Comparing/Contrasting Data Models

by Chenuri Gunaratne (V00938807)

Introduction

In this report I will be comparing three different data models according to the language that they use, queryability, structure, queryability, the importance of normalisation and when best to use each model.

The three models under consideration are:

- Relational Data Model
- Labelled Property Graph Model (LPG)
- Key Value Map

Query Language

The relational data model uses Structured Query Language (SQL). LPG makes use of the language Cypher. The key value map on the other hand does not have any query languages but it does make use of three operations put, get, and delete.

Structure

LMU Build (n.d.) in their article titled "The Relational Data Model: Structure" writes that each table in a relational data model has its own name. Each table is made up of several rows. Each row is a collection of values that are, by definition, related to one another in some way; these values correspond to the table's attributes or columns. Each table attribute defines a set of allowed values for that attribute; this set of allowed values is the domain of that attribute.

A labelled property graph (LPG) is a type of graph database. Oxford Semantic Technologies (n.d.) claims that the graph in LPG-style databases consists of nodes and relationships. Each node or relationship has an ID tag, one or more "labels" that describe its type or class, and a set of "properties" that present values and a corresponding key to allow referencing. Intuitively, two nodes are always joined by a relationship to form the larger structure of the graph.

According to the article “What Is a Key-Value Database?” by AWS (n.d.) a key-value database is a type of nonrelational database that uses a simple key/value method to store data. A key serves as a unique identifier, while the value can be anything. Both keys and values can be objects (simple or complex), or objects made up of other objects.

Although the structures of these three models are very different to one another the similarity I see is that they all have unique identifiers. Primary key in a relational data model, ID tag in an LPG and the keys in key-value databases that are all present for the ease of reference.

Queryability & Normalisation

Normalisation is a sort of transformation that improves the quality of the data by eliminating the possibility of anomalies. However, normalized data makes it hard to capture queryability of data, which is why users end up having to choose between these two.

The relational data model is preferred to be BCNF normalised as normalisation is more important than queryability. According to Chapple (2020) this is done to ensure that data dependencies make sense and to remove redundant data. When identical data is not saved across multiple tables and only related data is kept in a table, a database is said to be normalised.

Normalisation is also important in an LPG because graph databases, like relational databases, are built on the concept of relations between entities (tables in relational, vertexes in graph). As a result, all normalisation techniques that work for relational databases also work for property graph databases. (A., 2018)

Within the key value model. Queryability triumphs over normalisation. If requirements are read only, then everything can be denormalized. However, if the requirements require updating, we can normalise the data to some extent. When attempting to prioritise queryability or normalised data, we should ask ourselves this question. What data will be accessed together when we run our queries? It makes sense to combine these data together.

When to Use (Nature of Requirements)

Despot (2021) believes that in an LPG, the connections within the data should be the primary focus. It would not work if the data was transactional in nature. If your data model is unpredictable and requires frequent changes, a graph database may be the best option. Because graph databases provide some

flexibility. This is also a good model if you plan to retrieve data frequently. If you do not intend to use JOIN operations in your queries, a graph is not a must-have.

Mendis (2017) in his web article on “From RDBMS to Key-Value Store: Data Modeling Techniques” conveys if a large number of connections are required to map them to users, then a graph database may provide performance benefits; however, if the majority of your queries return a single node via a simple identifier (key), then graph databases should be avoided. Instead, use key value stores.

Mendis (2017) further states that if the data is more performance-oriented and the data model is not hierarchical then using key value stores, data can be modelled in a less complicated way.

Despot (2021) also mentions that on the other hand, there are often advantages to having a predefined and consistent table that is simple to understand. The fact that developers are accustomed to and comfortable with relational databases, cannot be downplayed. In her article about the key differences between graph databases and relational databases Joyce (2021) writes that according to IBM, relational databases are the most widely used query tool in business. Because of their tabular structure, they are ideal for records with simple relationships. Relational databases, for example, are popular for accounting and other transactional data because simple data with established relationships fits easily into the relational database format.

References

Arad. (2018, December 2). *Graph Data Modeling and Normalization*. Retrieved from LinkedIn: https://www.linkedin.com/pulse/graph-data-modeling-normalization-ayad-aliomer/

AWS. (n.d.). *The key-value database defined*. Retrieved December 22, 2022, from AWS: https://aws.amazon.com/nosql/key-value/#:~:text=A%20key%2Dvalue%20database%20is,objects%20to%20complex%20compound%20objects

Chapple, M. (2020, July 12). *What Is Boyce-Codd Normal Form (BCNF)?* Retrieved from Lifewire: https://www.lifewire.com/boyce-codd-normal-form-bcnf-1019245

Despot, I. (2021, March 4). *Graph Database vs Relational Database*. Retrieved from Memgraph: https://memgraph.com/blog/graph-database-vs-relational-database

Joyce, K. E. (2021, Jauary 15). *Graph database vs. relational database: Key differences*. Retrieved from TechTarget: https://www.techtarget.com/searchdatamanagement/feature/Graph-database-vs-relational-database-Key-differences#:~:text=The%20most%20notable%20difference%20between,data%20tables%2C%20not%20data%20points

LMU Build. (n.d.). *The Relational Data Model: Structure*. Retrieved December 22, 2022, from https://dondi.lmu.build/share/db/relational1.pdf

Mendis, W. (2017, October 29). *From RDBMS to Key-Value Store: Data Modeling Techniques*. Retrieved from Medium: https://medium.com/@wishmithasmendis/from-rdbms-to-key-value-store-data-modeling-techniques-a2874906bc46

Oxford Semantic Technologies. (n.d.). *What is a labeled property graph?* Retrieved December 22, 2022, from Oxford Semantic Technologies: https://www.oxfordsemantic.tech/fundamentals/what-is-a-labeled-property-graph