NEO 4J
SORTING

- ## SHORTEST JOB FIRST , LONGEST JOB FIRST

Given a set of Location ,we want to find the shortest path from one to another

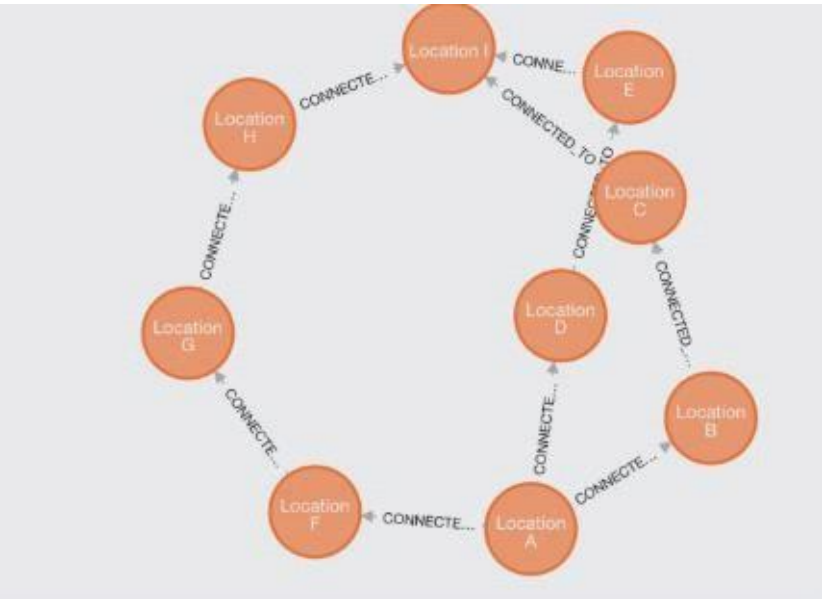
- We want to traverse from **Location A** to **Location I**.
- Nodes have a **name** and a relationship with at least one other node.
- Each relationship has a **distance** property.

```
CREATE ( LocationA : Location { name : ”Location A” } )
CREATE ( LocationB : Location { name : ”Location B” } )
CREATE ( LocationC : Location { name : ”Location C” } )
CREATE ( LocationD : Location { name : ”Location D” } )
CREATE ( LocationE : Location { name : ”Location E” } )
CREATE ( LocationF : Location { name : ”Location F” } )
CREATE ( LocationG : Location { name : ”Location G” } )
CREATE ( LocationH : Location { name : ”Location H” } )
CREATE ( LocationI : Location { name : ”Location I” } )
```

```
CREATE

( LocationA ) –[ : CONNECTED_TO { distance: 5 }] - > ( LocationB ) ,(
LocationB ) –[ : CONNECTED_TO { distance: 6 }] - > ( LocationC ) ,(
LocationC ) –[ : CONNECTED_TO { distance: 4 }] - > ( LocationI ) , (
LocationA ) –[ : CONNECTED_TO { distance: 3 }] - > ( LocationD ) ,(
LocationD ) –[ : CONNECTED_TO { distance: 4 }] - > ( LocationE ) ,(
LocationE ) –[ : CONNECTED_TO { distance: 5 }] - > ( LocationI ) , (
LocationA ) –[ : CONNECTED_TO { distance: 2 }] - > ( LocationF ) ,(
LocationF ) –[ : CONNECTED_TO { distance: 3 }] - > ( LocationG) , (
LocationG ) –[ : CONNECTED_TO { distance: 2 }] - > ( LocationH ) ,(
LocationH ) –[ : CONNECTED_TO { distance: 1 }] - > ( LocationI ) ,
```

```
MATCH (n) RETURN n
```

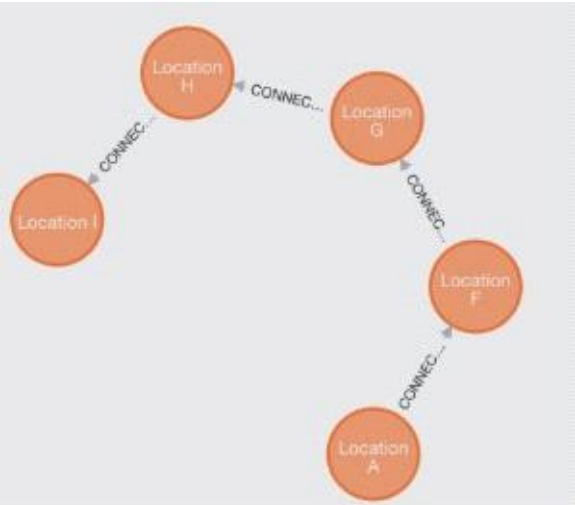


SHORTEST PATH :

MATCH (from:Location { name:"Location A" }) ,(to:Location { name:"Location I"}) , path =(from)-[:CONNECTED_TO*]- >(to)

RETURN path AS shortestPath ,

reduce(distance = 0 ,r in relationships(path) | distance + r.distance) AS totalDistance
ORDER BY totalDistance ASC
LIMIT 1



LONGEST PATH :

MATCH (from:Location { name:”Location A” }) ,(to:Location { name:”Location I”}) , path =(from)-[:CONNECTED_TO*]- >(to)

RETURN path AS longestPath ,

reduce(distance = 0 ,r in relationships(path) | distance + r.distance) AS totalDistance
ORDER BY totalDistance DESC
LIMIT 1