CS 343 Graph Data Science

- Create a projection for Movie database for Person and Movie labels and only ACTED_IN relationship without direction
- Calculate the degree for each node
- Calculate PageRank for each node
- Calculate closeness for each node
- Identify the top 5 nodes for each centrality metric
- Show the network of the node with the highest degree i.e. nodes attached with highest degree node
- Show the network of the node with the highest PageRank i.e. nodes attached with highest degree node
- Show the network of the node with the highest closeness i.e. nodes attached with highest degree node

Examples:

Calculating Degree:

```
CALL gds.degree.stream('interactionGraph')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).username AS username, score AS degree
ORDER BY degree DESC;
```

Calculating PageRank:

```
CALL gds.pageRank.stream('webGraph')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).url AS url, score AS pagerank
ORDER BY pagerank DESC
```

Calculating Closeness:

```
CALL gds.alpha.closeness.stream('interactionGraph')
YIELD nodeId, centrality
RETURN gds.util.asNode(nodeId).username AS username, centrality AS closeness
ORDER BY closeness DESC;
```

Calculating Shortest Path:

```
MATCH (a:Actor)
WHERE a.name IN ['Kevin Bacon', 'Denzel Washington']
WITH collect(id(a)) AS nodeIds
CALL gds.shortestPath.dijkstra.stream('proj', {sourceNode:nodeIds[0], TargetNode:nodeIds[1]})
YIELD sourceNode, targetNode, path
RETURN gds.util.asNode(sourceNode).name AS sourceNodeName,
        gds.util.asNode(targetNode).name AS targetNodeName,
        nodes(path) as path;
```

Calculating Breadth-first-search:

```
MATCH (source:Person{name:"Tom Hanks"})
call gds.dfs.stream("test2", {sourceNode: source})
YIELD  sourceNode,  nodeIds,path
RETURN *
```