1. match x = (p:peak{name:'Sandakphu'})<-[r\*]-(t:town{name:'Darjeeling'})   
   unwind r as relation  
   with x, count(relation) as stages  
   return x, stages  
   order by stages asc
2. match x = (p:peak{name:'Sandakphu'})<-[r\*{winter:'true'}]-(t:town{name:'Darjeeling'})

unwind r as relation

with x, count(relation) as stages

return x, stages

order by stages asc

1. match x = (p:peak{name:'Sandakphu'})<-[r\*]-(t:town{name:'Darjeeling'})

unwind r as relation

with x, sum(relation.distance) as dist

return x, dist

order by dist asc

1. match (d:town{name:'Darjeeling'})-[tw:twowheeler{summer:'true'}]-(m:village)

return m

union

match (d:town{name:'Darjeeling'})-[:twowheeler{summer:'true'}]-(m:town)

return m

union

match (d:town{name:'Darjeeling'})-[:twowheeler{summer:'true'}]-(m:peak)

return m

1. match (a:Airport)<-[:ORIGIN]-(f:Flight)

return a, count(f)

order by count(f) desc

1. match (:Airport{name:'LAX'})<-[:ORIGIN]-(:Flight)-[:DESTINATION]->(a1:Airport)

optional match (a1)<-[:ORIGIN | DESTINATION]-(:Flight)-[:ORIGIN | DESTINATION]->(a2:Airport)

where a1.name <> 'LAX'

and a2.name <> 'LAX'

and a1.name <> a2.name

optional match (a2)<-[:ORIGIN | DESTINATION]-(:Flight)-[:ORIGIN | DESTINATION]->(a3:Airport)

where a1.name <> 'LAX'

and a2.name <> 'LAX'

and a3.name <> 'LAX'

and a1.name <> a2.name

and a1.name <> a3.name

and a2.name <> a3.name

return distinct a1, a2, a3