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 r1 = (a0:Airport{name:'LAX'})<-[:ORIGIN]-(f1:Flight)-[:DESTINATION]->(a1:Airport)<-[:ORIGIN | DESTINATION\*]-(f2:Flight)-[:ORIGIN | DESTINATION\*]->(a2:Airport)<-[r\*0..5]->(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

match cf1 = (f1)<-[:ASSIGN]-(t1:Ticket)

match cf2 = (f2)<-[:ASSIGN]-(t2:Ticket)

unwind r as relation

with r1, cf1, cf2,

case

when a3 is not null then a3

when a2 is not null then a2

else a1

end as target,

case

when a3 is not null then sum(t1.price) + sum(t2.price) + sum(relation.price)

when a2 is not null then sum(t1.price) + sum(t2.price)

else sum(t1.price)

end as cost

where cost < 3000

return distinct target

1. match r1 = (a0:Airport{name:'LAX'})<-[:ORIGIN]-(f1:Flight)-[r2:ORIGIN|DESTINATION\*]->(a1:Airport{name:'DAY'})

match cf1 = (f1)<-[:ASSIGN]-(t1:Ticket)

with r1, cf1, sum(t1.price) as cost

return r1, cost

order by cost

1. match r1 = (a0:Airport{name:'LAX'})<-[:ORIGIN]-(f1:Flight)-[r2:ORIGIN|DESTINATION\*]->(a1:Airport{name:'DAY'})

match cf1 = (f1)<-[:ASSIGN]-(t1:Ticket)

with r1, cf1, sum(t1.price) as cost

return r1, cost

order by cost

limit 1

1. match r1 = (a1:Airport)<-[]-(f1:Flight)-[]->(a2:Airport)

with f1.airline as airline, a1 as city, count(a1) as flights

return airline, count(\*) as cities

order by airline

1. match r1 = (a1:Airport)<-[:ORIGIN]-(f1:Flight)-[:DESTINATION]->(a2:Airport)<-[:ORIGIN]-(f2:Flight)-[:DESTINATION]->(a3:Airport)

where a1 <> a2

and a1 <> a3

and a2 <> a3

match cf1 = (f1)<-[:ASSIGN]-(t1:Ticket)

match cf2 = (f2)<-[:ASSIGN]-(t2:Ticket)

with r1, cf1, cf2, t1.price+t2.price as cost

return r1, cost

order by cost

limit 1