and conciseness using a combination of existing descriptions, LLM judges, and human annotations. Similarly, constructing gold standards for schema matching required months of work by domain experts, while existing benchmarks have become increasingly saturated and no longer distinguish among the capabilities of modern LLM-based systems.

As semantic data systems become more complex, evaluation must also evolve. Beyond measuring the accuracy of individual tasks, future methodologies should assess the quality of reusable semantic artifacts, the effectiveness of human-AI collaboration, the robustness of workflow orchestration, and the ability of systems to improve over time through accumulated semantic memory.

7 CONCLUSION

For decades, the central challenge in data management was processing data efficiently. Today, the central challenge is understanding what data mean. This shift requires more than applying LLMs to existing problems. It calls for rethinking the architecture of data management systems around a new computational capability: semantic understanding.

The examples discussed in this paper illustrate one possible path toward this goal. Across problems ranging from dataset discovery and semantic type discovery to schema matching and data harmonization, a common pattern emerges. LLMs provide semantic reasoning and access to accumulated knowledge. Still, they are most effective when embedded within carefully designed systems that combine algorithmic methods, reusable workflows, human expertise, and increasingly, agentic orchestration. Throughout this process, semantic understanding is externalized into reusable artifacts—including dataset descriptions, semantic types, correspondences, and harmonization specifications—that support both human reasoning and automated workflows.

These observations suggest several principles for the design of future semantic data systems. They should externalize semantic understanding into reusable artifacts; combine algorithms and semantic reasoning through structured workflows; support efficient human validation and refinement of semantic decisions; and expose semantic capabilities as composable primitives that increasingly capable agents can orchestrate.

Although this paper has focused on data discovery and integration, we believe these principles extend well beyond these tasks. As semantic understanding becomes a first-class systems capability, it has the potential to reshape many aspects of data management, from data preparation and analytics to scientific discovery and decision support. The models and agents will continue to evolve. The enduring challenge for the database community is to design data management systems that can effectively represent, maintain, reason over, evaluate, and exploit semantic understanding as naturally as today's systems manage data itself. Meeting this challenge will require new representations, architectures, and evaluation methodologies that transform this understanding into reliable, reusable, and trustworthy data infrastructure.

ACKNOWLEDGMENTS

Although this paper presents my perspective on semantic data systems, the ideas it discusses were shaped by years of collaboration with an extraordinary group of students, postdoctoral researchers, research engineers, and collaborators at the NYU VIDA Center and beyond. Their creativity, curiosity, and willingness to explore new directions shaped much of the research described here. I am grateful for the opportunity to work with them and for everything they have taught me along the way.

This work was supported by NSF awards IIS-2106888 and OAC-2411221, the DARPA ASKEM program Agreement No. HR0011262087, DARPA D3M program, and the ARPA-H BDF program. The views, opinions, and findings expressed are those of the authors. They should not be interpreted as representing the official views or policies of DARPA, ARPA-H, the U.S. Government, or NSF.

REFERENCES

[1] Open AI. [n.d.]. OpenAI Deep Research. https://openai.com/index/introducing-deep-research
[2] Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *The World Wide Web Conference*. ACM, San Francisco CA USA, 1365–1375. https://doi.org/10.1145/3308558.3313685
[3] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: a dataset search engine for data discovery and augmentation. *Proceedings of the VLDB Endowment* 14, 12 (July 2021), 2791–2794. https://doi.org/10.14778/3476311.3476346
[4] CKAN. [n.d.]. The open source data portal software. https://ckan.org/
[5] Dataverse. [n.d.]. The Dataverse Project. https://dataverse.org
[6] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. TURL: Table Understanding through Representation Learning. *SIGMOD Rec. Association for Computing Machinery* 51, 1 (June 2022), 33–40.
[7] Yuyang Dong, Kunihiro Takeoka, Chuan Xiao, and Masafumi Oyamada. 2021. Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach. In *ICDE*. 456–467.
[8] Yuyang Dong, Chuan Xiao, Takuma Nozawa, Masafumi Enomoto, and Masafumi Oyamada. 2023. DeepJoin: Joinable Table Discovery with Pre-Trained Language Models. *Proc. VLDB Endow.* 16, 10 (2023), 2458–2470.
[9] Grace Fan, Eden Wu, Majid Daliri, and Juliana Freire. 2026. MosaicJoin: Compact Semantic Sketches for Value-Level Join Discovery. arXiv:2607.21781 [cs.DB] https://arxiv.org/abs/2607.21781
[10] R. Castro Fernandez, J. Min, D. Nava, and S. Madden. 2019. Lazo: A Cardinality-Based Method for Coupled Estimation of Jaccard Similarity and Containment. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 1190–1201. https://doi.org/10.1109/ICDE.2019.00109

[11] Benjamin Feuer, Yurong Liu, Chinmay Hegde, and Juliana Freire. 2024. ArcheType: A Novel Framework for Open-Source Column Type Annotation Using Large Language Models. *Proceedings of the VLDB Endowment* 17, 9 (2024), 2279–2292.

[12] Juliana Freire, Grace Fan, Benjamin Feuer, Christos Koutras, Yurong Liu, Eduardo Peña, Aécio Santos, Cláudio T Silva, and Eden Wu. 2025. Large language models for data discovery and integration: Challenges and opportunities. *IEEE Data Engineering Bulletin* 49, 1 (2025), 3 – 31.

[13] Google. [n.d.]. Gemini Deep Research. https://gemini.google/overview/deep-research

[14] Yuxiang Guo, Yuren Mao, Zhonghao Hu, Lu Chen, and Yunjun Gao. 2025. Snoopy: Effective and Efficient Semantic Join Discovery via Proxy Columns. *IEEE Trans. Knowl. Data Eng.* 37, 5 (2025), 2971–2985.

[15] Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 468–479.

[16] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2022), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

[17] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2024. Chorus: Foundation Models for Unified Data Discovery and Exploration. *Proceedings of the VLDB Endowment* 17, 8 (2024), 2104–2114.

[18] Laura M Koesten, Emilia Kacprzak, Jenifer FA Tennison, and Elena Simperl. 2017. The Trials and Tribulations of Working with Structured Data: -a Study on Information Seeking Behaviour. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 1277–1289.

[19] Keti Korini and Christian Bizer. 2023. Column type annotation using ChatGPT. In *CEUR Workshop Proceedings*, Vol. 3462. RWTH Aachen, 1–12.

[20] Christos Koutras and Juliana Freire. 2026. StraTyper: Automated Semantic Type Discovery and Multi-Type Annotation for Dataset Collections. https://arxiv.org/abs/2602.04004

[21] Yurong Liu, Eduardo H. M. Pena, Aécio Santos, Eden Wu, and Juliana Freire. 2025. Magneto: Combining Small and Large Language Models for Schema Matching. *Proceedings of the VLDB Endowment* 18, 8 (April 2025), 2681–2694. https://doi.org/10.14778/3742728.3742757

[22] Yurong Liu, Aécio Santos, Eduardo H. M. Pena, Roque Lopez, Eden Wu, and Juliana Freire. 2024. Enhancing Biomedical Schema Matching with LLM-based Training Data Generation. In *NeurIPS 2024 Third Table Representation Learning Workshop*. https://openreview.net/forum?id=RMZVUP7NgL

[23] Roque Lopez, Yurong Liu, Christos Koutras, and Juliana Freire. 2026. BDI-Kit Demo: A Toolkit for Programmable and Conversational Data Harmonization. In *Companion of the International Conference on Management of Data* (India) *(SIGMOD Companion '26)*. Association for Computing Machinery, New York, NY, USA, 94–97. https://doi.org/10.1145/3788853.3801608

[24] Roque Lopez, Aécio Santos, Christos Koutras, and Juliana Freire. 2026. BDI-Kit: An AI-powered toolkit for biomedical data harmonization. *Patterns* 7, 4 (April 2026), 101470. https://doi.org/10.1016/j.patter.2025.101470

[25] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? *Proc. VLDB Endow.* 16, 4 (Dec. 2022), 738–746. https://doi.org/10.14778/3574245.3574258

[26] Fatemeh Nargesian, Erkang Zhu, Renée J. Miller, Ken Q. Pu, and Patricia C. Arocena. 2019. Data lake management: challenges and opportunities. *Proceedings of the VLDB Endowment* 12, 12 (Aug. 2019), 1986–1989. https://doi.org/10.14778/3352063.3352116

[27] Fatemeh Nargesian, Erkang Zhu, Ken Q Pu, and Renée J Miller. 2018. Table union search on open data. *Proceedings of the VLDB Endowment* 11, 7 (2018), 813–825.

[28] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. 2008. Linking Data to Ontologies. In *Journal on Data Semantics X*, Stefano Spaccapietra (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 133–173.

[29] Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. 2021. Correlation Sketches for Approximate Join-Correlation Queries. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 1531–1544. https://doi.org/10.1145/3448016.3458456

[30] Aécio Santos, Eduardo Pena, Roque Lopez, and Juliana Freire. 2025. Interactive Data Harmonization with LLM Agents. In *ACM SIGMOD NOVAS workshop*. https://openreview.net/attachment?id=Ckn7Wo2TmS&name=pdf

[31] Aécio S. R. Santos, Aline Bessa, Christopher Musco, and Juliana Freire. 2022. A Sketch-based Index for Correlated Dataset Search. In *IEEE International Conference on Data Engineering (ICDE)*. IEEE, 2928–2941.

[32] Socrata. [n.d.]. The Socrata Open Data API. https://dev.socrata.com

[33] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating Columns with Pre-Trained Language Models. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 1493–1503.

[34] Eden Wu, Sonia Castelo, Yurong Liu, Cláudio T. Silva, and Juliana Freire. 2026. AgentTrails: Towards Trust and Reuse for Agentic Tasks. In *VLDB Workshop: DASHSys: Systems for Data-centric Agents with Human-in-the-loop*.

[35] Eden Wu, Christos Koutras, Cláudio T. Silva, and Juliana Freire. 2026. BDIViz in Action: Interactive Curation and Benchmarking for Schema Matching Methods. In *Companion of the International Conference on Management of Data* (India) *(SIGMOD Companion '26)*. Association for Computing Machinery, New York, NY, USA, 138–141. https://doi.org/10.1145/3788853.3801596

[36] Eden Wu, Dishita G Turakhia, Guande Wu, Christos Koutras, Sarah Keegan, Wenke Liu, Beata Szeitz, David Fenyo, Cláudio T. Silva, and Juliana Freire. 2026. BDIViz: An Interactive Visualization System for Biomedical Schema Matching with LLM-Powered Validation. *IEEE Transactions on Visualization and Computer Graphics* 32, 1 (Jan. 2026), 1208–1218. https://doi.org/10.1109/TVCG.2025.3634843

[37] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=WE_vluYUL-X

[38] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. 2020. Sato: Contextual Semantic Type Detection in Tables. *Proc. VLDB Endow.* 13, 12 (2020), 1835–1848.

[39] Haoxiang Zhang, Yurong Liu, Aécio Santos, Wei-Lun (Allen) Hung, and Juliana Freire. 2026. AutoDDG: Automated Dataset Description Generation using Large Language Models. *Proceedings of the ACM on Management of Data* 4, 1 (April 2026), 1–27. https://doi.org/10.1145/3786626

[40] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. 2019. JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. In *Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19)*. ACM, New York, NY, USA, 847–864. https://doi.org/10.1145/3299869.3300065