

Ten. Million. Questions. Let's celebrate all we've done together.

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free.

Take the 2-minute tour

## Graph database schema design - Is this suitable for neo4j?

## Stack Exchange

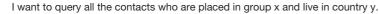
d join us on Facebook



Scenario: a simple address book where a user can create his own contacts and organize them by adding them in groups. A contact may have multiple addresses.



I have created the following diagram: ![schema-design][1]



Is this schema design good enough for those purposes (I want to use the neo4j database)?

database neo4j database-schema graph-databases schema-design share improve this question edited Apr 2 at 21:47

asked Apr 1 at 7:50 Moody **122 •** 9

active

oldest

votes

add a comment

## 2 Answers



It looks like the notion of country should be a first class citizen in your graph since your query depends on it. Graph model design typically gets influenced a lot by your query patterns.



So I suggest to have a node labeled Country for each country and connect the Address node with :LOCATED\_IN relationships to the country. (consequently drop the country property from the address nodes).



With that change your query is as easy as:

MATCH (:Group{name:'family'})<-[:placed\_in\_group]-(contact)-[:lives-at]->()-[:LOCATE RETURN contact

share improve this answer

answered Apr 1 at 8:14 Stefan Armbruster 23.1k • 2 • 28 • 51

Thank you! I appreciate it. - Moody Apr 1 at 8:30

By the way, Stefan, do you by any chance have any comparison of statistics concerning how much do dense nodes affect performance in cases like this? Say I have ~20K nodes with the Prop property (maybe indexed on it as well) and I'm querying the DB just to find all the nodes which have the Prop and then filter on the concrete Prop value (the amount of possible Prop values is rather small: I can enumerate them all (say 1000 possible values)). Would it be a better design if I create a label Prop and create 20K relationships to it? What would be a better choice performance-wise? - cdshines Apr 2 at 12:33

you need to be aware that you might run into locking issues if you connect concurrently a lot of addresses to the same country. Creating a relationship means locking its start and end node as well. However you can easily work around that by having couple of "docking nodes" around the country. The address then selects one docking node e.g. by consistent hashing on the thread id - this prevents locking issues. On the query side I don't have measurements available. - Stefan Armbruster Apr 2 at 12:49

add a comment

asked 4 months ago

viewed 77 times

active 4 months ago

Blog

Why Stack Overflow is a Good Workplace for Women



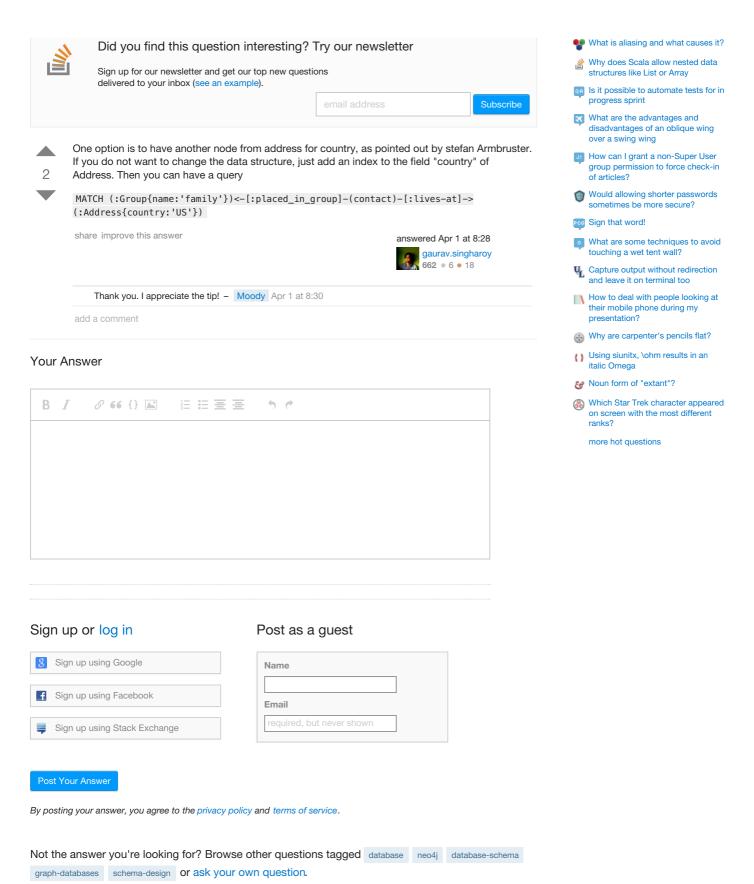
Sign up for the newsletter

see an example newsletter

## Related

- 101 Has anyone used Graphbased Databases (http://neo4j.org/)?
- 4 Good graph database for finding intersections (Neo4j? Pegasus? Allegro?
- 12 Storing multiple graphs in Neo4J
- 2 Neo4j graph database helping to store several time based data values, result of simulation
- 9 Graph Database Design Methodologies
- 1 Building a many-to-many db schema using only an unpredictable number of foreign keys
- 1 which role-permission database design?
- 0 Better ways to model this simple schema design (MongoDB)?
- 0 Schema design simple address book Hbase/Cassandra
- 1 Cassandra (CQL) schema/tables look the same as RDBMS for my scenario

**Hot Network Questions** 



a question feed



Web Applications Ask Ubuntu Webmasters Game Development TeX - LaTeX	WordPress Development Geographic Information Systems Electrical Engineering Android Enthusiasts Information Security	SharePoint User Experience Mathematica Salesforce ExpressionEngine® Answers more (13)	Graphic Design Movies & TV Seasoned Advice (cooking) Home Improvement Personal Finance & Money Academia more (9)	Mi Yodeya (Judaism) Travel Christianity Arqade (gaming) Bicycles Role-playing Games more (21)	Science Physics MathOverflow Chemistry Biology more (5)	Stack Overflow Careers
---	--	---	--	---	---	------------------------

site design / logo © 2015 Stack Exchange Inc; user contributions licensed under <u>cc by-sa 3.0</u> with <u>attribution required</u>