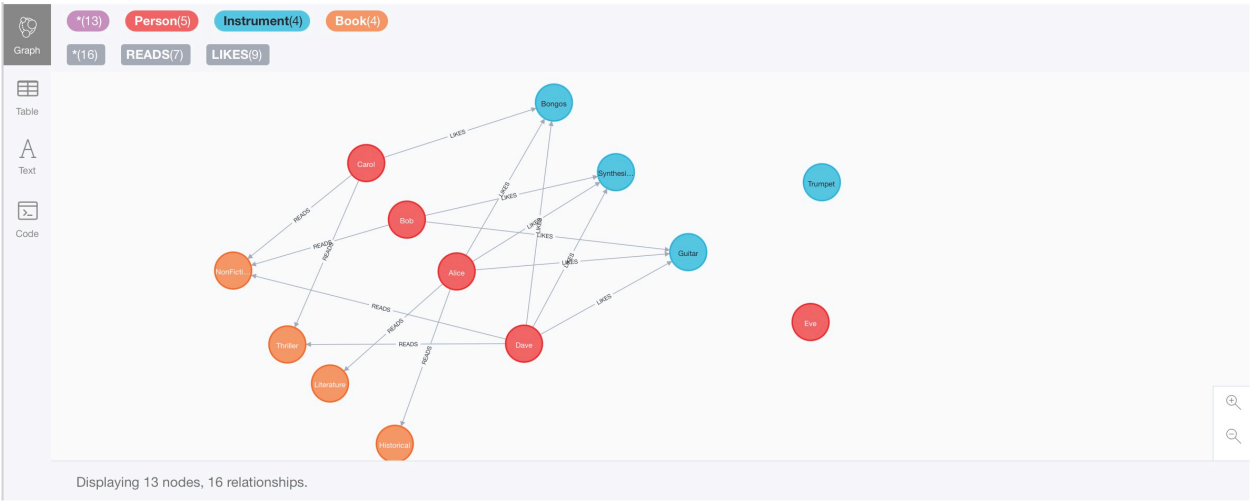
Question: 1 (4 points)
Create this graph in the Neo4j blank sandbox.
Paste the query that can render the entire graph - that is all different types of nodes with all different types of relationships in the CREATE statement above. (Note - start with a blank sandbox or make sure all entities, relationships created earlier are deleted.)

Answer:

```
match (n) return n
```

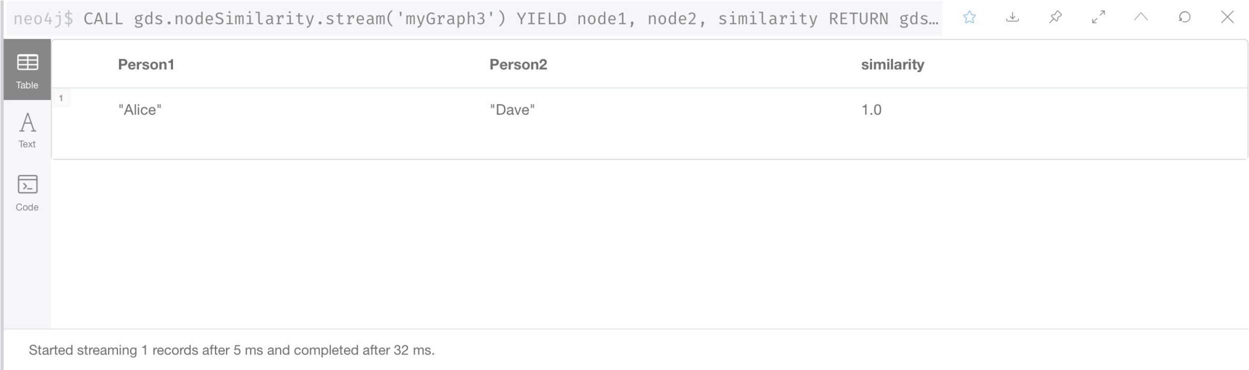


Displaying 13 nodes, 16 relationships.

Question: 2 (8 points)
Create a graph projection that only focuses on Instrument and Person and find out which 2 persons are most similar to each other.
Use https://neo4j.com/docs/graph-data-science/current/algorithms/node-similarity/
Please paste the CALL gds.graph.create command and the CALL gds graph analytics algorithm

Answer:

```
CALL gds.graph.create('myGraph3',
['Person', 'Instrument'], {
LIKES: {
type: 'LIKES'
} }
);
CALL gds.nodeSimilarity.stream('myGraph3')
YIELD node1, node2, similarity
RETURN gds.util.asNode(node1).name AS Person1,
gds.util.asNode(node2).name AS Person2, similarity
ORDER BY similarity DESCENDING, Person1, Person2
LIMIT 1
```



```
neo4j$ CALL gds.nodeSimilarity.stream('myGraph3') YIELD node1, node2, similarity RETURN gds…
```

	Person1	Person2	similarity
1	"Alice"	"Dave"	1.0

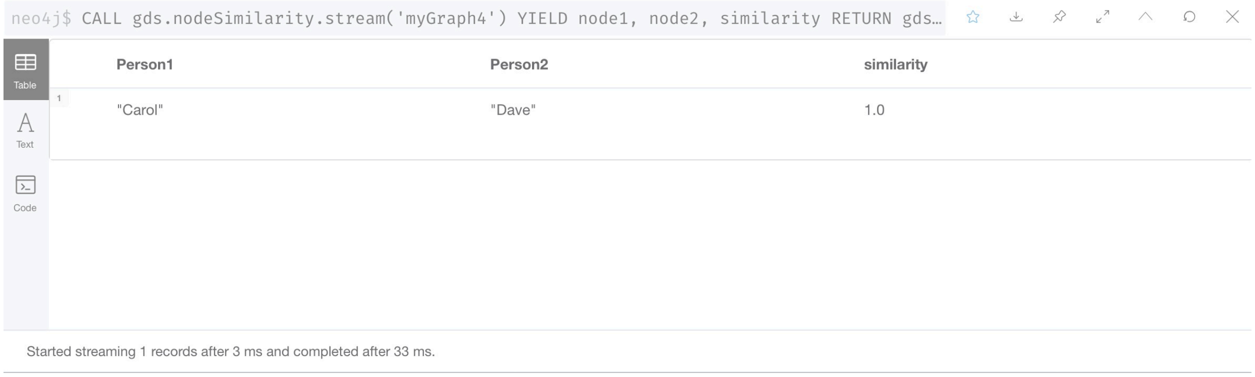
Started streaming 1 records after 5 ms and completed after 32 ms.

Question: 3 (8 points)

Create a graph projection that only focuses on Book and Person and find out which 2 persons are most similar to each other. Use https://neo4j.com/docs/graph-data-science/current/algorithms/node-similarity/
Please paste the CALL gds.graph.create command and the CALL gds graph analytics algorithm

Answer:

```
CALL gds.graph.create('myGraph4',
['Person', 'Book'], {
READS: {
type: 'READS'
} }
);
CALL gds.nodeSimilarity.stream('myGraph4')
YIELD node1, node2, similarity
RETURN gds.util.asNode(node1).name AS Person1,
gds.util.asNode(node2).name AS Person2, similarity ORDER BY
similarity DESCENDING, Person1, Person2
```



```
neo4j$ CALL gds.nodeSimilarity.stream('myGraph4') YIELD node1, node2, similarity RETURN gds…
```

	Person1	Person2	similarity
1	"Carol"	"Dave"	1.0

Started streaming 1 records after 3 ms and completed after 33 ms.