# NoSQL Databases

Daniel Rutz and Leon Schürmann (Eds.)

March 23, 2019

# Contents

# 1 GraphDB

Thore Krüss, Lennart Purucker, Johanna Sommer

## 1.1 Introduction

### 1.1.1 Motivation

First paragraph to state the general context and makes important points for the motivation and why its important.

But Graph Database research has been initiated already in the early 90s. During this time, a number of suggestions came up, trying to propose a semantic network to store data about the database. That was, because contemporary systems were failing to take into account the semantics of a database. The Locial Data Model [2] was proposed, where they tried to combine the advantages of relational, hierarchical and network approaches in that they modeled databases as directed graphs, in which leaves represent attributes and internal nodes as connections between the data. Similar to that the Functional Data Model [**?** ] was proposed with the same goal, focusing specifically on providing a conceptually natural database interface [1]. This this, most of the underlying theory of Graph Databases was created. It was most likely because of insufficient hardware support for big graphs that this research declined, only to be picked up again now, which is focused on actual practical systems because we have the hardware now but also on theoretical analysis of graph query languages. [1]

Storing, retrieving, and manipulating such complex data becomes onerous when using traditional RDBMS approaches. Schema based data models by their very definition put in place limits on how information will be stored. There is an involved manual process to redesign the schema in order to adapt to new data. Where the RDBMS is optimized for aggregated data, graph databases such as Neo4j are optimized for highly connected data.

GRAPHS ARE EVERYWHERE Graphs are extremely useful in understanding a wide diversity of datasets in fields such as science, government, and business. The real world—unlike the forms-based model behind the relational database—is rich and interrelated: uniform and rule-bound in parts, exceptional and irregular in others. Once we understand graphs, we begin to see them in all sorts of places. Gartner, for example, identifies five graphs in the world of business—social, intent, consumption, interest, and mobile—and says that the ability to leverage these graphs provides a “sustainable competitive advantage.” - Buch Graph Databases

Graph databases really shine when working in areas where information about data interconnectivity or topology is important. In such applications the relations between data and the data itself are usually at the same level [1]. Many companies have developed in-house implementations in order to cope with the need of graph database systems. Examples would be Facebook’s Open Graph, Google’s Knowledge Graph, Twitter’s FlockDB, any many more. The following are a few examples of systems that would benefit greatly from graph database approach. RDBMS can be used for such systems but in a much more limiting and expensive way (expensive meaning processing power caused by recursive JOINs for friend of a friend type problems). - Social Graph - Recommender Systems - Bioinformatik

## 1.2 Graph Database Theory

### 1.2.1 Description of data model and functionality

### 1.2.2 fields of application

### 1.2.3 CAP Theorem

### 1.2.4 GraphDB vs. RDBMS

## 1.3 Implementation with Neo4j

### 1.3.1 Use Case from the SQL world

### 1.3.2 Installation

### 1.3.3 modelling

### 1.3.4 usage, query language

### 1.3.5 short conclusion, summary

## 1.4 Reflection

### 1.4.1 alternative popular graphdbs

### 1.4.2 conclusion

reflect on advantages disadvantages with implementation references

# Bibliography

[1] Renzo Angles and Claudio Gutierrez. An introduction to graph data management. In *Graph Data Management*, 2018.

[2] Gabriel Kuper. The logical data model : a new approach to database logic /. 09 1985.