

BI / read / 18

BI 1
BI 2
BI 3
BI 4
BI 5
BI 6
BI 7
BI 8
BI 9
BI 10
BI 11
BI 12
BI 13
BI 14
BI 15
BI 16
BI 17
BI 18
BI 19
BI 20

query	BI / read / 18				
title	Friend recommendation				
pattern	<div><div>For each person1 compute top-k(person2) based on mutualFriendCount</div><pre>graph LR person1[person1: Person] -- knows --> Person[Person] Person -- knows --> person2[person2: Person] person2 -- hasInterest --> tag[tag: Tag] person1 -. «neg» knows .-> person2 subgraph Constraints C1[id = \$person1Id] C2[mutualFriendCount = count(*)] C3[name = \$tag] end</pre></div>				
desc.	<p>For a given Person (person1) and a Tag (tag), recommend new friends (person2) who</p> <ul style="list-style-type: none">• do not yet know person1• have many mutual friends with person1• are interested in tag. <p>Rank Persons person2 based on the number of mutual friends.</p>				
params	<div><div>1</div><div>person1Id</div><div>ID</div></div>	<div>Persons with a similar amount of friends are selected</div>			
	<div><div>2</div><div>tag</div><div>Long String</div></div>	<div>Tags with a similar amount of Messages are selected</div>			
result	<div><div>1</div><div>person2.id</div><div>ID</div></div>	<div>R</div>			
	<div><div>2</div><div>mutualFriendCount</div><div>32-bit Integer</div></div>	<div>A</div>			
sort	<div><div>1</div><div>mutualFriendCount</div><div>↓</div></div>				
	<div><div>2</div><div>person2.id</div><div>↑</div></div>				
limit	20				
CPs	2.5, 8.1				