Task: Add a node for London.

create (l:CITY {name:"London"})

Task: add the following relationships between London and Edinburgh:

match (l:CITY {name:"London"}), (e:CITY {name:"Edinburgh"}) create (e)-[:TRAIN {time: 5}]->(l)

match (l:CITY {name:"London"}), (e:CITY {name:"Edinburgh"}) create (e)-[:FLY {time: 1}]->(l)

Task: Add the town of Grangemouth, give it a TOWN role.

create (l:TOWN {name:"Grangemouth"})

Link Grangemouth to Edinburgh via a CAR relationship where the name is *m9*.

match (g:TOWN {name:"Grangemouth"}), (e:CITY {name:"Edinburgh"}) create (g)-[:DRIVE {time: 1}]->(e)

Task: For the London node, add the following information:

match (n:CITY {name:"London"}) set n.population=8630000 return n

match (n:CITY {name:"London"}) set n.size=1572 return n

match (n:CITY {name:"London"}) set n.founded="43 ad" return n

Task: List all the destinations you can reach with a multiple leg journey from Dundee

(n:CITY {name:"Dundee"})-[\*]->(d:CITY {name:"Cardiff"}) return distinct d.name

Task: what is the length of the shortest path between Dundee and Cardiff?

match p = (n:CITY {name:"Dundee"})-[\*]->(d:CITY {name:"Cardiff"}) return min(length(p))

Task: Create a FLY relationship between Edinburgh and Cardiff, with the relationship name *ba*. What is the length of the longest and shortest paths now?

match p = (n:CITY {name:"Dundee"})-[\*]->(d:CITY {name:"Cardiff"}) return min(length(p)),

max(length(p))

Task: What is the full query for using the TRAIN restriction?

match p = (n:CITY {name:"Dundee"})-[:TRAIN\*]->(d:CITY {name:"Cardiff"}) return min(length(p)),

max(length(p))

Task: What query can be used to find the destinations when you start from Edinburgh and take at journey with at least 2 legs but no more than 3?

match p = (n:CITY {name:"Edinburgh"})-[:TRAIN\*2..3]->(d:CITY) return distinct d.name