Fundamentals of Database Systems

[Graph and Browser Databases]

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

November, 2019

1. Graph Databases
   - Basics
   - Property Graph Model
   - Neo4j

2. Browser Databases
   - Basics
   - PouchDB
   - IndexedDB

Basics

Graph databases use graph structures for semantic queries with nodes, edges, and properties to represent and store data.

Basics

Graph databases use graph structures for semantic queries with nodes, edges, and properties to represent and store data.

Graph databases are part of the NoSQL databases created to address the limitations of the existing relational databases.

Basics

Graph databases use graph structures for semantic queries with nodes, edges, and properties to represent and store data.

Graph databases are part of the NoSQL databases created to address the limitations of the existing relational databases.

Querying relationships within a graph database is fast because they are perpetually stored within the database itself.

Property Graph Model

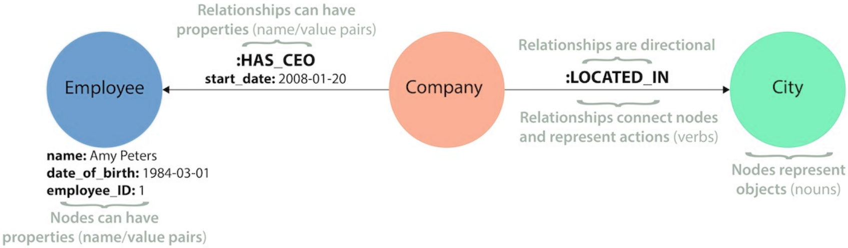
Property Graph Model is a key concept in Graph Databases. In this model, data is organized as nodes, relationships, and properties (data stored on the nodes or relationships).

Property Graph Model

Property Graph Model is a key concept in Graph Databases. In this model, data is organized as nodes, relationships, and properties (data stored on the nodes or relationships).

- `Nodes` are the entities in the graph. They can hold any number of attributes (key-value pairs) called properties. Nodes can be tagged with labels, representing their different roles in your domain. Node labels may also serve to attach metadata (such as index or constraint information) to certain nodes.
- `Relationships` provide directed, named, semantically-relevant connections between a pair of node entities. A relationship always has a direction, a type, a start node, and an end node. Like nodes, relationships can also have properties. In most cases, relationships have quantitative properties (e.g., weights, costs, ratings, time intervals, etc.).

Property Graph Model – An example


Relationships can have
properties (name/value pairs)
:HAS_CEO
start_date: 2008-01-20
Employee
Company
Relationships are directional
:LOCATED_IN
City
Relationships connect nodes
and represent actions (verbs)
name: Amy Peters
date_of_birth: 1984-03-01
employee_ID: 1
Nodes can have
properties (name/value pairs)
Nodes represent
objects (nouns)

Neo4j

Neo4j is an open-source, NoSQL based, native graph database that provides an ACID-compliant transactional back-end for various applications.

Neo4j efficiently implements the Property Graph Model down to the physical level (i.e., the data is stored exactly as you connect it), and the database uses pointers to navigate and traverse the graph.

Basics

Browser databases enable to persistently store data inside a user's browser.

Basics

Browser databases enable to persistently store data inside a user's browser.

Browser databases do not require persistent internet connectivity to work and hence applications build upon such databases can work both online and offline.

PouchDB

PouchDB is an open-source JavaScript database inspired by Apache CouchDB that is designed to run well within the browser.

PouchDB

PouchDB is an open-source JavaScript database inspired by Apache CouchDB that is designed to run well within the browser.

PouchDB was created to help web developers build applications that work as well offline as they do online.

PouchDB

PouchDB is an open-source JavaScript database inspired by Apache CouchDB that is designed to run well within the browser.

PouchDB was created to help web developers build applications that work as well offline as they do online.

It enables applications to store data locally while offline, then synchronize it with CouchDB and compatible servers when the application is back online, keeping the user's data in sync no matter where they next login.

IndexedDB

Unlike server-side databases, IndexedDB is a client-side database that stores data in the browser. Hence, the developers do not have direct access to it. But when a new version of an application is published, it is required to update the database.

IndexedDB

Unlike server-side databases, IndexedDB is a client-side database that stores data in the browser. Hence, the developers do not have direct access to it. But when a new version of an application is published, it is required to update the database.

IndexedDB has a built-in mechanism of *Schema Versioning*, absent in server-side databases.

IndexedDB

Unlike server-side databases, IndexedDB is a client-side database that stores data in the browser. Hence, the developers do not have direct access to it. But when a new version of an application is published, it is required to update the database.

IndexedDB has a built-in mechanism of *Schema Versioning*, absent in server-side databases.

Note: IndexedDB stores and retrieves objects that are indexed with a “key”.

Characteristics of IndexedDB

- IndexedDB databases store key-value pairs.
- IndexedDB is built on a transactional database model.
- The IndexedDB API is mostly asynchronous.
- IndexedDB uses a lot of requests.
- IndexedDB uses DOM events to notify you when results are available.
- IndexedDB is object-oriented.
- IndexedDB does not use Structured Query Language (SQL).
- IndexedDB adheres to a same-origin policy.