How Human-Centric Are Our Graph Data Abstractions?

Angela Bonifati
Lyon 1 University, CNRS LIRIS, France
IUF, Lyon, France

Anastasia Dimou
KU Leuven, Flanders Make@KULeuven, Leuven.AI, Belgium

Stefania Dumbrava
ENSIIE, SAMOVAR, Télécom SudParis, France
IRIF, INRIA Paris, France

George Fletcher
Eindhoven University of Technology, The Netherlands

Katja Hose
TU Wien, Austria

George Konstantinidis
University of Southampton, UK

Jose Emilio Labra Gayo
WESO Lab, University of Oviedo, Spain

Wim Martens
University of Bayreuth, Germany

Nina Pardal
University of Edinburgh, UK

Liat Peterfreund
Hebrew University of Jerusalem, Israel

Katherine Thornton
Yale University Library, New Haven, CT, USA

Maria-Esther Vidal
Data Science Institute, Leibniz University Hannover, Germany
TIB-Leibniz Institute for Science and Technology, Hannover, Germany
L3S-Research Center, Hannvcer, Germany

Hannes Voigt
Neo4j, Leipzig, Germany

Abstract

Graph data abstractions are often assumed to be intuitive, but experience shows that they are not equally understandable or usable in practice. In this vision and challenges paper, we examine the human-centricity of contemporary graph data abstractions through four lenses: researchability, usability, teachability, and societal impact. Drawing on diverse real-world use cases, ranging from clinical data and collaborative knowledge bases to biological and pangenomic graphs, we distill insights from database research, human-computer interaction, and education. Based on this analysis, we identify open research challenges that must be addressed to make graph abstractions easier to study, use, learn, and reason about.

Transactions on Graph Data and Knowledge, Vol. 4, Issue 2, Article No. 2, pp. 2:1–2:31
TGDK Transactions on Graph Data and Knowledge
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2012 ACM Subject Classification Information systems → Graph-based database models; Information systems → Resource Description Framework (RDF); Human-centered computing → Human computer interaction (HCI); Social and professional topics → Computing education; Information systems → Query languages; Computing methodologies → Knowledge representation and reasoning

Keywords and phrases graph data abstractions, property graphs, RDF, human-centric data management, usability, computing education

Digital Object Identifier 10.4230/TGDK.4.2.2

Category Position

Funding *Angela Bonifati*: Partially supported by the ANR Verigraph (ANR-21-CE48-0015) project.
Stefania Dumbrava: Partially supported by the ANR JCJC VERDI (ANR-24-CE25-1109) project.
Katja Hose: Partially supported by the Vienna Science and Technology Fund (WWTF) [DORSET, grant id: 10.47379/ICT25032].
Jose Emilio Labra Gayo: Partially funded by projects SHAKIGRA: Shaping Knowledge and Interoperable Graphs, code: PID2024-157010OBI00, and project SEK25-GRU-GIC-24-089.
Nina Pardal: Partially funded by DFG grant VI 1045-1/1, and Huawei Strategic Talent Program.
Liat Peterfreund: Supported by Israel Science Foundation 2355/24.
Maria-Esther Vidal: Partially funded by Leibniz Association, program "Leibniz Best Minds: Programme for Women Professors", project TrustKG-Transforming Data in Trustable Insights (Grant P99/2020), and by the Lower Saxony Ministry of Science and Culture (MWK) with funds from the Volkswagen Foundation's zukunft.niedersachsen program (CAIMed – Lower Saxony Center for AI and Causal Methods in Medicine; GA No. ZN4257).

Acknowledgements This work builds on discussions and outcomes from the Dagstuhl Seminar 24102 *Shapes in Graph Data: Theory and Implementation* [2].

Received 2026-02-27 **Accepted** 2026-06-16 **Published** 2026-09-03

Editors Stefania Dumbrava, George Fletcher, Matteo Lissandrini, and Riccardo Tommasini

Special Issue Data Management for (Knowledge) Graphs

Generative AI Declaration During the preparation of this work, the authors only used generative AI assistants for language and style improvement, for support in structuring parts of the text, and for polishing visual elements (Canva AI). No references were generated by these tools. All AI-assisted output was reviewed and edited by the authors, who take full responsibility for the content of this article.

1 Introduction

Graph data abstractions have emerged as a central paradigm for representing and querying interconnected data. From ER diagrams [34] and Sowa's conceptual graphs [92] to data models such as XML [28], RDF [57] and property graphs [6, 82], and their associated query languages, graph abstractions are now widely used across domains, including data integration, collaborative knowledge bases, and scientific data management [7, 83]. Their appeal is often attributed to the explicit representation of relationships, which supports flexible schema evolution and arguably offers more intuition and easier constructs than relational or hierarchical models.

Figure 1 provides a high-level historical perspective on the evolution of graph data abstractions. Successive abstractions have accumulated in this historical progression, resulting in a complex and diverse ecosystem of coexisting graph data models and languages. Practical examples of graph abstractions have often evolved in close interaction with human needs, practices, and communities. Wikidata [98] is such a bottom-up initiative that has emerged while inherently accounting for human-centric aspects of graph data representations such as usability, learnability, and explainability. Cypher offers a particularly instructive example.

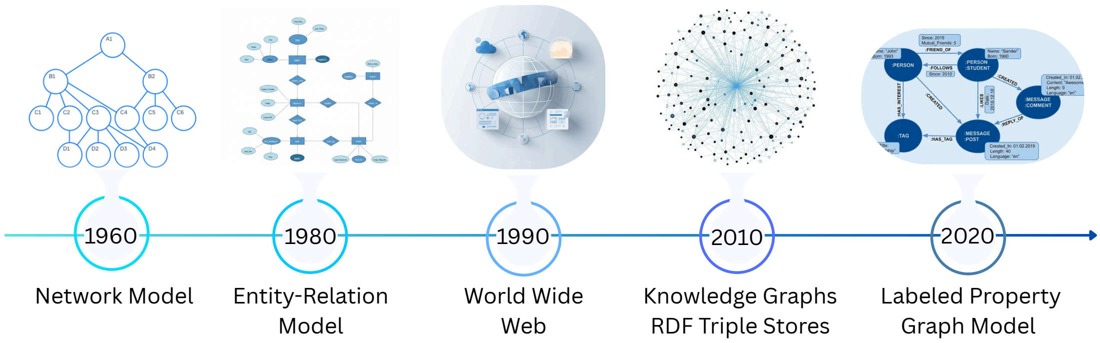
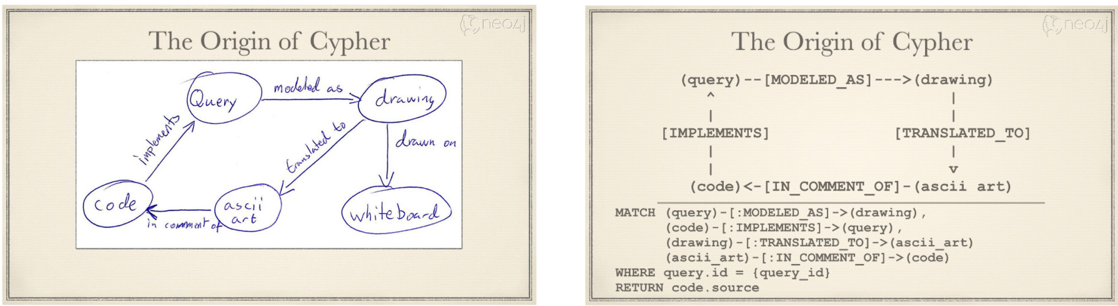


Figure 1 Historical evolution of graph data abstractions: from the CODASYL Network Data Model (NDL), introducing explicit navigational, pointer-based record graphs; through Entity–Relationship (ER) modeling, elevating graphs to a high-level conceptual design formalism; to the World Wide Web, which operationalized a global, large-scale hyperlink graph over semi-structured documents; followed by Knowledge Graphs, emphasizing semantic integration and ontology-driven interoperability; and culminating in the Labeled Property Graph model and ongoing standardization efforts (e.g., GQL), marking the consolidation of expressive, interoperable, and queryable graph data systems.



(a) Conceptual origin whiteboard drawing.

(b) 2D ASCII-art comment (top) and Cypher ASCII-art patterns syntax (bottom).

Figure 2 Evolution of Cypher from informal sketches to declarative pattern matching (from [60]).

▶ **Example 1** (The Human-Centric Design of Cypher)**.** The origins of the Cypher graph query language illustrate how graph data abstractions can emerge in a human-centric, bottom-up manner. As documented in its historical evolution [60], Cypher was shaped not by a pre-existing formal model, but by observing how practitioners interacted with graph data in early applications. Users reasoned about traversing chains of relationships and interesting patterns of connected entities on whiteboard or scratchpad drawings (cf. Figure 2a) and documented them in 2D ASCII-art comments next to their imperative traversal code (cf. top of Figure 2b). Cypher ported this practice to a query language. It introduced pattern-matching constructs with an ASCII-art syntax [69], as well as labeled nodes and relationships that are closely aligned with the graph mental models of users (cf. bottom of Figure 2b). These features abstracted recurring usage patterns into language constructs, reducing the gap between the intended query formulation and its execution. From a human-centric perspective, Cypher's design prioritizes interpretability and learnability alongside expressiveness, enabling users with limited formal training to interact with graph data [66, 69].

Early approaches in the neighboring discipline of relational databases had explicitly treated usability as a first-class property of a data abstraction, even at a system-design level, alongside formal semantics and query evaluation. In this area, human-centric aspects have been studied in a principled manner. Seminal empirical studies of query languages such as SQUARE, SEQUEL, and Query-by-Example [30, 112], and more recently of Relational Diagrams [42], systematically examined how users understood, learned, and applied these abstractions in practice, grounding language design in human factors rather than purely formal considerations [80, 95]. These studies established a tradition in which abstraction quality was evaluated not only by expressiveness or efficiency but also by accessibility.

In graph data management approaches, these aspects have not always been first-class design principles nor have they been scrutinized by the research community in the same principled way.

Thus, human-facing properties of graph abstractions, e.g., their interpretability, usability, learnability, and the way they differentiate user roles, remain unevenly understood. Empirical studies in graph data visualization and exploration show substantial variation in how users interpret graph structure, semantics, and query results, particularly across different levels of technical expertise [4, 11, 19, 20, 59, 61, 110]. Research in education and human-computer interaction further suggests that graph-based representations pose distinct cognitive challenges compared to tabular or tree-based abstractions, affecting users' ability to form accurate mental models [21, 48, 108]. Crucially, empirical evidence shows that the ability to successfully execute graph queries or complete interaction tasks does not necessarily imply a sound conceptual understanding of the underlying data abstraction [21, 71].

These limitations increasingly matter in practice. Community-maintained knowledge bases such as Wikidata rely on graph schemas, constraints, and validation mechanisms to coordinate large and heterogeneous contributor populations [51, 63, 78, 98]. Scientific applications in domains such as biology, ecology, and genomics use graph abstractions to integrate heterogeneous datasets and support complex analytical workflows [29, 36]. Recently, graph-structured data has been used with large language models (LLMs) and AI systems to support downstream reasoning and interaction, widening the audience of graph abstractions to include non-expert users and automated agents [72]. Given the increasing synergy between graph abstractions and the booming AI landscape, for tasks ranging from data annotation and model training to supporting human-in-the-loop explainability and decision-making, aligning the human mental model with graph abstractions becomes crucial. In these settings, shortcomings in the learnability, interpretability, or usability of graph abstractions directly affect data quality, system reliability, and downstream behavior.

Vision and Contributions

We use the term *graph data abstraction* to refer to the conceptualization of data as a graph, where nodes represent entities and edges represent relationships. In such abstractions, information emerges not only from individual data elements, but also from the structure of their interconnections, enabling the explicit representation of patterns, dependencies, and contextual relationships within the data. We draw the boundary deliberately: an abstraction is *graph-based* in our sense when relationships are first-class, independently addressable modeling primitives that may themselves carry identity, type, or attributes. This includes RDF and property graphs and excludes models in which inter-entity links are implicit or value-derived, e.g., the relational model, where a foreign key is a value-based join rather than a first-class edge.

Graph abstractions are increasingly adopted because certain data and tasks, e.g., densely interlinked entities, multi-hop reasoning over relationships, and schema-flexible integration of heterogeneous sources, are represented more faithfully and queried more naturally as graphs.

Yet, the expressive power of graph abstractions does not automatically translate into human accessibility or understanding. Because such abstractions are *already* widely deployed across collaborative, scientific, and AI-mediated settings, the question we pursue is not whether to adopt them, but how to make them human-accessible to the diverse audiences that now encounter them.

Human-centricity is not the only lens through which graph abstractions can be evaluated. Such abstractions also serve machine-oriented purposes, including automated reasoning, recommendation, and learning, where considerations such as expressiveness and performance are central. We view human-facing properties as complementary to these concerns rather than as a replacement for them. Our focus on human accessibility reflects the observation that graph abstractions are increasingly authored, queried, taught, and relied upon by people, and that these human-facing aspects have received comparatively less systematic attention.

We argue that graph abstractions should be designed not only to represent interconnected data, but also to treat human-accessible aspects as first-class design principles, supporting how users observe, navigate, interpret, manipulate, and reason about such interconnections.

We use the term *human-centric graph data abstractions* for graph abstractions that treat human-facing properties as first-class design and evaluation concerns. This specializes to graphs a broader human-data interaction agenda, which has produced consolidated design recommendations [97]. We identify these as *interpretability* (users can understand what the abstraction represents and why a result was produced), *usability* (users can perform intended tasks efficiently and with few errors), *learnability* (newcomers can acquire competence with the abstraction), *role differentiation* (distinct roles, such as schema designers, query writers, analysts, consumers, and indirectly affected parties, interact with the same abstraction with different goals, expertise, and responsibilities, and the abstraction accommodates them), and *societal impact* (consequences for people who are not direct users are made visible and accountable). We examine these properties through the four analysis lenses used consistently throughout the paper: researchability, usability, teachability, and societal impact. In these abstractions, the relational and topological properties of graph representations are not merely expressive features of the data model, but cognitively accessible structures that users can effectively exploit for reasoning, interaction, and decision-making.

We examine real-world graph-based use cases from clinical data management, collaborative knowledge bases, and biological and pangenomic analysis to surface recurring human-facing tensions, particularly around usability and interpretability, that persist across domains and abstraction paradigms (Section 2). As our main contribution, we distill these tensions into research challenges, spanning institutional, methodological, operational, and applicative concerns. These challenges are grounded in prior results from database theory, human-computer interaction, and education (Section 3) and we refine them into a concrete roadmap of theory- and application-driven research questions (Section 4).

Methodology

This paper is a *vision and challenges contribution*. We synthesize observations from diverse use cases and prior empirical work to delineate an agenda for human-centric graph abstraction research. Our approach proceeds in three stages.

First, we selected use cases covering two distinct paradigms of graph data abstractions: open, collaborative, and community-driven graph construction (Wikidata, Section 2.1), and expert-curated, integration-heavy scientific graph data management from clinical and biological domains (Section 2.2). Several co-authors are directly involved in these projects, providing first-hand knowledge of their human-facing challenges.

Second, for each use case, we synthesized published empirical findings, e.g., controlled user studies [71], system evaluations [76], and community surveys [78, 81, 85], along our four human-centric dimensions: researchability, usability, teachability, and societal impact (Section 4).

Third, recurring tensions observed across two or more use cases were elevated to research challenges (Section 3), organized by institutional, methodological, operational, and applicative categories; these are subsequently refined into actionable research questions in Section 4.

To support traceability, each challenge in Section 3 is annotated with the specific use case(s) that motivated it, and Table 1 consolidates these into the connective tissue of the paper. Reading it left to right: the institutional, methodological, operational, and applicative *challenges* of this section (IQ/MQ/OQ/AQ) are refined into the researchability, usability, teachability, and societal *roadmap* dimensions of Section 4; each dimension concerns a specific human *subject*, i.e., researcher, direct user, learner, or indirect user, and is pursued through theory-driven and application-driven research questions (RQ/UQ/TQ/SQ).

Table 1 Overview of research challenges, roadmap, concerned human subjects, and the main underlying theory-driven and application-driven research questions.

	Institutional (Sec. 3.1)	**Methodological** (Sec. 3.2)	**Operational** (Sec. 3.3)	**Applicative** (Sec. 3.4)
Challenges (Sec. 3)	IQ1, IQ2	MQ1, MQ2, MQ3	OQ1, OQ2, OQ3	AQ1, AQ2
Roadmap (Sec. 4)	**Researchability** (Sec. 4.1)	**Usability** (Sec. 4.2)	**Teachability** (Sec. 4.3)	**Societal** (Sec. 4.4)
Subject	**Researcher**	**Direct User** (data modeler, query writer, etc.)	**Learner** (person learning to become a practitioner or researcher)	**Indirect User** (person affected by use of database technology)
Theory	RQ1	UQ1, UQ2, UQ3	TQ1, TQ2	SQ1, SQ2
Application	RQ2	UQ4, UQ5	TQ3	SQ3, SQ4

2 Motivational Use Cases

We present a set of use cases that illustrate how graph data abstractions are designed, learned, and used across different domains, including collaborative knowledge bases, as well as clinical and biological data management. These applications show that graph abstractions can successfully support complex tasks and serve diverse communities. At the same time, they reveal recurring difficulties related to abstraction understanding, modeling choices, and role differentiation. In several settings, users rely on documentation, peer support, or specialized tools to work effectively with these abstractions. While such practices demonstrate that graph abstractions can be adopted successfully in practice, they do not yet amount to a systematic understanding of how these abstractions align with users' mental models across roles and domains.

2.1 Collaborative Graph Abstractions in Wikidata

Wikidata is a collaboratively curated knowledge base of structured data with an active editor community [98]. Data can be accessed through the Wikidata Query Service [107] as well as through public API endpoints [105]. As a cross-domain knowledge graph, Wikidata enables unrestricted reuse of its data under the Creative Commons 0 (CC0) license [106].

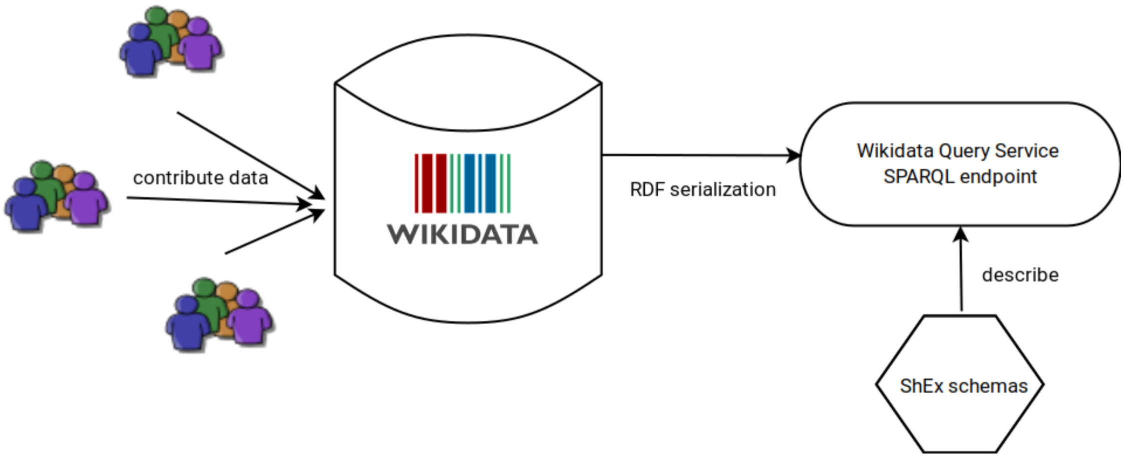


Figure 3 Wikidata editors contribute data to the knowledge base and contribute ShEx schemas to encode data models and support validation of entity data.

We focus on Shape Expressions (ShEx) schemas here deliberately: they are the explicit, human-authored abstraction layer that the Wikidata community *co-creates and consults*, which makes them the natural object of our human-centric analysis.

In May 2019, the Wikidata development team announced a new namespace in Wikidata, called EntitySchema, whose entries carry the prefix E (analogous to Q for items and P for properties) and store schemas encoded in the Shape Expressions (ShEx) language. The ShEx language is a formal language for data modeling and validation of RDF graphs [24, 93]. ShEx syntax adopts tokens from Turtle and SPARQL, and its semantics are inspired by XML Schema and RELAX NG [55]. ShEx has been successfully used for RDF validation in the FHIR community for exchanging medical records [91]. Wikidata editors encode data models as schemas, exemplifying community co-creation of practical graph abstractions. Other editors consult these schemas to understand how data is modeled by editors, thereby using these abstractions for clarity about sub-sections of the Wikidata graph. In addition to encoding data models, Wikidata editors use ShEx schemas to validate entity data from the knowledge base. Community members use Wikidata's schema namespace to publish schemas for validating data conformance [24, 67]. The successful use of ShEx schemas for building consensus around the alignment of biodiversity data with Wikidata is discussed in [100]. ShEx validation has shown promise in increasing data quality by addressing low-level modeling inconsistencies in Wikidata [78].

Researchers have also developed several different applications to extract schemas from Wikidata [23, 101]. These tools suggest a schema based on an analysis of how data for similar items have been modeled. Editors may choose to further refine the suggested schema or reuse it.

Considering the aspect of **usability**, the fact that several contributors have authored schemas after consulting documentation or example schemas suggests that ShEx authoring is usable enough for task completion within the community context. No systematic usability study has yet evaluated whether it is also efficient and low-error.

Considering the aspect of **teachability**, educators use Wikidata as a learning platform [90]. The Wikidata community has created resources for learning how to write schemas and how to use schemas to validate data. Researchers observed that members of the Wikidata community without previous experience with the SPARQL query language taught themselves SPARQL through interacting with Wikidata's SPARQL endpoint [63]. Members of the Wikidata community learn ShEx to author schemas. Community organizers have hosted interactive sessions where people who are already familiar with ShEx demonstrate how to use ShEx schemas within Wikidata, such as Data Quality Days 2021 [102], Data Quality Days 2022 [103], and Data Modeling Days 2023 [104]. Following these events, contributors who had not previously engaged with the schema namespace began editing existing schemas and creating new ones, demonstrating how the Wikidata community facilitates peer-to-peer learning and skill development.

While these developments demonstrate the dynamism of the Wikidata community, they also point to a barrier to entry for newcomers. The very evidence that ShEx schemas are learnable here also shows how that competence is acquired: through documentation, example schemas, onboarding events, mentorship, and tooling, rather than because the schemas are self-explanatory. Familiarity with the schema language and the community's modeling conventions thus remains a hurdle. Since no systematic usability study has yet been conducted, the success of schema adoption is better read as a sign of strong community support.

Due to the fact that members of the Wikidata community publish their schemas on-wiki using the CC0 license, researchers have access to the data in the knowledge base as well as the related human-defined schemas. This combination of openly licensed data and community-authored schemas provides a rare opportunity to empirically study how human-defined abstractions evolve in large-scale collaborative environments.

People from diverse backgrounds contribute data to Wikidata [53, 74, 81]. Editors range from highly active, technically skilled power editors to occasional contributors with limited technical background [85]. This blend of contributors makes Wikidata a useful community for investigating differences in user affinity toward existing graph abstractions. Wikidata is also a cross-domain knowledge base [46]. The fact that such a wide range of conceptual areas are represented in Wikidata allows researchers to examine how well existing graph abstractions support mainstream use cases across domains by comparing data with community-authored schemas.

While Wikidata illustrates that graph abstractions can be adopted and extended in large-scale collaborative settings, this success leverages sustained community coordination, self-directed learning, and the efforts of technically skilled contributors. Schema design, validation logic, and modeling conventions are not uniformly accessible to all participants, and abstraction complexity remains unevenly distributed across roles. These observations raise recurring questions about user affinity, roles, and accountability in collaborative graph systems.

Beyond direct contributors, Wikidata increasingly serves as a backend for AI-mediated systems, introducing a new category of *indirect users* who interact with graph abstractions not through query interfaces but through outputs generated by LLMs. This is relevant to human-centricity because indirect users inherit whatever biases, coverage gaps, or modeling limitations the graph abstraction imposes, without visibility into those constraints.

2.2 Graph Abstractions for Clinical and Biological Data

Clinical and biological domains provide a compelling context for studying the human-centric properties of graph abstractions due to their highly interconnected, heterogeneous, and evolving nature, as well as diverse user roles. The following use-cases highlight how graph abstractions support analytical tasks, while exposing recurrent challenges in usability and learnability.

2.2.1 MIMIC-III and Epidemiological Nosocomial Infections

Nosocomial infections represent a major challenge for epidemiological analysis, as their emergence and propagation depend on complex interactions between patients, hospital spaces, and time-dependent clinical events. Tracing potential transmission pathways requires reconstructing who was where, when, and under which clinical conditions, often across heterogeneous data sources

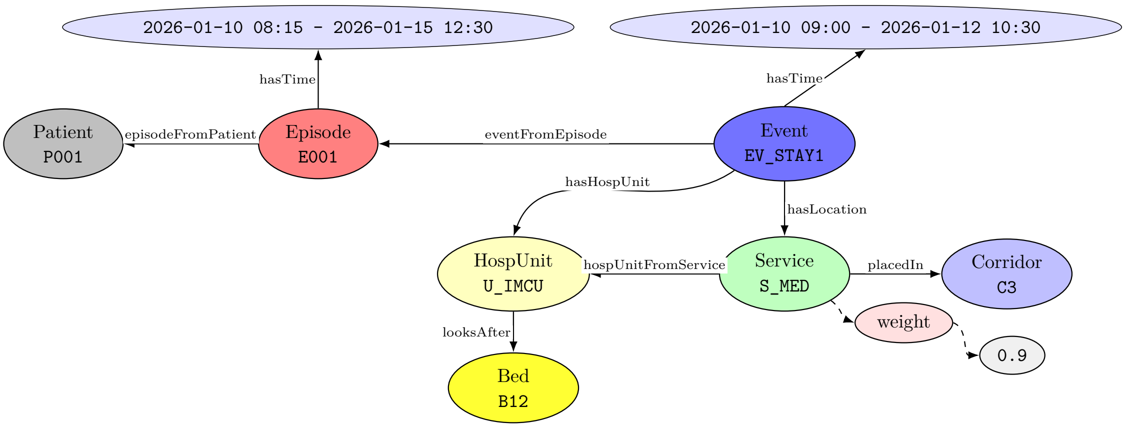


Figure 4 Instance graph (single patient, single episode, single event) illustrating the integration of temporal and spatial dimensions in the hospitalization model. Node colors distinguish different entity types (patient, episode, event, spatial locations, temporal literals, relation attributes, and numeric values). To enhance readability, the corresponding class-level representations are intentionally omitted, and only instance-level elements are depicted in this visualization.

and fine-grained spatial layouts. This setting demands expressive modeling approaches capable of jointly capturing temporal dynamics, spatial hierarchies, and clinical context, while remaining amenable to analysis, querying, and visualization.

Graph abstractions were employed to model and analyze patients' movements during their hospital stays in the context of spatio-temporal epidemiological research [76]. The use case captures the interplay between temporal evolution and hospital space to support the detection and analysis of nosocomial infection patterns. The model integrates two interconnected dimensions. The *spatial dimension* describes the hospital environment as a structured hierarchy of locations, ranging from beds to rooms, corridors, areas, floors, and buildings. In addition to hierarchical containment, spatial adjacency relations (e.g., *next to*, *opposite*) capture proximity between locations at the same level, enabling fine-grained reasoning about potential infection transmission paths. The spatial model further distinguishes between the *physical space* (locations) and the *logical space* (hospitalization units and clinical services responsible for patient care). The *temporal dimension* models each patient's hospitalization as a sequence of episodes, where each episode corresponds to a hospital stay composed of time-stamped events. Events represent movements (e.g., room transfers), medical procedures, or diagnostic outcomes such as microbiological test results. Each episode and event is associated with a start and end time, enabling the representation of both instantaneous and interval-based occurrences. Events linked to locations constitute the integration point between temporal and spatial dimensions. Figure 4 illustrates a fragment of this model as a directed edge-labeled graph at the instance level.

Various graph data abstractions, including RDF, property graphs, and RDF-star, were evaluated for clinical epidemiological tasks [76]. Data from MIMIC-III [52] encompassed inpatient movements, clinical tests, and epidemiological outcomes, including patient stays, ward transitions, services, and microbiological test results. To complete the domain model, synthetic data were generated to replicate hospital layouts, categorizing beds into surgical, medical, mixed, and neonatal units. The resulting graphs exhibited increasing event sets and varying structural densities while preserving consistent layouts and service configurations. Three epidemiological tasks were modeled: (i) detecting infection outbreaks, (ii) tracking outbreak propagation, and (iii) monitoring potentially infectious locations. All tasks were successfully supported across the considered graph abstractions.

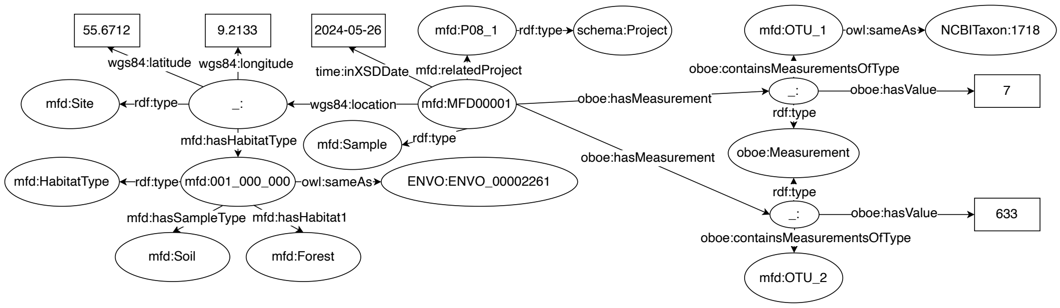


Figure 5 Microflora Danica RDF Graph.

Following the user taxonomy proposed by Li et al. [59], stakeholders were categorized into three groups: *knowledge builders* (experts in graph construction, data modeling, and query optimization), *analysts* (proficient in graph data models and visualization techniques), and *consumers* (infectious disease researchers and healthcare professionals). Knowledge builders required efficient query processing mechanisms for traversing graphs and computing adjacency, reachability, and shortest paths. Analysts needed expressive data models that support visualization of queries and results, while consumers primarily sought accurate outputs and interpretable visual representations for clinical decision-making.

From a **usability** perspective, recent work has reported [76] that Cypher queries over property graphs were more concise and easier to interpret visually than the equivalent SPARQL formulations over the RDF encodings. Consumers understood high-level graphical representations and the notion of traversing relationships to answer epidemiological questions; however, the underlying graph abstractions (schemas, encodings, and formal query formulations) remained inaccessible without technical mediation.

From a **teachability** perspective, modeling statements about relationships (particularly when edges required attributes or when n-ary relations were involved) has proven challenging. RDF's lack of native support for such constructs requires reification, i.e., transforming relationships into first-class nodes with associated properties. While cognitively demanding, reification enables expressive modeling of annotated relations. RDF-star provides a more direct mechanism for expressing statements about statements, but its nested syntax is still difficult for newcomers [7, 55].

From the perspective of **knowledge builders**, the evaluation revealed that graph management systems supporting different abstractions exhibited distinct memory management strategies and indexing mechanisms. Query complexity affected execution performance differently across implementations, highlighting that abstraction choices have both conceptual and measurable system-level implications.

2.2.2 Microflora Danica

Life sciences constitute an active research area in which large volumes of data are shared with the community. While some initiatives publish data natively as ontologies or RDF, such as Bio2RDF [29], the majority of biological datasets originate from empirical studies and are shared in tabular formats, such as spreadsheets or CSV files.

Microflora Danica integrates large-scale microbiological data with external environmental datasets [36]. The core dataset results from a nationwide field study in Denmark and includes DNA sequences and associated metadata such as location, time, and sampling processes. Identified microorganisms are linked to established ontologies and taxonomies, and the data is integrated with external datasets describing soil properties, such as EcoDes-DK15 [13].

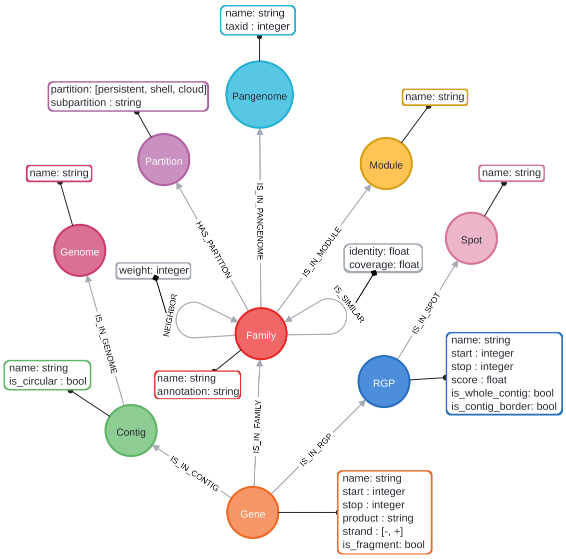


Figure 6 Graph schema structure of the pangenome dataset.

This use case illustrates a recurring human-centric friction rather than a **usability** success. Although the data is published as an RDF graph (Figure 5), most bioscience workflows, including modern machine-learning pipelines, remain optimized for tabular data. As a result, researchers outside computer science often struggle to adopt the graph abstraction directly: despite its integration benefits, the representation raises rather than lowers the barrier to entry for its intended scientific users.

From the perspective of **teachability**, the dataset offers opportunities for teaching SPARQL, spatio-temporal graph querying, and graph-based machine learning. The data supports OLAP-style analyses over graphs, such as identifying geographic regions with high microbial diversity. Although these opportunities are not yet fully exploited, the use case highlights the pedagogical potential of graph abstractions in the life sciences.

2.2.3 Pangenomic Graphs

In genomics, the rapid growth of sequencing projects has resulted in hundreds of thousands of available genomes. Comparative genomics increasingly relies on large-scale analyses of pangenomes to study gene presence, absence, and variation across species. Graphs provide a natural abstraction for capturing the complex relationships between multiple pangenomes and for analyzing evolutionary mechanisms, such as the transmission of antimicrobial resistance.

Recent work [12] has proposed representing multi-pangenomic data as property graphs. One such dataset, PanGraphDB, integrates data produced using the panRGP method and the PPanGGOLiN framework into a unified property graph stored in Neo4j, whose schema is given in Figure 6. This representation allows domain experts to efficiently express and execute complex genomic analyses in Cypher, including cross-species queries that were previously difficult.

Regarding **teachability**, this use case was the subject of a controlled student study [71]. The study aimed to assess how effectively students with varying backgrounds could discover the schema of a complex graph dataset and formulate graph queries of increasing complexity. After a six-hour session consisting of an intensive introduction to graph databases followed by hands-on tasks, participants were able to write basic queries. However, errors primarily occurred due to misunderstandings of semantic language constructs and difficulties in distinguishing between schema and data, highlighting challenges in abstraction comprehension rather than syntax.

3 Research Challenges

Across domains, graph abstractions succeed technically yet reveal recurring human-centric frictions. As Section 2 shows, these frictions are not isolated usability issues but structural consequences of how abstractions are formalized, taught, governed, and deployed. That is, graph data abstractions are socio-technical constructs shaped by institutional contexts, methodological choices, user roles, and application scenarios. In Wikidata (Section 2.1), abstraction design and validation are embedded in community governance and peer learning. In clinical epidemiology (Section 2.2.1), abstraction choices affect interpretability and role differentiation. In biological and pangenomic settings (Sections 2.2.2 and 2.2.3), abstraction expressiveness interacts with tooling ecosystems and learners' conceptual understanding. These tensions motivate the research challenges below.

Each challenge below follows a uniform *Why/What/How* structure: *Why* states the human-centric tension and which use case(s) from Section 2 motivate it; *What* surveys the relevant state of the art; and *How* outlines potential *research directions*. Concrete, prioritized research questions derived from these challenges are deferred to the roadmap in Section 4.

3.1 Institutional Challenges

Institutional challenges relate to educational structures, curriculum guidelines, incentive systems, and evaluation criteria that shape which research questions are prioritized and which skills are cultivated in future practitioners. We discuss these as follows.

IQ1: Incorporating human-centric studies into our curricula

Why. Advancing human-centric graph abstractions requires early and sustained exposure to human-centered thinking within educational pathways. However, current curricula, particularly in the context of the rapid expansion of AI and data-driven technologies, tend to emphasize technical proficiency, algorithmic performance, and system scalability, often at the expense of human factors such as usability, interpretability, and social impact. Without curricular redesign, future practitioners may lack the tools to reason about graph abstractions as socio-technical rather than purely technical artifacts.

What. To address this gap, human-centric methodologies must be systematically embedded into graph-related curricula across educational levels, from introductory courses to advanced graduate programs. This includes exposing students to user-centered design principles, qualitative and mixed-methods evaluation, and interdisciplinary perspectives that complement formal modeling and algorithmic techniques. Existing work in education and data abstraction has shown that challenge-based and visually grounded learning approaches can improve conceptual understanding and transferability across domains [5, 50, 62].

How. Implementing these changes requires coordinated curricular and institutional efforts. Courses and programs should be co-designed and, where possible, co-taught with experts from socio-humanistic disciplines, education research, and HCI to ensure meaningful integration rather than superficial coverage. Project-based and challenge-based learning should be adopted at multiple stages of training, allowing students to engage with graph abstractions through real-world, interdisciplinary problems. At the institutional level, sustained collaboration between data researchers, mathematics educators, and computational pedagogy experts is needed to develop reusable teaching materials, assessment strategies, and learning outcomes. Finally, articulating clear educational value propositions and business cases can help engage academic leadership and accreditation bodies, ensuring that human-centric competencies are supported.

IQ2: Institutional and disciplinary barriers to centering human perspectives

Why. Institutional and disciplinary structures can make it difficult to systematically integrate human perspectives into socio-technical research, including user-centric data management. These barriers arise from entrenched funding priorities, evaluation criteria, and disciplinary cultures, particularly in computer science, where scalability, performance, and formal correctness are traditionally emphasized, while user studies and qualitative methods receive comparatively less recognition. As a result, researchers adopting human-centered approaches may face limited visibility, fewer publication opportunities, and constrained career incentives.

What. Prior socio-technical discussions in the data management and computing communities have highlighted the lack of widely recognized venues and evaluation frameworks that explicitly value multidisciplinary contributions. Metrics such as ergonomics, learnability, interpretability, and user satisfaction are rarely treated as first-class research outcomes, despite their central role in the design and adoption of data systems. This imbalance reinforces disciplinary silos and discourages sustained engagement with human-centric research questions.

How. Addressing these barriers requires coordinated institutional efforts. These include broadening review and evaluation criteria to explicitly recognize human-centered contributions, fostering publication venues and tracks that welcome mixed qualitative–quantitative methodologies, and encouraging collaboration across data management, HCI, social sciences, and education. By aligning incentives with the goals of human-centric data management, institutions can better support research that integrates technical rigor with meaningful engagement with human needs.

3.2 Methodological Challenges

Methodological challenges address how graph abstractions are studied and evaluated. They highlight the need to borrow, adapt, and integrate methods from disciplines such as HCI, cognitive science, and visualization in order to better understand users' mental models, interaction strategies, and sources of error.

MQ1: Learning from how other disciplines use graph abstractions

Why. Disciplines such as biology, the social sciences, network analysis, and data science rely heavily on graph abstractions for problem solving and data representation [37, 83]. Studying how practitioners in these fields actually build, query, and reason over graphs reveals the cognitive demands graph abstractions place on real users, grounding design in established practice rather than the assumptions of a single community.

What. These communities have developed mature, domain-specific conventions for modeling and analyzing graph data [37]. Examining them clarifies which abstraction features support or hinder domain reasoning and which conventions recur across fields, providing an externally validated basis for graph-abstraction design.

How. Future work should systematically survey graph practices across data-intensive disciplines and distill the recurring modeling patterns, tool choices, and points of friction into transferable design guidance, identifying common human-facing requirements.

MQ2: Adapting empirical methods from HCI and cognitive science

Why. Understanding how users perceive, interpret, and struggle with graph abstractions requires empirical study, yet graph-data research rarely employs the methods that HCI and cognitive science have established for exactly this purpose; bringing cognitive-science methods into data management has been argued to be a productive and underused direction [17, 21, 108].

What. These fields offer a mature toolkit, e.g., surveys, structured and semi-structured interviews, Grounded Theory, sketching, think-aloud protocols, eye-tracking, and mixed-methods usability evaluation, for eliciting users' mental models and locating sources of friction. Bigelow et al. [21] and Williams et al. [108] use Grounded Theory and sketching to show how personal and professional context shapes the data abstractions users form; Li et al. [59] and Lissandrini et al. [61] document domain-specific usability and visualization challenges for knowledge-graph practitioners; mixed-methods usability evaluations have been applied to knowledge-graph–based systems [58]; eye-movement studies of visual versus textual representations [5] quantify differences in how users read each; and abstraction ability itself can be treated as a learnable, assessable competence rather than an assumed one [54].

How. Adapting these methods to graph abstractions means using surveys and interviews to surface mental models, expectations, and pain points; applying Grounded Theory to interaction traces to expose the social and cognitive factors behind usability problems; pairing think-aloud and eye-tracking with statistical analysis to evaluate behavior across user groups; and conducting mixed-methods usability evaluations of graph systems in the spirit of [58].

MQ3: Applying HCI interaction- and visualization-design principles

Why. Beyond methods for *studying* users, HCI and visualization research offer principles for *building* abstractions that match how people think. Iterative testing, support for exploration, attention to the diversity of users' mental models, and theory-grounded visual design can make graph abstractions and their interfaces more comprehensible and usable [19, 20, 62, 68, 109].

What. Evidence points to concrete design levers. Ma et al. [62] show that grounding the visual abstraction of property graph queries in established principles produces more effective and appealing representations. Chanthatrojwong et al. [32] extend this principle to property graph schemas, showing that composite, theory-grounded visual encodings can make schema features such as inheritance, constraints, and label expressions more comprehensible. Nouraei et al. [68] stress supporting different user types and balancing productivity with exploration. Williams et al. [108] show that divergent mental models call for multiple, customizable representations of the same data. Xie et al. [109] quantify the gap between users' and designers' mental models, establishing that it directly affects task performance.

How. These principles translate into adaptive, customizable visualizations, schema-visualization interfaces such as PASCAL [33], role- and goal-aware knowledge-graph exploration interfaces [96], and iterative, user-tested design cycles. Mental model elicitation techniques such as path diagrams [109] can align abstractions with users' expectations and narrow the user–designer gap.

3.3 Operational Challenges

Graph abstractions are designed, used, and maintained by a heterogeneous set of human and non-human actors. These challenges focus on identifying relevant user roles, understanding how their goals and expertise differ, and examining how transitions between roles affect abstraction needs, learning processes, and accountability.

OQ1: Which types of users rely on graph abstractions?

Why. As illustrated in Section 2, graph abstractions are used by heterogeneous actors. In Wikidata (Section 2.1), contributors, schema authors, tool developers, and data consumers interact with the same abstraction at different levels. In the MIMIC-III epidemiological case (Section 2.2.1), knowledge builders, analysts, and healthcare professionals engage with graph representations in distinct ways. In the pangenomic study (Section 2.2.3), students transition from learners to query writers. These examples show that graph abstractions are no longer confined to expert database designers but are appropriated by users with varying goals, expertise, and responsibilities.

What. Prior work distinguishes roles such as knowledge builders, analysts, and consumers, particularly in the context of knowledge graphs and visualization systems [59]. Similar role differentiation has been observed in collaborative knowledge bases such as Wikidata, where contributors, curators, tool developers, and data consumers interact with the same abstractions in different ways [63, 74, 98]. However, these roles are often treated implicitly, and most graph abstractions expose a uniform interface that assumes homogeneous expertise and goals.

How. Future research should develop explicit, empirically grounded taxonomies of user roles for graph abstractions, linking abstraction features to user goals, capabilities, and responsibilities. Such taxonomies should inform the design of role-aware abstractions, interfaces, and documentation that adapt to different user needs without fragmenting the underlying data model.

OQ2: How do transitions between roles shape abstraction needs over time?

Why. Users frequently transition between roles, for example, from learner to analyst, from analyst to schema designer, or from data consumer to contributor. These transitions often involve abrupt increases in abstraction complexity, which can hinder learning, reduce confidence, and lead to misuse or avoidance of graph-based tools. The controlled study on pangenomic graphs (Section 2.2.3) illustrates how learners struggle when transitioning from schema discovery to query formulation, highlighting the cognitive discontinuities that accompany role shifts.

What. Empirical studies in database education and graph querying show that novices struggle with abstraction boundaries, schema discovery, and the semantics of query constructs [71]. Similar challenges have been observed in knowledge graph exploration systems, where users report difficulty transitioning from exploratory interaction to precise analytical tasks [61]. Most systems expose the full complexity of abstraction from the outset, offering limited support for progressive learning.

How. Supporting role transitions requires graph abstractions and tools that enable the progressive disclosure of complexity. This includes scaffolded interaction modes, learning-oriented query assistance, and interfaces that evolve as users gain expertise. Longitudinal studies are needed to understand how users' mental models of graph abstractions change over time and how abstraction design can facilitate these transitions.

OQ3: How should responsibility and accountability be distributed across roles?

Why. Different user roles carry different levels of responsibility for data quality, modeling decisions, and downstream effects. For example, schema designers and constraint authors influence which data are considered valid, while analysts and automated systems rely on these decisions for interpretation and inference.

What. Current graph abstractions often obscure responsibility boundaries. Collaborative environments, such as Wikidata, show that modeling decisions are unevenly distributed among contributors, with limited visibility into who is responsible for specific abstraction choices [74, 85]. Similarly, constraint languages such as SHACL and ShEx introduce implicit assumptions that may not be apparent to downstream users [51, 55, 78].

How. Role-aware graph abstractions should make responsibility explicit by linking modeling and validation actions to provenance, feedback mechanisms, and governance processes. This includes supporting traceability from abstraction decisions to their effects on data quality and analysis outcomes, as well as clarifying accountability in collaborative and automated settings.

3.4 Applicative Challenges

Use case challenges examine whether the scenarios commonly used to motivate graph abstractions are representative of mainstream practice, and how evolving construction, refinement, and maintenance workflows, particularly those involving automation and AI, reshape human involvement.

AQ1: Do we need several types of mainstream graph construction scenarios?

Why. It is important to understand the landscape of existing graph construction methods and use cases to clarify the interplay between human effort and automation in these scenarios [49]. With the surge of LLMs, the problem arises regarding which parts of a graph can be constructed by pre-trained models and which need to be constructed by humans.

What. In the past, the construction of knowledge graphs (KG) relied heavily on manual effort and crowd-sourcing methods, such as in highly curated common-sense KGs, e.g., Wikidata [98] and Freebase [22]. As discussed in Section 2.1, Wikidata exemplifies a highly curated, community-driven construction paradigm, whereas the clinical and biological use cases in Section 2.2 illustrate integration-heavy, domain-specific construction processes. These contrasting scenarios suggest that multiple construction paradigms coexist and should be analyzed explicitly.

More recently, deep learning techniques in NLP have led to substantial progress in automatic KG construction, enabling tasks such as entity recognition, entity linking, coreference resolution, and relation extraction [111]. The emergence of pre-trained models, spanning from BERT [38] to the recent LLMs, has allowed us to study the problem of automating knowledge graph construction from large text-based corpora [72]. Knowledge acquisition from other sources, such as images and structured data, has also been considered in Retrieval Augmented Generation (RAG) models.

How. Combining automatic knowledge graph construction with manually curated approaches is still an open problem that requires the attention of researchers in the coming years. While automation in KG extraction is desirable to cope with large-scale data, the quality of these approaches and adherence to fairness, equity, and regulatable AI results should be checked [87].

AQ2: Which graph refinement and maintenance scenarios are most relevant?

Why. This question about the quality of graph-based use cases and their evolution is crucial, as graph data needs to be checked for quality and maintained upon changes, similarly to data in other formats. What is distinctive about graph data use cases is that the underlying relations are *first-class, explicit, and directly traversable edges*; cleaning and refinement operations therefore propagate along these edges across the graph rather than being mediated by value-based joins. The user also plays a central role in graph refinement and cleaning tasks, as fully automated approaches might not reflect what the user has in mind. The Wikidata case (Section 2.1) demonstrates community-driven validation and schema-based quality control, while the epidemiological and biological settings (Section 2.2) highlight domain-driven refinement and analytical correctness requirements. This suggests that refinement and maintenance tasks vary greatly across contexts.

What. Recent approaches, such as knowledge graph completion (or link prediction) [64], rule-based graph repairing [35], user-centric repairs [70], and graph data normalization techniques [86] have been proposed to address use cases in which graph data needs to be sanitized. All these approaches assume that rules are either formulated interactively with the help of users or learned from examples on the knowledge graph through Inductive Logic Programming (ILP) approaches. In both cases, the role of users is central to driving the graph completion and repairing tasks. Beyond repairing data, recent work also tests the reliability of graph database applications, for instance by fuzzing them with constraint-violating graphs [39].

How. Scenarios that aim to combine symbolic approaches based on graph sampling with interactive graph rule repairing approaches are needed in order to design use cases that reduce human effort in graph data curation. For instance, link prediction can be applied to obtain links that can repair a graph violation, as opposed to edge deletion and node/edge relabeling in current graph repairing use cases.

4 Research Roadmap

While the previous sections focus on overarching and cross-cutting challenges, this section refines them into concrete research questions organized by researchability, usability, teachability, and societal impact. These questions distinguish between fundamental investigations and application-driven concerns.

4.1 Researchability Challenges

Researchability challenges concern how graph abstractions can be formally defined, compared, and reasoned about by researchers. They address the tension between expressive power, theoretical tractability, interoperability, and accessibility across research communities.

Here the human subject is the researcher. An abstraction becomes a shared object of formal study only once it is specified precisely enough that communities with different backgrounds, e.g., database theory, knowledge representation, systems, can agree on what is being reasoned about and reproduce results over it. Expressiveness, tractability, and interoperability are thus not purely technical desiderata: a property established once over a commonly accepted formalization transfers across communities, whereas under-specified models oblige each group to re-derive its own foundations. Researchability, in this sense, concerns the researcher who studies and extends an abstraction rather than the end user who queries it.

4.1.1 Theory-driven questions

We discuss what makes a graph abstraction suitable for formal study, focusing on how expressiveness, conceptualization, interoperability, and extensibility affect theoretical research on graph data models and query languages.

RQ1: What influences the suitability of graph abstractions for formal study?

Why. Good graph abstractions allow us to systematically study different aspects of graph data models and query languages, such as complexity and expressiveness, compatibility with the standards, and interoperability and extensibility issues. Depending on our research question, we need to make the appropriate abstractions and understand the important features that we must include in our modeling. Graph abstractions are unifying paradigms that are also necessary for understanding how humans conceptualize graphs.

What. The space of graph abstractions is broad. At the data-model level, it forms a lattice over graph-shaped data (the so-called "abstraction soup"), spanning RDF and several property-graph variants [84]. At the query-language level, it comprises a similarly rich landscape of formalisms [7]. From a theoretical viewpoint, recent work has characterized the expressive power of the standardized graph query languages GQL and SQL/PGQ [44], introduced unified formalisms for property graphs schemas and constraints such as PG-Schema [8] and PG-Keys [9], designed declarative formalisms for graph pattern matching and transformation, such as the Graph Pattern Calculus (GPC) [41] and the transformation-oriented GPC+ formalism [27], and proposed a path-based algebra for property graphs [10].

How. Despite the above efforts to study graph abstractions suitable for formal studies, the latter mainly focused on theoretical models to delineate the expressive boundaries of graph data models and graph query languages. Further research aspects, such as the conceptualization of graph abstractions, standards interoperability, and their extensibility, need further attention. The focus should be on creating theoretically robust abstractions that are accessible to researchers from different communities.

4.1.2 Application-driven question

We discuss the practical limitations of existing graph abstractions, highlighting the scalability, performance, and expressiveness challenges that arise when these are deployed in large-scale, heterogeneous, and evolving applications.

RQ2: What are the practical limitations of existing graph abstractions?

Why. In large-scale, highly connected datasets and distributed computing environments, graph abstractions encounter scalability, storage, and communication bottlenecks. For example, large graphs, such as those representing social networks or web-scale data, require massive storage capacities while maintaining fast access and processing times. At the same time, distributed and GPU systems need extensive inter-node communication, which can become a bottleneck for highly connected graphs. In terms of expressiveness, graph abstractions often simplify relationships and structures to fit into node-edge models, while many real-world problems require richer semantics, such as multi-layer or heterogeneous graphs and streaming, dynamic, or temporal structures.

What. Beyond scalability and expressiveness, graph abstractions face practical limitations in evolution, validation, and cross-system portability in long-lived applications. Knowledge graphs evolve continuously [75], requiring constraint management and schema validation mechanisms such as SHACL and ShEx, whose complexity and adoption challenges are well documented [24, 51, 55, 78, 79, 93]. Collaborative ecosystems like Wikidata further show that schema design and quality control depend on sustained community coordination and tooling support [63, 98]. Recent work also explores local-first, replicated property graphs that can be edited collaboratively, with concurrent edits converging without central coordination [73]. Meanwhile, heterogeneity across RDF, property graphs, and emerging standards complicates interoperability and reproducibility [1, 60, 84]. Recently, approaches for cross-model graph transformations have been proposed, e.g., from relational to property graphs [94], from RDF to property graphs [77], and from property graphs to RDF [89]. However, more work is needed to cover all possible interoperability issues. These frictions directly impact analytical reliability, lifecycle management, and long-term governance in domains such as healthcare and life sciences [43, 83], so addressing RQ2 requires controlled evolution mechanisms, portable constraints, and abstraction-aware lifecycles.

How. Framed through our lens, these are not only performance questions: the practical limits of an abstraction govern whether researchers can study realistic, large-scale instances at all, and therefore which human-facing questions about graph data even become investigable. We need to maintain and enhance the research community's focus on developing efficient graph operations, such as traversals or centrality computations, and efficient mechanisms to partition or compress graphs without losing essential structural information. Graph databases can handle moderately complex queries, but they struggle with large, deep traversals or global queries (e.g., finding shortest paths in continent-scale graphs), and we need to address such problems. We need to manage irregular graph structures (e.g., power-law distributions in social networks) and the uneven workload distribution across computing nodes. Problems such as community detection, pattern finding, or graph neural network training on large graphs require substantial resources and new graph abstractions. For distributed and GPU computing, communication and load-balancing overhead deserve more focus. Future research should extend graph abstractions to heterogeneous, dynamic, and multi-layer structures while preserving tractable query evaluation and performance.

4.2 Usability Challenges

Usability challenges focus on how different categories of users interact with graph abstractions in practice. They examine cognitive load, task alignment, abstraction fit, and the limits of existing mechanisms such as views and interoperability layers.

4.2.1 Theory-driven questions

We explore how different categories of users perceive and appropriate graph abstractions, questioning assumptions on intuitiveness and analyzing whether mechanisms such as views suffice to mitigate cognitive load and abstraction complexity.

UQ1: User affinity of existing graph abstractions

Why. Graph abstractions are often perceived as intuitive and closely aligned with human ways of reasoning about relationships. However, their actual usability varies significantly across user groups. It remains insufficiently understood who benefits from graph abstractions, for which tasks, and under which conditions. In particular, non-technical domain experts may face substantial barriers that limit their ability to effectively engage with graph-based representations. Developing a clearer understanding of user affinity is essential to identify exclusion patterns and to ensure that graph abstractions do not inadvertently privilege certain users or forms of expertise over others. Such insights are also critical for tailoring abstractions to the distinct needs of graph data builders, analysts, consumers, and other user roles discussed later in this section.

What. Graph abstractions are well established in technical contexts, such as software engineering (e.g., UML) and conceptual data modeling (e.g., ER diagrams), where their usability is primarily assessed in terms of expressiveness, correctness, and formal properties. By contrast, broader cognitive, accessibility, and learnability aspects have received comparatively less attention. At the same time, graph abstractions are increasingly encountered by non-technical users as data becomes more widely available. This is visible in data-journalism platforms such as the ICIJ Data Explorer[1] and graph-centric investigative tools[2]. A recurring lesson in this setting is that raw graph visualizations (e.g., force-directed layouts) are hard to use, motivating systems that *hide*

[1] `https://offshoreleaks.icij.org/power-players/tony-blair`
[2] `https://team.inria.fr/cedar/connectionlens/`

the underlying graph behind global overviews and tabular exports while retaining graph-based integration [15]. Yet such research is largely qualitative and typically focuses on improving tools for specific, narrowly defined user groups. There is still a lack of systematic, cross-domain studies examining how different populations, particularly users without formal training in data systems, perceive, interpret, and appropriate graph abstractions.

How. Addressing these gaps requires combining qualitative and quantitative research methods. In-depth qualitative studies can uncover users' mental models, expectations, and sources of friction when interacting with graph abstractions. At the same time, large-scale surveys are needed to map user demographics, expertise levels, goals, and usage patterns across technical and non-technical communities. Such studies should explicitly investigate factors influencing engagement or disengagement, including perceived complexity, cognitive alignment, task suitability, and tool support. Accessibility concerns must also be considered, including visualization challenges for differently abled users and language barriers for non-English speakers. Collecting feedback from underserved or excluded groups is essential for informing the redesign of graph abstractions. Finally, measuring cognitive load differences across user groups[3] and analyzing learning curves and long-term usage patterns can provide actionable guidance for designing graph abstractions that better align with diverse mental models and workflows.

UQ2: Limitations of the view-based approach for usability

Why. The view-based approach simplifies the representation of complex graph data by focusing only on specific portions that are relevant for a given context. This helps reduce the cognitive load [17] of users, making the data more accessible, especially to non-experts who need to explore or interact with large-scale, densely interconnected, and highly annotated graph data.

What. The view-based approach creates dynamic perspectives over the input graph that correspond to particular tasks or user needs [25, 27, 47]. These are often implemented through queries that can be further refined, depending on the functionalities provided by each system. While such methods simplify user interaction, they have several limitations. First, the support for view-based mechanisms is limited and fragmented. Second, given the evolution of the underlying graph data, one needs to ensure that the views remain consistent and that modifications to the views are reflected in the initial graph.

How. First, to overcome the fragmented support for view-based mechanisms, it is essential to develop standardized methods and tools for view creation, maintenance, and refinement, leveraging existing standard graph query languages such as GQL and SQL/PGQ. It is important that these are designed in an intuitive, user-friendly manner, allowing users to engage with the graph dynamically in ways that minimize complexity. For example, visual graph view builders or drag-and-drop interfaces can simplify their formulation, making view generation accessible to non-experts [18].

Second, to ensure that the input graph data and the views defined over it remain consistent with one another upon modifications, existing techniques for view maintenance [45] and updates [16] can be adapted from the relational setting. Furthermore, providing users with tools to visualize the effects of view changes, as well as integrating their feedback, can help create more reliable abstractions. Interactive dashboards can help visualize the effects of updates on both the graph and its views, enabling users to understand and validate changes in real time. Integrating user feedback loops ensures that the generated views remain aligned with their evolving needs. Also, adaptive guidance mechanisms, such as recommendations for relevant subgraphs or query templates based on user behavior and historical interactions, can empower users to refine views effectively.

[3] `https://en.wikipedia.org/wiki/Cognitive_load#Measurement`

UQ3: Suitability of existing graph abstractions for mainstream use cases

Why. Graph abstractions [84] are foundational for representing and processing interconnected data across various domains. Their suitability for specific scenarios directly determines their effectiveness in addressing real-world challenges. These scenarios span a wide spectrum, including data augmentation, integration, semantic reasoning, recommendation systems, and advanced machine learning applications such as link prediction and graph representation learning.

What. Different types of graph abstractions are needed to effectively address the requirements of these popular tasks. For data augmentation, abstractions that support dynamic and extensible schemas, such as RDF and property graphs, are needed, as they enable the incorporation of diverse data sources and the enrichment of the graph with new relationships and attributes. For data integration, multi-modal graphs that combine heterogeneous data types and support schema alignment are essential, as they facilitate merging datasets into a unified representation. In semantic reasoning, RDF graphs support inference tasks, leveraging ontologies that capture domain-specific semantics. For recommendation systems, multi-relational graph abstractions, such as edge-labeled and property graphs, are well-suited, as they capture user-item interactions, contextual metadata, and dynamic preferences that drive recommendation algorithms. Finally, advanced machine learning applications, such as link prediction and graph representation learning, require abstractions that can efficiently model high-dimensional attributes, dynamic graphs, and hierarchical structures. Abstractions like attributed property graphs or hypergraphs are particularly effective in these scenarios, as they facilitate encoding rich structural and semantic information, enabling the generation of expressive embeddings and robust predictive models.

How. To ensure that graph abstractions are well suited for the scenarios in which they are deployed, user-centric approaches should be developed. For example, users can be empowered to combine features from different graph abstractions depending on their specific application requirements. Tools and interfaces should prioritize accessibility for non-expert users through features such as interactive schema creation, guided workflows, and embedded recommendations tailored to common use cases. Task-specific optimizations, such as native support for efficient embeddings in machine learning applications or mechanisms for handling dynamic edge weights in recommendation systems, are crucial for performance. Finally, user feedback loops can help refine abstractions dynamically, allowing systems to adapt to evolving patterns and needs.

4.2.2 Application-driven questions

We focus on how well graph abstractions support real-world analytical workflows, examining how they handle iterative analysis, workflow composition, interoperability, and long-term usability.

UQ4: How do graph abstractions support real-world analytical workflows?

Why. In practice, users rarely interact with graph abstractions through isolated queries or single operations. Instead, graph-based analysis is embedded in broader analytical workflows that involve data exploration, hypothesis refinement, iterative querying, validation of intermediate results, and collaboration among multiple users. If graph abstractions fail to support these end-to-end workflows, users may experience friction, increased cognitive load, or be forced to rely on ad-hoc workarounds that undermine both usability and correctness.

What. Most usability studies of graph systems focus on localized interactions, such as the expressiveness of a query language, the correctness of query results, or the efficiency of individual tasks. While valuable, these evaluations often abstract away from how users combine multiple interactions over time to achieve higher-level analytical goals. Hence, important workflow-related aspects, such as how users refine queries incrementally, manage intermediate results, recover from errors, or coordinate analysis across tools and collaborators, remain underexplored. Furthermore, current graph abstractions typically provide limited explicit support for workflow structuring, provenance tracking, or the reuse and adaptation of analysis fragments across tasks and users.

How. Addressing this challenge requires moving beyond interaction-level evaluation toward workflow-oriented and longitudinal studies of graph usage. These should examine how different abstractions support iterative exploration, sense-making, and decision-making over extended periods, as well as how they facilitate error diagnosis, correction, and rollback. From a system design perspective, graph abstractions should expose mechanisms for structuring analytical workflows, such as composable queries, reusable views, and explicit representations of intermediate results and dependencies. Support for collaboration, including shared abstractions, annotation, and provenance-aware analysis, is also essential. By aligning graph abstractions more closely with real-world analytical workflows, systems can improve their usability and reliability.

UQ5: Limitations of existing graph abstractions for usability

Why. Existing graph abstractions tend to lock users and applications into a specific ecosystem centered around those abstractions. This is often determined by the particular use case and the needs regarding how the data will be used. For example, RDF graphs are designed to work well in scenarios where ontologies are available and are thus well suited for reasoning. Property graphs, however, tend to support network analyses and connectivity queries more efficiently.

What. While use cases often determine which graph abstraction is better suited, the same graph might be used in different use cases in different ways. Hence, there has been some work on interoperability, e.g., by transforming one abstraction into another or developing a data model that serves as multiple graph abstractions that can be used to access the data [56, 99]. However, this so far only considers translation between query languages and data models.

How. Since existing graph abstractions differ in their limitations and strengths, we need to find a way to combine the strengths of the existing ones to overcome their limitations. This can either lead to the development of a new abstraction that encompasses existing ones or to the development of a framework that achieves full interoperability at all levels, including query languages, schema languages, data models, etc.

4.3 Education/Teachability Challenges

Teachability challenges address how graph abstractions are learned and taught, and how their cognitive demands compare to other abstractions. They emphasize the need for strategies that explicitly support abstraction formation, mental model alignment, and progressive learning.

4.3.1 Theory-driven questions

We discuss the pedagogical aspects and the fundamental cognitive difficulties learners face when reasoning about graph-based representations.

TQ1: Criteria for selecting a suitable graph abstraction for teaching

Why. Abstractions are central to computational thinking [3, 54, 88], alongside decomposition, design, debugging, iteration, and generalization [88], as they allow one to ignore non-essential features, extract similarities between concepts, and generate further knowledge [31].

Given their educational role, it is crucial to choose appropriate graph abstractions to enhance learner understanding and make materials comprehensible, relatable, and applicable to real-world problems.

What. The criteria should emphasize clarity, context, visual intuition, and computational relevance. A suitable abstraction presents graph objects and the additional features and/or constraints attached to these in ways that students can quickly comprehend and manipulate, ideally with real-world examples that underscore practical applications.

How. To achieve this, instructors can prioritize abstractions that offer clear visualizations and gradually introduce more advanced topics to ensure that learners build a strong conceptual foundation before exploring more complex concepts. Practical exercises, real-world analogies, and intuitive visual aids are key strategies that foster engagement, reinforce comprehension, and make abstract graph concepts more tangible and meaningful.

TQ2: Inherent difficulties in learning and understanding graph abstractions

Why. Unlike tabular or hierarchical models, which align with natural cognitive tendencies for sequential or categorical reasoning, graph-based representations emphasize interdependencies and structural flexibility. This shift demands a different cognitive approach, making graphs less intuitive to grasp [48, 65]. Additionally, the diversity of mental models among learners further complicates comprehension, requiring tailored instructional methods [14].

What. The challenge stems from the way graph abstractions prioritize relationships over individual data points, requiring learners to understand connectivity rather than isolated entities. Research shows that humans often struggle with networked representations due to their reliance on linear or hierarchical thinking [48, 65, 109]. Additionally, as Williams et al. [108] highlight, learners approach graph-based models with varying conceptual frameworks, making it difficult to standardize effective teaching strategies.

How. To improve learning, educators should employ progressive abstraction, beginning with concrete, real-world graph representations before transitioning to abstract models [48]. Utilizing visualization-driven learning, such as interactive exploration and multiple representation formats, can accommodate diverse cognitive styles [14]. Additionally, integrating path diagrams and mental model elicitation techniques [109] helps align instructional methods with how learners conceptualize graphs. Project-based and challenge-based learning approaches provide hands-on engagement, reinforcing practical understanding and making abstract graph reasoning more accessible.

4.3.2 Application-driven questions

We analyze the limitations of existing methods for teaching graph abstractions, focusing on mismatches between pedagogical practices, learners' mental models, and the cognitive demands imposed by graph abstractions in educational settings.

TQ3: Limitations of current methods for teaching graph abstractions

Why. Teaching graph abstractions is essential for computational thinking, as they model complex systems and relationships. However, current methods often fail to address the cognitive complexity of graph abstractions and are further limited by a weak focus on user-centered approaches and inadequate support for learners' diverse mental models [48, 65]. Moreover, Bach et al. [14] emphasize that visualization education must cater to a diverse audience while addressing methodological and contextual challenges. Additionally, Hazzan [48] highlights that teaching abstraction requires overcoming uncertainty and cognitive complexity, especially for concepts like graphs, which demand conscious navigation across abstraction levels.

What. These limitations stem from a lack of systematic frameworks for teaching graph abstractions and from instructional methods poorly aligned with learners' cognitive processes. Moore et al. [65] extend Piaget's theories of abstraction to show how instructional materials shape the development of stable knowledge structures, while Bach et al. [14] identify challenges such as fostering visual representation skills, addressing learner diversity, and assessing project-based learning. This misalignment between instructors' and learners' mental models significantly impacts comprehension and performance [108, 109].

How. Addressing these limitations requires integrating human-centric, evidence-based strategies for teaching graph abstractions. Educators must adopt methods such as project-based learning and challenge-based learning, as suggested by Bach et al. [14], to foster engagement and contextual understanding. Hazzan [48] advocates for raising awareness of abstraction levels and teaching learners to navigate these levels consciously to mitigate cognitive complexity. Drawing on Moore et al. [65], instructional materials should be designed to support abstraction processes, focusing on the gradual development of knowledge structures. Incorporating tools like mental model elicitation, such as path diagrams [109], and iterative user testing [108] can align teaching methods with learners' understanding. These approaches help educators bridge abstract graph concepts and learner comprehension, making graph abstractions more accessible.

4.4 Societal Challenges

Societal challenges consider the broader consequences of graph abstractions beyond direct users. They include fairness, accountability, transparency, and the indirect impact of graph-based systems on individuals and communities affected by their deployment.

4.4.1 Theory-driven questions

We examine how graph abstraction choices shape fairness, bias, and transparency, analyzing how modeling decisions influence downstream reasoning and affect stakeholders who are not direct users of graph-centric data systems.

SQ1: Fairness and bias induced by graph abstractions

Why. Graph abstractions play a central role in shaping how data is structured, connected, and interpreted. Choices about which entities, relationships, attributes, and constraints are modeled influence downstream reasoning, analytics, and automated decisions. When graph abstractions are used in sensitive domains or as part of AI pipelines, such design choices can amplify existing biases or introduce new ones, with consequences that extend beyond technical users to affected individuals and groups.

What. Bias may be introduced at multiple stages of abstraction design, including schema modeling, data integration, constraint definition, and inference mechanisms. For example, large shared knowledge graphs such as Wikidata or Freebase embed collective modeling decisions that reflect the perspectives, priorities, and coverage of their contributing communities [22, 74, 98]. Similarly, choices in validation languages and constraints (e.g., SHACL or ShEx) can privilege certain data patterns while marginalizing others [55]. Despite their impact, these biases are often opaque to end users and difficult to trace back to specific abstraction decisions.

How. Addressing fairness in graph abstractions requires systematic methods for identifying how abstraction choices propagate bias across graph-based workflows. This includes developing auditing techniques that relate schema and constraint design to observed disparities in query results or automated outcomes, as well as supporting comparative analyses across alternative abstractions. Human-centric approaches should also expose abstraction assumptions to users, enabling critical revisions of modeling decisions, particularly in high-stakes applications.

SQ2: Transparency and interpretability for affected parties

Why. Graph-based systems increasingly influence decisions that affect people who are not the direct users of graph abstractions, such as patients, citizens, or content consumers. In these settings, the lack of transparency regarding how graph abstractions shape results can undermine trust, accountability, and the ability to contest or understand outcomes.

What. Existing graph abstractions and query languages are primarily designed to be interpretable by technical experts, emphasizing formal semantics and expressive power [7]. However, non-expert stakeholders often lack access to meaningful explanations of how data is structured, how inferences are derived, and how uncertainty and incompleteness are handled. This gap is particularly evident when graph abstractions are combined with machine learning or language models, where abstraction layers and automated reasoning further obscure causal links between data and outcomes [72, 87].

How. Human-centric graph abstractions should support explanation mechanisms that translate technical representations into accessible accounts of modeling assumptions, limitations, and uncertainty. This may include abstraction-aware explanations of query results, provenance tracking across abstraction layers, and role-specific views that communicate relevant information to affected parties without requiring technical expertise. Such mechanisms are essential for enabling informed oversight and meaningful engagement with graph-based systems.

4.4.2 Application-driven questions

We overview societal and governance challenges associated with graph abstractions, identifying their role in automated decision making and in managing shared abstractions within collaborative and institutional settings.

SQ3: Societal impact of graph abstractions in automated decision systems

Why. Graph abstractions are increasingly embedded in automated decision-making pipelines, including those involving machine learning, recommender systems, and large language models. In these contexts, abstraction choices can influence not only system performance but also real-world outcomes: access to resources, prioritization of information, and clinical and policy decisions.

What. While there is growing awareness of the societal risks associated with automated systems, the specific role played by graph abstractions within these pipelines remains poorly understood. Studies often focus on algorithmic fairness or model behavior in isolation, overlooking how abstraction design, e.g., schema structure, integration of heterogeneous data, or constraint enforcement, shapes the inputs and outputs of automated processes. Case studies in domains such as healthcare and epidemiology illustrate both the potential and the risks of graph-based decision support when abstraction choices are insufficiently scrutinized [52, 76].

How. Application-level research should examine the societal impact of graph abstractions through empirical studies of real deployments. This includes analyzing how different abstraction choices affect decision outcomes, error propagation, and disparate impacts across populations. Such studies should be complemented by design guidelines that help practitioners assess the societal implications of abstraction choices when integrating graphs into automated systems.

SQ4: Governance of shared graph abstractions

Why. Many large-scale graph systems rely on shared abstractions maintained by communities or institutions, such as collaboratively curated knowledge graphs or domain-wide schemas. The long-term sustainability and societal trustworthiness of these systems depend on how changes to abstractions are governed, negotiated, and communicated.

What. Existing governance mechanisms for shared graph abstractions are often informal, uneven, or implicit. Studies of collaborative platforms such as Wikidata show that a relatively small subset of contributors exerts disproportionate influence over modeling decisions, schema evolution, and editorial norms [74, 85]. While this enables scalability, it also raises concerns about representation, accountability, and inclusiveness, particularly as these abstractions are reused across domains and applications.

How. Designing graph abstractions with explicit governance and stewardship mechanisms is essential for responsible and sustainable use. This includes support for versioning, documented design rationales, community feedback, and mechanisms to assess the impact of abstraction changes on downstream users and applications. Embedding governance considerations into abstraction design can help align technical evolution with societal values and long-term accountability.

5 Conclusion

We have examined the human-centricity of graph abstractions, analyzing several current paradigms through use cases ranging from collaborative knowledge graphs to clinical and biological applications, and we have distilled the research challenges they raise from a data management perspective.

Making graph abstractions more user-centric opens several directions for future work. The most immediate is empirical: targeted user studies, each focused on a single dataset and processing paradigm, should assess how well current graph paradigms serve particular applications and, in aggregate, characterize what users expect from a given data or query model. A second concerns expressiveness, which we argue should be measured jointly with users' perception of that expressive power. The analysis of Wikidata query logs [26] is illustrative in this respect: queries are posed by both humans and bots, and although recursive operators are common, the regular path queries that actually appear are structurally simple. Extending such analyses beyond RDF would help close the gap between formal power and practical use. A final direction is to study machine-centric abstractions alongside human-centric ones, including how well generative models reason over graph models beyond simple ones [40].

We believe the next generation of graph abstractions must treat expressiveness and human accessibility equally as first-class design goals.

References

1 Shqiponja Ahmetaj, Iovka Boneva, Jan Hidders, Katja Hose, Maxime Jakubowski, José Emilio Labra Gayo, Wim Martens, Fabio Mogavero, Filip Murlak, Cem Okulmus, Axel Polleres, Ognjen Savkovic, Mantas Simkus, and Dominik Tomaszuk. Common foundations for SHACL, ShEx, and PG-Schema. In *WWW*, pages 8–21. ACM, 2025. doi:10.1145/3696410.3714694.

2 Shqiponja Ahmetaj, Slawomir Staworko, Jan Van den Bussche, and Maxime Jakubowski. Shapes in graph data: Theory and implementation (Dagstuhl Seminar 24102). *Dagstuhl Reports*, 14(3):9–30, 2024. doi:10.4230/DAGREP.14.3.9.

3 Alfred V. Aho and Jeffrey D. Ullman. Abstractions, their algorithms, and their compilers. *Commun. ACM*, 65(2):76–91, 2022. doi:10.1145/3490685.

4 Sihem Amer-Yahia, Jasmina Bogojeska, Roberta Facchinetti, Valeria Franceschi, Aristides Gionis, Katja Hose, Georgia Koutrika, Roger D. Kouyos, Matteo Lissandrini, Silviu Maniu, Katsiaryna Mirylenka, Davide Mottin, Themis Palpanas, Mattia Rigotti, and Yannis Velegrakis. Towards reliable conversational data analytics. In *EDBT*, pages 962–969. OpenProceedings.org, 2025. doi:10.48786/EDBT.2025.78.

5 Magdalena Andrzejewska and Anna Stolinska. Do structured flowcharts outperform pseudocode? evidence from eye movements. *IEEE Access*, 10:132965–132975, 2022. doi:10.1109/ACCESS.2022.3230981.

6 Renzo Angles. The property graph database model. In *AMW*, CEUR Workshop Proceedings. CEUR-WS.org, 2018. URL: https://ceur-ws.org/Vol-2100/paper26.pdf.

7 Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan L. Reutter, and Domagoj Vrgoc. Foundations of modern query languages for graph databases. *ACM Comput. Surv.*, 50(5):68:1–68:40, 2017. doi:10.1145/3104031.

8 Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Stefan Plantikow, Ognjen Savkovic, Michael Schmidt, Juan Sequeda, Slawek Staworko, Dominik Tomaszuk, Hannes Voigt, Domagoj Vrgoc, Mingxi Wu, and Dusan Zivkovic. Pg-schema: Schemas for property graphs. *Proc. ACM Manag. Data*, 1(2):198:1–198:25, 2023. doi:10.1145/3589778.

9 Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Keith W. Hare, Jan Hidders, Victor E. Lee, Bei Li, Leonid Libkin, Wim Martens, Filip Murlak, Josh Perryman, Ognjen Savkovic, Michael Schmidt, Juan F. Sequeda, Slawek Staworko, and Dominik Tomaszuk. Pg-keys: Keys for property graphs. In *SIGMOD Conference*, pages 2423–2436. ACM, 2021. doi:10.1145/3448016.3457561.

10 Renzo Angles, Angela Bonifati, Roberto García, and Domagoj Vrgoc. Path-based algebraic foundations of graph query languages. In *EDBT*, pages 783–795. OpenProceedings.org, 2025. doi:10.48786/EDBT.2025.63.

11 Marcelo Arenas, Enrico Franconi, Janik Hammerer, Olaf Hartig, Katja Hose, Laura Koesten, George Konstantinidis, Leonid Libkin, Wim Martens, Yuya Sasaki, Stefanie Scherzinger, Katherine Thornton, and Hsiang-Yun Wu. Exploring exploratory querying. *Proc. VLDB Endow.*, 18(13):5731–5739, 2025. URL: https://www.vldb.org/pvldb/vol18/p5731-konstantinidis.pdf.

12 Jérôme Arnoux, Angela Bonifati, Alexandra Calteau, Stefania Dumbrava, and Guillaume Gautreau. Integrating complex pangenome graphs. In *ICDEW*, pages 350–354. IEEE, 2024. doi:10.1109/ICDEW61823.2024.00052.

13 Jakob J Assmann, Jesper E Moeslund, Urs A Treier, and Signe Normand. EcoDes-DK15: High-resolution ecological descriptors of vegetation and terrain derived from Denmark's national airborne laser scanning data set. *Earth System Science Data*, 2022.

14 Benjamin Bach, Mandy Keck, Fateme Rajabiyazdi, Tatiana Losev, Isabel Meirelles, Jason Dykes, Robert S. Laramee, Mashael AlKadi, Christina Stoiber, Samuel Huron, Charles Perin, Luiz Morais, Wolfgang Aigner, Doris Kosminsky, Magdalena Boucher, Søren Knudsen, Areti Manataki, Jan Aerts, Uta Hinrichs, Jonathan C. Roberts, and Sheelagh Carpendale. Challenges and opportunities in data visualization education: A call to action. *IEEE Trans. Vis. Comput. Graph.*, 30(1):649–660, 2024. doi:10.1109/TVCG.2023.3327378.

15 Oana Balalau, Nelly Barret, Simon Ebel, Théo Galizzi, Ioana Manolescu, and Madhulika Mohanty. Graph lenses over any data: the ConnectionLens experience. In *ICDEW*, pages 370–374. IEEE, 2024. doi:10.1109/ICDEW61823.2024.00056.

16 François Bancilhon and Nicolas Spyratos. Update semantics of relational views. *ACM Trans. Database Syst.*, 6(4):557–575, 1981. doi:10.1145/319628.319634.

17 Sourav S. Bhowmick, S. H. Annabel Chen, and Divesh Srivastava. Cognitive psychology meets data management: State of the art and future directions. In *SIGMOD Conference Companion*, pages 590–596. ACM, 2024. doi:10.1145/3626246.3654682.

18 Sourav S. Bhowmick and Byron Choi. Data-driven visual query interfaces for graphs: Past, present, and (near) future. In *SIGMOD Conference*, pages 2441–2447. ACM, 2022. doi:10.1145/3514221.3522562.

19 Sourav S. Bhowmick, Byron Choi, and Chengkai Li. Graph querying meets HCI: state of the art and future directions. In *SIGMOD Conference*, pages 1731–1736. ACM, 2017. doi:10.1145/3035918.3054774.

20 Sourav S. Bhowmick, Byron Choi, and Chengkai Li. *Human Interaction with Graphs: A Visual Querying Perspective*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2018. doi:10.2200/S00855ED1V01Y201805DTM047.

21 Alex Bigelow, Katy Williams, and Katherine E. Isaacs. Guidelines for pursuing and revealing data abstractions. *IEEE Trans. Vis. Comput. Graph.*, 27(2):1503–1513, 2021. doi:10.1109/TVCG.2020.3030355.

22 Kurt Bollacker, Robert P. Cook, and Patrick Tufts. Freebase: A shared database of structured general human knowledge. In *AAAI*, pages 1962–1963. AAAI Press, 2007. URL: http://www.aaai.org/Library/AAAI/2007/aaai07-355.php.

23 Iovka Boneva, Jérémie Dusart, Daniel Fernández-Álvarez, and José Emilio Labra Gayo. Shape designer for ShEx and SHACL constraints. In *ISWC (Satellites)*, CEUR Workshop Proceedings, pages 269–272. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2456/paper70.pdf.

24 Iovka Boneva, José Emilio Labra Gayo, Samuel Hym, Eric G. Prud'hommeaux, Harold R. Solbrig, and Slawek Staworko. Validating RDF with shape expressions. *CoRR*, abs/1404.1270, 2014. arXiv:1404.1270.

25 Angela Bonifati, Stefania Dumbrava, and Emilio Jesús Gallego Arias. Certified Graph View Maintenance with Regular Datalog. *Theory Pract. Log. Program.*, 18(3-4):372–389, 2018. doi:10.1017/S1471068418000224.

26 Angela Bonifati, Wim Martens, and Thomas Timm. Navigating the Maze of Wikidata Query Logs. In *WWW*, pages 127–138. ACM, 2019. doi:10.1145/3308558.3313472.

27 Angela Bonifati, Filip Murlak, and Yann Ramusat. Transforming property graphs. *Proc. VLDB Endow.*, 17(11):2906–2918, 2024. doi:10.14778/3681954.3681972.

28 Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible Markup Language (XML) 1.0, W3C. *http://www.w3.org/TR/REC-xml/*, 1998.

29 Alison Callahan, Jose Cruz-Toledo, Peter Ansell, and Michel Dumontier. Bio2RDF release 2: Improved coverage, interoperability and provenance of life science linked data. In *ESWC*, LNCS, pages 200–212. Springer, 2013. doi:10.1007/978-3-642-38288-8_14.

30 Donald D. Chamberlin and Raymond F. Boyce. SEQUEL: A structured english query language. In *SIGMOD Workshop, Vol. 1*, pages 249–264. ACM, 1974. doi:10.1145/800296.811515.

31 John H. Chambers. The difference between the abstract concepts of science and the general concepts of empirical educational research. *Journal of Educational Thought / Revue de la Pensée Educative*, 2018. URL: https://api.semanticscholar.org/CorpusID:151011055.

32 Kasidis Chanthatrojwong, Sourav S. Bhowmick, and Byron Choi. LICS: towards theory-informed effective visual abstraction of property graph schemas. *Proc. ACM Manag. Data*, 3(3):180:1–180:26, 2025. doi:10.1145/3725410.

33 Kasidis Chanthatrojwong, Sourav S. Bhowmick, and Byron Choi. PASCAL: A theory-informed visual interface for property graph schema visualization. In *SIGMOD Conference Companion*, pages 67–70. ACM, 2025. doi:10.1145/3722212.3725123.

34 Peter P. Chen. The entity-relationship model - toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36, 1976. doi:10.1145/320434.320440.

35 Yurong Cheng, Lei Chen, Ye Yuan, and Guoren Wang. Rule-Based Graph Repairing: Semantic and Efficient Repairing Methods. In *ICDE*, pages 773–784. IEEE Computer Society, 2018. doi:10.1109/ICDE.2018.00075.

36 Mads Corfixen, Thomas Heede, Tomer Sagi, Mads Albertsen, Thomas D. Nielsen, and Katja Hose. Integrating Multi-Modal Spatial Data using Knowledge Graphs - a Case Study of Microflora Danica. In *SeWeBMeDA@ESWC*, CEUR Workshop Proceedings. CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3726/paper1.pdf.

37 Resul Das and Mucahit Soylu. A key review on graph data science: The power of graphs in scientific studies. *Chemometrics and Intelligent Laboratory Systems*, 240:104896, September 2023. doi:10.1016/j.chemolab.2023.104896.

38 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi:10.18653/V1/N19-1423.

39 Stefania Dumbrava, Melchior W. M. Oudemans, and Burcu Kulahcioglu Ozkan. Fuzzing graph database applications with graph transformations. In *ICGT*, volume 15720 of *Lecture Notes in Computer Science*, pages 135–156. Springer, 2025. doi:10.1007/978-3-031-94706-3_7.

40 Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a Graph: Encoding Graphs for Large Language Models. In *ICLR*. OpenReview.net, 2024. URL: https://openreview.net/forum?id=IuXR1CCrSi.

41 Nadime Francis, Amélie Gheerbrant, Paolo Guagliardo, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Liat Peterfreund, Alexandra Rogova, and Domagoj Vrgoc. GPC: A pattern calculus for property graphs. In *PODS*, pages 241–250. ACM, 2023. doi:10.1145/3584372.3588662.

42 Wolfgang Gatterbauer and Cody Dunne. Relational diagrams and the pattern expressiveness of relational languages. *SIGMOD Rec.*, 54(1):80–88, April 2025. doi:10.1145/3733620.3733637.

43 Sandra Geisler, Cinzia Cappiello, Irene Celino, David Chaves-Fraga, Anastasia Dimou, Ana Iglesias-Molina, Maurizio Lenzerini, Anisa Rula, Dylan Van Assche, Sascha Welten, and Maria-Esther Vidal. From genesis to maturity: Managing knowledge graph ecosystems through life cycles. *Proc. VLDB Endow.*, 18(5):1390–1397, 2025. doi:10.14778/3718057.3718067.

44 Amélie Gheerbrant, Leonid Libkin, Liat Peterfreund, and Alexandra Rogova. GQL and SQL/PGQ: theoretical models and expressive power. *Proc. VLDB Endow.*, 18(6):1798–1810, 2025. doi:10.14778/3725688.3725707.

45 Ashish Gupta and Inderpal Singh Mumick. Maintenance of materialized views: Problems, techniques, and applications. *IEEE Data Eng. Bull.*, 18(2):3–18, 1995. URL: http://sites.computer.org/debull/95JUN-CD.pdf.

46 Armin Haller and Axel Polleres. Are we better off with just one ontology on the Web? *Semantic Web*, 11(1):87–99, 2020. doi:10.3233/SW-190379.

47 Soonbo Han and Zachary G. Ives. Implementation strategies for views over property graphs. *Proc. ACM Manag. Data*, 2(3):146, 2024. doi:10.1145/3654949.

48 Orit Hazzan. Reflections on teaching abstraction and other soft ideas. *ACM SIGCSE Bull.*, 40(2):40–43, 2008. doi:10.1145/1383602.1383631.

49 Marvin Hofer, Daniel Obraczka, Alieh Saeedi, Hanna Köpcke, and Erhard Rahm. Construction of Knowledge Graphs: Current State and Challenges. *Inf.*, 15(8):509, 2024. doi:10.3390/INFO15080509.

50 Gwo-Jen Hwang, Haoran Xie, Benjamin W. Wah, and Dragan Gašević. Vision, challenges, roles and

research issues of artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 1:100001, 2020. doi:10.1016/j.caeai.2020.100001.

51 Maxime Jakubowski, Dominik Tomaszuk, and Katja Hose. Bridging Gaps in RDF Validation–Insights and Innovation Opportunities in RDF Validation Practices. In *Linking Meaning: Semantic Technologies Shaping the Future of AI*, pages 70–84. IOS Press, 2025. doi:10.3233/SSW250011.

52 Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. Mimic-III, a freely accessible critical care database. *Scientific Data*, 3(1), May 2016. doi:10.1038/sdata.2016.35.

53 Lucie-Aimée Kaffee, Alessandro Piscopo, Pavlos Vougiouklis, Elena Simperl, Leslie Carr, and Lydia Pintscher. A Glimpse into Babel: An Analysis of Multilinguality in Wikidata. In *OpenSym*, pages 14:1–14:5. ACM, 2017. doi:10.1145/3125433.3125465.

54 Jeff Kramer. Is abstraction the key to computing? *Commun. ACM*, 50(4):36–42, 2007. doi:10.1145/1232743.1232745.

55 Jose Emilio Labra-Gayo, Herminio García-González, Daniel Fernández-Alvarez, and Eric Prud'hommeaux. Challenges in RDF validation. In *Current Trends in Semantic Web Technologies: Theory and Practice*, pages 121–151. Springer, 2019. doi:10.1007/978-3-030-06149-4_6.

56 Ora Lassila, Michael Schmidt, Olaf Hartig, Brad Bebee, Dave Bechberger, Willem Broekema, Ankesh Khandelwal, Kelvin Lawrence, Carlos Manuel López Enríquez, Ronak Sharda, and Bryan B. Thompson. The OneGraph vision: Challenges of breaking the graph model lock-in. *Semantic Web*, 14(1):125–134, 2023. doi:10.3233/SW-223273.

57 Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. *https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/*, 1999.

58 Minmin Leng, Yue Sun, Ce Li, Shuyu Han, and Zhiwen Wang. Usability evaluation of a knowledge graph–based dementia care intelligent recommender system: Mixed methods study. *Journal of Medical Internet Research*, 25:e45788, September 2023. doi:10.2196/45788.

59 Harry X. Li, Gabriel Appleby, Camelia Daniela Brumar, Remco Chang, and Ashley Suh. Knowledge graphs in practice: Characterizing their users, challenges, and visualization opportunities. *IEEE Trans. Vis. Comput. Graph.*, 30(1):584–594, 2024. doi:10.1109/TVCG.2023.3326904.

60 Tobias Lindaaker. An overview of the recent history of graph query languages. Technical report dm32.2-2018-00085r1, ISO/IEC JTC 1 Database Languages WG3, May 2018. Prepared for ISO/IEC JTC 1 WG3. URL: https://s3.amazonaws.com/artifacts.opencypher.org/website/materials/DM32.2/DM32.2-2018-00085R1-recent_history_of_property_graph_query_languages.pdf.

61 Matteo Lissandrini, Davide Mottin, Katja Hose, and Torben Bach Pedersen. Knowledge graph exploration systems: are we lost? In *CIDR*. www.cidrdb.org, 2022. URL: https://vldb.org/cidrdb/2022/knowledge-graph-exploration-systems-are-we-lost.html.

62 Jiebing Ma, Sourav S. Bhowmick, Byron Choi, and Lester Tay. Theories and principles matter: Towards visually appealing and effective abstraction of property graph queries. *Proc. ACM Manag. Data*, 1(2):132:1–132:26, 2023. doi:10.1145/3589277.

63 Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. Getting the Most Out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph. In *ISWC (2)*, LNCS, pages 376–394. Springer, 2018. doi:10.1007/978-3-030-00668-6_23.

64 Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckenschmidt. Anytime bottom-up rule learning for large-scale knowledge graph completion. *VLDB J.*, 33(1):131–161, 2024. doi:10.1007/S00778-023-00800-5.

65 Kevin C. Moore, Erin Wood, Shaffiq Welji, Mike Hamilton, Anne Waswa, Amy B. Ellis, and Halil I. Tasova. Using abstraction to analyze instructional tasks and their implementation. *The Journal of Mathematical Behavior*, 74:101153, 2024. doi:10.1016/j.jmathb.2024.101153.

66 Nebula Graph Documentation. An introduction to graph databases. https://docs.nebula-graph.io/3.4.3/1.introduction/0-1-graph-database/, 2023. Accessed: 2026-02.

67 Finn Årup Nielsen, Katherine Thornton, and José Emilio Labra Gayo. Validating Danish Wikidata lexemes. In *SEMANTiCS (Posters & Demos)*, CEUR Workshop Proceedings. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2451/paper-20.pdf.

68 Farnaz Nouraei, Alexa F. Siu, Ryan A. Rossi, and Nedim Lipka. Thinking outside the box: Non-designer perspectives and recommendations for template-based graphic design tools. In *CHI Extended Abstracts*, pages 326:1–326:9. ACM, 2024. doi:10.1145/3613905.3650967.

69 openCypher Implementers Group. openCypher. https://s3.amazonaws.com/artifacts.opencypher.org/website/, 2015. Accessed: 2026-02.

70 Amedeo Pachera, Angela Bonifati, and Andrea Mauri. User-centric property graph repairs. *Proc. ACM Manag. Data*, 3(1):85:1–85:27, 2025. doi:10.1145/3709735.

71 Amedeo Pachera, Stefania Dumbrava, Angela Bonifati, and Andrea Mauri. Understanding student errors in graph query formulation. *ACM Trans. Comput. Educ.*, 25(3):1–35, 2025. doi:10.1145/3743687.

72 Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. Large language models and knowledge graphs: Opportunities and challenges. *TGDK*, 1(1):2:1–2:38, 2023. doi:10.4230/TGDK.1.1.2.

73 Ayush Pandey, Stefania Dumbrava, Marc Shapiro, Carla Ferreira, Mário Pereira, and Nuno M. Preguiça. Towards local-first distributed property graphs. In *PaPoC@EuroSys*, pages 22–29. ACM, 2025. doi:10.1145/3721473.3722139.

74 Alessandro Piscopo and Elena Simperl. Who Models the World?: Collaborative Ontology Creation and User Roles in Wikidata. *Proc. ACM Hum. Comput. Interact.*, 2(CSCW):141:1–141:18, 2018. doi:10.1145/3274410.

75 Axel Polleres, Romana Pernisch, Angela Bonifati, Daniele Dell'Aglio, Daniil Dobriy, Stefania Dumbrava, Lorena Etcheverry, Nicolas Ferranti, Katja Hose, Ernesto Jiménez-Ruiz, Matteo Lissandrini, Ansgar Scherp, Riccardo Tommasini, and Johannes Wachs. How Does Knowledge Evolve in Open Knowledge Graphs? *TGDK*, 1(1):11:1–11:59, 2023. doi:10.4230/TGDK.1.1.11.

76 Lorena Pujante-Otalora, Manuel Campos, Jose M. Juarez, and Maria-Esther Vidal. Comparison between graph databases and RDF engines for modelling epidemiological investigation of nosocomial infections. In *BIOSTEC (2)*, pages 23–36. SCITEPRESS, 2024. doi:10.5220/0012319900003657.

77 Kashif Rabbani, Matteo Lissandrini, Angela Bonifati, and Katja Hose. Transforming RDF Graphs to Property Graphs using Standardized Schemas. *Proc. ACM Manag. Data*, 2(6), December 2024. doi:10.1145/3698817.

78 Kashif Rabbani, Matteo Lissandrini, and Katja Hose. SHACL and ShEx in the wild: A community survey on validating shapes generation and adoption. In *WWW (Companion Volume)*, pages 260–263. ACM, 2022. doi:10.1145/3487553.3524253.

79 Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Extraction of Validating Shapes from very large Knowledge Graphs. *Proc. VLDB Endow.*, 16(5):1023–1032, 2023. doi:10.14778/3579075.3579078.

80 Phyllis Reisner, Raymond F. Boyce, and Donald D. Chamberlin. Human factors evaluation of two data base query languages: SQUARE and SEQUEL. In *AFIPS National Computer Conference*, AFIPS Conference Proceedings, pages 447–452. AFIPS Press, 1975. doi:10.1145/1499949.1500036.

81 WMDE research. Questions, Methods and Results for the Wikidata Community Survey, 2021. URL: https://commons.wikimedia.org/wiki/File:Wikidata_Community_Survey_2021.pdf.

82 Ian Robinson, Jim Webber, and Emil Eifrem. *Graph databases: new opportunities for connected data.* O'Reilly Media, Inc., 2015.

83 Siddhartha Sahu, Amine Mhedhbi, Semih Salihoglu, Jimmy Lin, and M. Tamer Özsu. The ubiquity of large graphs and surprising challenges of graph processing: extended survey. *VLDB J.*, 29(2-3):595–618, 2020. doi:10.1007/S00778-019-00548-X.

84 Sherif Sakr, Angela Bonifati, Hannes Voigt, Alexandru Iosup, Khaled Ammar, Renzo Angles, Walid G. Aref, Marcelo Arenas, Maciej Besta, Peter A. Boncz, Khuzaima Daudjee, Emanuele Della Valle, Stefania Dumbrava, Olaf Hartig, Bernhard Haslhofer, Tim Hegeman, Jan Hidders, Katja Hose, Adriana Iamnitchi, Vasiliki Kalavri, Hugo Kapp, Wim Martens, M. Tamer Özsu, Eric Peukert, Stefan Plantikow, Mohamed Ragab, Matei Ripeanu, Semih Salihoglu, Christian Schulz, Petra Selmer, Juan F. Sequeda, Joshua Shinavier, Gábor Szárnyas, Riccardo Tommasini, Antonino Tumeo, Alexandru Uta, Ana Lucia Varbanescu, Hsiang-Yun Wu, Nikolay Yakovets, Da Yan, and Eiko Yoneki. The future is big graphs: a community view on graph processing systems. *Commun. ACM*, 64(9):62–71, 2021. doi:10.1145/3434642.

85 Cristina Sarasua, Alessandro Checco, Gianluca Demartini, Djellel Eddine Difallah, Michael Feldman, and Lydia Pintscher. The Evolution of Power and Standard Wikidata Editors: Comparing Editing Behavior over Time to Predict Lifespan and Volume of Edits. *Comput. Support. Cooperative Work.*, 28(5):843–882, 2019. doi:10.1007/S10606-018-9344-Y.

86 Johannes Schrott, Maxime Jakubowski, and Katja Hose. A Graph-Native Approach to Normalization. *CoRR*, abs/2603.02995, 2026. doi:10.48550/arXiv.2603.02995.

87 Xudong Shen, Hannah Brown, Jiashu Tao, Martin Strobel, Yao Tong, Akshay Narayan, Harold Soh, and Finale Doshi-Velez. Directions of technical innovation for regulatable AI systems. *Commun. ACM*, 67(11):82–89, 2024. doi:10.1145/3653670.

88 Valerie J. Shute, Chen Sun, and Jodi Asbell-Clarke. Demystifying computational thinking. *Educational Research Review*, 22:142–158, 2017. doi:10.1016/j.edurev.2017.09.003.

89 Sophia Sideri, Vasilis Efthymiou, Dimitris Plexousakis, and Haridimos Kondylakis. PG2RDF: schema-guided transformation of property graphs to RDF. In *IJCKG*, Lecture Notes in Computer Science, pages 358–373. Springer, 2025. doi:10.1007/978-981-95-5009-8_24.

90 Shani Evenstein Sigalov and Rafi Nachmias. Investigating the potential of the semantic web for education: Exploring Wikidata as a learning platform. *Educ. Inf. Technol.*, 28(10):12565–12614, 2023. doi:10.1007/S10639-023-11664-1.

91 Harold R. Solbrig, Eric Prud'hommeaux, Grahame Grieve, Lloyd McKenzie, Joshua C. Mandel, Deepak K. Sharma, and Guoqian Jiang. Modeling and validating HL7 FHIR profiles using semantic web shape expressions (ShEx). *J. Biomed. Informatics*, 67:90–100, 2017. doi:10.1016/J.JBI.2017.02.009.

92 John F. Sowa. Conceptual graphs for a data base interface. *IBM J. Res. Dev.*, 20(4):336–357, 1976. doi:10.1147/RD.204.0336.

93 Slawek Staworko, Iovka Boneva, José Emilio Labra Gayo, Samuel Hym, Eric G. Prud'hommeaux, and Harold R. Solbrig. Complexity and expressiveness of ShEx for RDF. In *ICDT*, LIPIcs, pages 195–211. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.ICDT.2015.195.

94 Radu Stoica, George H. L. Fletcher, and Juan F. Sequeda. On directly mapping relational databases to property graphs. In *AMW*, CEUR Workshop Proceedings. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2369/short06.pdf.

95 John C. Thomas and John D. Gould. A psychological study of query by example. In *AFIPS Na-*

tional Computer Conference, AFIPS Conference Proceedings, pages 439–445. AFIPS Press, 1975. doi:10.1145/1499949.1500035.

96 Hernán Vargas, Carlos Buil-Aranda, Aidan Hogan, and Claudia López. A user interface for exploring and querying knowledge graphs (extended abstract). In *IJCAI*, pages 4785–4789. ijcai.org, 2020. doi:10.24963/IJCAI.2020/666.

97 Eliane Zambon Victorelli, Júlio Cesar dos Reis, Heiko Horst Hornung, and Alysson Bolognesi Prado. Understanding human-data interaction: Literature review and recommendations for design. *Int. J. Hum. Comput. Stud.*, 134:13–32, 2020. doi:10.1016/J.IJHCS.2019.09.004.

98 Denny Vrandecic. Wikidata: a new platform for collaborative data collection. In *WWW (Companion Volume)*, pages 1063–1064. ACM, 2012. doi:10.1145/2187980.2188242.

99 Domagoj Vrgoc, Carlos Rojas, Renzo Angles, Marcelo Arenas, Diego Arroyuelo, Carlos Buil-Aranda, Aidan Hogan, Gonzalo Navarro, Cristian Riveros, and Juan Romero. MillenniumDB: An open-source graph database system. *Data Intell.*, 5(3):560–610, 2023. doi:10.1162/DINT_A_00229.

100 Andra Waagmeester, Lynn Schriml, and Andrew Su. Wikidata as a linked-data hub for biodiversity data. *Biodiversity Information Science and Standards*, 3, 2019. doi:10.3897/biss.3.35206.

101 Lucas Werkmeister. Schema inference on Wikidata. Master's thesis, Karlsruhe Institute of Technology, 2018. URL: https://github.com/lucaswerkmeister/master-thesis/releases/download/final/master-thesis-Lucas-Werkmeister.pdf.

102 Wikidata Community. Data quality days 2021. https://www.wikidata.org/wiki/Wikidata:Events/Data_Quality_Days_2021, 2021. Accessed: 2026-02-16.

103 Wikidata Community. Data quality days 2022. https://www.wikidata.org/wiki/Wikidata:Events/Data_Quality_Days_2022, 2022. Accessed: 2026-02-16.

104 Wikidata Community. Data modelling days 2023. https://www.wikidata.org/wiki/Wikidata_talk:Events/Data_Modelling_Days_2023, 2023. Accessed: 2026-02-16.

105 Wikimedia Foundation. Wikidata: Data access. https://www.wikidata.org/wiki/Wikidata:Data_access, 2026. Accessed: 2026-02-16.

106 Wikimedia Foundation. Wikidata: Licensing. https://www.wikidata.org/wiki/Wikidata:Licensing, 2026. Accessed: 2026-02-16.

107 Wikimedia Foundation. Wikidata query service. https://query.wikidata.org/, 2026. Accessed: 2026-02-16.

108 Katy Williams, Alex Bigelow, and Katherine E. Isaacs. Data abstraction elephants: The initial diversity of data representations and mental models. In *CHI*, pages 803:1–803:24. ACM, 2023. doi:10.1145/3544548.3580669.

109 Bingjun Xie, Jia Zhou, and Huilin Wang. How influential are mental models on interaction performance? exploring the gap between users' and designers' mental models through a new quantitative method. *Adv. Hum. Comput. Interact.*, 2017:3683546:1–3683546:14, 2017. doi:10.1155/2017/3683546.

110 Vahan Yoghourdjian, Daniel Archambault, Stephan Diehl, Tim Dwyer, Karsten Klein, Helen C. Purchase, and Hsiang-Yun Wu. Exploring the limits of complexity: A survey of empirical studies on graph visualisation. *Vis. Informatics*, 2(4):264–282, 2018. doi:10.1016/J.VISINF.2018.12.006.

111 Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.*, 56(4):94:1–94:62, 2024. doi:10.1145/3618295.

112 Moshé M. Zloof. Query-by-Example: the Invocation and Definition of Tables and Forms. In *VLDB*, pages 1–24. ACM, 1975. doi:10.1145/1282480.1282482.