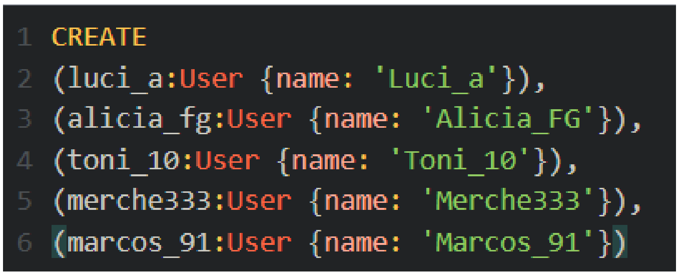
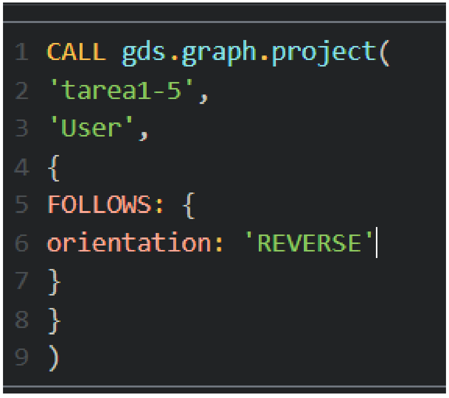
SOLUCIÓN TAREA 1-5

Creamos los nodos



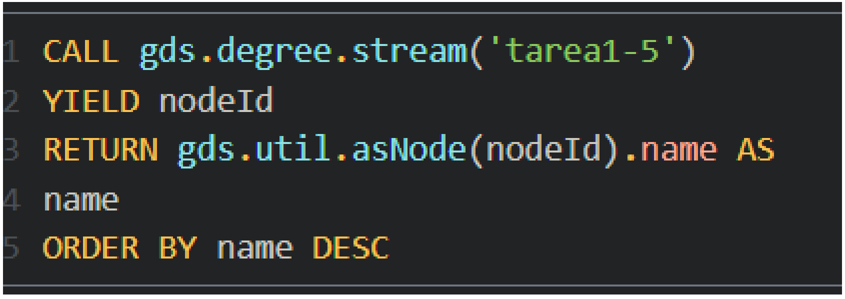
```
CREATE
(luci_a:User {name: 'Luci_a'}),
(alicia_fg:User {name: 'Alicia_FG'}),
(toni_10:User {name: 'Toni_10'}),
(merche333:User {name: 'Merche333'}),
(marcos_91:User {name: 'Marcos_91'})
```

Lo guardamos

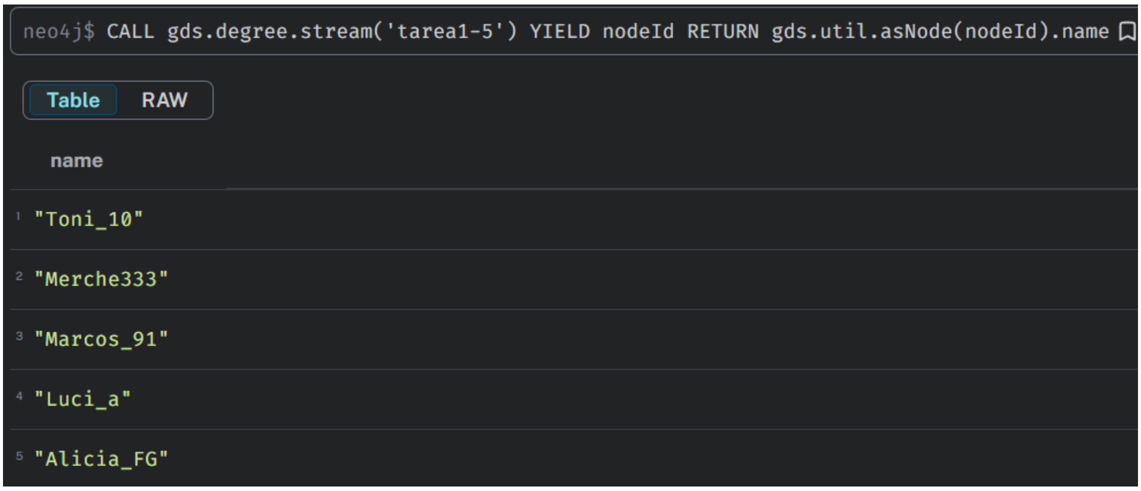


```
CALL gds.graph.project(
'tarea1-5',
'User',
{
FOLLOWS: {
orientation: 'REVERSE'
}
}
)
```

Calculamos las medidas de centralidad de grado



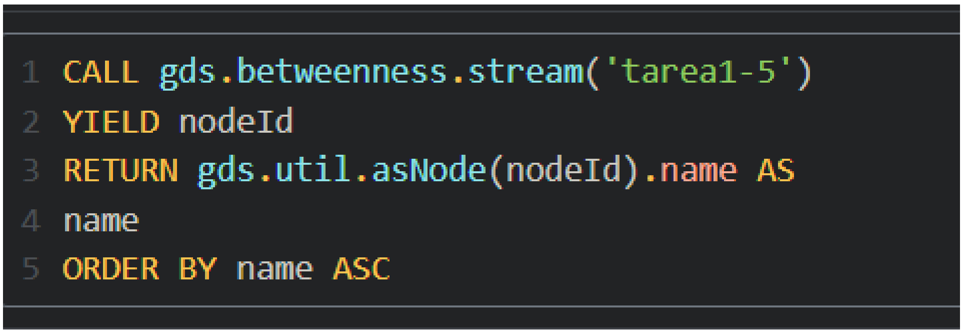
```
CALL gds.degree.stream('tarea1-5')
YIELD nodeId
RETURN gds.util.asNode(nodeId).name AS
name
ORDER BY name DESC
```



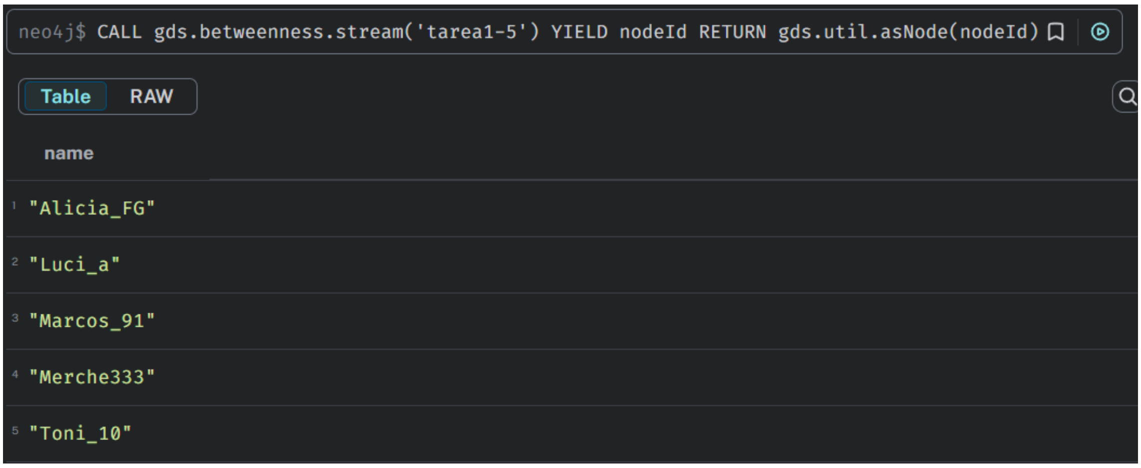
```
neo4j$ CALL gds.degree.stream('tarea1-5') YIELD nodeId RETURN gds.util.asNode(nodeId).name
```

```
name
1 "Toni_10"
2 "Merche333"
3 "Marcos_91"
4 "Luci_a"
5 "Alicia_FG"
```

Calculamos la intermediación para casa uno de los nodos



```
CALL gds.betweenness.stream('tarea1-5')
YIELD nodeId
RETURN gds.util.asNode(nodeId).name AS
name
ORDER BY name ASC
```

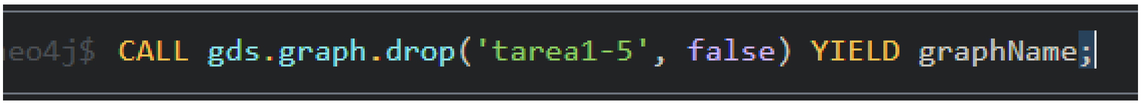


```
neo4j$ CALL gds.betweenness.stream('tarea1-5') YIELD nodeId RETURN gds.util.asNode(nodeId)
```

	name
1	"Alicia_FG"
2	"Luci_a"
3	"Marcos_91"
4	"Merche333"
5	"Toni_10"

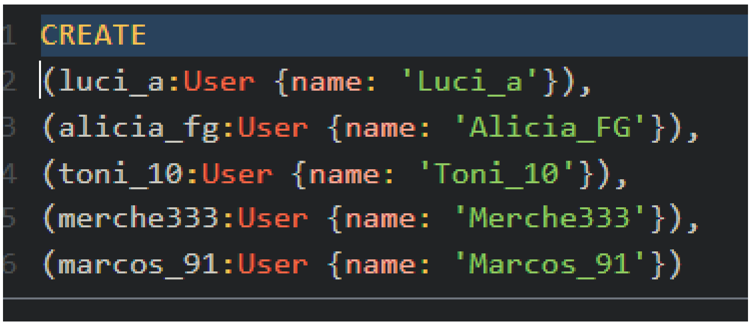
Conteo de triángulos

- Borramos el proyecto ya creado



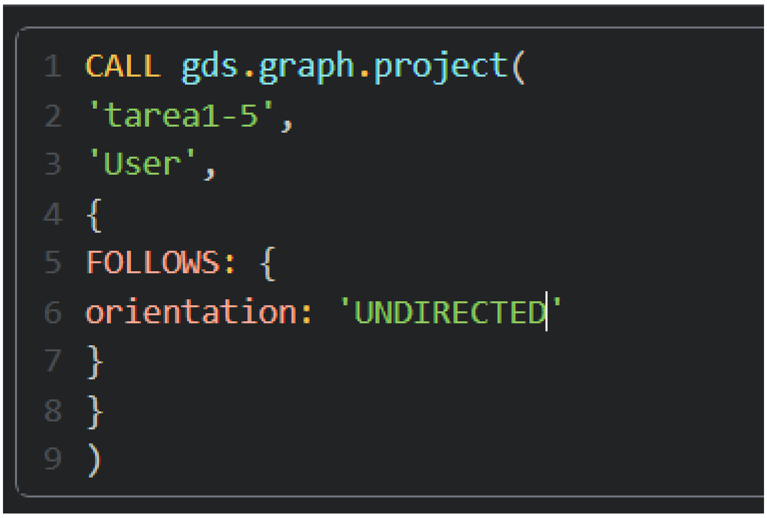
```
eo4j$ CALL gds.graph.drop('tarea1-5', false) YIELD graphName;
```

- Volvemos a crear los nodos



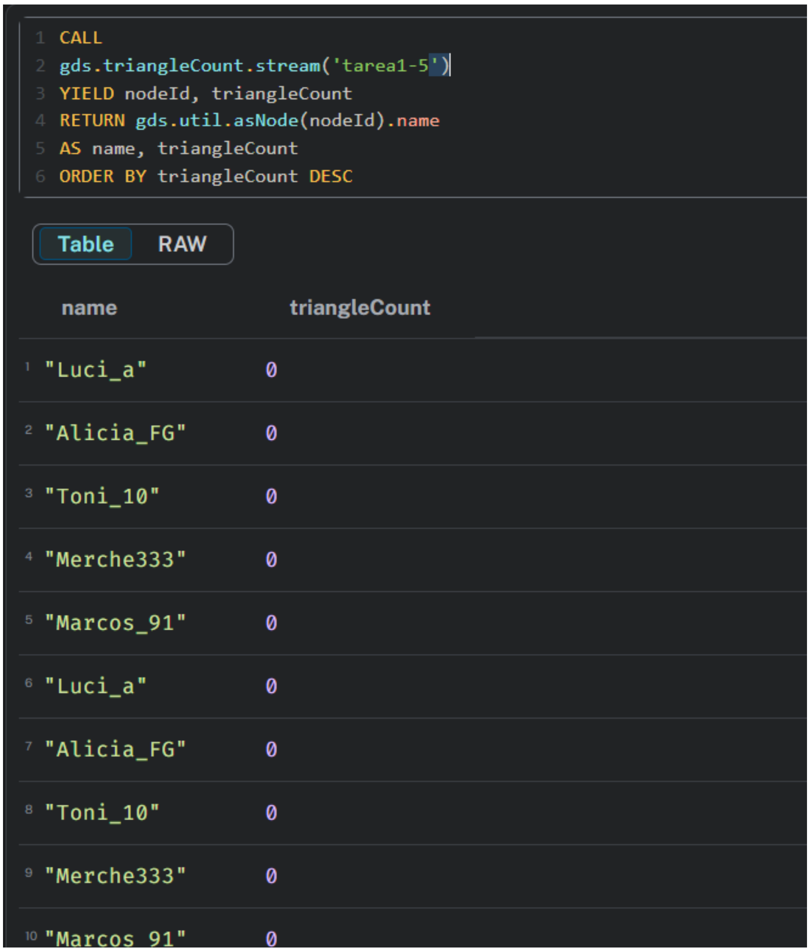
```
CREATE
(luci_a:User {name: 'Luci_a'}),
(alicia_fg:User {name: 'Alicia_FG'}),
(toni_10:User {name: 'Toni_10'}),
(merche333:User {name: 'Merche333'}),
(marcos_91:User {name: 'Marcos_91'})
```

- Guardamos el proyecto con orientation “undirected”



```
CALL gds.graph.project(
'tarea1-5',
'User',
{
FOLLOWS: {
orientation: 'UNDIRECTED'
}
}
)
```

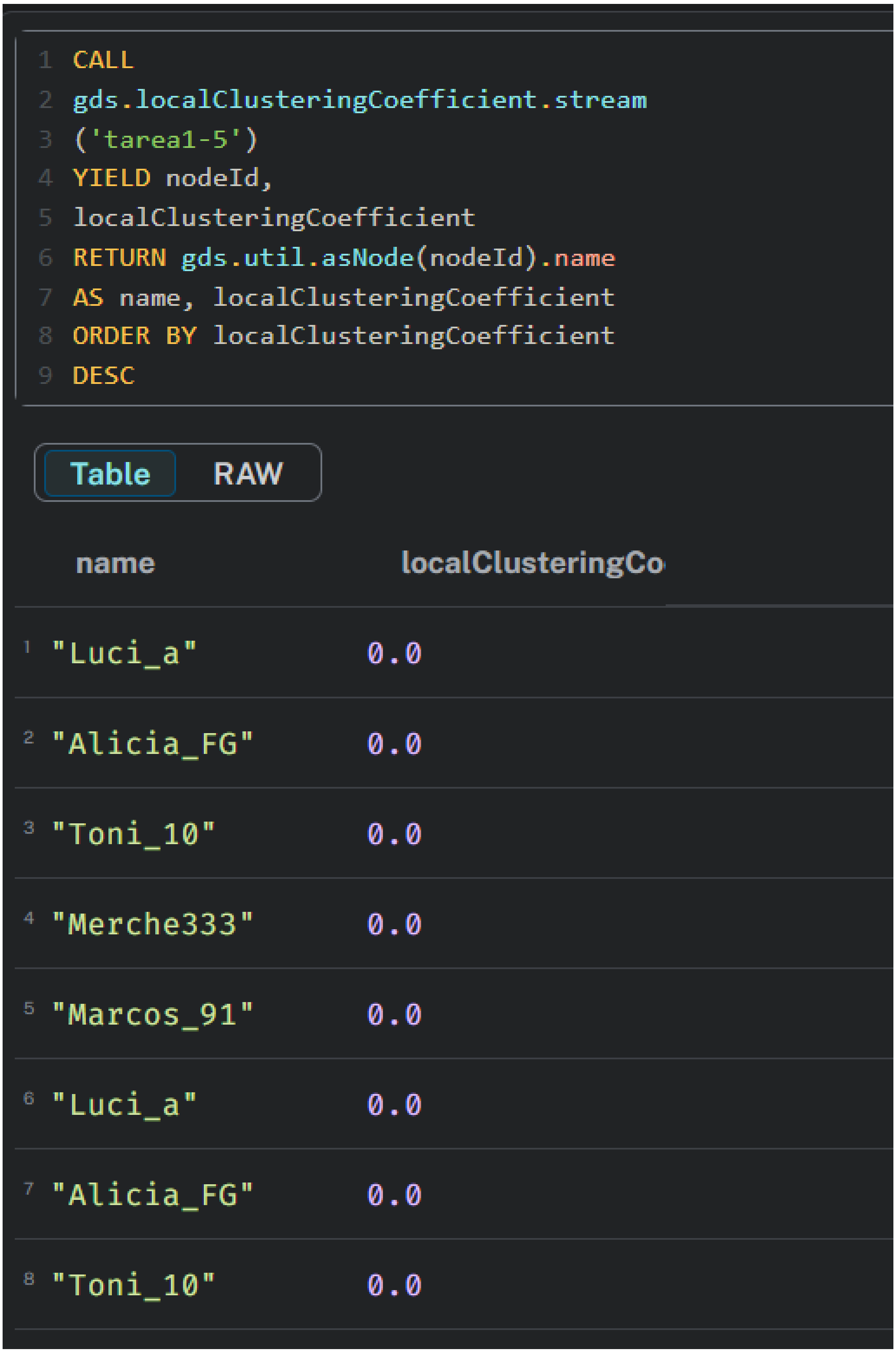
- Llamamos a la función para contear los triángulos



```
CALL
gds.triangleCount.stream('tarea1-5')
YIELD nodeId, triangleCount
RETURN gds.util.asNode(nodeId).name
AS name, triangleCount
ORDER BY triangleCount DESC
```

	name	triangleCount
1	"Luci_a"	0
2	"Alicia_FG"	0
3	"Toni_10"	0
4	"Merche333"	0
5	"Marcos_91"	0
6	"Luci_a"	0
7	"Alicia_FG"	0
8	"Toni_10"	0
9	"Merche333"	0
10	"Marcos_91"	0

Calculamos el coeficiente local de clustering

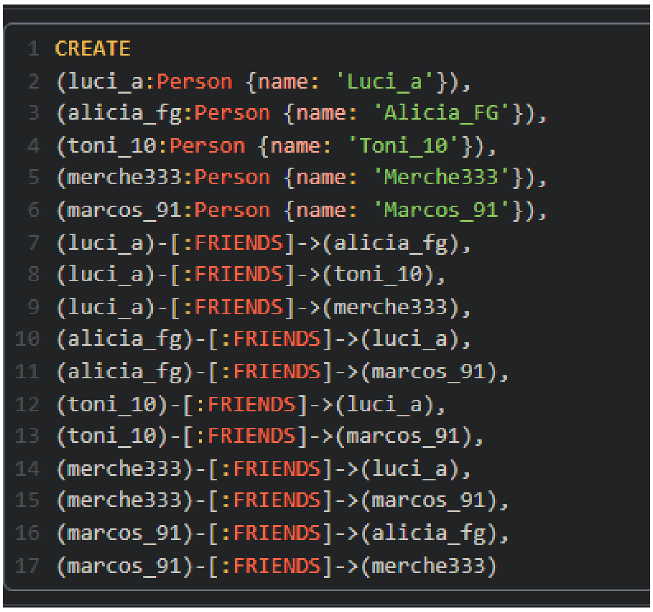


```
CALL
gds.localClusteringCoefficient.stream
('tarea1-5')
YIELD nodeId,
localClusteringCoefficient
RETURN gds.util.asNode(nodeId).name
AS name, localClusteringCoefficient
ORDER BY localClusteringCoefficient
DESC
```

	name	localClusteringCo
1	"Luci_a"	0.0
2	"Alicia_FG"	0.0
3	"Toni_10"	0.0
4	"Merche333"	0.0
5	"Marcos_91"	0.0
6	"Luci_a"	0.0
7	"Alicia_FG"	0.0
8	"Toni_10"	0.0

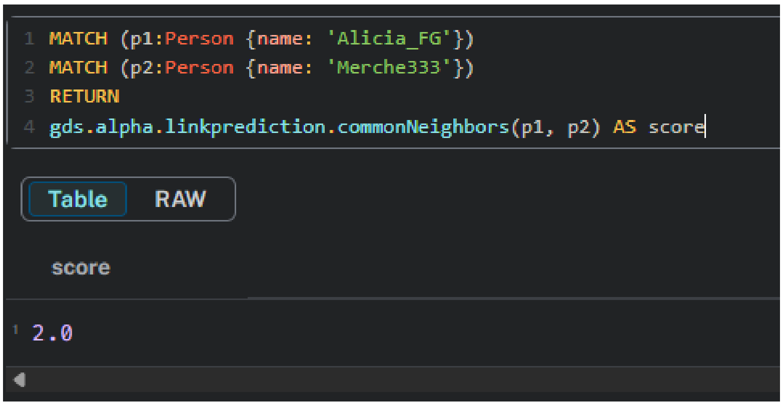
Posibilidad de que se produzca un enlace entre Alicia_FG y Merche333

- Volvemos a crear los nodos con sus amigos



```
CREATE
(luci_a:Person {name: 'Luci_a'}),
(alicia_fg:Person {name: 'Alicia_FG'}),
(toni_10:Person {name: 'Toni_10'}),
(merche333:Person {name: 'Merche333'}),
(marcos_91:Person {name: 'Marcos_91'}),
(luci_a)-[:FRIENDS]->(alicia_fg),
(luci_a)-[:FRIENDS]->(toni_10),
(luci_a)-[:FRIENDS]->(merche333),
(alicia_fg)-[:FRIENDS]->(luci_a),
(alicia_fg)-[:FRIENDS]->(marcos_91),
(toni_10)-[:FRIENDS]->(luci_a),
(toni_10)-[:FRIENDS]->(marcos_91),
(merche333)-[:FRIENDS]->(luci_a),
(merche333)-[:FRIENDS]->(marcos_91),
(marcos_91)-[:FRIENDS]->(alicia_fg),
(marcos_91)-[:FRIENDS]->(merche333)
```

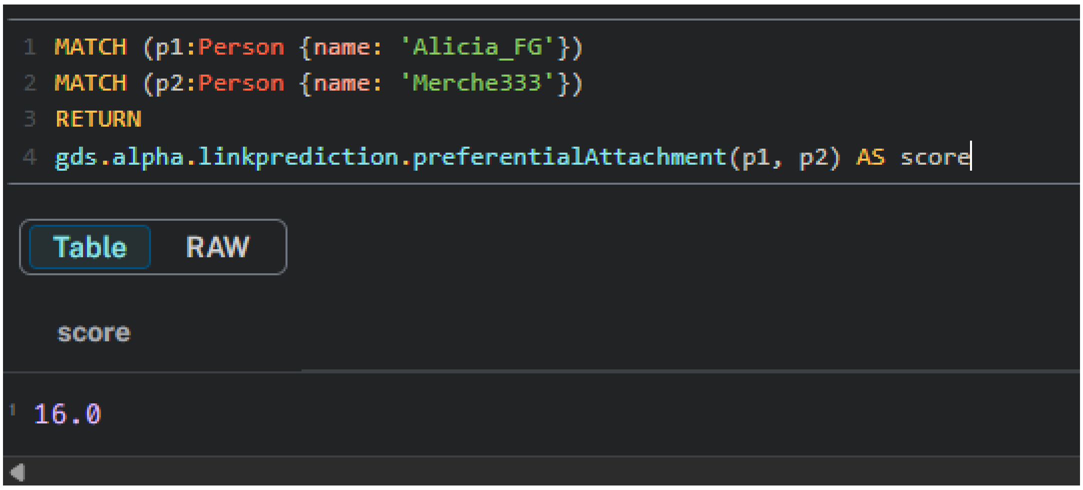
- Método vecinos comunes



```
MATCH (p1:Person {name: 'Alicia_FG'})
MATCH (p2:Person {name: 'Merche333'})
RETURN
gds.alpha.linkprediction.commonNeighbors(p1, p2) AS score
```

```
score
1 2.0
```

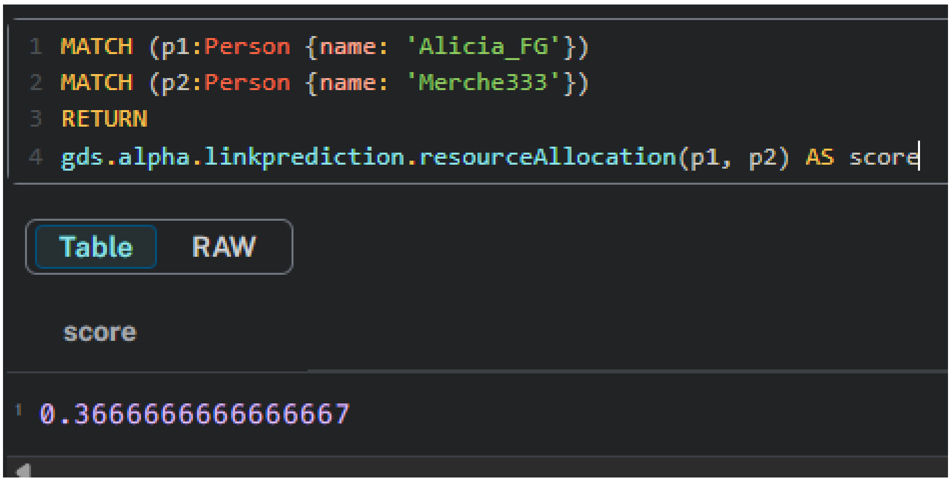
- Método de adhesión preferencial



```
MATCH (p1:Person {name: 'Alicia_FG'})
MATCH (p2:Person {name: 'Merche333'})
RETURN
gds.alpha.linkprediction.preferentialAttachment(p1, p2) AS score
```

```
score
1 16.0
```

- Método de asignación de recursos

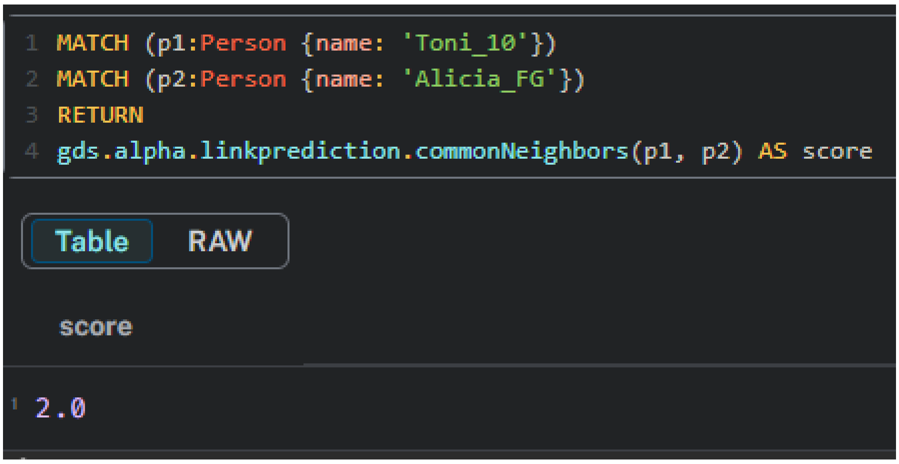


```
MATCH (p1:Person {name: 'Alicia_FG'})
MATCH (p2:Person {name: 'Merche333'})
RETURN
gds.alpha.linkprediction.resourceAllocation(p1, p2) AS score
```

```
score
0.3666666666666667
```

Posibilidad de que se produzca un enlace entre Toni_10 y Alicia_FG

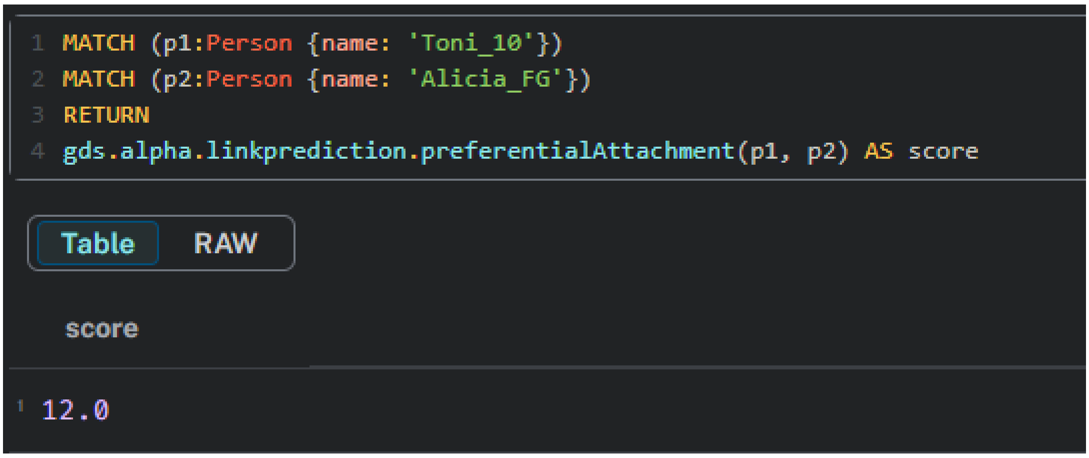
- Método de vecinos comunes



```
MATCH (p1:Person {name: 'Toni_10'})
MATCH (p2:Person {name: 'Alicia_FG'})
RETURN
gds.alpha.linkprediction.commonNeighbors(p1, p2) AS score
```

```
score
2.0
```

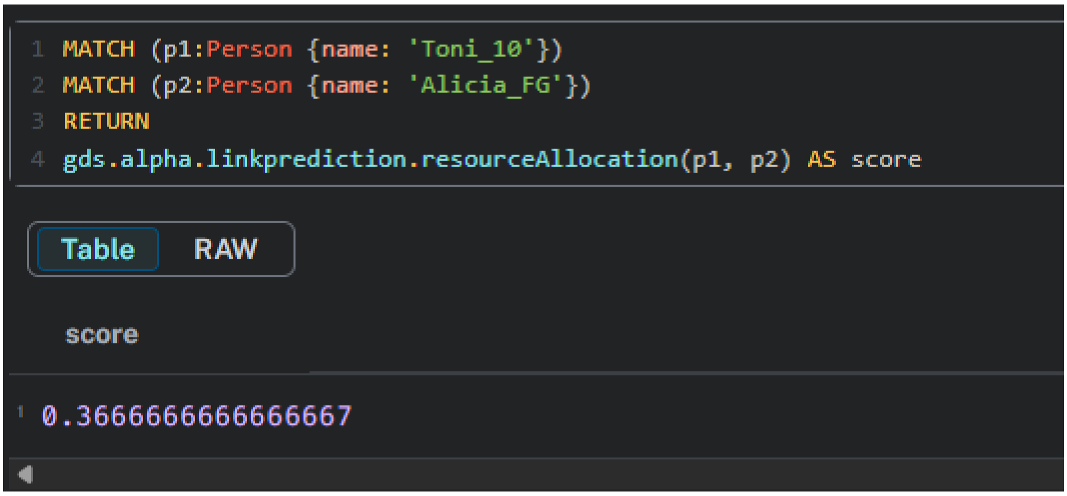
- Método de adhesión preferencial



```
MATCH (p1:Person {name: 'Toni_10'})
MATCH (p2:Person {name: 'Alicia_FG'})
RETURN
gds.alpha.linkprediction.preferentialAttachment(p1, p2) AS score
```

```
score
12.0
```

- Método de asignación de recursos



```
MATCH (p1:Person {name: 'Toni_10'})
MATCH (p2:Person {name: 'Alicia_FG'})
RETURN
gds.alpha.linkprediction.resourceAllocation(p1, p2) AS score
```

```
score
0.3666666666666667
```