Adding new properties to an existing map

5 posts by 2 authors

bi...@levelstory.com Apr 7

Is there any way to add additional properties to an existing Map object?

I thought I could use SET, but it doesn't work on maps:

```
WITH {a: "foo"} AS result
SET result.b = "bar"
RETURN result
```

```
Type mismatch: expected Node or Relationship but was Map (line 1, column 31)
```

Because of this, I'm currently doing this everywhere which makes the queries hard to read:

```
WITH {a: "foo"} AS result
WITH {a: result.a, b: "bar"} AS result
RETURN result
```

Thanks,
bill

Michael Hunger Apr 7

Other recipients: stefan.p...@neotechnology.com

Hey Bill,

interesting what's your use-case for this?

Thanks so much,

Michael

- show quoted text -
- show quoted text -

bi...@levelstory.com Apr 7

Other recipients: stefan.p...@neotechnology.com

Hi Michael,

In order to reduce the number or rows resulting from complex queries, I bundle up the results of sub queries into a Map object. I then return a collection of Map objects which can then be easily converted into Go structs. I believe that you were the one that suggested this approach to me.

Here is how I am doing most of my queries:

```
  MATCH (f:file {id: "foobar"})
  WITH f, {accessedrole: "", accessedby: ""} AS r

  OPTIONAL MATCH (f)-[:CREATED]->(a:activity)
  WITH f, {
    accessedrole: r.accessedrole, accessedby: r.accessedby,
    createdat: a.createdat, createdby: {id: a.contactid, object: "contact"}} AS
r

  OPTIONAL MATCH (f)-[:HISTORY]->(a:activity)
  WITH f, {
    accessedrole: r.accessedrole, accessedby: r.accessedby,
    createdat: r.createdat, createdby: r.createdby,
    updatedat: a.createdat, updatedby: {id: a.contactid, object: "contact"}} AS
r

  OPTIONAL MATCH (f)-[:TAG]->(x)
  WITH f, COLLECT(x) AS xs, r
  WITH f, {
    accessedrole: r.accessedrole, accessedby: r.accessedby,
    createdat: r.createdat, createdby: r.createdby,
    updatedat: r.updatedat, updatedby: r.updatedby,
    tags: [x IN xs | {id: x.id, object: x.object, name: x.name}]
    } AS r

  RETURN COLLECT({
    id: f.id, object: f.object, projectid: f.projectid,
    no: f.no, name: f.name, description: f.description, url: f.url,
    mime: f.mime, thumbnail: f.thumbnail, size: f.size,
    accessedrole: r.accessedrole, accessedby: r.accessedby,
    createdat: r.createdat, createdby: r.createdby,
    updatedat: r.updatedat, updatedby: r.updatedby,
    tags: r.tags
  }) AS r
```

- show quoted text -

Michael Hunger Apr 7

Other recipients: stefan.p...@neotechnology.com

Hmm,

what you could do instead of modifying the map in-place to create the new map with the additional data around the original one:

e.g.

> WITH {a: "foo"} AS result
> RETURN {inner: result, b: "bar"} AS result

> We're also looking into a feature like:

> SET map.key = value
>
> But I can't give any timeline for that right now.
>
> Cheers, Michael

- show quoted text -

bi...@levelstory.com Apr 7

Other recipients: stefan.p...@neotechnology.com

Yes, I thought about that but then I would need to unpack everything again before I return the results. Being able to do SET map.key would be ideal, I guess I'll just wait until that becomes available.

- show quoted text -