



FernUniversität Hagen

– Faculty of Mathematics and Computer
Science –

Different query languages for graph databases

Seminar „Discovering Big Data“

Chair of Databases and Information Systems

presented by

Daniel Langhann

Registration number: 3788687

Mentoring : Prof. Dr. Uta Störl

INTRODUCTION

1.1 RESEARCH TOPIC

Graph databases have developed into a powerful tool for modeling and querying complex relations in various data types. They are particularly suitable for dealing with linked data and complex network structures that essentially consist of nodes and their edges (relations). Examples of use are e.g [Yua22]:

- Relationships in social networks
- Recommendation applications
- Biological networks

In this paper, the topic of graph databases and selected query languages designed for graph databases will be presented and contextualized. In particular, the path from different query languages to a uniform ISO standard is pointed out.

1.2 MOTIVATION

The aim of this paper is to provide an overview of the different languages and to highlight the respective advantages and disadvantages, their capabilities, and limitations, as well as their specific use cases.

1.3 STATE OF RESEARCH

Graph databases are a comparatively young technology in the field of database systems. Over time, various query languages have been developed with specific focuses and corresponding strengths and weaknesses. A common ISO standard should help to keep the technology and its application more generic [Har24]. In this paper, important proprietary languages are presented and differentiated from each other. In the further course, a overview of the different languages is created.

1.4 APPROACH

Basically, relevant documentation on the various query languages is analyzed and the key points are highlighted. To support this, selected languages are tried out in a test project and the corresponding learnings are integrated into the work.

CRITERIA FOR COMPARISON

First, comparative criteria are introduced against which the different languages are compared.

The evaluation is done purely qualitatively as this is a complex topic that is difficult to evaluate on the basis of a uniform scale. Nevertheless, the aim is to evaluate the languages as objectively and reproducibly as possible on the basis of the defined criteria.

2.1 EXPRESSIVENESS

Expressiveness in the context of query languages for graph databases means the ability of a language to allow a wide range of queries on graphs and their underlying databases. An expressive language can be used with

- complex patterns
- relationships

even across various different graphs. In addition, an expressive language offers users the corresponding flexibility when querying graph structures.

2.2 EASE OF USE

For users, it is important that the corresponding query language is intuitive to understand, follows familiar and learned patterns and is as close as possible to the de facto industry standard. On the other hand, a less intuitive language means an increased learning curve, which makes the technology more difficult to use.

2.3 PERFORMANCE AND PERFORMANCE OPTIMIZATION

Performance is a decisive factor when dealing with the analysis of large amounts of data. Especially when it comes to evaluations that need to be carried out in real time or very quickly. A recommendation system does not help anyone if it does not deliver an answer in a reasonable time. This criterion evaluates in particular the ability of the corresponding language to enable techniques that improve the performance of queries on graph databases. Particular attention is paid to the following two factors:

- Implementation of index-based structures
- Parallel processing

* Implementation of index-based structures * Parallel processing

2.4 INTEROPERABILITY

How well a query language integrates with other tools, techniques and API's is an important criteria for the decision for and against the query language and often for the entire database system.

2.5 CLOSENESS TO THE ISO STANDARD

In particular, attention is paid to the extent to which the respective language is oriented towards the ISO standard for query languages for graph databases. A language that is strongly oriented towards the ISO standard or whose model and structure has a high proportion of the ISO standard automatically ensures, that the user has to acquire a low level of proprietary knowledge, which is beneficial to the user-friendliness of the language. In addition, a strong orientation towards the ISO standard ensures interoperability with other technologies and systems, as it can be assumed that these are also oriented towards at least the basic concepts of the ISO standard.

In the next chapter the ISO-standard for Graph databases will be introduced to enable the reader to compare the corresponding language with the ISO-standard when reading chapter XXX.

THE ISO-STANDARD FOR QUERY LANGUAGES FOR GRAPH DATABASES

3.1 OVERVIEW OF THE GQL-STANDARD

In short, GQL is a new standard for a property graph database language developed by international standards committee. GQL is an acronym for Graph Query Language. It is the first time in more than 35 years, that the ISO released a new standard for database query language.

In opposite to SQL, where data is organized in tables, graph databases structure data in graphs. This enables new ways to analyze and recognize patterns in very large amounts of data without having specific knowledge about the data itself. Various use cases for graph databases have already been mentioned in chapter xxx.

3.2 SCOPE OF THE GQL-STANDARD

As already mentioned graph databases store and retrieve nodes (vertexes) and edges between nodes (relationships). GQL is a declarative language influenced both by existing property graph database products and by the SQL standard. The GQL standard is a complete database language that supports

- creating
- reading
- updating
- deleting
- modifying

property graph data.

Graph data can be organized in two different ways:

- Schema free or
- constrained by an database administrator.

Schema free graph data has no restrictions for adding and changing graph data. Graph data that is subject to a predefined schema must fulfill this at all times, otherwise an error is triggered.

The schema of graph data is specified via so-called graph types, which on the one hand specify the structure of the nodes as well as that of the edges, i.e. the relationships between the nodes.

3.3 EXAMPLES OF COMMON GQL QUERY OPERATIONS

This chapter provides a brief introduction to GQL capabilities including

- Queries and Graph Pattern Matching
- Add, modify and delete operations
- Transactions

3.3.1 *Queries and Graph Pattern Matching*

GQL queries are based on rich Graph Pattern Matching (GPM) language. The below example finds all nodes with a one-hop relationship to a node with a productId of `6594301c4aa9b3232889e7c3`

```
{
  MATCH (a {productId: "6594301c4aa9b3232889e7c3"}) - [b] -> (c)
  RETURN a, b, c
}
```

The GQL-standard does not define how the returned data has to be displayed to the user. Two possible ways can be:

- Graph Data Visualization tools
- Textoutput

Here is suggestion how the above listing TODO can be displayed: This below table shows how the output can be displayed as text
TODO table erstellen

3.4 SCHEMA FREE VS GRAPH TYPES

A schema free graph accepts any form of graph data which makes it relatively fast to use but on the other hand gives the user control over the data and therefore a certain amount of data proliferation is accepted and has to be managed.

A graph type is a kind of template that specifies the structure of the graph, which must be adhered to at all times. The structure of the nodes as well as the edges or both can be specified. The following example illustrates the concept of a graph type

```
{
  CREATE GRAPH TYPE /folder
  (client: Client => {cid::STRING, cname::STRING}),
  (agent:Agent => {no::STRING, office::STRING}),
  (client)-[:SUPERVISED_BY]->(agent)
}
```

LITERATUR

- [Har24] Keith W. Hare. *ISO/IEC 39075:2024 Information Technology – Database languages*. Apr. 2024. URL: <https://jtc1info.org/slug/gql-database-language/> (besucht am 28.04.2024).
- [Yua22] Tian Yuanyuan. *The World of Graph Databases from An Industry Perspective*. Nov. 2022. URL: <https://arxiv.org/pdf/2211.13170> (besucht am 28.04.2024).