> Bases de Datos 2, Resumen 5
> Miguel Ku Liang - 2019061913

Chapter 1: Graphs are the future

Why You Should Care about Graph Databases

- Performance: stays consistent even when data grows
- Flexibility: you can add to the existing structure without endangering the current functionality
- Agility: allows applications to evolve with the changing requirements

How Graph Databases Work

- Graph Storage: some use native storages designed to store and manage graphs, others use relational or object-oriented databases
- Graph Processing Engine: native graph processing is the most efficient means of processing data in a graph. Non-native graph processing engines use other means to process CRUD operations.

Chapter 2: Why Data Relationships Matter

The Irony of Relational Databases

Relational databases aren't effective at handling data relationships. Their schema is too inflexible. As data grows, relational databases become burdened with large JOIN tables that affect its performance.

Why Other NoSQL Databases Don't Fix the Problem Either

Their disconnected construction makes it harder to harness data relationships properly.

Graph databases store data relationships as relationships, meaning fewer disconnects between the schema and the database. The flexibility of a graph model allows you to add nodes and relationships without affecting the existing network or expensively migrating the data.

Chapter 3: Data Modeling Basics

What is Modeling Exactly?

Data modeling is an abstraction process. Obtain business and user needs and map those needs in a structure for storing and organizing the data.

A Relational Vs. Graph Data Modeling Match-Up

In a relational data modeling, having foreign keys adds complexity and having JOIN tables slow down the queries' speed. In a graph data modeling, the model is enriched with the business and user needs.

Why Data Modeling Isn't a One-Off Activity

Over time, applications change, data models evolves to meet new business and user needs. Relational databases aren't good for rapid change. Graph databases adapt your data models to the changing needs.

Chapter 4: Data Modeling Pitfalls to Avoid

Example Data Model: Fraud Detection in Email Communications

Fraud detection application that analyzes users' email communications. Looking for frequently using blind-copying and using aliases to conduct fake conversations that mimic legitimate interactions. Using a graph data model to capture all the elements and activities.

The email isn't represented in the data model. Adding properties to the EMAILED relationship won't allow to correlate connections between EMAILED, CC and BCC relationships, which are needed for the solution.

The Fix: A Stronger Fraud Detection Data Model

Add nodes to the graph model that represent the emails exchanged with their relationships, to track who wrote and to whom the email was sent.

The Next Step: Tracking Email Replies

Adding FORWARDED and REPLIED_TO relationships to the graph model will create the same problem of the EMAILED relationship.

A reply can be represented as a Email and Reply node. Also, using TO, CC and BCC relationships to map whether the reply was sent to the original sender, all recipients or a subset of recipients. REPLY_TO relationship to reference the original email.