Graph Traversal Languages

for Large-Scale Graph Processing on Modern Hardware

Pascal Zittlau

pascal.zittlau@ovgu.de

April 28, 2025

Faculty of Computer Science
Otto von Guericke University Magdeburg

16th Student Conference on Software Engineering and Database Systems
Supervisor: Paul Blockhaus

Relevance

Why is it relevant?

- Ubiquity of Graphs
- Massive datasets
- Hardware Evolution

Key Problem:

How effectively do high-level graph languages map to optimized execution on diverse, modern hardware?

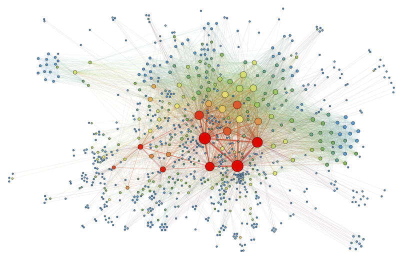


Figure 1: A sample Social Graph [Commons, 2016]

What is the survey about?

- Querying **large-scale graphs**
  - RDF and Property Graphs
- Surveying **graph query languages**
  - SPARQL, Cypher, Gremlin, GQL
- Performance on **modern hardware**
  - Multi-core CPUs, GPUs, FPGAs, . . .

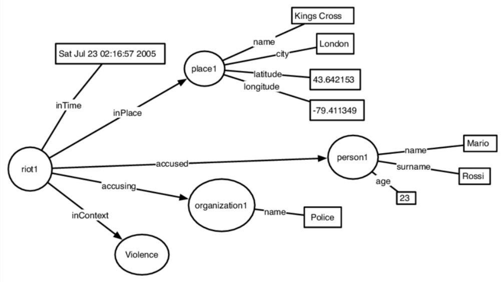


Figure 2: Example of a RDF Graph [Vincenzo et al., 2014]

Key Areas of Investigation:

- Traversal & Pattern Matching
- Parallel Execution
- Hardware Acceleration
- Algorithm Integration
- Backend Abstractions
- Query Optimization

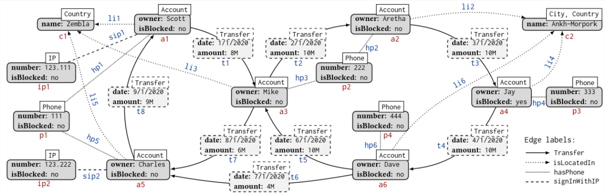


Figure 3: Example of a labelled Property Graph [Deutsch et al., 2021]

```
MATCH (a:Account {isBlocked:'no'})-[:isLocatedIn]->
      (g: City {name:'Ankh-Morpork'})<-[:isLocatedIn]-
      (b:Account {isBlocked:'yes'}),
      p = (a)-[:Transfer*1..]->(b)
RETURN a.owner
```

Listing 1: Cypher Query on graph in Figure 3

Figure 4: [Angles et al., 2017]

Foundations of Modern Query Languages for Graph Databases

RENZO ANGLES, Universidad de Talca & Center for Semantic Web Research
MARCELO ARENAS, Pontificia Universidad Católica de Chile & Center for Semantic Web Research
PABLO BARCELÓ and AIDAN HOGAN, DCC, Universidad de Chile & Center for Semantic Web Research
JUAN REUTTER and DOMAGOJ VRGOČ, Pontificia Universidad Católica de Chile & Center for Semantic Web Research

We survey foundational features underlying modern graph query languages. We first discuss two popular graph data models: *edge-labelled graphs*, where nodes are connected by directed, labelled edges, and *property graphs*, where nodes and edges can further have attributes. Next we discuss the two most fundamental graph querying functionalities: *graph patterns* and *navigational expressions*. We start with graph patterns, in which a graph-structured query is matched against the data. Thereafter, we discuss navigational expressions, in which patterns can be matched recursively against the graph to navigate paths of arbitrary length; we give an overview of what kinds of expressions have been proposed and how they can be combined with graph patterns. We also discuss several semantics under which queries using the previous features can be evaluated, what effects the selection of features and semantics has on complexity, and offer examples of such features in three modern languages that are used to query graphs: SPARQL, Cypher, and Gremlin. We conclude by discussing the importance of formalisation for graph query languages; a summary of what is known about SPARQL, Cypher, and Gremlin in terms of expressivity and complexity; and an outline of possible future directions for the area.

CCS Concepts: • **Information systems → Query languages**; • **Theory of computation → Database query languages (principles)**;

Additional Key Words and Phrases: Property graphs, graph databases, query languages, graph patterns, navigation, aggregation

ACM Reference format:
Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. 2017. Foundations of Modern Query Languages for Graph Databases. *ACM Comput. Surv.* 50, 5, Article 68 (September 2017), 40 pages.
https://doi.org/10.1145/3104031

This work was funded by the Millennium Nucleus Center for Semantic Web Research under Grant No. NC120004.
Authors' addresses: R. Angles, Universidad de Talca, Camino a los Niches Km. 1, Curicó, Chile; email: rangles@utalca.cl; M. Arenas, J. Reutter, and D. Vrgoč, Department of Computer Science, School of Engineering, Pontificia Universidad Católica de Chile, Vicuna Mackenna 4860, Edificio San Agustin, 4to piso, Macul 7820436, Santiago, Chile; emails: {marenas, jreutter, dvrgoc} @ing.puc.cl; P. Barceló and A. Hogan, Department of Computer Science, Universidad de Chile, Avenida Blanco Encalada 2120, Santiago, Chile; emails: pbarcelo@dcc.uc.cl, ahogan@dcc.uchile.cl.

https://doi.org/10.1145/3104031

68

Figure 4: [Angles et al., 2017]

Foundations of Modern Query Languages for Graph Databases

RENZO ANGLES, Universidad de Talca & Center for Semantic Web Research
MARCELO ARENAS, Pontificia Universidad Católica de Chile & Center for Semantic Web Research
PABLO BARCELÓ and AIDAN HOGAN, DCC, Universidad de Chile & Center for Semantic Web Research
JUAN REUTTER and DOMAGOJ VRGOČ, Pontificia Universidad Católica de Chile & Center for Semantic Web Research

We survey foundational features underlying modern graph query languages. We first discuss two popular graph data models: *edge-labelled graphs*, where nodes are connected by directed, labelled edges, and *property graphs*, where nodes and edges can further have attributes. Next we discuss the two most fundamental graph querying functionalities: *graph patterns* and *navigational expressions*. We start with graph patterns, in which a graph-structured query is matched against the data. Thereafter, we discuss navigational expressions, in which patterns can be matched recursively against the graph to navigate paths of arbitrary length; we give an overview of what kinds of expressions have been proposed and how they can be combined with graph patterns. We also discuss several semantics under which queries using the previous features can be evaluated, what effects the selection of features and semantics has on complexity, and offer examples of such features in three modern languages that are used to query graphs: SPARQL, Cypher, and Gremlin. We conclude by discussing the importance of formalisation for graph query languages; a summary of what is known about SPARQL, Cypher, and Gremlin in terms of expressivity and complexity; and an outline of possible future directions for the area.

CCS Concepts: • **Information systems → Query languages**; • **Theory of computation → Database query languages (principles)**;

Additional Key Words and Phrases: Property graphs, graph databases, query languages, graph patterns, navigation, aggregation

ACM Reference format:
Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. 2017. Foundations of Modern Query Languages for Graph Databases. *ACM Comput. Surv.* 50, 5, Article 68 (September 2017), 40 pages.
https://doi.org/10.1145/3104031

This work was funded by the Millennium Nucleus Center for Semantic Web Research under Grant No. NC120004.

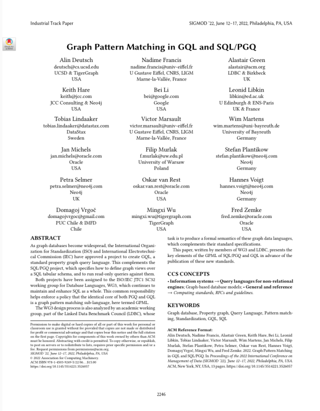
Authors' addresses: R. Angles, Universidad de Talca, Camino a los Niches Km. 1, Curicó, Chile; email: rangles@utalca.cl; M. Arenas, J. Reutter, and D. Vrgoč, Department of Computer Science, School of Engineering, Pontificia Universidad Católica de Chile, Vicuna Mackenna 4860, Edificio San Agustin, 4to piso, Macul 7820436, Santiago, Chile; emails: {marenas, jreutter, dvrgoc} @ing.puc.cl; P. Barceló and A. Hogan, Department of Computer Science, Universidad de Chile, Avenida Blanco Encalada 2120, Santiago, Chile; emails: pbarcelo@dcc.uc.cl, ahogan@dcc.uchile.cl.

https://doi.org/10.1145/3104031

68

ACM Computing Surveys, Vol. 50, No. 5, Article 68. Publication date: September 2017.

Figure 5: [Deutsch et al., 2021]



Industrial Track Paper — SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

Graph Pattern Matching in GQL and SQL/PGQ

Alin Deutsch, deutsch@cs.ucsd.edu, UCSD & TigerGraph, USA
Nadime Francis, nadime.francis@univ-eiffel.fr, U Gustave Eiffel, CNRS, LIGM, Marne-la-Vallée, France
Alastair Green, alastair@acm.org, LDBC & Birkbeck, UK
Keith Hare, keith@jcc.com, JCC Consulting & Neo4j, USA
Bei Li, bei@google.com, Google, USA
Leonid Libkin, libkin@ed.ac.uk, U Edinburgh & ENS-Paris, UK & France
Tobias Lindaaker, tobias.lindaaker@datastax.com, DataStax, Sweden
Victor Marsault, victor.marsault@univ-eiffel.fr, U Gustave Eiffel, CNRS, LIGM, Marne-la-Vallée, France
Wim Martens, wim.martens@uni-bayreuth.de, University of Bayreuth, Germany
Jan Michels, jan.michels@oracle.com, Oracle, USA
Filip Murlak, f.murlak@uw.edu.pl, University of Warsaw, Poland
Stefan Plantikow, stefan.plantikow@neo4j.com, Neo4j, Germany
Petra Selmer, petra.selmer@neo4j.com, Neo4j, UK
Oskar van Rest, oskar.van.rest@oracle.com, Oracle, USA
Hannes Voigt, hannes.voigt@neo4j.com, Neo4j, Germany
Domagoj Vrgoč, domagojvrgoc@gmail.com, PUC Chile & IMFD, Chile
Mingxi Wu, mingxi.wu@tigergraph.com, TigerGraph, USA
Fred Zemke, fred.zemke@oracle.com, Oracle, USA

ABSTRACT

As graph databases become widespread, the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) have approved a project to create GQL, a standard property graph query language. This complements the SQL/PGQ project, which specifies how to define graph views over a SQL tabular schema, and to run read-only queries against them.

Both projects have been assigned to the ISO/IEC JTC1 SC32 working group for Database Languages, WG3, which continues to maintain and enhance SQL as a whole. This common responsibility helps enforce a policy that the identical core of both PGQ and GQL is a graph pattern matching sub-language, here termed GPML.

The WG3 design process is also analyzed by an academic working group, part of the Linked Data Benchmark Council (LDBC), whose task is to produce a formal semantics of these graph data languages, which complements their standard specifications.

This paper, written by members of WG3 and LDBC, presents the key elements of the GPML of SQL/PGQ and GQL in advance of the publication of these new standards.

ACM ISBN 978-1-4503-9249-5/22/06…$15.00
https://doi.org/10.1145/3514221.3526057

CCS CONCEPTS

• **Information systems → Query languages for non-relational engines**; Graph-based database models; • **General and reference** → *Computing standards, RFCs and guidelines.*

KEYWORDS

Graph database, Property graph, Query Language, Pattern matching, Standardization, GQL, SQL

ACM Reference Format:
Alin Deutsch, Nadime Francis, Alastair Green, Keith Hare, Bei Li, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Wim Martens, Jan Michels, Filip Murlak, Stefan Plantikow, Petra Selmer, Oskar van Rest, Hannes Voigt, Domagoj Vrgoč, Mingxi Wu, and Fred Zemke. 2022. Graph Pattern Matching in GQL and SQL/PGQ. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3514221.3526057

2246

Figure 4: [Angles et al., 2017]

Foundations of Modern Query Languages for Graph Databases

RENZO ANGLES, Universidad de Talca & Center for Semantic Web Research
MARCELO ARENAS, Pontificia Universidad Católica de Chile & Center for Semantic Web Research
PABLO BARCELÓ and AIDAN HOGAN, DCC, Universidad de Chile & Center for Semantic Web Research
JUAN REUTTER and DOMAGOJ VRGOČ, Pontificia Universidad Católica de Chile & Center for Semantic Web Research

We survey foundational features underlying modern graph query languages. We first discuss two popular graph data models: *edge-labelled graphs*, where nodes are connected by directed, labelled edges, and *property graphs*, where nodes and edges can further have attributes. Next we discuss the two most fundamental graph querying functionalities: *graph patterns* and *navigational expressions*. We start with graph patterns, in which a graph-structured query is matched against the data. Thereafter, we discuss navigational expressions, in which patterns can be matched recursively against the graph to navigate paths of arbitrary length; we give an overview of what kinds of expressions have been proposed and how they can be combined with graph patterns. We also discuss several semantics under which queries using the previous features can be evaluated, what effects the selection of features and semantics has on complexity, and offer examples of such features in three modern languages that are used to query graphs: SPARQL, Cypher, and Gremlin. We conclude by discussing the importance of formalisation for graph query languages; a summary of what is known about SPARQL, Cypher, and Gremlin in terms of expressivity and complexity; and an outline of possible future directions for the area.

CCS Concepts: • **Information systems** → **Query languages**; • **Theory of computation** → **Database query languages (principles)**;

Additional Key Words and Phrases: Property graphs, graph databases, query languages, graph patterns, navigation, aggregation

ACM Reference format:
Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. 2017. Foundations of Modern Query Languages for Graph Databases. *ACM Comput. Surv.* 50, 5, Article 68 (September 2017), 40 pages.
https://doi.org/10.1145/3104031

This work was funded by the Millennium Nucleus Center for Semantic Web Research under Grant No. NC120004.

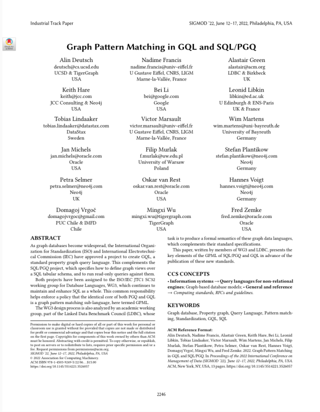
Authors' addresses: R. Angles, Universidad de Talca, Camino a los Niches Km. 1, Curicó, Chile; email: rangles@utalca.cl; M. Arenas, J. Reutter, and D. Vrgoč, Department of Computer Science, School of Engineering, Pontificia Universidad Católica de Chile, Vicuna Mackenna 4860, Edificio San Agustin, 4to piso, Macul 7820436, Santiago, Chile; emails: {marenas, jreutter, dvrgoc} @ing.puc.cl; P. Barceló and A. Hogan, Department of Computer Science, Universidad de Chile, Avenida Blanco Encalada 2120, Santiago, Chile; emails: pbarcelo@dcc.uchile.cl, ahogan@dcc.uchile.cl.

https://doi.org/10.1145/3104031

68

ACM Computing Surveys, Vol. 50, No. 5, Article 68. Publication date: September 2017.

Figure 5: [Deutsch et al., 2021]



Industrial Track Paper SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

Graph Pattern Matching in GQL and SQL/PGQ

Alin Deutsch, deutsch@cs.ucsd.edu, UCSD & TigerGraph, USA
Nadime Francis, nadime.francis@univ-eiffel.fr, U Gustave Eiffel, CNRS, LIGM, Marne-la-Vallée, France
Alastair Green, alastair@acm.org, LDBC & Birkbeck, UK
Keith Hare, keith@jcc.com, JCC Consulting & Neo4j, USA
Bei Li, bei@google.com, Google, USA
Leonid Libkin, libkin@ed.ac.uk, U Edinburgh & ENS-Paris, UK & France
Tobias Lindaaker, tobias.lindaaker@datastax.com, DataStax, Sweden
Victor Marsault, victor.marsault@univ-eiffel.fr, U Gustave Eiffel, CNRS, LIGM, Marne-la-Vallée, France
Wim Martens, wim.martens@uni-bayreuth.de, University of Bayreuth, Germany
Jan Michels, jan.michels@oracle.com, Oracle, USA
Filip Murlak, f.murlak@uw.edu.pl, University of Warsaw, Poland
Stefan Plantikow, stefan.plantikow@neo4j.com, Neo4j, Germany
Petra Selmer, petra.selmer@neo4j.com, Neo4j, UK
Oskar van Rest, oskar.van.rest@oracle.com, Oracle, USA
Hannes Voigt, hannes.voigt@neo4j.com, Neo4j, Germany
Domagoj Vrgoč, domagojvrgoc@gmail.com, PUC Chile & IMFD, Chile
Mingxi Wu, mingxi.wu@tigergraph.com, TigerGraph, USA
Fred Zemke, fred.zemke@oracle.com, Oracle, USA

ABSTRACT

As graph databases become widespread, the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) have approved a project to create GQL, a standard property graph query language. This complements the SQL/PGQ project, which specifies how to define graph views over a SQL tabular schema, and to run read-only queries against them.

Both projects have been assigned to the ISO/IEC JTC1 SC32 working group for Database Languages, WG3, which continues to maintain and enhance SQL as a whole. This common responsibility helps enforce a policy that the identical core of both PGQ and GQL is a graph pattern matching sub-language, here termed GPML.

The WG3 design process is also analyzed by an academic working group, part of the Linked Data Benchmark Council (LDBC), whose task is to produce a formal semantics of these graph data languages, which complements their standard specifications.

This paper, written by members of WG3 and LDBC, presents the key elements of the GPML of SQL/PGQ and GQL in advance of the publication of these new standards.

ACM ISBN 978-1-4503-9249-5/22/06…$15.00
https://doi.org/10.1145/3514221.3526057

CCS CONCEPTS

• **Information systems** → **Query languages for non-relational engines**; Graph-based database models; • **General and reference** → *Computing standards, RFCs and guidelines.*

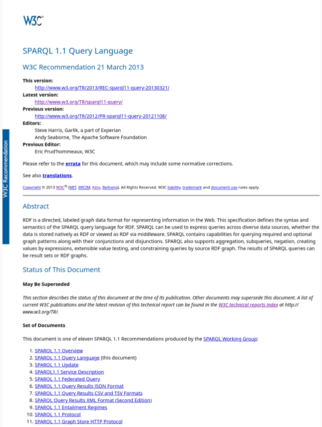
KEYWORDS

Graph database, Property graph, Query Language, Pattern matching, Standardization, GQL, SQL

ACM Reference Format:
Alin Deutsch, Nadime Francis, Alastair Green, Keith Hare, Bei Li, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Wim Martens, Jan Michels, Filip Murlak, Stefan Plantikow, Petra Selmer, Oskar van Rest, Hannes Voigt, Domagoj Vrgoč, Mingxi Wu, and Fred Zemke. 2022. Graph Pattern Matching in GQL and SQL/PGQ. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3514221.3526057

2246

Figure 6: [W3C, 2013]



SPARQL 1.1 Query Language

W3C Recommendation 21 March 2013

This version:
http://www.w3.org/TR/2013/REC-sparql11-query-20130321/
Latest version:
http://www.w3.org/TR/sparql11-query/
Previous version:
http://www.w3.org/TR/2012/PR-sparql11-query-20121108/

Editors:
Steve Harris, Garlik, a part of Experian
Andy Seaborne, The Apache Software Foundation
Previous Editor:
Eric Prud'hommeaux, W3C

Please refer to the **errata** for this document, which may include some normative corrections.

See also **translations**.

W3C Recommendation

Abstract

RDF is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports aggregation, subqueries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph. The results of SPARQL queries can be result sets or RDF graphs.

Status of This Document

May Be Superseded

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

Set of Documents

This document is one of eleven SPARQL 1.1 Recommendations produced by the SPARQL Working Group:

1. SPARQL 1.1 Overview
2. SPARQL 1.1 Query Language (this document)
3. SPARQL 1.1 Update
4. SPARQL1.1 Service Description
5. SPARQL 1.1 Federated Query
6. SPARQL 1.1 Query Results JSON Format
7. SPARQL 1.1 Query Results CSV and TSV Formats
8. SPARQL Query Results XML Format (Second Edition)
9. SPARQL 1.1 Entailment Regimes
10. SPARQL 1.1 Protocol
11. SPARQL 1.1 Graph Store HTTP Protocol

Figure 7: [Neo4j, 2025b]

Figure 8: [Neo4j, 2025a]

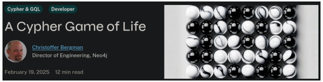


Figure 7: [Neo4j, 2025b]

Figure 8: [Neo4j, 2025a]

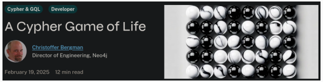


Figure 9: [Memgraph, 2025]

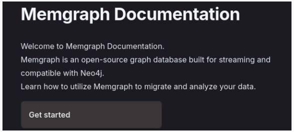


Figure 10: [Memgraph, 2024]

Figure 7: [Neo4j, 2025b]

Figure 9: [Memgraph, 2025]

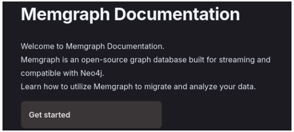


Figure 11: [kuzudb, 2023b]

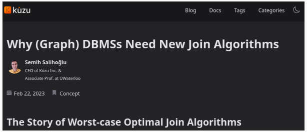


Figure 8: [Neo4j, 2025a]

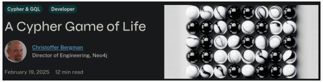
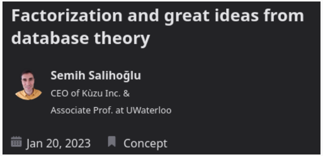


Figure 10: [Memgraph, 2024]

Figure 12: [kuzudb, 2023a]



Structure & Discussion Approach

Planned Paper Structure:

1. Introduction
2. Background on Data Models
3. Established Languages Overview
4. Upcoming Standards Languages
5. Parallelization & Hardware Acceleration
6. Optimization & Backends
7. Algorithm Integration
8. Synthesis: State-of-the-Art, Challenges, Future Directions

Conclusion & Questions

Recap: Surveying graph traversal languages, focusing on mappings to modern hardware for large-scale processing.

Goal: Provide a structured overview of the state-of-the-art, identifying key challenges and opportunities.

Conclusion & Questions

Recap: Surveying graph traversal languages, focusing on mappings to modern hardware for large-scale processing.

Goal: Provide a structured overview of the state-of-the-art, identifying key challenges and opportunities.

Thank you!

Conclusion & Questions

Recap: Surveying graph traversal languages, focusing on mappings to modern hardware for large-scale processing.

Goal: Provide a structured overview of the state-of-the-art, identifying key challenges and opportunities.

Thank you!

Questions?

References

[Angles et al., 2017] Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., and Vrgoč, D. (2017). **Foundations of modern query languages for graph databases.** *ACM Comput. Surv.*, 50(5).

[Commons, 2016] Commons, W. (2016). **Socialnetworkanalysis.** File: `SocialNetworkAnalysis.png`.

[Deutsch et al., 2021] Deutsch, A., Francis, N., Green, A., Hare, K. W., Li, B., Libkin, L., Lindaaker, T., Marsault, V., Martens, W., Michels, J., Murlak, F., Plantikow, S., Selmer, P., Voigt, H., van Rest, O., Vrgoc, D., Wu, M., and Zemke, F. (2021). **Graph pattern matching in GQL and SQL/PGQ.** *CoRR*, abs/2112.06217.

[kuzudb, 2023a] kuzudb (2023a). **Factorization and great ideas from database theory.**

[kuzudb, 2023b] kuzudb (2023b). **Why (graph) dbmss need new join algorithms.**

[Memgraph, 2024] Memgraph (2024). **How memgraph uses skip lists for fast indexes and unique constraints.**

References ...

[Memgraph, 2025] Memgraph (2025). **Documentation.**

[Neo4j, 2025a] Neo4j (2025a). **A cypher game of life.**

[Neo4j, 2025b] Neo4j (2025b). **Use neo4j parallel spark loader to improve large-scale ingestion jobs.**

[Vincenzo et al., 2014] Vincenzo, C., Di Lascio, R., Foggia, P., and Vento, M. (2014). **A semantic reasoner using attributed graphs based on intelligent fusion of security multi-sources information.** volume 8703.

[W3C, 2013] W3C (2013). **Sparql 1.1 query language.**