Property Graph Standards: State of the Art & Open Challenges

Haridimos Kondylakis
Computer Science Department, UOC
kondylak@csd.uoc.gr

Stefania Dumbrava
ENSIIE & SAMOVAR - Télécom SudParis
stefania.dumbrava@ensiie.fr

Matteo Lissandrini
University of Verona
matteo.lissandrini@univr.it

Nikolay Yakovets
Eindhoven University of Technology
hush@tue.nl

Angela Bonifati
Lyon 1 University & IUF
angela.bonifati@liris.cnrs.fr

Vasilis Efthymiou
Harokopio University of Athens
vefthym@hua.gr

George Fletcher
Eindhoven University of Technology
g.h.l.fletcher@tue.nl

Dimitris Plexousakis
FORTH-ICS
dp@ics.forth.gr

Riccardo Tommasini
INSA-Lyon
riccardo.tommasini@insa-lyon.fr

Georgia Troullinou
CNRS, Univ. Grenoble Alpes
georgia.troullinou@univ-grenoble-alpes.fr

Elisjana Ymeralli
FORTH-ICS
ymeralli@ics.forth.gr

ABSTRACT

Property Graphs are a versatile and expressive data model that has gained widespread adoption due to their flexibility in supporting labeled and attributed nodes and edges. They are well-established in research communities and are becoming widespread in companies and organizations across various sectors. They have been boosted by a fervent ISO/IEC standardization activity, leading to dedicated query and schema languages. While the current standards are still evolving, opportunities remain to enrich them with features such as composability. The plethora of existing query languages reflects a rich and diverse ecosystem, which ongoing unification efforts aim to align. This tutorial aims to deepen the understanding of Property Graph standards by showcasing their strengths, highlighting recent unification efforts, clarifying the central role of schema constraints, and exploring the rich landscape of research and industrial opportunities shaping the future of graph data management.

PVLDB Reference Format:
Haridimos Kondylakis, Stefania Dumbrava, Matteo Lissandrini, Nikolay Yakovets, Angela Bonifati, Vasilis Efthymiou, George Fletcher, Dimitris Plexousakis, Riccardo Tommasini, Georgia Troullinou, and Elisjana Ymeralli. Property Graph Standards:
State of the Art & Open Challenges. PVLDB, 18(12): 5477 - 5481, 2025.
doi:10.14778/3750601.3750698

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://propertygraphstandardstutorial.github.io/.

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750698

1 INTRODUCTION

Graphs are a flexible and expressive data model for representing complex network-structured data across diverse application domains, including social networks, biological networks, bioinformatics, cheminformatics, medical data, and knowledge management. As “unifying abstractions”, graphs leverage interconnectedness to represent, explore, predict, and explain both real-world and digital phenomena. We are witnessing unprecedented growth in interconnected data that highlights the vital societal role of graph processing, as emphasized in the CACM article "The future is big graphs" [18]. The impact of this growth is evident, as big data processing systems are being applied to many complex graph data management tasks.

Currently, most graph data management systems use the Resource Description Framework (RDF) or Property Graph (PG) models. RDF, the W3C standard for data sharing and interoperability, is central to the Linked Data and FAIR data initiatives of the Semantic Web community. Property graphs emerged in enterprise data management as multigraphs whose nodes and edges can have labels and properties (key-value pairs). The PG model has become increasingly popular, with PG solutions serving 75% of Fortune 500 companies [12]. Notably, at the foundational level, all models underlying graph database systems are subsumed by the PG model.

The industrial popularity of the PG model led to its selection for standardization by ISO (International Organization for Standardization). While the standard is still evolving, it provides a solid foundation for future enhancements. In recent years, research in the field has grown significantly, marking a period of rapid progress and highlighting several key challenges.

This tutorial aims to provide a comprehensive understanding of the PG model and its unique features, along with the current standardization endeavors. It explores the diverse landscape of PG processing by surveying current capabilities, analyzing schema constraints, outlining open challenges, and highlighting knowledge gaps in advancing research and industrial applications.

Intended audience. This tutorial is designed for a diverse audience, including researchers, students, IT professionals, practitioners, and developers. It will particularly benefit those who work extensively with graphs and property graphs. Students and researchers will receive a comprehensive introduction to the field and an overview of current research challenges that could guide their future work. Practitioners will gain insight into the latest algorithms, techniques, and systems that can enhance their understanding of the domain. System developers working with PGs will learn how to enhance property graph products through improved querying, visualization, and comprehension.

Prerequisites. This tutorial requires only a basic familiarity with graph theory and algorithms. Advanced knowledge of query evaluation techniques (from standard undergraduate curricula), familiarity with W3C RDF standards, Linked Open Data, data mining, or data visualization may help participants grasp certain concepts more quickly but is not required.

Tutorial Material. The tutorial will use slides with code samples, datasets, and best practices. All content will be available on the tutorial web page before August 20 under a CC license. *No additional audiovisual or technical equipment is required.*

Related Events. No part of this tutorial has been presented before. Although there have been several tutorials in related areas in the past five years at major database conferences, none have focused specifically on PGs. *Our tutorial is the first to offer a unified view on PG standards, schema evolution, and interoperability across the graph model spectrum.* Previous tutorials have covered graph query techniques [2, 26, 30, 31], advances in graph data management systems [27], visual query interfaces [4], indexing [39], knowledge graph embeddings, analytics, and machine learning [11, 37, 38], the historical development of knowledge graphs [24], and continuous querying in the graph landscape [10]. These tutorials complement rather than overlap with our focus on property graphs.

Tutorial Structure and Duration. This tutorial is 3 hours long and divided into three parts. Part I (Section 2) is an introduction to the industrial context around property graphs and their query languages. Part II (Section 3) provides a detailed exploration of property graph languages, covering key topics, such as the Data Manipulation Language, the Data Definition Language, and transformations. Part III (Section 4) focuses on in-use aspects concerning property graphs, e.g., graph schema discovery, data integration, education, and training, and outlines emerging challenges that go beyond current standards. In particular, we will highlight open issues related to compositionality, scalability, interoperability, and the integration of property graphs with AI-driven techniques, such as embeddings and large language models. The two 1.5-hour sessions will cover Parts I and II in the first session, and Part III in the second, allowing a natural progression from establishing foundational knowledge of the PG landscape and languages to advanced applications and future research directions.

2 INDUSTRIAL CONTEXT

The emergence of standardized property graph query languages - GQL and SQL/PGQ - marks an important milestone in graph database development. Although the adoption of these standards is still ongoing, with varying levels of support across different systems and versions, we expect broader implementation throughout the graph database ecosystem as the standards mature.

Database System	(open)Cypher	SQL/PGQ	GQL
Native Graph Databases			
Neo4j	✓	–	Planned
TigerGraph	–	–	Planned
NebulaGraph	✓	–	✓
Kuzu	✓	–	–
MillenniumDB	✓	–	Planned
Apache HugeGraph	✓	–	–
Quine	✓	–	Planned
In-Memory Graph Databases			
RedisGraph	✓	–	–
Memgraph	✓	–	–
Graph Extensions for Relational Databases			
DuckDB	–	✓	–
SAP HANA Graph	✓	–	–
Oracle PGX	–	✓	–
Cloud-first Multi-model Databases			
Amazon Neptune	✓	–	Planned
Spanner Graph	–	✓	✓

Table 1: Adoption of Graph Query Languages

Table 1 aims to provide an overview of current support for GQL and SQL/PGQ, along with notable implementations of an established but not standardized (open)Cypher query language.

Since the GQL ISO standardization in 2024, several major GDBMS have already announced plans to implement it. This signals a shift in the graph database industrial landscape. *Neo4j*, a key contributor to the GQL standard, plans to align its offerings to include full GQL support, leveraging its extensive experience with Cypher to shape the evolution of GQL. *Amazon Neptune*, a fully managed service on AWS with features such as high availability, scalability, and robust security, has also declared its intention to incorporate GQL, further validating the importance of the standard in cloud environments. *MillenniumDB*, an open-source multi-model graph database known for advanced query processing features supported by solid theoretical foundations (e.g., worst-case optimal joins), has also committed to GQL support. In addition, *NebulaGraph*, with its focus on high performance and flexible data modeling, already provides native GQL support in its Enterprise edition (since v5.0), demonstrating early traction among distributed graph database systems. *Google Cloud's Spanner Graph*, a globally distributed database with a unified relational and graph model, further advances the adoption of GQL, highlighting its enterprise-scale capabilities and potential for managing both relational and graph data. Furthermore, established systems such as *SAP HANA Graph*, with its comprehensive feature set, *TigerGraph*, known for its high-performance graph analytics and native parallel graph platform, and *Memgraph*, with its open-source, in-memory design and focus on real-time analysis, have committed themselves to support, underscoring a broader industry trend toward standardization. As GQL matures, its implementation is anticipated across a wider range of graph systems.

3 PROPERTY GRAPH LANGUAGES

Data Manipulation Language. A Data Manipulation Language (DML) allows users to retrieve and modify graph data, such as insert or delete nodes, edges, and properties. In the context of property graphs, such languages allow users to leverage the rich, interconnected structure of graph data, which is especially useful for managing complex networks, such as social networks, product catalogs, etc. To a large extent, the reason why property graphs exist is the ability to handle the data as graphs instead of tables (as in traditional relational databases), going beyond simple joins to allow more intuitive pattern matching and traversal-based queries (e.g., friends-of-friends queries). In this tutorial, we will go over some of the most popular property graph query languages, including Cypher, GCORE, and the recently standardized property graph query languages, GQL and SQL/PGQ. Following a common property graph as a running example throughout the tutorial, we will see how each query language can formulate the same user query, as well as the high-level strengths of each language.

Data Definition Language. A Data Definition Language (DDL) is fundamental for any data model as it provides the formal framework to define, structure, and enforce data integrity rules. In the context of property graphs, while the technology has reached maturity with multiple robust DBMSs and ongoing ISO standardization efforts for GQL, schema support remains limited in both existing systems and the first GQL Standard version. In this area, two significant developments are shaping the future of property graph schemas and constraints. First, PG-Keys [14] emerged from a collaborative community effort in LDBC and introduced flexible key constraints with different modes (combinations of exclusive, mandatory, and singleton restrictions) that can be applied to nodes, edges, and properties. Based on PG-Keys, PG-Schema [17] was developed in LDBC as a comprehensive schema formalism that combines flexible type definitions supporting multi-inheritance with expressive constraints based on the PG-Keys framework. These developments are expected to influence the second version of the GQL standard, which is expected to include a rich DDL, making graph database systems more useful, powerful, and expressive while ensuring proper data integrity and object identification capabilities.

Transformations. Graph transformations allow to systematically convert an input graph into another by applying custom rules or operations. Particularly in graph data management, they are important for data integration, exchange, evolution, and migration.

Supporting property graph transformations (PGTs) is a key future requirement of the GQL standardization effort, as PGTs have the potential to enable seamless interoperability among heterogeneous graph models. Research in this area is still in its infancy, but graph transformations have recently been used to support automatic translations between Property Graphs and RDF, while preserving structural and semantic integrity [20, 35]. The S3PG system leverages schema definitions, i.e., SHACL for RDF and PG-Schema for Property Graphs, to establish bidirectional mappings and schema alignments, addressing the key challenge of representing RDF triples with the nodes, edges, and key-value property primitives of the Property Graph model [35].

Graph transformations also play a crucial role in supporting database evolution within a given graph data model, for example, using rules with data values expressed in GPC [9] and in abstracting and adapting graph data to user requirements through view formulation. In the latter setting, recent work focuses on strategies for implementing and maintaining property graph views [6, 25]. Moreover, for high-velocity updates, stream processing techniques were also adopted in the PG setting [32], with recent work also extending Cypher for window-based continuous queries [15].

Understanding the incorporation of the PG-Schema and PG-Key constraints for property graph transformations in the above areas is essential in the domain. The tutorial will pinpoint interesting future research directions, e.g., in the spirit of graph data exchange for labeled graphs [3].

4 APPLICATIONS AND OPEN CHALLENGES

Schema Discovery.

Despite the widespread adoption of property graphs (PGs), support for schema constraints remains limited and fragmented in both GDBMSs [17] and applications [13], hindering tasks like data integration, query optimization, and ML feature extraction. Schema discovery seeks to automatically infer PG structure, a problem previously explored to improve data quality and querying [22, 23]. Approaches like [28] use label- or property-based heuristics, though each has drawbacks. GMMSchema [8] addresses these by using Gaussian Mixture Models to cluster nodes by both labels and properties, producing accurate and scalable hierarchical schemas. Its extension in DiscoPG [7] supports schema visualization and incremental updates. A key challenge remains: developing hybrid symbolic-statistical methods for robust schema discovery in evolving PGs.

Data Integration. Data integration for property graphs presents significant challenges, particularly in achieving seamless interoperability with RDF graphs. This requires either mapping both models to a common unifying representation, such as the recently introduced OneGraph [16] meta-model, or enabling direct transformations between the two. A key challenge with direct transformations is preserving semantics, i.e., ensuring that RDF triples, relationships, and associated inference rules are accurately mapped to the nodes, edges, and properties of property graphs and vice versa. Additionally, such transformations must also account for features such as multi-valued attributes, diverse edge types, hierarchical relationships, and schema constraints. Finally, maintaining graph views poses an additional challenge, requiring mechanisms to ensure that derived or materialized views of graph data remain up-to-date with minimal performance overhead as the underlying graph evolves.

In this context, schema alignment plays a crucial role. RDF leverages well-established standards, such as SHACL for defining constraints, whereas property graphs are positioned to adopt emerging standards like PG-Schema. Ensuring that graph transformations uphold these standards is essential to maintaining data integrity and compatibility across different graph data models.

Education and Training. Although graph databases are increasingly used in various domains, including social networks, life sciences, logistics, and finance, their educational aspects require more attention. Indeed, the development of graph management systems (and data management systems, in general) has historically prioritized performance and functionality. Increasingly, these efforts must also place more emphasis on improving usability, in particular from a learning and training perspective. The tutorial will highlight key educational challenges posed by property graphs [33]. These include understanding the intricacies of graph data modeling, the role and application of graph schema concepts for tasks such as data navigation and query formulation, as well as grasping graph pattern matching mechanisms and custom graph query language primitives. In addition to addressing these issues, the tutorial will propose best practices for teaching the fundamentals of PGs, such that learners are apt to use these in real-world scenarios.

Beyond PG Standards. The publication of the GQL/PGQ standards was a historical event. However, the research community is already discussing how to go beyond such standards, which requires addressing critical functionality, performance, and usability gaps. Standards provide a common foundation yet often fail to support advanced use cases, such as dynamic schema evolution, heterogeneous data integration, or real-time analytics.

To date, graph data normalization and standard forms remain fundamentally under-researched. Normalizing property graphs has only recently been the center of discussion [36], with current proposals failing to address the full spectrum of issues related to identifying and resolving redundancies in a graph schema. Furthermore, the question of how to express, encode, and perform inference in a property graph so as to support the same functionalities as for RDF-based Knowledge Graphs remains an unresolved challenge.

Future research and industry efforts must focus on complementing these standards with robust frameworks capable of supporting (multi-)graph *compositionality*, *scalability*, and *interoperability*.

Compositionality. The key takeaway of a very recent study [21] is that current graph query languages lack full compositionality, as they primarily operate as graph-to-relation transformations.

Scalability. As query languages become more expressive and standardization efforts have been established with the emergence of GQL and SQL/PGQ, a significant challenge in property graph querying continues to be scalability. GSQL was designed to allow massively parallel execution of property graph queries. Yet, the data deluge and the ever-increasing need to analyze data from various sources are hard to compete with today's technology, making graph query answering relatively slow. A potential solution to the scalability challenges is integrating quantum computing [19].

Interoperability. Moreover, such standards should allow seamless integration with broader data ecosystems, including knowledge graphs and relational systems. Worth mentioning is the development of streaming graph processing languages and techniques that have emerged as a critical area of focus. As we mentioned, there are extensions of property graph languages with specialized constructs for streaming [15, 32], but interoperability with standardized languages such as GQL or SQL / PGQ remains limited. A promising research direction is developing hybrid processing paradigms that integrate streaming and static graph capabilities. It is also essential to establish standardized methods for expressing temporal graph queries and explore approaches for incremental, continuous query answering in high-throughput environments [10].

AI, Embeddings & LLMs. The convergence of PG and GenAI technologies has the potential to drive innovation in data analysis, knowledge discovery, and decision-making across a wide range of domains. In particular, large language models [1, 29, 34] and knowledge graph embeddings [5], which are now integral to modern AI pipelines, can enhance tasks such as semantic search over knowledge graphs, knowledge graph completion, and reasoning. However, as emphasized in recent work [5], embeddings still face significant challenges in incorporating structural constraints. Exploring how these techniques can leverage schema formalisms, such as PG-Schema and PG-Constraints, offers a promising direction to improve their accuracy and interpretability.

5 SHORT BIOGRAPHIES OF THE PRESENTERS

Haridimos Kondylakis is an Associate Professor of Big Data Engineering in the Department of Computer Science, University of Crete, and an Affiliated Researcher at FORTH-ICS. He has delivered multiple high-profile tutorials (e.g., EDBT, ISWC) on related topics.

Stefania Dumbrava is an Associate Professor at ENSIIE and Télécom SudParis, France. She is interested in reliable graph data management and recently lectured at the VLDB Summer School.

Matteo Lissandrini is an Associate Professor at the University of Verona, specializing in data exploration, knowledge graphs, and intelligent data management. He has also presented tutorials on exploratory search at ESWC'20, SIGIR'19, SIGMOD'19, and VLDB'17.

Nikolay Yakovets is an Assistant Professor in the Department of Computer Science and Mathematics at Eindhoven University of Technology. He has experience delivering tutorials and invited lectures both within the data management community and beyond.

Angela Bonifati is a Distinguished Professor at Lyon 1 University, a Senior Member at the French University Institute, and an Adjunct Professor at the University of Waterloo. Her current research interests are in graph databases, knowledge graphs, and data integration along with data science and AI.

Vasilis Efthymiou is an Assistant Professor at the Harokopio University of Athens (HUA). He has been co-organizing the SemTab challenges at ISWC and the Tabular Data Analysis (TaDA) workshop at VLDB.

George Fletcher is a Professor at Eindhoven University of Technology, where he chairs the Database Research Group and the Data & AI Cluster. He is studying human and social aspects of data systems, query languages, and schema languages.

Dimitris Plexousakis is a Professor of Computer Science at the University of Crete and the Director of the Institute of Computer Science, FORTH, and Head of the Information Systems Laboratory. He has delivered tutorials at international conferences and summer schools.

Riccardo Tommasini is an Associate Professor at INSA Lyon. His research interests cover continuous queries, streaming graph processing, and programming languages. He has presented many tutorials on the above topics, e.g., SIGMOD'24, ISWC'24, RW'19.

Georgia Troullinou is a postdoctoral researcher at CNRS in the Laboratoire d'Informatique de Grenoble (LIG). She is focusing on knowledge representation and management, with a particular interest in summarizing semantic knowledge graphs, big data partitioning, and semantic schema discovery.

Elisjana Ymeralli is a PhD candidate in the Computer Science Department at the University of Crete. She contributed to several European and National projects and is interested in knowledge representation, big data partitioning, and the Semantic Web.

ACKNOWLEDGMENTS

The work reported in this paper was partially supported by the H.F.R.I. Project Number: 16819, by Horizon DataGEMS (No 101188416). and by the grant ANR-24-CE25-1109.

REFERENCES

[1] Sihem Amer-Yahia, Angela Bonifati, Lei Chen, Guoliang Li, Kyuseok Shim, Jianliang Xu, and Xiaochun Yang. 2023. From Large Language Models to Databases and Back: A Discussion on Research and Education. *SIGMOD Rec.* 52, 3 (2023), 49–56.

[2] Marcelo Arenas, Claudio Gutierrez, and Juan F Sequeda. 2021. Querying in the age of graph databases and knowledge graphs. In *SIGMOD*. 2821–2828.

[3] Pablo Barceló, Jorge Pérez, and Juan L. Reutter. 2013. Schema mappings and data exchange for graph databases. In *EDBT/ICDT*. 189–200.

[4] Sourav S Bhowmick and Byron Choi. 2022. Data-driven visual query interfaces for graphs: past, present, and (near) future. In *SIGMOD*. 2441–2447.

[5] Russa Biswas, Lucie-Aimée Kaffee, Michael Cochez, Stefania Dumbrava, Theis E. Jendal, Matteo Lissandrini, Vanessa López, Eneldo Loza Mencía, Heiko Paulheim, Harald Sack, Edlira Vakaj, and Gerard de Melo. 2023. Knowledge Graph Embeddings: Open Challenges and Opportunities. *TGDK* 1, 1 (2023), 4:1–4:32.

[6] Angela Bonifati, Stefania Dumbrava, and Emilio Jesús Gallego Arias. 2018. Certified Graph View Maintenance with Regular Datalog. *Theory Pract. Log. Program.* 18, 3-4 (2018), 372–389.

[7] Angela Bonifati, Stefania-Gabriela Dumbrava, Emile Martinez, Fatemeh Ghasemi, Malo Jaffré, Pacome Luton, and Thomas Pickles. 2022. DiscoPG: Property Graph Schema Discovery and Exploration. *Proc. VLDB Endow.* 15, 12 (2022), 3654–3657.

[8] Angela Bonifati, Stefania Dumbrava, and Nicolas Mir. 2022. Hierarchical Clustering for Property Graph Schema Discovery. In *EDBT*. 2:449–2:453.

[9] Angela Bonifati, Filip Murlak, and Yann Ramusat. 2024. Transforming Property Graphs. *VLDB* 17, 11 (2024), 2906–2918.

[10] Angela Bonifati and Riccardo Tommasini. [n.d.]. An Overview of Continuous Querying in (Modern) Data Systems. In *Companion of ACM SIGMOD/PODS*. 605–612.

[11] Felipe T Brito, André LC Mendonça, and Javam C Machado. 2024. A Differentially Private Guide for Graph Analytics. In *EDBT*. 850–853.

[12] David Cushman. 2021. Record investment in Neo4j suggests, maybe it IS all about relationships. https://www.hfsresearch.com/research/record-investment-in-neo4j-suggests-maybe-it-is-all-about-relationships/ [Online; posted June-2021].

[13] Stefania Dumbrava, Melchior W. M. Oudemans, and Burcu Kulahcioglu Ozkan. 2025. Fuzzing Graph Database Applications with Graph Transformations. In *Graph Transformation*, Jörg Endrullis and Matthias Tichy (Eds.). Springer Nature Switzerland, 135–156. http://doi.org/10.1007/978-3-031-94706-3_7

[14] Angles Renzo et al. 2021. PG-Keys: Keys for Property Graphs. In *SIGMOD*. 2423–2436.

[15] Christopher Rost et al. [n.d.]. Seraph: Continuous Queries on Property Graph Streams. In *EDBT*. 234–247.

[16] Ora Lassila et al. 2023. The OneGraph vision: Challenges of breaking the graph model lock-in. *Semantic Web* 14, 1 (2023), 125–134.

[17] Renzo Angles et al. 2023. PG-Schema: Schemas for Property Graphs. *Proc. ACM Manag. Data* 1, 2 (2023), 198:1–198:25.

[18] Sherif Sakr et al. 2021. The future is big graphs: a community view on graph processing systems. *Commun. ACM* 64, 9 (2021), 62–71.

[19] Tobias Winker et al. [n.d.]. Quantum Machine Learning: Foundation, New Techniques, and Opportunities for Database Research. In *SIGMOD/PODS Companion*. 45–52.

[20] Ewout Gelling, George Fletcher, and Michael Schmidt. 2024. Statement Graphs: Unifying the Graph Data Model Landscape. In *DASFAA*. 364–376.

[21] Amélie Gheerbrant, Leonid Libkin, Liat Peterfreund, and Alexandra Rogova. 2024. GQL and SQL/PGQ: Theoretical Models and Expressive Power. *CoRR* abs/2409.01102 (2024).

[22] Silvio Normey Gómez, Lorena Etcheverry, Adriana Marotta, and Mariano P. Consens. 2018. Findings from Two Decades of Research on Schema Discovery using a Systematic Literature Review. In *AMW (CEUR Workshop Proceedings)*, Vol. 2100.

[23] Benoît Groz, Aurélien Lemay, Slawek Staworko, and Piotr Wieczorek. 2022. Inference of Shape Graphs for Graph Databases. In *ICDT*. 7:1–7:20.

[24] Claudio Gutierrez and Juan F Sequeda. 2020. Knowledge graphs: a tutorial on the history of knowledge Graph's Main ideas. In *CIKM*. 3509–3510.

[25] Soonbo Han and Zachary G. Ives. 2024. Implementation Strategies for Views over Property Graphs. *SIGMOD* 2, 3 (2024), 146.

[26] Aidan Hogan and Domagoj Vrgoć. 2024. Querying Graph Databases at Scale. In *SIGMOD*. 585–589.

[27] Arijit Khan and Cuneyt Gurcan Akcora. 2022. Graph-based management and mining of blockchain data. In *CIKM*. 5140–5143.

[28] Hanâ Lbath, Angela Bonifati, and Russ Harmer. 2021. Schema Inference for Property Graphs. In *EDBT*. 499–504.

[29] Hoa Thi Le, Angela Bonifati, and Andrea Mauri. 2025. Graph Consistency Rule Mining with LLMs: an Exploratory Study. In *EDBT*. OpenProceedings.org, 748–754.

[30] Ioana Manolescu and Madhulika Mohanty. 2023. Full-Power Graph Querying: State of the Art and Challenges. *VLDB* 16, 12 (2023), 3886–3889.

[31] Amine Mhedhbi and Semih Salihoglu. 2022. Modern Techniques for Querying Graph-Structured Relations: Foundations, System Implementations, and Open Challenges. *Proc. VLDB Endow.* 15, 12 (2022), 3762–3765. https://doi.org/10.14778/3554821.3554894

[32] Anil Pacaci, Angela Bonifati, and M. Tamer Özsu. [n.d.]. Evaluating Complex Queries on Streaming Graphs. In *ICDE*. 272–285.

[33] Amedeo Pachera, Stefania Dumbrava, Angela Bonifati, and Andrea Mauri. 2025. Understanding Student Errors in Graph Query Formulation. *ACM Transactions on Computing Education* (2025). https://doi.org/10.1145/3743687

[34] 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. 2023. Large Language Models and Knowledge Graphs: Opportunities and Challenges. *TGDK* 1, 1 (2023), 2:1–2:38.

[35] Kashif Rabbani, Matteo Lissandrini, Angela Bonifati, and Katja Hose. 2025. Transforming RDF Graphs to Property Graphs using Standardized Schemas. In *ACM SIGMOD*.

[36] Philipp Skavantzos and Sebastian Link. 2023. Normalizing Property Graphs. *Proc. VLDB Endow.* (2023).

[37] Bo Xiong, Mojtaba Nayyeri, Daniel Daza, and Michael Cochez. 2023. Reasoning beyond Triples: Recent Advances in Knowledge Graph Embeddings. In *CIKM*. 5228–5231.

[38] Da Yan, Lyuheng Yuan, Akhlaque Ahmad, Chenguang Zheng, Hongzhi Chen, and James Cheng. 2024. Systems for Scalable Graph Analytics and Machine Learning: Trends and Methods. In *SIGKDD*. 6627–6632.

[39] Chao Zhang, Angela Bonifati, and M Tamer Özsu. 2023. An overview of reachability indexes on graphs. In *SIGMOD*. 61–68.