> *Bases de Datos 2, Resumen 7 y 8*
> *Miguel Ku Liang - 2019061913*

Use Cases for Graph Technology

When your data is complex, the solution is in the graph. **Neo4j** helps the world make sense of data by identifying connections and plotting relationships otherwise unseen amidst never-ending strams of data.

1. Fraud Detection

Storing voluminous transaction detail in a graph database captures connections that already exist. The data is then ripe for graph-native machine learning techniques that surface fraud.

2. Real-Time Recommendations

To be relevant and prevent a potential customer from clicking to a competitor, recommendations must be contextual and instantaneous. Running in Neo4j, Bechtle’s recommendation system generates a detailed model of all the actions customers take on the website.

3. Bill of Materials

Tracking every component and its cost, what equipment the components relate to, and the expected product lifespan or average time to failure into a bill of materials is a behemoth operation for an organization.

4. Track & Trace

Graph data models enable traceability for a variety of industries and use cases. Track and trace enables you to find out the status of a product.

5. Network & IT Ops

Graph databases are a natural fit for network and IT ops. The graph's connected structure enables network managers to conduct sophisticated impact analyses. A simple graph query determines which applications, services, and customers will be affected if a particular network element fails.

Digital transformation requires modernizing the IT landscape and demands crystal clear visibility into current systems.

The Return on Connected Data

Connections Unlock the Value of Data

The value of data comes from an organization’s ability to understand it in relation to other data. Increasing data's connectedness further increases its value through additional context. Because relational databases don't persist relationship information in storage or any other stage of their analytic exercises, finding connections requires an enormous amount of processing.

Data, Data Everywhere, Nor any Drop to Drink

Organizations are surrounded by data, but they’re limited in their ability to do anything with it. They are failing to realize big data’s value because the value isn't in disparate data, but in the relationships between the data.

Data becomes connected when you treat relationship information as a first-class entity.

The Power of Connected Data

The biggest benefit of connected data is the ability to provide a connected view of data to the analytic and operational applications. The connections can be made available or revealed to applications or business users to make operational decisions.

You obtain context that allows you to more deeply or better refine the pieces of information collected or the recommendations produced. The more understanding you have of the connectedness or relationships between data, the better and more refined the system is downstream.

You can better understand the flow of money to detect fraud and money laundering and assess the risk of a network outage across computer networks. Connected data is most powerful when it provides operational, real-time insights. It allows business users and applications to make business decisions and act in real-time.

Quench Your Thirst for Insights with Connected Data

A graph database makes it easy to express and persist relationships across many types of data elements. A graph database is a highly scalable transactional and analytic database that stores data relationships as first-class entities.

Graph databases are simple and agile due to their schema-optional nature. Graphs can be easily changed or updated. A graph database can address the challenge of migrating all of your data into one connected place.

You can get more value out of your big data technology by using graphs to transfer knowledge of what the organization has done across other departments. By connecting data, graphs enable you to leverage all of the data you’ve dumped into the data lake.

Neo4j: Your Path to Connected Data

Neo4j offers a native graph platform that reveals and maintains the integrity of connected data. It offers this connections-first approach as a new paradigm to reveal and utilize relationships among data in order to help organizations along their path to new generations of applications and analytic advancements. It makes connected data easy.

A Brief Introduction to Graph Data Platforms

Introduction

Relational databases' structure fails to adapt easily to the complexity of data, its context and its interconnections. Real-time recommendations rely on connected data. This would require linking many tables.

What Are Graph Databases Good for?

Graph database technology is specifically designed and optimized for highly interconnected datasets to identify patterns and hidden connections. A graph database efficiently stores and queries connected data in a node-and-relationships format.

The most common graph use cases and solutions include:

- Fraud Detection & Analytics
- Artificial Intelligence & Machine Learning
- Real-Time Recommendation Engines
- Knowledge Graphs
- Network & Database Infrastructure Monitoring
- Master Data Management

Their success requires solving complex problems with dynamic and interconnected datasets. Graph databases tackle the most harrowing of data problems, including:

- Different views of the data model between business and technology teams
- Lack of schema flexibility and adaptability
- The "JOIN problem"

Traditional Technology Choices Do Not Consider How Data Is Interrelated

Relational databases are not well-suited for modeling and storing today’s highly connected and agile datasets. Traditional RDBMS technology has a difficult time expressing and revealing how real and virtual entities are related.

Collections vs. Connections

SQL & NoSQL Systems Focus on Data Aggregation & Collection

Collection-centric storage designs as implemented by SQL and Not only SQL databases are designed to efficiently divide and store data.

In SQL’s case, the normalization of data into a tabular schema aims to minimize storage of duplicate data objects, types and values. The RDBMS design achieved this normalized goal by linking tables of data via foreign keys to their associated records from other tables.

NoSQL systems carry those concepts forward by simplifying their models in exchange for higher levels of scale and simplicity. A lack of concern about relationships leads to looser data guarantees, plain APIs and straightforward scaling schemes. These systems take on the “store and retrieve” problem at scale for simple data.

Graph Systems Focus on Data Connections

Graph database technologies focus on how data elements are interrelated and contextualized as connected data. Connected data is the materialization and harnessing of relationships between data elements, modeled as a property graph. A property graph is a data model designed to express data connectedness as nodes connected via relationships to other nodes.

In the graph model, data relationships are persisted so they can be navigated or traversed along connected paths to gain context. Relationships are both typed and directional.

Property Graphs Are Intentionally Simple

- You can draw property graphs on whiteboards and map that design directly into a graph database
- You can change or update a property graph easily
- You can quickly program property graphs
- You can visualize and navigate property graphs efficiently
- You can rapidly determine data context when property graph queries are executed in hyper-fast native graph platforms

Benefits of Graph Databases

- Simple and natural data modeling
- Flexibility for evolving data structures
- Simultaneous support for real-time updates and queries
- Better, faster and more powerful querying and analytics