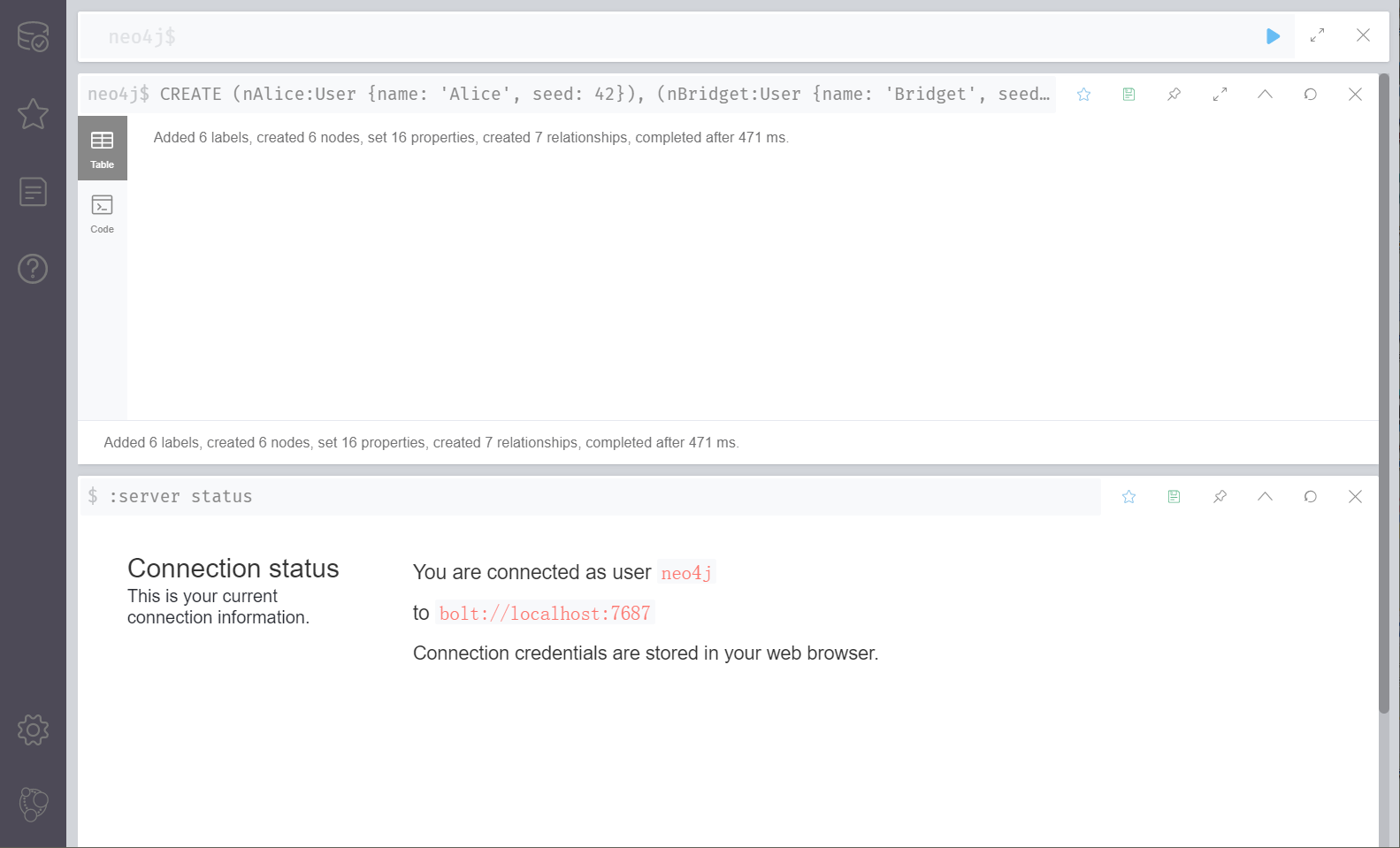
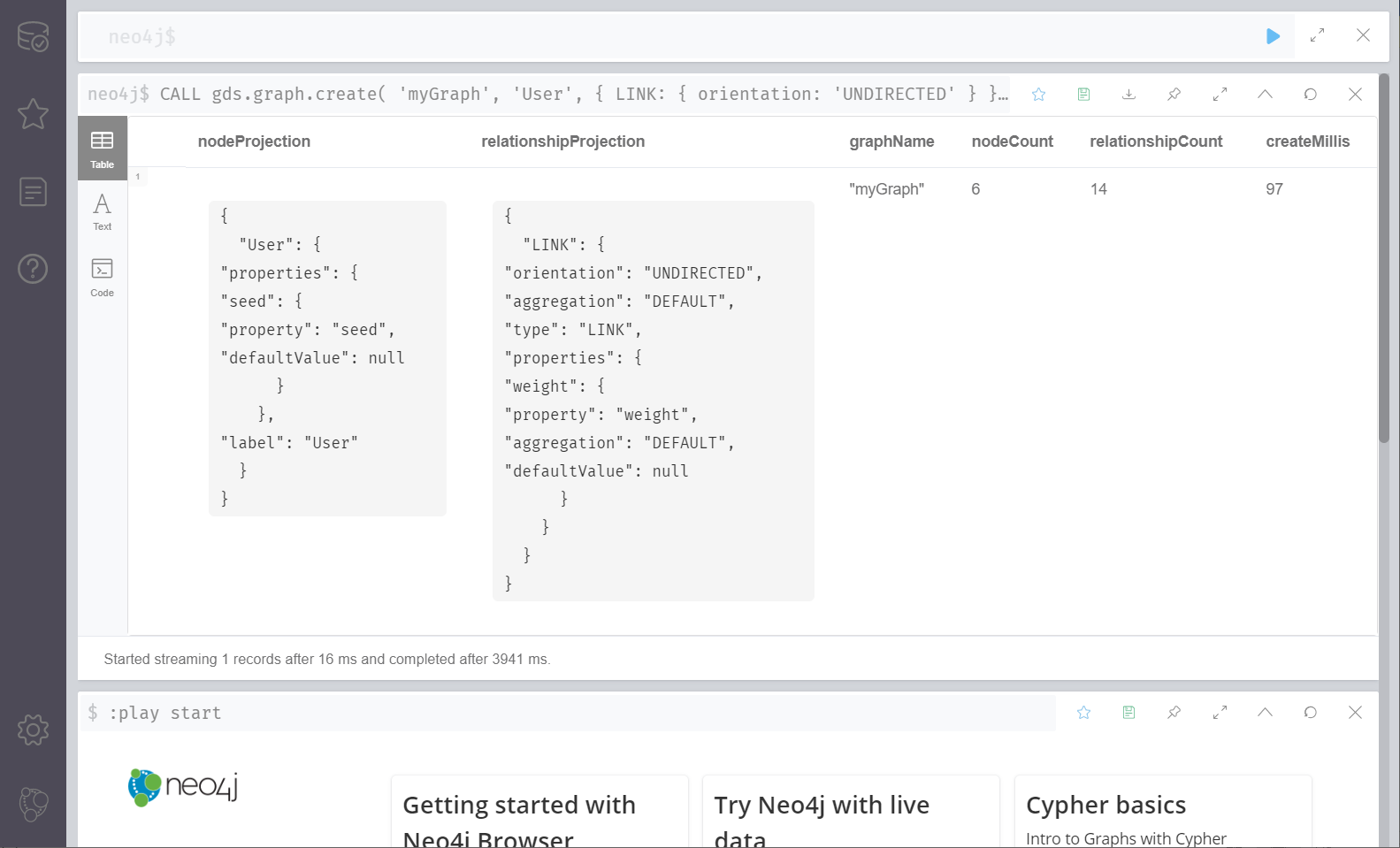
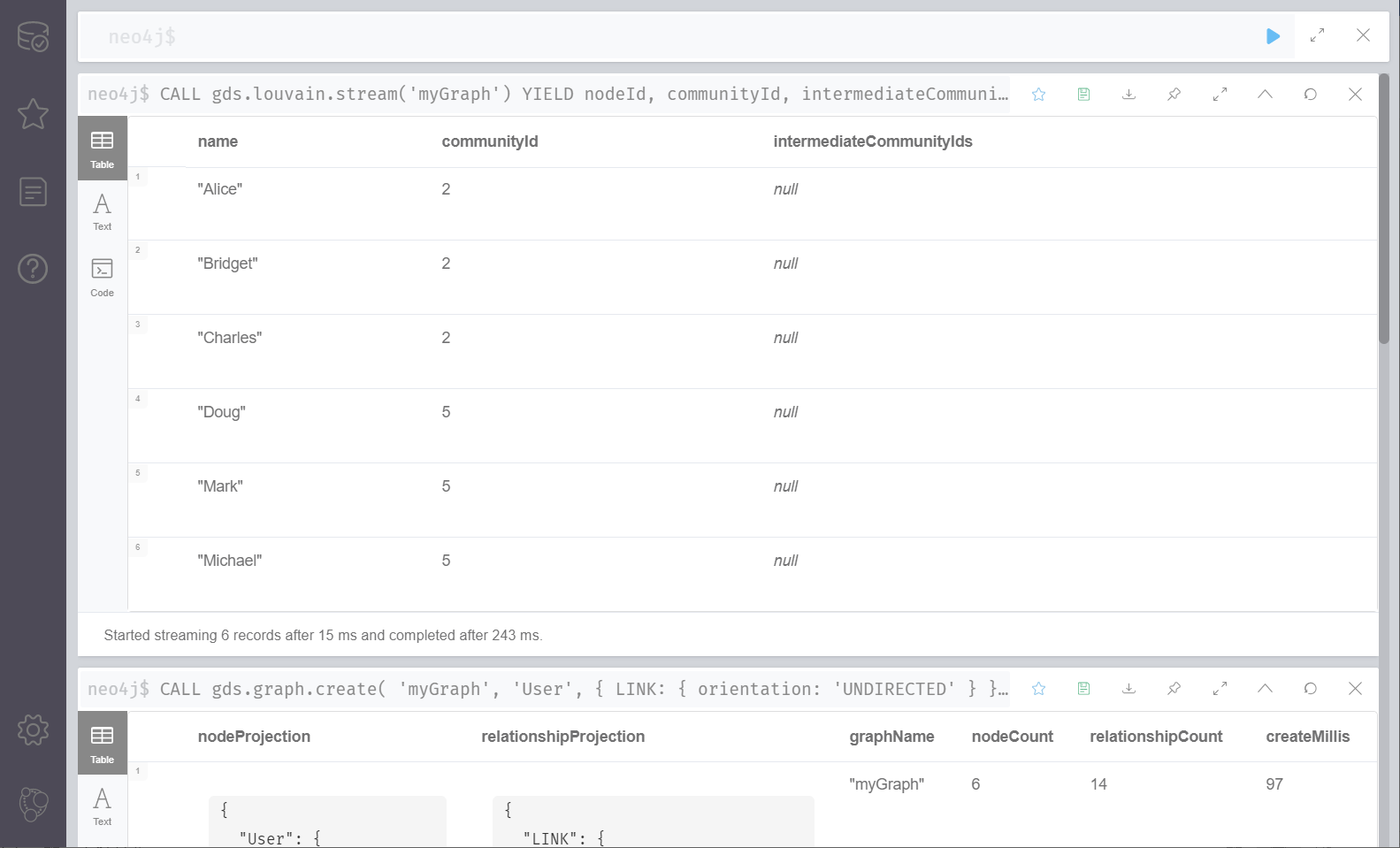
|  |
| --- |
| CREATE |
| (nAlice:User {name: 'Alice', seed: 42}), |
| (nBridget:User {name: 'Bridget', seed: 42}), |
| (nCharles:User {name: 'Charles', seed: 42}), |
| (nDoug:User {name: 'Doug'}), |
| (nMark:User {name: 'Mark'}), |
| (nMichael:User {name: 'Michael'}), |
|  |
| (nAlice)-[:LINK {weight: 1}]->(nBridget), |
| (nAlice)-[:LINK {weight: 1}]->(nCharles), |
| (nCharles)-[:LINK {weight: 1}]->(nBridget), |
|  |
| (nAlice)-[:LINK {weight: 5}]->(nDoug), |
|  |
| (nMark)-[:LINK {weight: 1}]->(nDoug), |
| (nMark)-[:LINK {weight: 1}]->(nMichael), |
| (nMichael)-[:LINK {weight: 1}]->(nMark); |



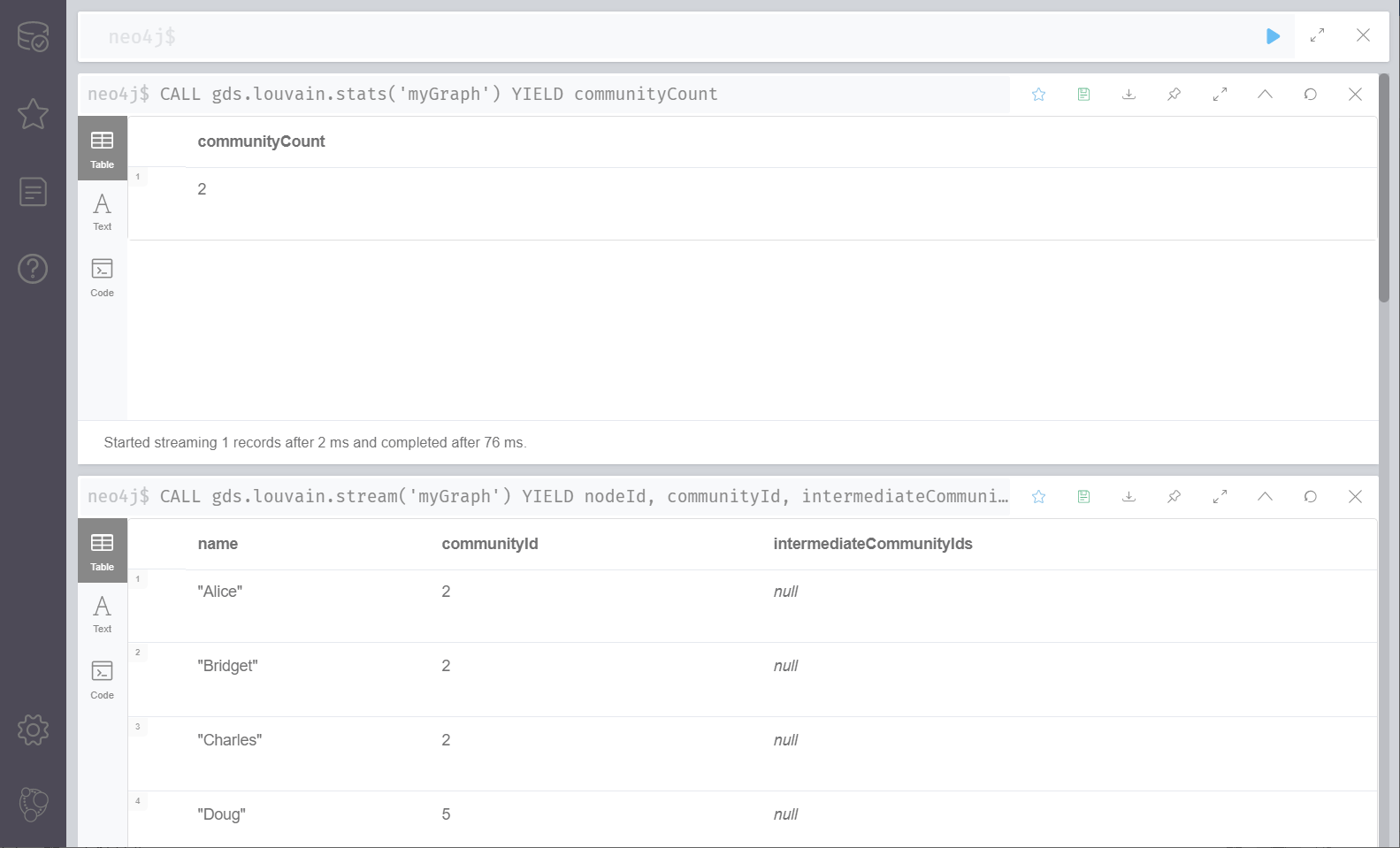
|  |
| --- |
| CALL gds.graph.create( |
| 'myGraph', |
| 'User', |
| { |
| LINK: { |
| orientation: 'UNDIRECTED' |
| } |
| }, |
| { |
| nodeProperties: 'seed', |
| relationshipProperties: 'weight' |
| } |
| ) |



|  |
| --- |
| CALL gds.louvain.stream('myGraph') |
| YIELD nodeId, communityId, intermediateCommunityIds |
| RETURN gds.util.asNode(nodeId).name AS name, communityId, intermediateCommunityIds |
| ORDER BY name ASC |



|  |
| --- |
| CALL gds.louvain.stats('myGraph') |
| YIELD communityCount |



|  |
| --- |
| CALL gds.louvain.mutate('myGraph', { mutateProperty: 'communityId' }) |
| YIELD communityCount, modularity, modularities |

