Bipartite graph projection via Cypher query

7 posts by 2 authors

Dan

Apr 24

I was wondering if it is possible to generate a single mode graph projection from a bipartite (2-mode) graph via a Cypher query?

e.g., http://upload.wikimedia.org/wikipedia/commons/f/f9/Bipartite_network_projection.png

So let's say I have text mining related data such as entities (X) linked to documents (Y) in a neo4j graph database. And I want a query result set that links X's to X's (for example) based on links to common Y's. Can such be done via a Cypher query sequence?

Thanks, Dan

Michael Hunger

Apr 24

Do you mean something like this?

```
MATCH (entity:Entity)-[:LINKS_TO]->(doc:Document)
RETURN doc, collect(entity) as entities
```

Michael

- show quoted text -
- show quoted text -

--

Dan

Apr 24

Thanks for info Michael. I believe that query will return the list of entities common to each document. What I'm really after is generating dynamically (via Cypher) a new graph result (relationships) comprised of only one node type derived from the underlying bipartite graph (two node types).

e.g.,

http://image.slidesharecdn.com/pn2013-130704162301-phpapp01/95/sna-in-technologyenhanced-learning-a-shift-to-personal-network-perspective-using-clustered-graphs-9-638.jpg?cb=1373000400

So if neo4j database includes links between entities and documents, I want to generate a new graph result that would just contain links between one entity type: entities to entities or documents to documents. These new dynamic (query time) links would be based on shared connections between entities and documents in the underlying graph database.

So if E1 is linked to D1 and D2, E2 is linked to D1, and E3 is linked to D3, a Cypher query would return:

1) 1) For a entity only graph result: E1 linked to E2 and E3 by itself.

2) 2) For a document only graph result: D1 linked to D2 and D3 by itself.

Thanks, Dan

- show quoted text -

Michael Hunger Apr 24

ah, so what I said

```
MATCH (entity:Entity)-[:LINKS_TO]->(doc:Document)
WITH doc, collect(entity) as entities
FOREACH (e1 in entities | FOREACH (e2 in filter(e in entities WHERE id(e) > id(e1)) | CREATE (e1)-[:RELATED_TO]->(e2)))
```

Michael
- show quoted text -

Dan Apr 25

Thanks Michael, that definitely works.

A final question is there a way to do similar but to only return the links, not create them? I see that FOREACH statement is apparently used to write to database (update, create, etc). I was trying to see how to do similar but only return the links (not actually create them in database). Not sure if some nested EXTRACT statements would do it. Thanks.
- show quoted text -
- show quoted text -

Michael Hunger Apr 26

Foreach is just iterating over a list and in this case creates a "cross-product".

instead of using foreach you can also use unwind, which turns collections back into rows, like this

```
MATCH (entity:Entity)-[:LINKS_TO]->(doc:Document)
WITH doc, collect(entity) as entities
UNWIND entities as e1
UNWIND entities as e2
WHERE id(e2) > id(e1)
RETURN e1,e2,doc
```

- show quoted text -

Dan Apr 26

That is great info. This is a useful graph transformation common in SNA tools like Pajek and Gephi (with a plugin), so nice I can do it directly in Neo4j without needing to do post-processing at the application side. I did encounter a syntax error on the "WHERE" clause when trying out the query, e.g., Neo.ClientError.Statement.InvalidSyntax, Invalid input 'H': expected 'i/I' (line 5, column 2 (offset: 133)) "WHERE id(e2) > id(e1)"

But after injecting a "WITH" clause it worked fine:

```
MATCH (entity:Entity)-[:LINKS_TO]->(doc:Document)
WITH doc, collect(entity) as entities
UNWIND entities as e1
UNWIND entities as e2
WITH e1, e2, doc
WHERE id(e2) > id(e1)
RETURN e1, e2, doc
```

- show quoted text -