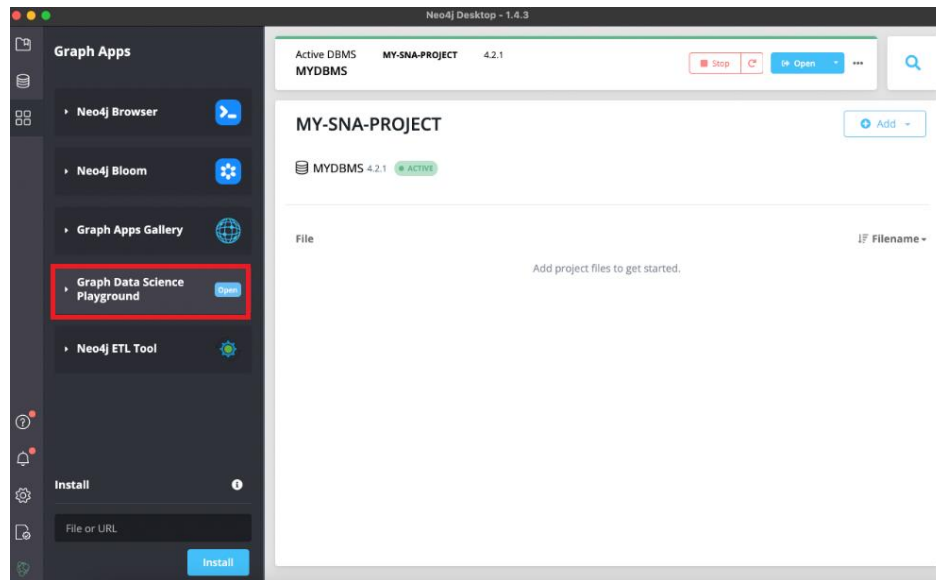
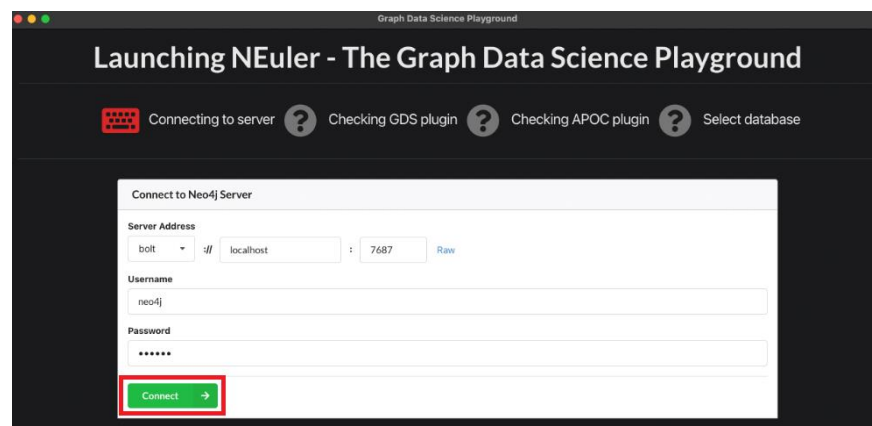


Centrality Algorithms

1. Open Graph Data Science Playground for Run Centrality Algorithm



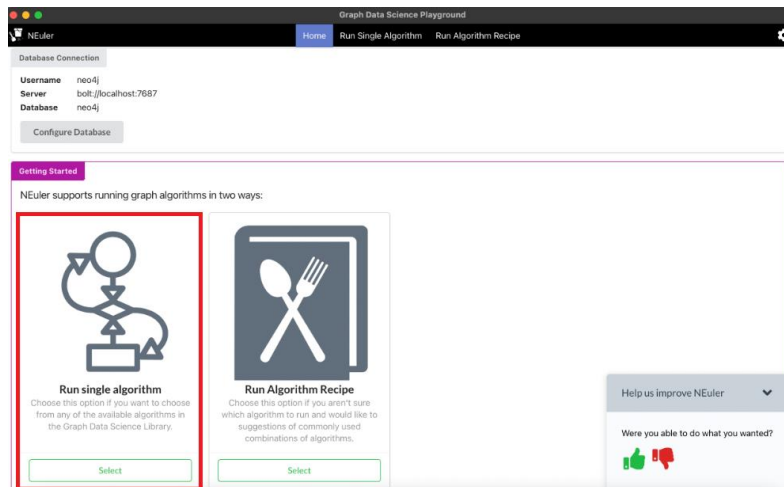
2. Launching NEuler – Connect to server



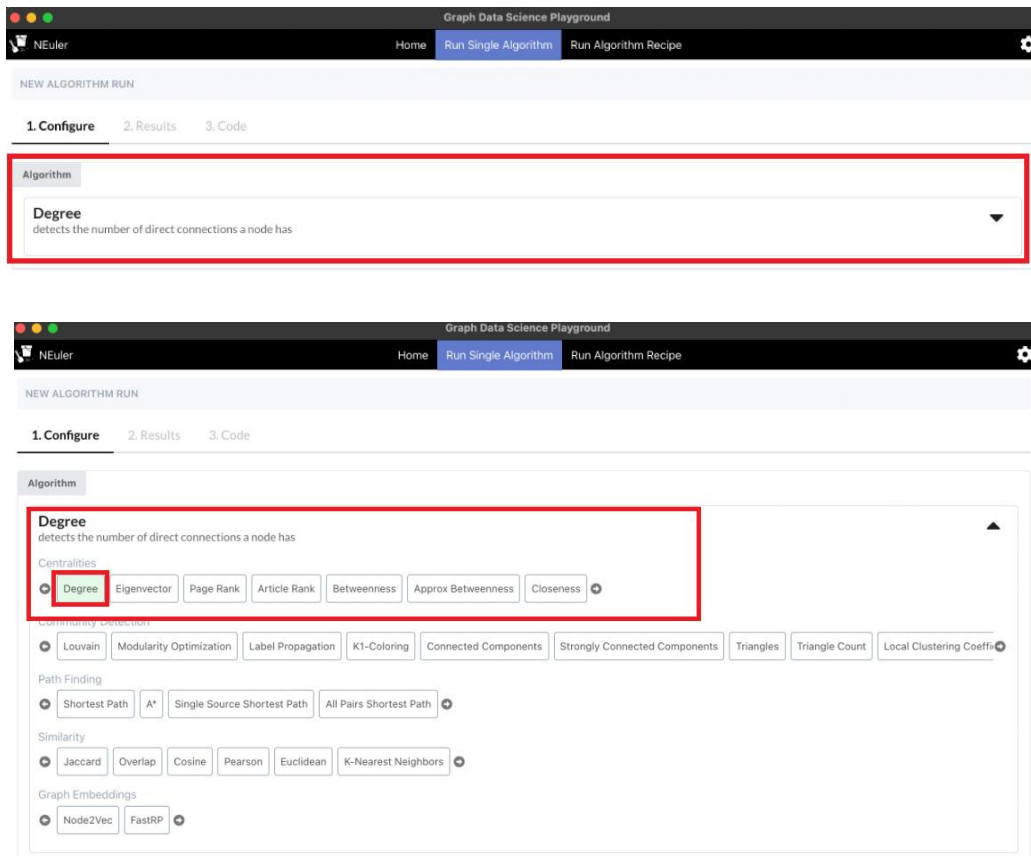
3. Select Database



4. Select Run Single Algorithm



5. Select Degree Centrality Algorithm can help us find popular nodes in a graph.



6. And you can set values by following screen then Click “Run Algorithm”

The screenshot shows the '1. Configure' tab of the NEuler interface. The 'Algorithm' dropdown is set to 'Degree', which is described as 'detects the number of direct connections a node has'. Under the 'Projected Graph' section, several settings are highlighted with a red box: 'Label' is 'dx', 'Relationship Type' is 'Risk', 'Relationship Orientation' is 'Natural', 'Weight Property' is 'n_id', and 'Default weight' is '1'. In the 'Results' section, 'Store results?' is a toggle switch, and 'Rows to show' is set to '42'. A red box highlights the 'Run Algorithm' button at the bottom left. A 'Help us improve NEuler' link is at the bottom right.

7. You'll see your code fragment for Degree Centrality

The screenshot shows the '3. Code' tab of the NEuler interface. It displays the generated code for the Degree Centrality algorithm. The code is organized into three sections, each with a 'Copy' button. The first section contains the initial setup:

```
:use neo4j;
```

 The second section, highlighted with a red box, contains the configuration parameters:

```
:param limit => ( 42);  
:param config => ({  
  nodeProjection: 'dx',  
  relationshipProjection: {  
    relType: {  
      type: 'Risk',  
      orientation: 'NATURAL',  
      properties: {  
        n_id: {  
          property: 'n_id',  
          defaultValue: 1  
        }  
      }  
    },  
  },  
  relationshipWeightProperty: 'n_id'  
});  
:param communityNodeLimit => ( 10);
```

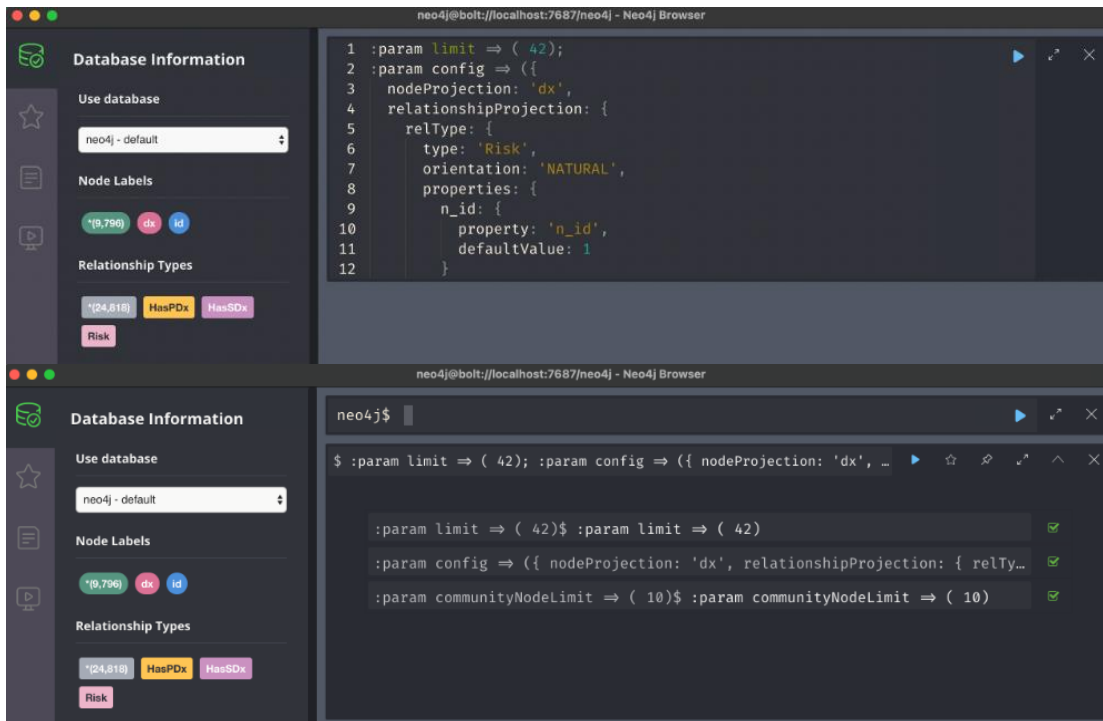
 The third section, also highlighted with a red box, contains the Cypher query:

```
CALL gds.alpha.degree.stream($config) YIELD nodeId, score  
WITH gds.util.asNode(nodeId) AS node, score  
RETURN node, score  
ORDER BY score DESC  
LIMIT toInteger($limit);
```

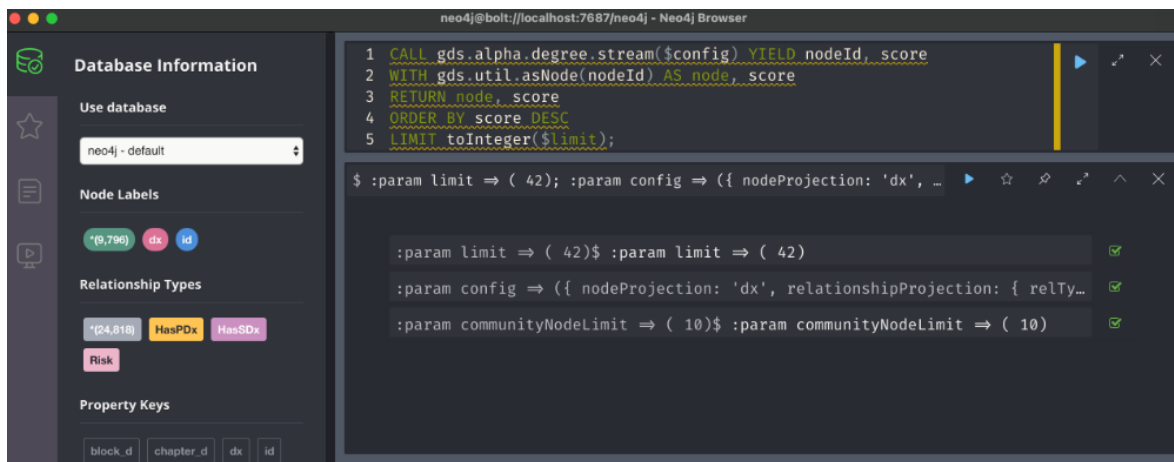
 A 'Help us improve NEuler' link is at the bottom right.

8. You can copy your code fragment for Degree Centrality in order to paste on Neo4j Brower

Setting parameters:



Anonymous Graph:



That's already. You'll see result following screen:

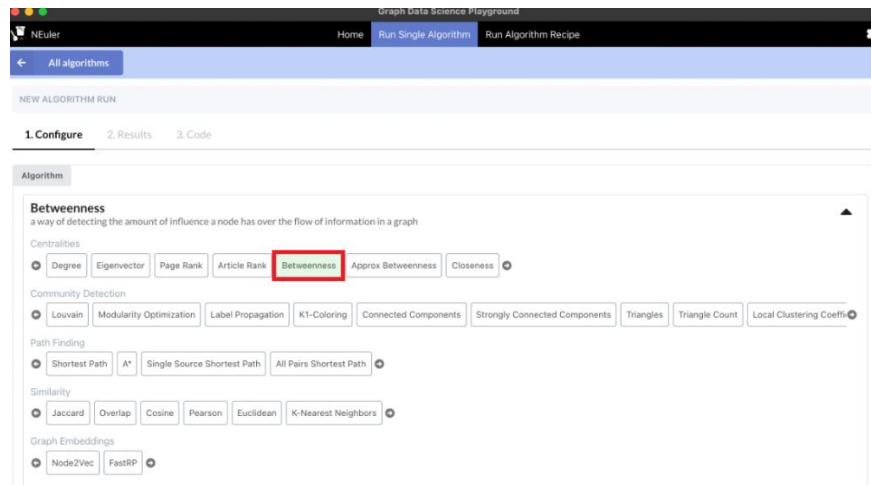
neo4j\$ CALL gds.alpha.degree.stream(\$config) YIELD nodeId, score

node	score
[{"block_d": "Hypertensive diseases", "dx": "I10", "chapter_d": "Diseases of the circulatory system", "term_d": "Essential (primary) hypertension"}]	158.0
[{"block_d": "Metabolic disorders", "dx": "E789", "chapter_d": "Endocrine, nutritional and metabolic diseases", "term_d": "Disorder of lipoprotein metabolism, unspecified"}]	130.0
[{"block_d": "Diabetes mellitus", "dx": "E119", "chapter_d": "Endocrine, nutritional and metabolic diseases", "term_d": "Type 2 diabetes mellitus, without complications"}]	125.0
[{"block_d": "Metabolic disorders", "dx": "E876", "chapter_d": "Endocrine, nutritional and metabolic diseases", "term_d": "Hypokalaemia"}]	94.0
[{"block_d": "Renal failure", "dx": "N185", "chapter_d": "Diseases of the genitourinary system", "term_d": "Chronic kidney disease, stage 5"}]	79.0
[{"block_d": "Renal failure", "dx": "N179", "chapter_d": "Diseases of the genitourinary system", "term_d": "Acute renal failure, unspecified"}]	79.0
[{"block_d": "Other forms of heart disease", "dx": "I48", "chapter_d": "Diseases of the circulatory system", "term_d": "Atrial fibrillation and flutter"}]	78.0

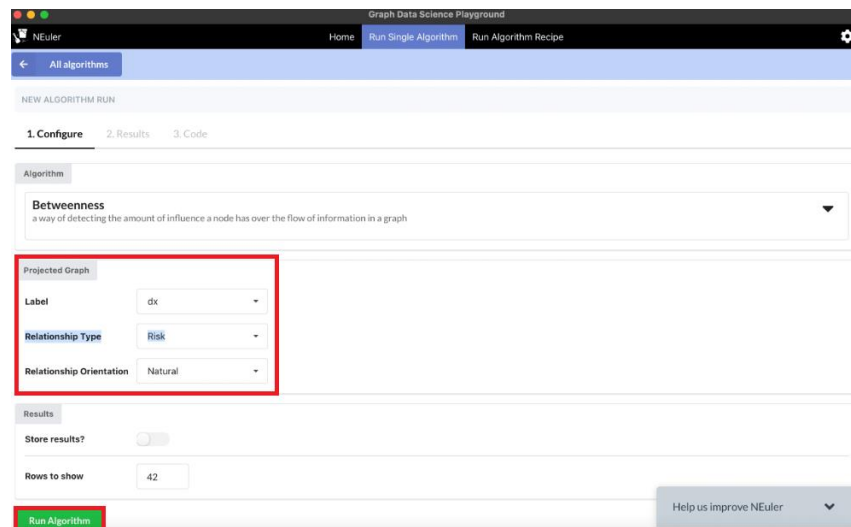
MAX COLUMN WIDTH: 100

9. You can find **Betweenness Centrality** same as step on Degree Centrality from **step 5 to 8**

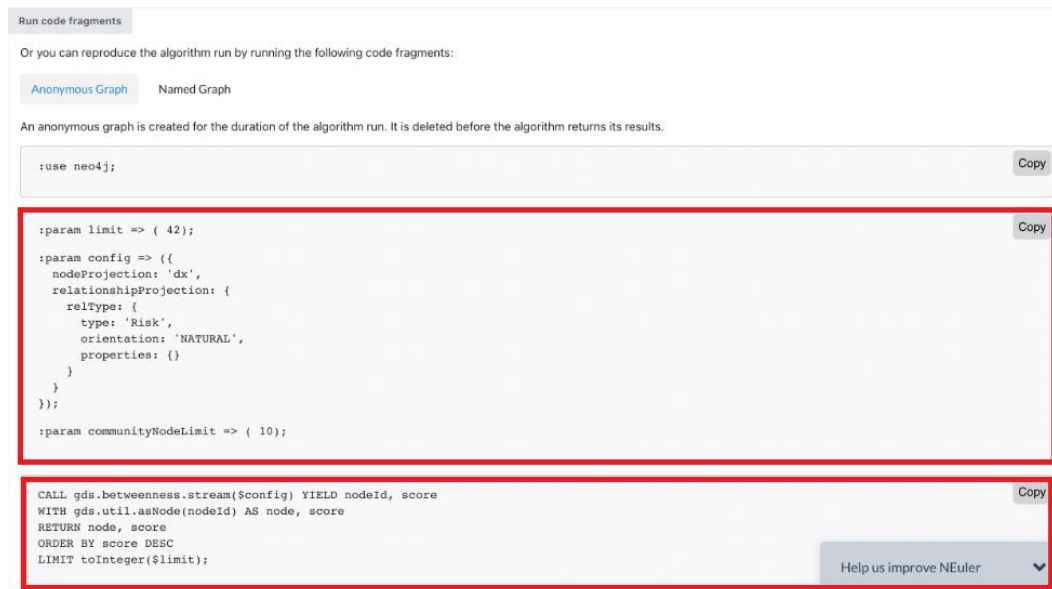
Select **Betweenness Centrality Algorithm** used to find nodes that serve as a bridge from one part of a graph to another.



And you can set values by following screen then Click “Run Algorithm”



You'll see your code fragment for Betweenness Centrality



You can copy your code fragment for Degree Centrality in order to paste on Neo4j Brower

Setting parameters:

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' panel is visible. The main editor area contains the following Cypher query:

```
1 :param limit => ( 42);
2 :param config => ({
3   nodeProjection: 'dx',
4   relationshipProjection: {
5     relType: {
6       type: 'Risk',
7       orientation: 'NATURAL',
8       properties: {}
9     }
10  }
11 });
12 :param communityNodeLimit => ( 10);
```

The screenshot shows the Neo4j Browser interface after the query has been executed. The main editor area displays the following output:

```
neo4j$
$ :param limit => ( 42); :param config => ({ nodeProjection: 'dx',...
:param limit => ( 42) :param limit => ( 42)
:param config => ({ nodeProjection: 'dx', relationshipProjection: { rel...
:param communityNodeLimit => ( 10) :param communityNodeLimit => ( 10)
```

Anonymous Graph:

The screenshot shows the Neo4j Browser interface. The main editor area contains the following Cypher query:

```
1 CALL gds.betweenness.stream($config) YIELD nodeId, score
2 WITH gds.util.asNode(nodeId) AS node, score
3 RETURN node, score
4 ORDER BY score DESC
5 LIMIT toInteger($limit);
```

That's already. You'll see result following screen:

The screenshot shows the Neo4j Browser interface with the query result displayed in table format. The table has two columns: "node" and "score".

"node"	"score"
{"block_d":"Ischaemic heart diseases","dx":"I251","chapter_d":"Diseases of the circulatory system","term_d":"Atherosclerotic heart disease"}	74840.07050359374
{"block_d":"Other forms of heart disease","dx":"I500","chapter_d":"Diseases of the circulatory system","term_d":"Congestive heart failure"}	46106.22052198258
{"block_d":"Hypertensive diseases","dx":"I10","chapter_d":"Diseases of the circulatory system","term_d":"Essential (primary) hypertension"}	43450.85809436045
{"block_d":"Chronic rheumatic heart diseases","dx":"I071","chapter_d":"Diseases of the circulatory system","term_d":"Tricuspid insufficiency"}	35649.25691214926
{"block_d":"Other forms of heart disease","dx":"I48","chapter_d":"Diseases of the circulatory system","term_d":"Atrial fibrillation and flutter"}	34045.69607198199
{"block_d":"Ischaemic heart diseases","dx":"I214","chapter_d":"Diseases of the circulatory system","term_d":"Acute subendocardial myocardial infarction"}	33666.064916922114
{"block_d":"Diseases of veins, lymphatic vessels and lymph nodes, not"}	32063.5806132377