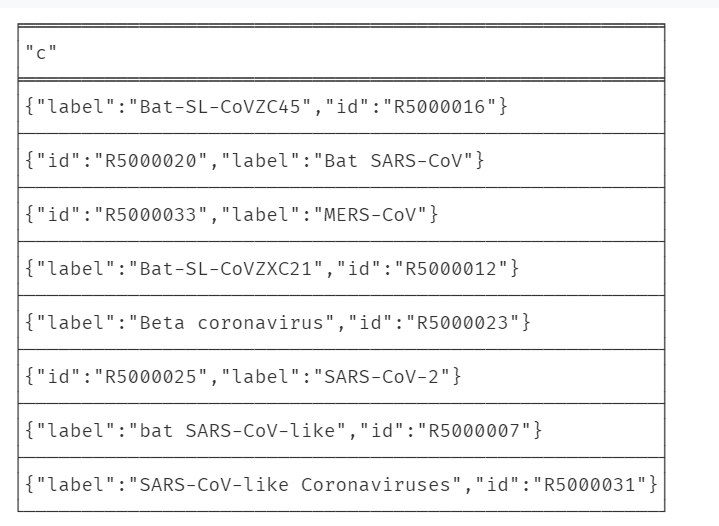
For Virus,Given label/id

1. Show its similar virus

Match p=(a)-[r:SIMILAR]->(c)

Where a.label='SARS-CoV'

Return c;



2.Show it’s neighbor nodes other than virus

2.1Drug

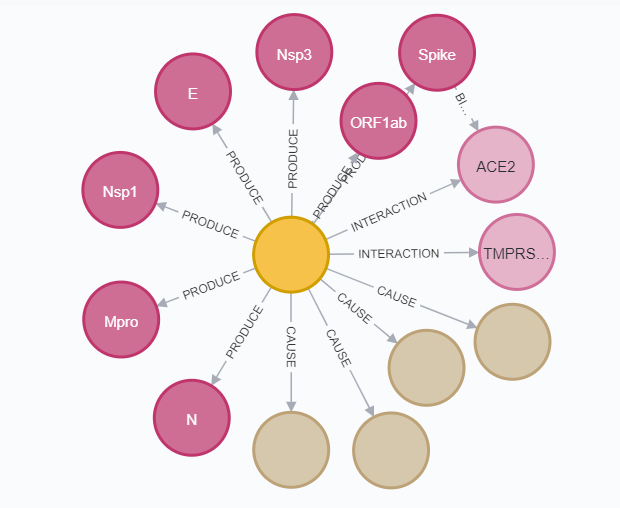
2.2Disease

2.3VP

Match p=(a)-[r]->(c)

Where not c:Virus AND a.label='SARS-CoV'

Return \*;



3.Show some inferred nodes:

3.1 Similarity

3.1.1 On disease

CALL gds.graph.create(

'Disease\_graph',

['Virus', 'Disease'],

{

CAUSE: {

type: 'CAUSE',

properties: {

}

}

}

);

CALL gds.nodeSimilarity.stream('Disease\_graph')

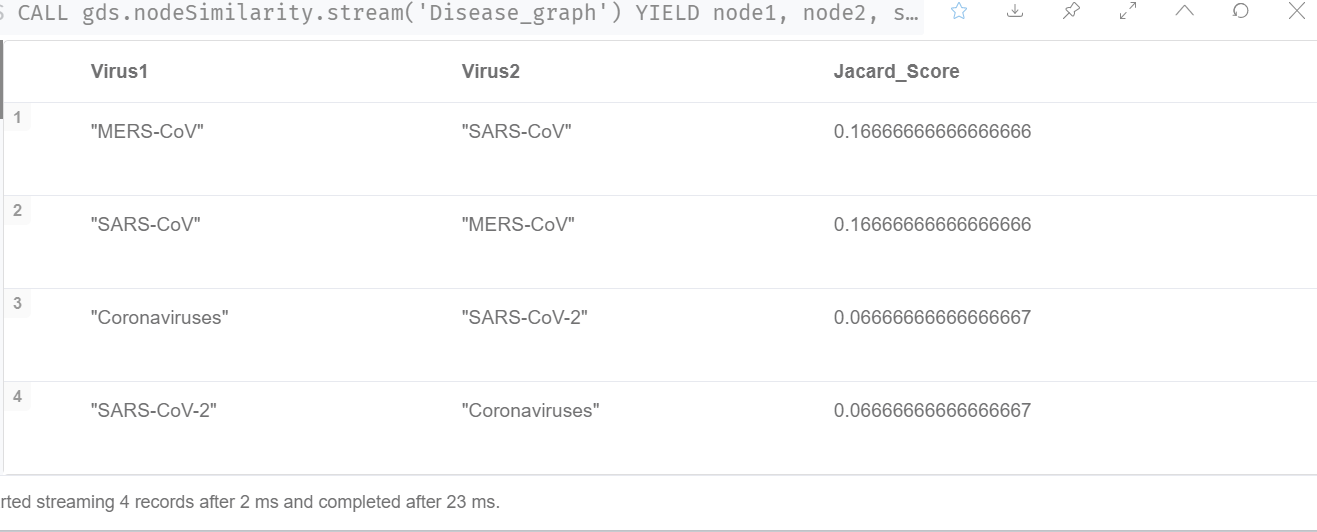
YIELD node1, node2, similarity

RETURN gds.util.asNode(node1).label AS Virus1, gds.util.asNode(node2).label AS Virus2, similarity as Jacard\_Score

ORDER BY Jacard\_Score DESCENDING, Virus1, Virus2

3.1.2 On Drug

3.1.3 On Virus Protein



3.2 Powerful Drug ­­­­­­

3.2.1 Find Drug which can effect on itself and viruses that are similar to it.

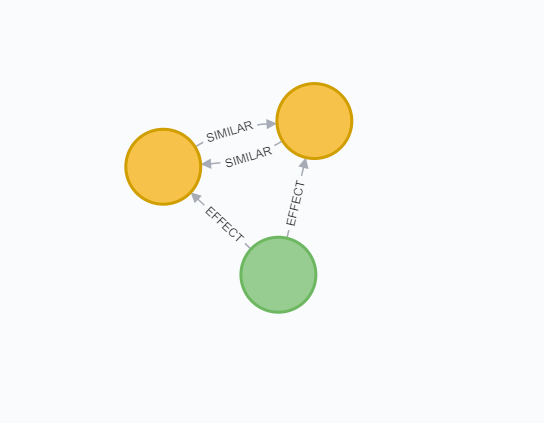
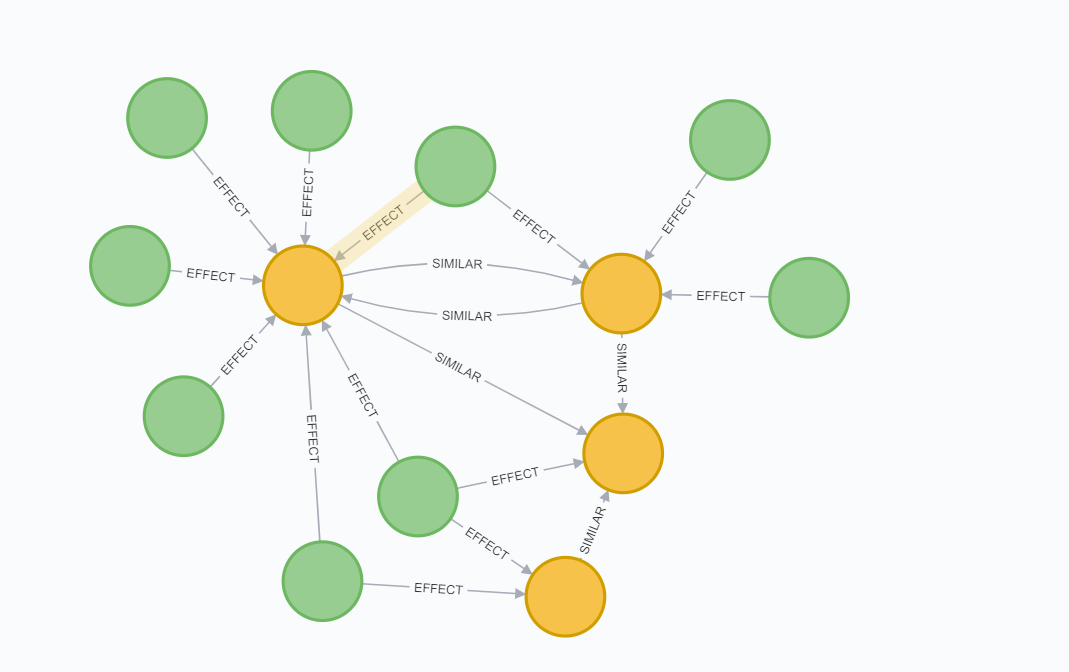
MATCH p1=(d:Drug)-[r1:EFFECT]->(v1:Virus),

p2=(d:Drug)-[r2:EFFECT]->(v2:Virus)

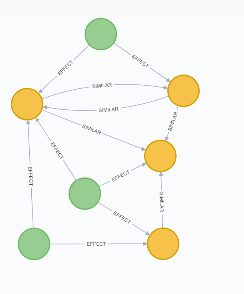
where v1.label='SARS-CoV' AND

(exists((v1)-[:SIMILAR]->(v2)) or exists((v2)-[:SIMILAR]->(v1)))

return p1,p2

An algorithm

Virus=[]

Drug=[]