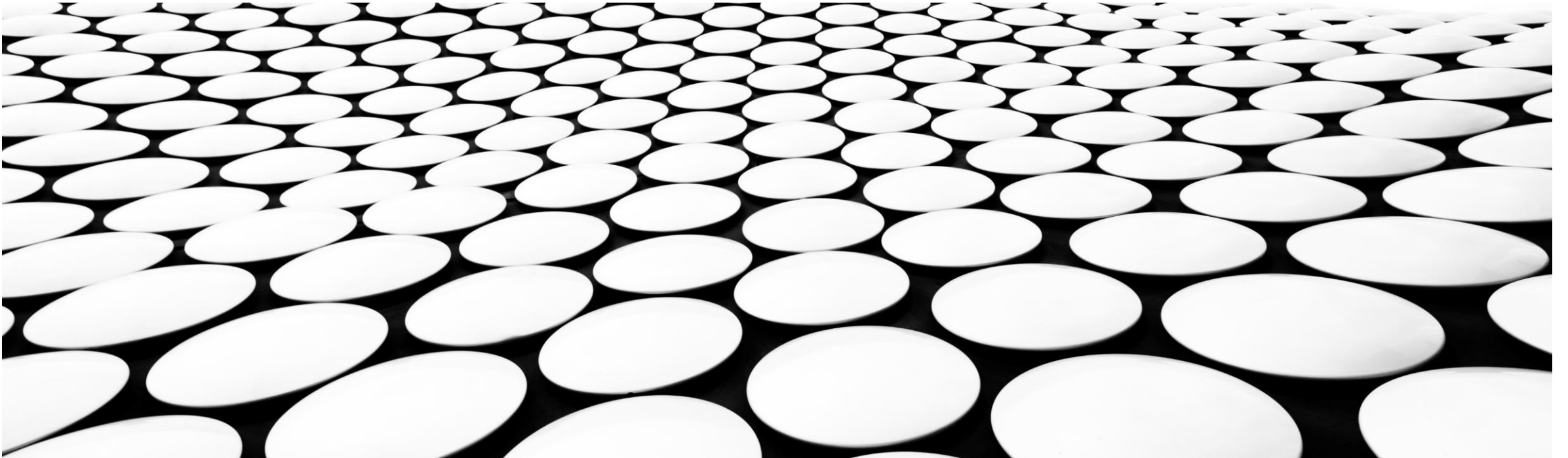

NEO4J –APOC_GRAPH GROUPING

Jee-Soo Kim / jeesoo.kim@en-core.com



Example on movie graph Cypher

Copy to Clipboard

Run in Neo4j Browser

```
MATCH (n)                                released
SET n.century = toInteger(coalesce(n.born,n.released)/100) * 100;

CALL apoc.nodes.group(['Person','Movie'], ['century']);
```

There is no guide/dataset to create data. Fortunately, I remembered that I can download movie1.csv files from 'Importing Data' lectures of Graph Academy. And I just want to let you know I tried to practice this example on SandBox with Movies dataset.

Anyway, I created data using movie1.csv. I created the dataset by importing csv. I intend to use 'released' and 'born' as props name to use the cypher on APOC documentation. There is a misspelling 'relased' and I encountered an error. I repeated to delete & create dataset and executed the apoc procedures. Later I found the misspelling.

I can fix the mistake and keep studying, but I think the content uploaders/developers who are in charge should double-check by putting themselves in the user's shoes.

```
1 LOAD CSV WITH HEADERS FROM 'file:///movies1.csv' AS row
2 MERGE (m:Movie {movieId: toInteger(row.movieId), title:row.title, avgVote:toFloat(row.avgVote),
  released: toInteger(row.releaseYear), genres:split(coalesce(row.genres,""), ":")})
3
4 LOAD CSV WITH HEADERS FROM 'file:///people.csv' AS row
5 MERGE (p:Person {personId:toInteger(row.personId), name:row.name,
  born:coalesce(toInteger(row.birthYear), 0)})
```

Also, I want to get the information about the dataset. I could create 'Person' and 'Movie' nodes but I don't know how to connect all of them randomly for only practicing apoc procedures.

The only required parameters are a *label-list* (can also be ['*']) and a *list of property names* to group by (both for rels/nodes).

I think * can be used only when all of different labels have a common property for grouping.

```
Graph Setup Cypher Copy to Clipboard Run in Neo4j Browser

CREATE
  (alice:Person {name:'Alice', gender:'female', age:32, kids:1}),
  (bob:Person {name:'Bob', gender:'male', age:42, kids:3}),
  (eve:Person {name:'Eve', gender:'female', age:28, kids:2}),
  (graphs:Forum {name:'Graphs', members:23}),
  (dbs:Forum {name:'Databases', members:42}),
  (alice)-[:KNOWS {since:2017}]->(bob),
  (eve)-[:KNOWS {since:2018}]->(bob),
  (alice)-[:MEMBER_OF]->(graphs),
  (alice)-[:MEMBER_OF]->(dbs),
  (bob)-[:MEMBER_OF]->(dbs),
  (eve)-[:MEMBER_OF]->(graphs)
```

```
Cypher Copy to Clipboard Run in Neo4j Browser

CALL apoc.nodes.group([*], 'gender',
  [{*:'count', age:'min'}, {*:'count'} ])
```

In my opinion, *(asterisk, means “all”) should be changed to **‘Person’** because the label ‘Forum’ data don’t have ‘gender’ property.

When I put ‘Person’ in the [label field], I got the result with grouping by ‘gender’ property.

If developers intended to use * for a label-list, I would like to let them know it’s not working.

Something went wrong: "TypeError: Cannot read property 'join' of null" and the application can't recover.

Reload application

```
with ["US","DE","UK","FR","CA","BR","SE"] as tld
unwind range(1,1000) as id
create (u:User {id:id, age : id % 100, female: rand() < 0.5, name: "Name "+id, country:tld[toInteger(id)])
with collect(u) as users
unwind users as u
unwind range(1,10) as r
with u, users[toInteger(rand()*size(users))] as u2
where u <> u2
merge (u)-[:KNOWS]-(u2);
```

This is **redundant** because alias r(number from 1 to 10) is not used in the cypher.

I can pass this redundant line, but I think it makes beginners confused.