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

# cypher indexing common property on all nodes







As per new indexing rules, the auto\_index will go away in future and its expected to create indexes using cypher. According to this new way, to index a node property, you MUST provide a Node Label.



I have a 'nodeld' property present on all types of Node Labels - User, Employee, Bank, Car, etc. I used to auto-index this property to retrieve any type of node if its nodeld is known. Please note that since auto-index did not require me to give a Node Label, it was possible for me to do what I did.

ReadableIndex<Node> readableIndex = this.graphDatabaseService.index().getNodeAutoInd readableIndex.get("nodeId", "0").getSingle();

But with new style, I have to create index on nodeld property for each and every Node Label. So I have to do this:

```
create index on :User(nodeId)
create index on :Employee(nodeId)
```

Moreover, my method getByNodeId(String nodeId) is useless now because this cypher query IMHO will not be able to use the index anymore since I am not passing any node label.

match (node) where node.nodeId = {nodeId} return node;

Since the whole point of my getByNodeId() method was to be generic across all nodes, I cannot give this cypher query a node label. So what should I do here. My 2 questions are:

- How do I tell neo4j via cypher to index on all node labels
- · How do I write a cypher query which uses index not based on node label, but based on node property.

### Note:

- It is essential for me to use cypher because I am using neo4j-jdbc and they have no method to create auto-index or access the auto-indexer (atleast not that I know of).
- Some might suggest me to change the neo4j.properties to enable auto-indexing there, but I dont like changing configuration files. I want to do it in my program. Anyway, that would have only solved the first issue. Second issue is still there.

java neo4j cypher share improve this question



oldest

votes

active

add a comment



1 Answer

Thus, if you make all your nodes share a common label, say Base (in addition to whatever labels they currently have), you can just have a single index that covers all your nodes:

CDEATE THIREY ON . Baco ( nodeTd)

A node can have multiple labels.

asked 4 months ago

viewed 38 times

active 4 months ago

Blog

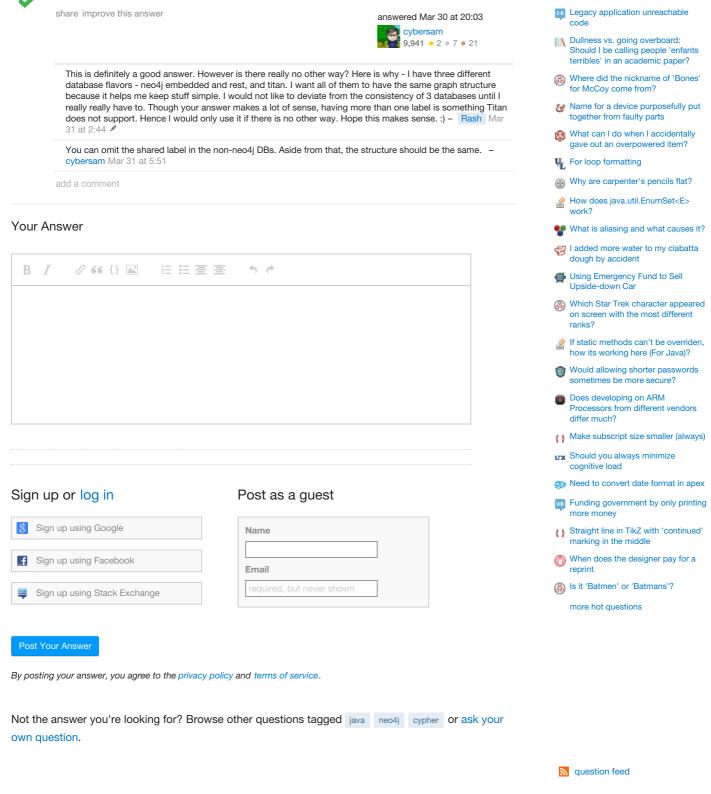
Why Stack Overflow is a Good Workplace for Women



## Related

- 1 Neo4j: fulltext indices and auto indexing in Cypher
- 1 Create relationship between nodes having same property value in common, using one Cypher query
- 1 Get all nodes with given type by ( select \* from tablename in SQL world) with Cypher Neo4j
- O Cypher aggregating nodes by string in a property
- 0 how to remove node itself in NOT logical cypher query?
- 0 Fetching nodes with a label and index using cypher
- 1 How to return all properties of a node with their name and their value using Cypher
- 0 Cypher: Create relationships between nodes based on a common property key id
- 1 Cypher query for all nodes that lack a certain property
- 0 deleteing a node cypher

**Hot Network Questions** 



Noun form of "extant"?

CKEATE INDEX ON : DdSe(HOUELU)

tour help blog chat data legal privacy policy work here advertising info mobile contact us feedback CULTURE / **TECHNOLOGY** LIFE / ARTS SCIENCE OTHER RECREATION Stack Apps Programmers Photography English Language & Administrators Meta Stack Exchange Server Fault Unix & Linux Cross Validated (stats) Drupal Answers Fantasy Skeptics Ask Different (Apple) Super User **Theoretical Computer** SharePoint Stack Overflow Careers Web Applications WordPress Development Movies & TV Ask Uhuntu Geographic Information Seasoned Advice MathOverflow Mathematica Christianity Webmasters Arqade (gaming) Electrical Engineering **ExpressionEngine® Bicycles** Biology TeX - LaTeX Android Enthusiasts Personal Finance &

Information Security more (13) Money Role-playing Games more (5)

Academia more (21)

more (9)

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