Implementation of 'beads-style' connections in Neo4j

asked 4 months ago
viewed 22 times
active 4 months ago

0

I am new to Neo4j, but it looks like relationships are only possible between their start and end nodes there without any other nodes in the middle. In other words, I cannot create a relationship 'from A to B via C', only 'A to C, C to B' instead.

Now say we have the following node types (or labels in Neo4j terms): buyer, product, measuring units. How to build the graph if the use case is 'give me measuring units this buyer used when buying that product'?

Naturally the relationships go like this:

```
(:buyer)-[:ordered]->(:product)-[:measured]->(:unit)
```

But in this model we can't tell which of the units the product was ever measured in (by all buyers) were actually used by the buyer we are interested in. We can only trace buyer to product and then when we trace product to units we 'forget' the buyer and can only get all the units used by every buyer ever ordered that product.

The way I implement that now is that my `:ordered` relationships going from buyer to product carry a property 'unit', so I can do queries like:

```
MATCH (b:buyer)-[o:ordered]->(p:product) WHERE b.id = 123 AND p.id = 345 RETURN DISTINCT o.unit
```

This, however looks very ugly because logically units should be nodes (and them being nodes comes convenient in other use cases). Here, instead, we have to duplicate them as relationship properties because there is no way to tell 'this relationship starts at the buyer node, goes through product node and without interrupting ends at measuring unit node'.

My question is if there are nicer ways to design a graph for similar queries, and if maybe that level of data duplication is not considered ugly when dealing with graph databases. Or maybe the fact I have such use cases means graph is a wrong tool for the job?

neo4j cypher graph-databases

share improve this question

asked Apr 10 at 15:39
Yuriy
1,370 • 9 • 19

add a comment

1 Answer

active oldest votes

1

Generally when you encounter the need to cross relationships shows a modelling concern.

I can't remember any e-commerce like applications not having dedicated domain model for orders, would be neo4j or other persistence layers. Orders are an important part of your domain and would then have their dedicated node.

A relationship between the order and the product would represent what is called in common ecommerce apps an order line item which can holds the unit as property.

share improve this answer

answered Apr 10 at 15:50
Christophe Willemsen
5,115 • 1 • 4 • 17

In my case there is a traditional SQL database with the orders, buyers etc. The graph is built on top of it to

Related

43 Neo4j - Cypher vs Gremlin query language
3 Integrating mongodb with neo4j, is there any API that will link them?
1 Optimize Neo4j Cypher path finding with limited paths in an undirected graph
1 Neo4j - Cypher: merge duplicate relationships
0 Neo4j - NOT IN query
1 Most efficient way to find connections between the nodes in Cypher / Neo4J?
0 How to write a Cypher query with a condition on one of relationship properties in Neo4j database?
0 How to select relationships spreading from neo4j?
13 Neo4j super node issue - fanning out pattern
1 Is incremental adds are possible in neo4j?

Hot Network Questions

If a stock doesn't pay dividends, then why is the stock worth anything?

speed up some specific reports that are slow and cumbersome on that source model. So far it's a pilot 'proof of concept' project dealing with aggregated and not individual orders in covered use cases, hence there are no nodes for every order. If we have nodes as buyers connected to orders, orders connected to products and orders connected to units, it will work, but what would be the difference from SQL apart from different medium? – Yuriy Apr 10 at 15:56

the unit can be a relationship property. the main difference is that you will not have join tables. also a common modelling technique in neo4j is to first write the questions you want to ask to the database and then you model for these questions – Christophe Willemsen Apr 10 at 17:48

So my solution is acceptabke then, I reckon. Thanks. – Yuriy Apr 10 at 17:49

every solution is acceptable, the difference when your model is not optimal or graph oriented is that you will loose the performance capabilities that the db can provide – Christophe Willemsen Apr 10 at 17:51

What I was not sure about is if using attributes here doesn't scream wrong. I need to think if I really need to keep those unit nodes though. – Yuriy Apr 10 at 17:53

add a comment

Your Answer

Sign up or log in

Sign up using Google

Sign up using Facebook

Sign up using Stack Exchange

Post as a guest

Name

Email

required, but never shown

Post Your Answer

By posting your answer, you agree to the privacy policy and terms of service.

Not the answer you're looking for? Browse other questions tagged neo4j cypher graph-databases or ask your own question.

Can a Glaive be stowed in a Ruby Scabbard?

What is aliasing and what causes it?

How to ask dumb questions

Approximation of Borel sets by a countable collection of majorants

Important formulas in Combinatorics

How do I remove Candy Crush Saga from Windows 10?

Words of the Buddha

Aerofoil that gives reasonably good lift for both flow directions: forward and backward

Legacy application unreachable code

Double integral problem with sin

Make subscript size smaller (always)

What does OWA stand for?

How can I grant a non-Super User group permission to force check-in of articles?

Religion After the Discovery of a Multiverse

Select polygons one by one and export them using arcpy

Is there a single-word antoym for "Opportunity"?

Two definitions of Green's function

An "outside the box" sequence

Is it possible to automate tests for in progress sprint

Suggested alternatives for that horrible new noun 'nice-to-have'?

Draw horizontal lines like open and close brackets

Does Lebanon ban the diary of Anne Frank?

Using Emergency Fund to Sell Upside-down Car

question feed