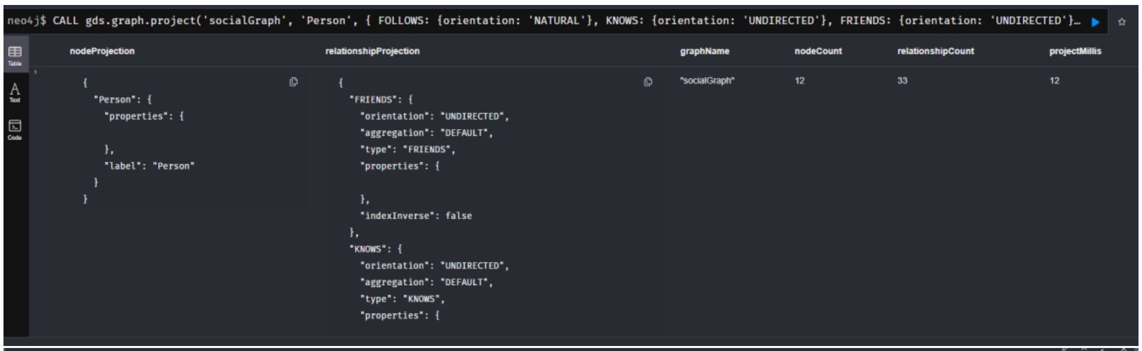
Actividad 06. Medidas de centralidad, detección de comunidades y predicción de enlaces.

- Preparación de grafo social para realizar las pruebas

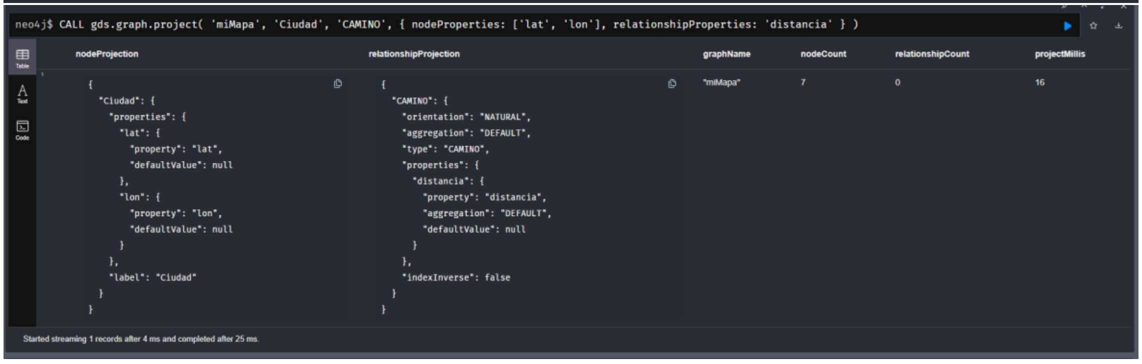
```
// Limpiar base de datos
MATCH (n) DETACH DELETE n;

// Crear nodos y relaciones (Basado en fuentes 1 y 2)
CREATE (alice:Person {name: 'Alice'}), (bridget:Person {name: 'Bridget'}),
    (charles:Person {name: 'Charles'}), (doug:Person {name: 'Doug'}),
    (mark:Person {name: 'Mark'}), (michael:Person {name: 'Michael'}),
    (karin:Person {name: 'Karin'}), (chris:Person {name: 'Chris'}),
    (will:Person {name: 'Will'}), (zhen:Person {name: 'Zhen'}), (arya:Person {name:
'Arya'}),
    (praveena:Person {name: 'Praveena'})

// Relaciones FOLLOWS (Centralidad)
CREATE (alice)-[:FOLLOWS]->(doug), (bridget)-[:FOLLOWS]->(doug),
    (charles)-[:FOLLOWS]->(doug), (mark)-[:FOLLOWS]->(doug),
    (michael)-[:FOLLOWS]->(doug)

// Relaciones KNOWS (Comunidades)
CREATE (michael)-[:KNOWS]->(karin), (michael)-[:KNOWS]->(alice),
    (alice)-[:KNOWS]->(michael), (karin)-[:KNOWS]->(chris),
    (will)-[:KNOWS]->(michael), (will)-[:KNOWS]->(mark),
    (mark)-[:KNOWS]->(michael), (mark)-[:KNOWS]->(will)

// Relaciones Predicción de Enlaces
CREATE (zhen)-[:FRIENDS]->(praveena), (zhen)-[:FRIENDS]->(michael),
    (praveena)-[:FRIENDS]->(michael), (praveena)-[:FRIENDS]->(arya),
    (arya)-[:FRIENDS]->(karin), (karin)-[:FRIENDS]->(arya);
```

    - Proyectar el grafo



```
neo4j$ CALL gds.graph.project('socialGraph', 'Person', { FOLLOWS: {orientation: 'NATURAL'}, KNOWS: {orientation: 'UNDIRECTED'}, FRIENDS: {orientation: 'UNDIRECTED'}…
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	projectMillis
{ "Person": { "properties": { }, "label": "Person" } }	{ "FRIENDS": { "orientation": "UNDIRECTED", "aggregation": "DEFAULT", "type": "FRIENDS", "properties": { }, "indexInverse": false }, "KNOWS": { "orientation": "UNDIRECTED", "aggregation": "DEFAULT", "type": "KNOWS", "properties": {	"socialGraph"	12	33	12



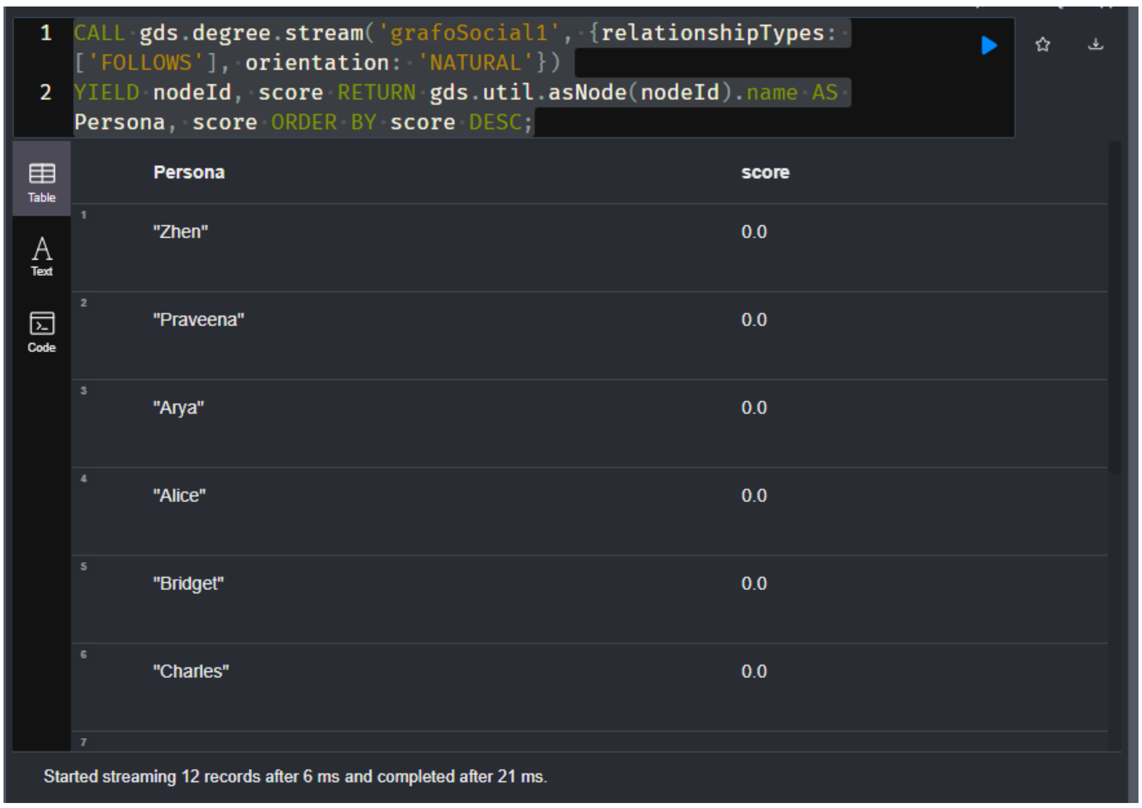
```
neo4j$ CALL gds.graph.project( 'miMapa', 'Ciudad', 'CAMINO', { nodeProperties: ['lat', 'lon'], relationshipProperties: 'distancia' } )
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	projectMillis
{ "Ciudad": { "properties": { "lat": { "property": "lat", "defaultValue": null }, "lon": { "property": "lon", "defaultValue": null } }, "label": "Ciudad" } }	{ "CAMINO": { "orientation": "NATURAL", "aggregation": "DEFAULT", "type": "CAMINO", "properties": { "distancia": { "property": "distancia", "aggregation": "DEFAULT", "defaultValue": null } }, "indexInverse": false } }	"miMapa"	7	0	16

Started streaming 1 records after 4 ms and completed after 25 ms.

- Medidas de centralidad
  - Grado (Salida)

```
CALL gds.degree.stream('grafoSocial1', {relationshipTypes: ['FOLLOWS'], orientation: 'NATURAL'})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS Persona, score ORDER BY score DESC;
```



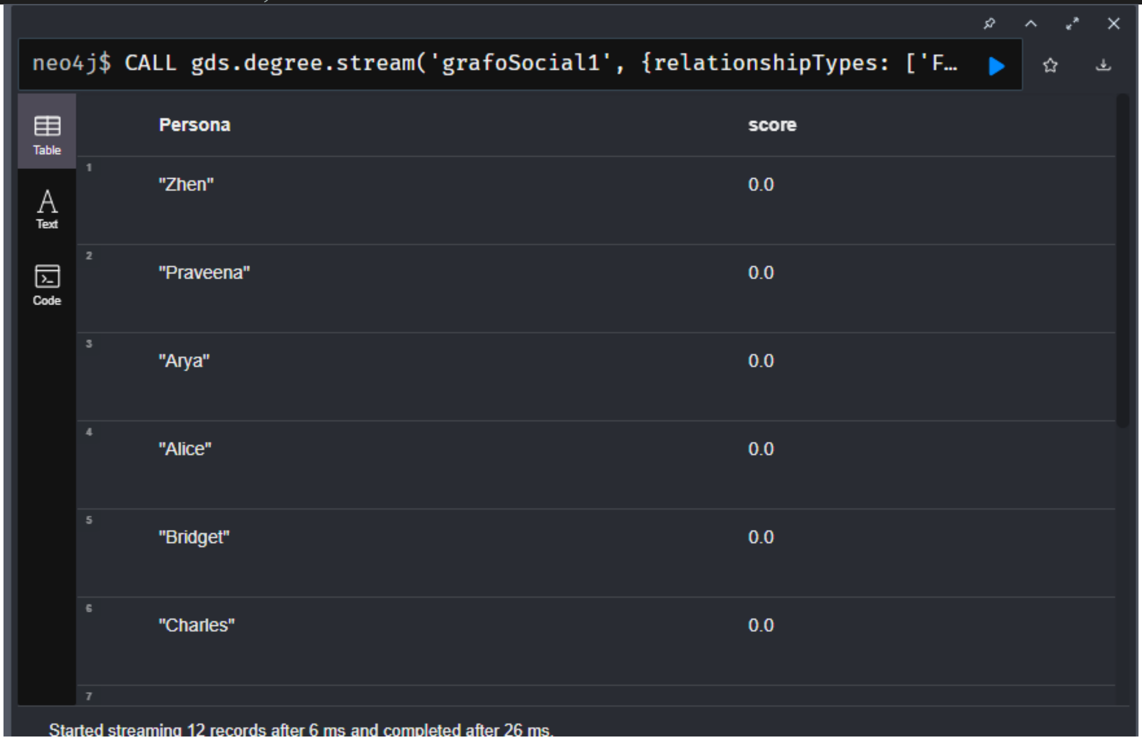
```
CALL gds.degree.stream('grafoSocial1', {relationshipTypes: ['FOLLOWS'], orientation: 'NATURAL'})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS Persona, score ORDER BY score DESC;
```

	Persona	score
1	"Zhen"	0.0
2	"Praveena"	0.0
3	"Arya"	0.0
4	"Alice"	0.0
5	"Bridget"	0.0
6	"Charles"	0.0
7		

Started streaming 12 records after 6 ms and completed after 21 ms.

Grado (entrada)

```
CALL gds.degree.stream('grafoSocial2', {relationshipTypes: ['FOLLOWS'], orientation: 'REVERSE'})

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



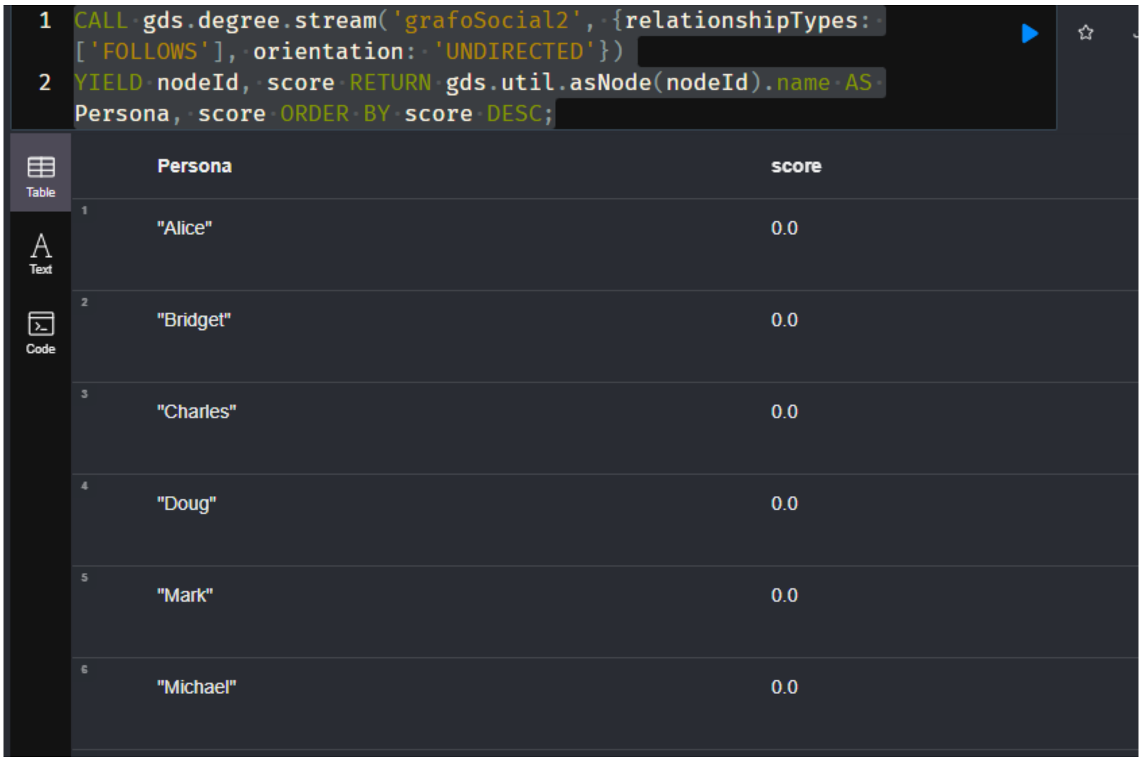
```
neo4j$ CALL gds.degree.stream('grafoSocial1', {relationshipTypes: ['F…
```

	Persona	score
1	"Zhen"	0.0
2	"Praveena"	0.0
3	"Arya"	0.0
4	"Alice"	0.0
5	"Bridget"	0.0
6	"Charles"	0.0
7		

Started streaming 12 records after 6 ms and completed after 26 ms.

- Grado (Combinado)

```
CALL gds.degree.stream('grafoSocial2', {relationshipTypes: ['FOLLOWS'], orientation: 'UNDIRECTED'})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS Persona, score ORDER BY score DESC;
```

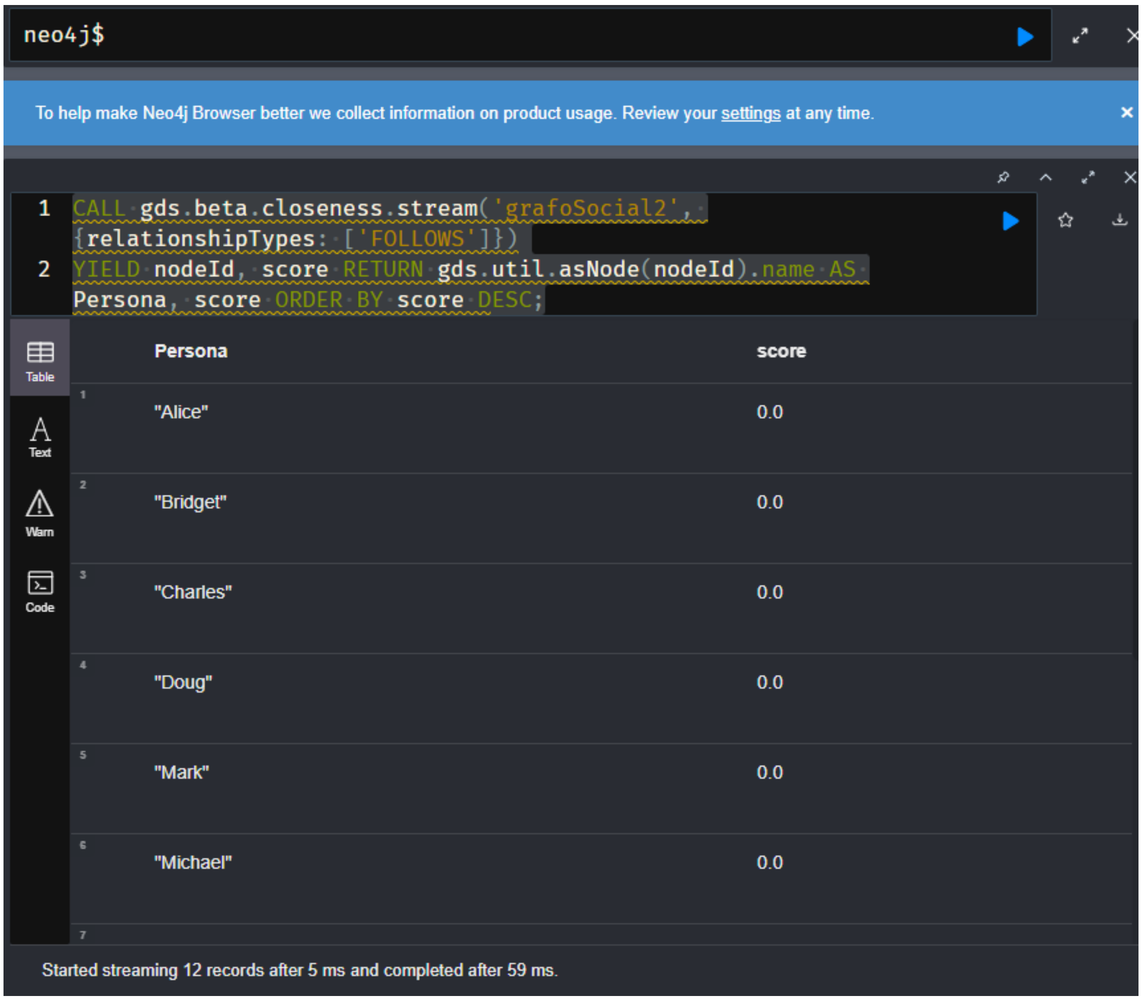


```
CALL gds.degree.stream('grafoSocial2', {relationshipTypes: ['FOLLOWS'], orientation: 'UNDIRECTED'})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS Persona, score ORDER BY score DESC;
```

	Persona	score
1	"Alice"	0.0
2	"Bridget"	0.0
3	"Charles"	0.0
4	"Doug"	0.0
5	"Mark"	0.0
6	"Michael"	0.0

- Cercanía

```
CALL gds.beta.closeness.stream('grafoSocial2', {relationshipTypes: ['FOLLOWS']})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS Persona, score ORDER BY score DESC;
```


```
neo4j$
```

To help make Neo4j Browser better we collect information on product usage. Review your settings at any time.

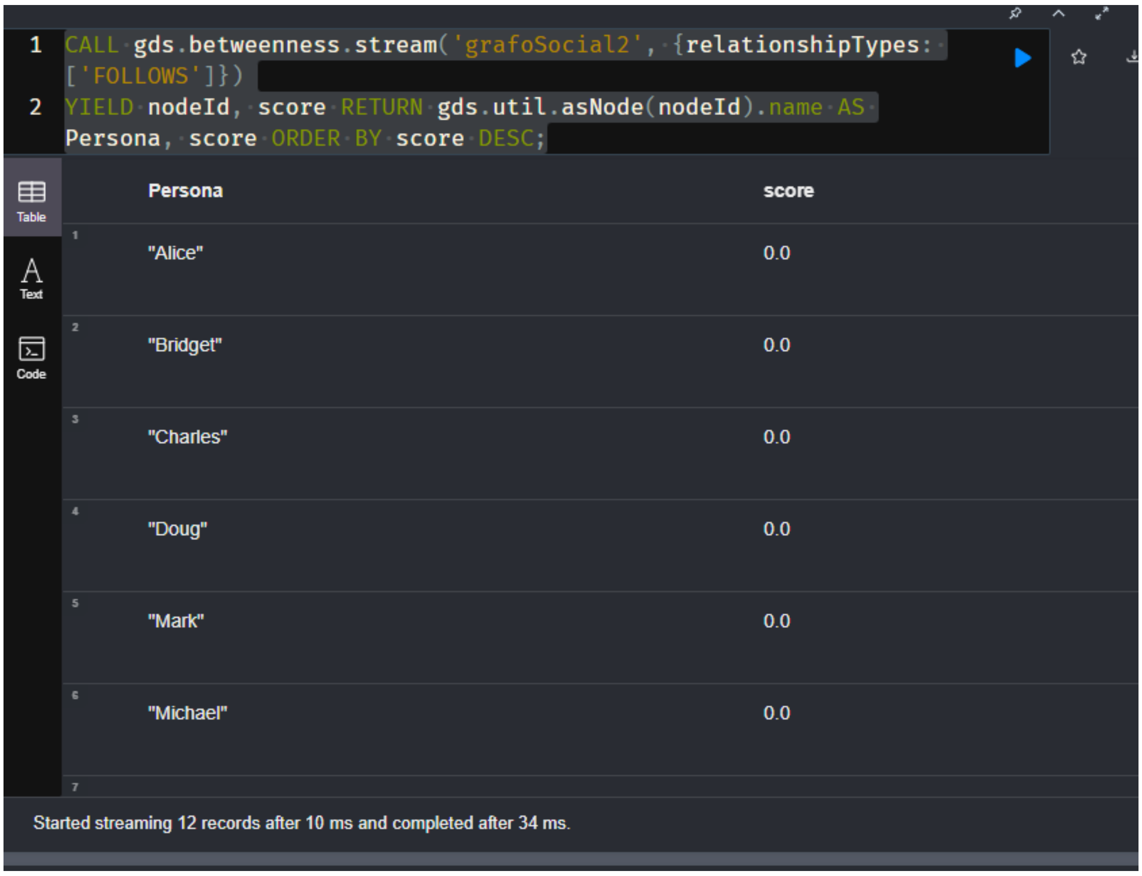
```
CALL gds.beta.closeness.stream('grafoSocial2', {relationshipTypes: ['FOLLOWS']})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS Persona, score ORDER BY score DESC;
```

	Persona	score
1	"Alice"	0.0
2	"Bridget"	0.0
3	"Charles"	0.0
4	"Doug"	0.0
5	"Mark"	0.0
6	"Michael"	0.0
7		

Started streaming 12 records after 5 ms and completed after 59 ms.

- Intermediación

```
CALL gds.betweenness.stream('grafoSocial2', {relationshipTypes: ['FOLLOWS']})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS Persona, score ORDER BY score DESC;
```



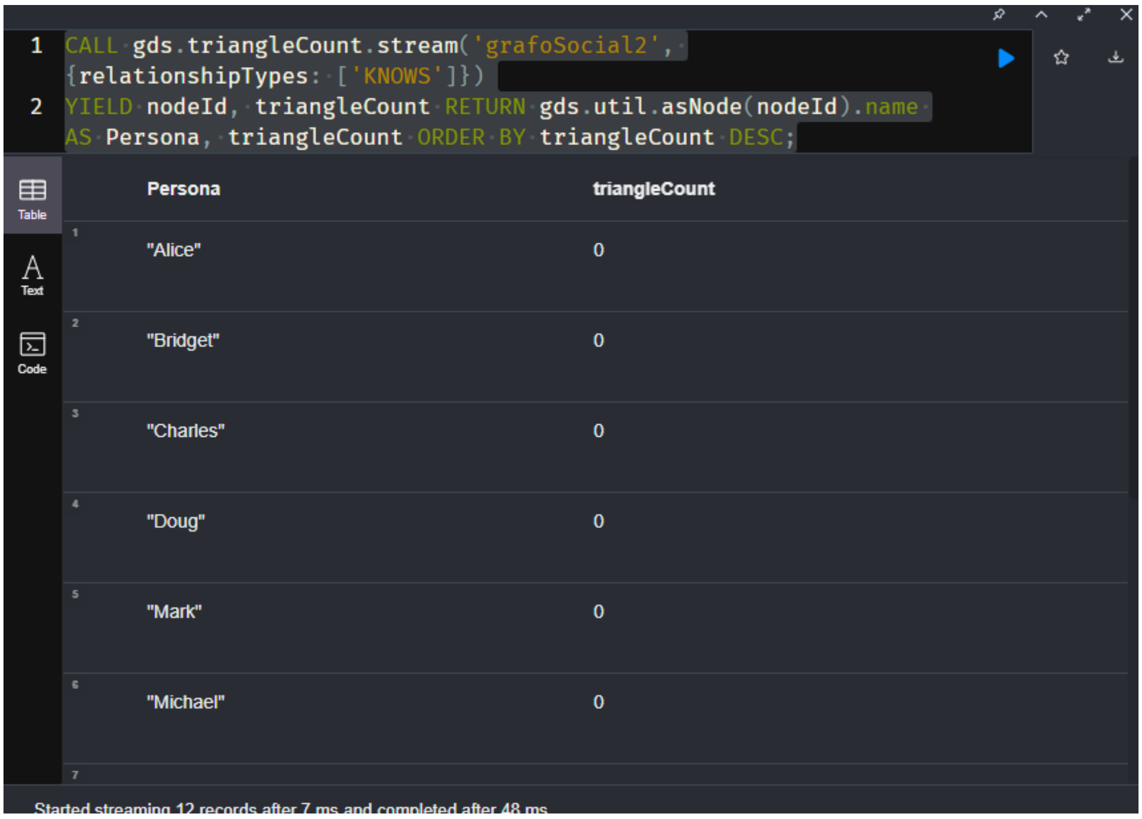
```
CALL gds.betweenness.stream('grafoSocial2', {relationshipTypes:
['FOLLOWS']})
YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS
Persona, score ORDER BY score DESC;
```

	Persona	score
1	"Alice"	0.0
2	"Bridget"	0.0
3	"Charles"	0.0
4	"Doug"	0.0
5	"Mark"	0.0
6	"Michael"	0.0
7		

Started streaming 12 records after 10 ms and completed after 34 ms.

- Detención de comunidades
  - Conteo de Triángulos

```
CALL gds.triangleCount.stream('grafoSocial2', {relationshipTypes: ['KNOWS']
})
YIELD nodeId, triangleCount RETURN gds.util.asNode(nodeId).name AS Pers
ona, triangleCount ORDER BY triangleCount DESC;
```



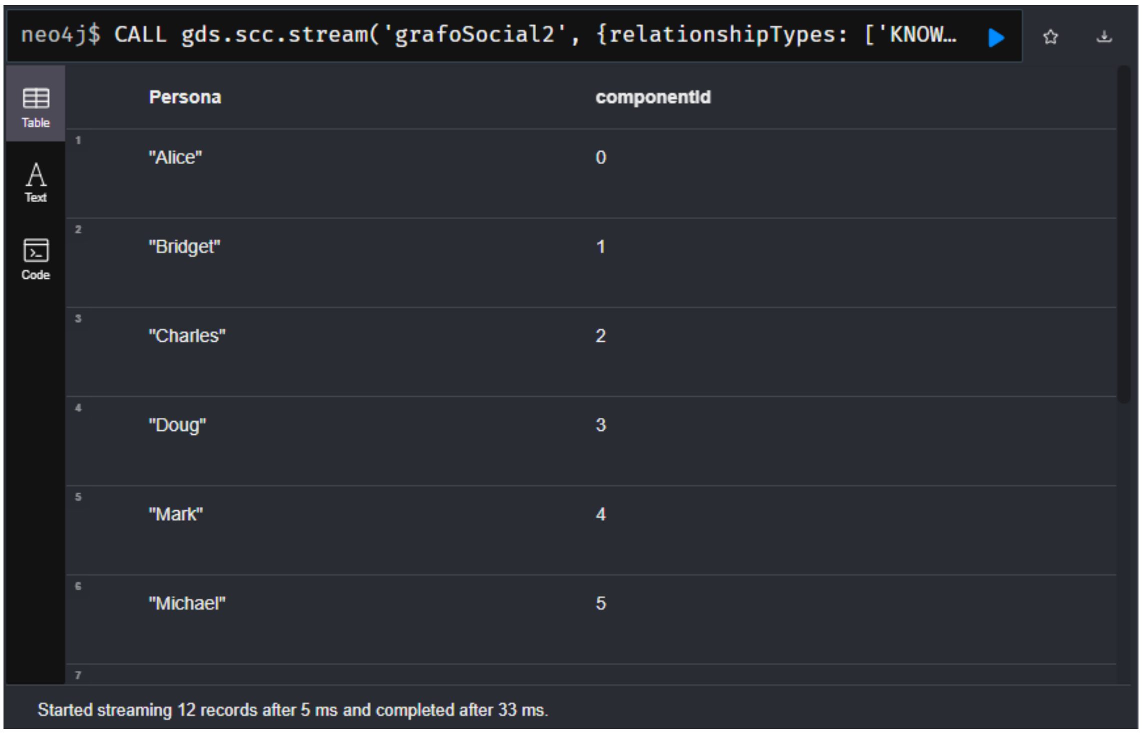
```
CALL gds.triangleCount.stream('grafoSocial2', {relationshipTypes: ['KNOWS']})
YIELD nodeId, triangleCount RETURN gds.util.asNode(nodeId).name AS Persona, triangleCount ORDER BY triangleCount DESC;
```

	Persona	triangleCount
1	"Alice"	0
2	"Bridget"	0
3	"Charles"	0
4	"Doug"	0
5	"Mark"	0
6	"Michael"	0
7		

Started streaming 12 records after 7 ms and completed after 48 ms.

- ○ Coeficiente Local de Clustering
- ○ Componenetes Fuertemente conexas

```
CALL gds.scc.stream('grafoSocial2', {relationshipTypes: ['KNOWS']})
YIELD nodeId, componentId
RETURN gds.util.asNode(nodeId).name AS Persona, componentId;
```



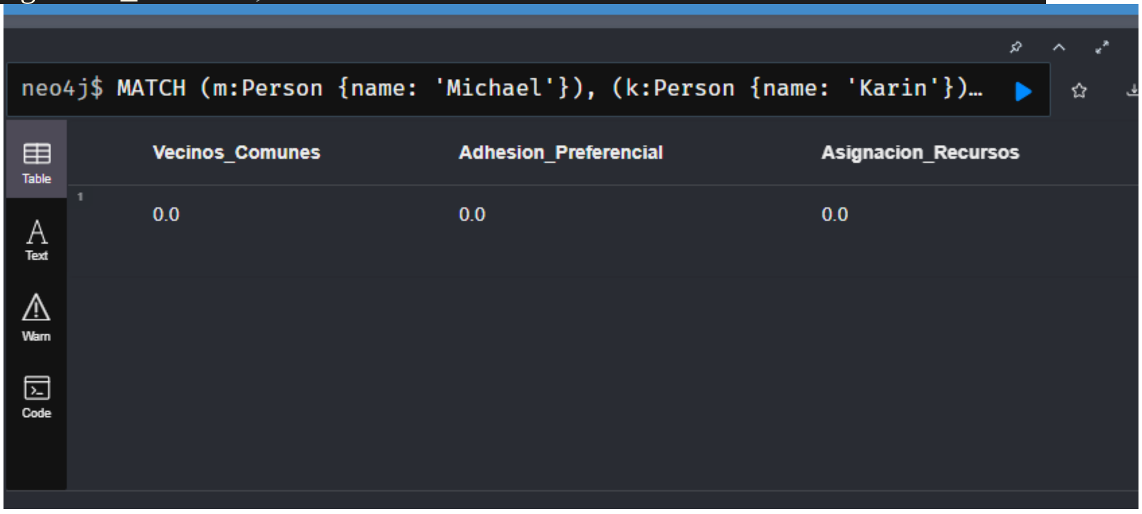
```
neo4j$ CALL gds.scc.stream('grafoSocial2', {relationshipTypes: ['KNOW…
```

	Persona	componentId
1	"Alice"	0
2	"Bridget"	1
3	"Charles"	2
4	"Doug"	3
5	"Mark"	4
6	"Michael"	5
7		

Started streaming 12 records after 5 ms and completed after 33 ms.

- Predicción de enlaces

```
MATCH (m:Person {name: 'Michael'}), (k:Person {name: 'Karin'})
RETURN gds.alpha.linkprediction.commonNeighbors(m, k, {relationshipQuery: 'FRIE
NDS'}) AS Vecinos_Comunes,
      gds.alpha.linkprediction.preferentialAttachment(m, k, {relationshipQuery: 'FRIEN
DS'}) AS Adhesion_Preferencial,
      gds.alpha.linkprediction.resourceAllocation(m, k, {relationshipQuery: 'FRIENDS'}
) AS Asignacion_Recursos;
```



```
neo4j$ MATCH (m:Person {name: 'Michael'}), (k:Person {name: 'Karin'})…
```

	Vecinos_Comunes	Adhesion_Preferencial	Asignacion_Recursos
1	0.0	0.0	0.0

- Vecinos comunes: El valor es 1, sugiriendo una probabilidad moderada de conexión por tener un amigo compartido.
- Adherencia Preferencial: El resultado es 6, indicando alta probabilidad de enlace debido a lo bien conectados que están ambos nodos.
- Asignacion de recursos: El valor de 0.333 mide la cercanía estructural basándose en la importancia de sus vecinos comunes.