

Neo4j modeling : better keep data unique and linked or duplicate it ?

3 posts by 2 authors 



Gaël N.

Apr 16



Hi dear community,

I was wondering if there were some good practices about how you modelize your graph when you have different kind of nodes (so different labels) that have properties in common.

Let me explain with an example :

Let's say I have a node containing localization data (city, country, street, zipcode...) which is related to a "person" node.

I also have an other node, a "company", which have "countries" as a property.

I can see many kind of modelization :

1. (person)-[:IS_LOCATED_IN]->(location {city:"", country:"france", zipcode:""}), (company {countries: : ["france", "USA"]})
2. (person)-[:IS_LOCATED_IN]->(location {city:"", country:"france", zipcode:""}), (company)-[whatever]->(country {name:"france"})
3. (person)-[:IS_LOCATED_IN]->(location {city:"", zipcode:""})-[whatever2]->(country {name:"france"})<- [whatever]-(company)

You will probably say that it depends on my needs, but I was just wondering if there was a rule like "When you have data in common, you'd better extract it to an other node and link them all to have a highly connected graph".

Considering my needs, my application is not specifically focused on localization, it is just some datas that could be share between nodes by extracting them to node and create relations.

This question came to my mind after seeing some "time representation model" like the GraphAware TimeTree.

- Is there a best modelization between the 3 I suggest ?
- Does it only depends on what will be the queries ?
- What about global Neo4j performances and disk(or memory) usage between those 3 ?
- Do you have a better modelization ? :)

Thank you guyz



Michael Hunger

Apr 16



I think it depends on the use-case

if your use-cases and queries make use of that structure to provide some functionality that is hard to achieve otherwise, then go for the graph structure.

If it is just for the storage and some property lookup or comparison it might not be worth it.

There is no "general rule" :) Some types of nodes e.g. Countries can be connected to **a lot** of other nodes, so querying across them might not be as efficient, then you'd rather check from the other direction, if something is **also** connected to the same country.

The data modeling training has some infos on that, as well as Ian Robinsons talks (recorded on watch.neo4j.org) on data modeling

See also: <http://neo4j.com/developer/guide-data-modeling>

HTH

Cheers, Michael

- show quoted text -
- show quoted text -

--

You received this message because you are subscribed to the Google Groups "Neo4j" group. To unsubscribe from this group and stop receiving emails from it, send an email to [neo4j+un...@googlegroups.com](mailto:neo4j+unsubscribe@googlegroups.com). For more options, visit <https://groups.google.com/d/optout>.



Gaël N.

Apr 22



Thank you for your answer, I will look at the guide you linked :)