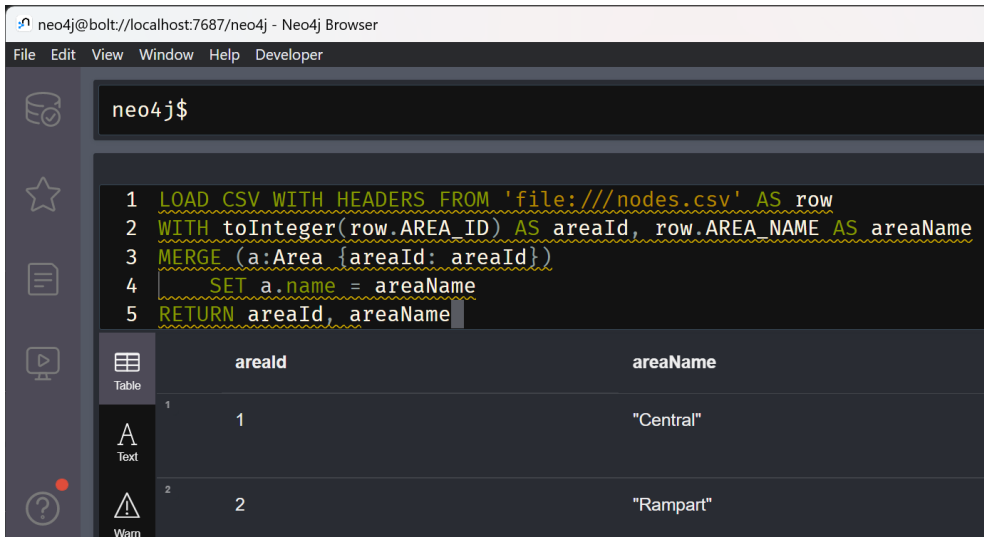
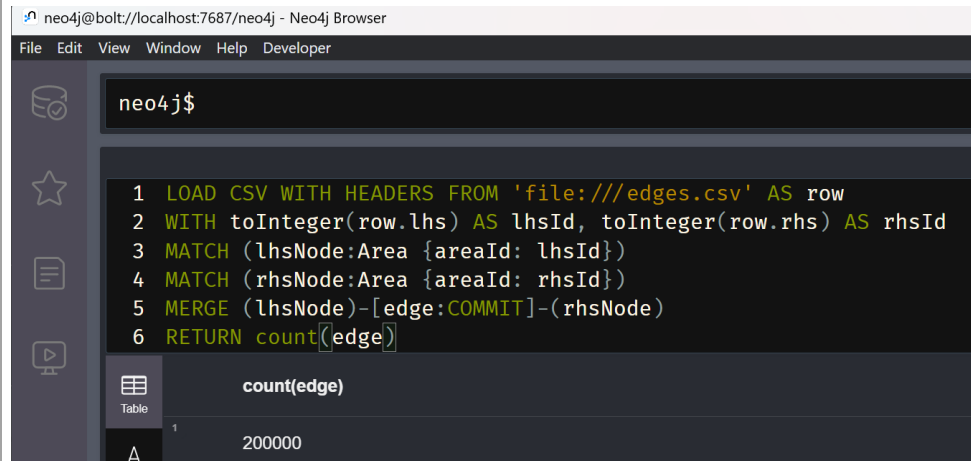


Export für Neo4j	<pre>select t1.area_id, t1.crime_code, t2.area_id, t2.crime_code from crime t1, crime t2 where t1.crime_code = t2.crime_code and t1.area_id &gt; t2.area_id</pre> <pre>call CSVWRITE ( 'C:/users/beni/downloads/edges.csv', 'select top 200000 t1.area_id, t2.area_id from crime t1, crime t2 where t1.crime_code = t2.crime_code and t1.area_id &gt; t2.area_id' );</pre> <pre>call CSVWRITE ( 'C:/users/beni/downloads/nodes.csv', 'select area_id, area_name from area' );</pre>
Neo4j import csv (Nodes)	<pre>LOAD CSV WITH HEADERS FROM 'file:///nodes.csv' AS row WITH toInteger(row.AREA_ID) AS areald, row.AREA_NAME AS areaName MERGE (a:Area {areald: areald}) SET a.name = areaName RETURN areald, areaName</pre> 
Neo4j import csv (Edges)	<pre>LOAD CSV WITH HEADERS FROM 'file:///edges.csv' AS row WITH toInteger(row.lhs) AS lhsId, toInteger(row.rhs) AS rhsId MATCH (lhsNode:Area {areald: lhsId})</pre>

```
MATCH (rhsNode:Area {areald: rhsId})  
  
MERGE (lhsNode)-[edge:COMMIT]-(rhsNode)  
  
RETURN count(edge)
```



Neo4j  
create  
graph  
(gds)

```
CALL gds.graph.project(  
  'myGraphProjection',  
  'Area',  
  {  
    COMMIT: {  
      orientation: "NATURAL"  
    }  
  })
```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

neo4j\$

```

1 CALL gds.graph.project(
2   'myGraphProjection',
3   'Area',
4   {
5     COMMIT: {
6       orientation: "UNDIRECTED"
7     }
8   })

```

	nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount
1	{ "Area": { "label": "Area", "properties":	{ "COMMIT": { "aggregation": "DEFAULT", "orientation":	"myGraphProjection"	21	342

Table

Text

Code

Neo4j  
calculat  
e  
centrali  
ty

```

CALL gds.degree.stream(
  "myGraphProjection"
) YIELD
  nodeId, score

RETURN gds.util.asNode(nodeId).name AS name, score AS degree
ORDER BY degree DESC, name DESC;

```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

neo4j\$

```

1 CALL gds.degree.stream(
2   "myGraphProjection"
3 ) YIELD
4   nodeId, score
5 RETURN gds.util.asNode(nodeId).name AS name, score AS degree
6 ORDER BY degree DESC, name DESC;

```

	name	degree
1	"Wilshire"	18.0
2	"West Valley"	18.0

Table

Text

Code

