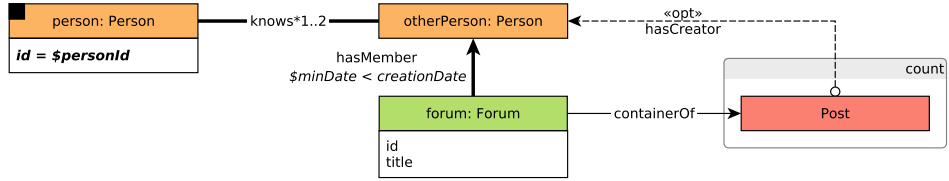


Interactive / complex / 5

query	Interactive / complex / 5														
title	New groups														
pattern	 <pre>graph LR person["person: Person
id = \$personId"] -- "knows*1..2" --> otherPerson["otherPerson: Person"] otherPerson -- "hasMember
\$minDate < creationDate" --> forum["forum: Forum
id
title"] forum -- "containerOf" --> Post["Post"] Post -- "«opt» hasCreator" --> otherPerson subgraph count Post end</pre>														
desc.	Given a start Person, find the Forums which that Person’s friends and friends of friends (excluding start Person) became Members of after a given date. For each Forum find the number of Posts that were created by any of these Persons. For each Forum and consider only those Persons which joined that particular Forum after the given date (minDate).														
params	<table><tr><td>1</td><td>personId</td><td>ID</td><td></td></tr><tr><td>2</td><td>minDate</td><td>Date</td><td></td></tr></table>					1	personId	ID		2	minDate	Date			
1	personId	ID													
2	minDate	Date													
result	<table><tr><td>1</td><td>forum.title</td><td>Long String</td><td>R</td><td></td></tr><tr><td>2</td><td>postCount</td><td>32-bit Integer</td><td>A</td><td>Number of Posts made in forum that were created by friends</td></tr></table>					1	forum.title	Long String	R		2	postCount	32-bit Integer	A	Number of Posts made in forum that were created by friends
1	forum.title	Long String	R												
2	postCount	32-bit Integer	A	Number of Posts made in forum that were created by friends											
sort	<table><tr><td>1</td><td>postCount</td><td>↓</td><td></td></tr><tr><td>2</td><td>forum.id</td><td>↑</td><td></td></tr></table>					1	postCount	↓		2	forum.id	↑			
1	postCount	↓													
2	forum.id	↑													
limit	20														
CPs	2.3, 3.3, 8.2, 8.5														
relevance	This query looks for paths of length two and three, starting from a given Person, moving to friends and friends of friends, and then getting the Forums they are members of. Besides testing the ability of the query optimizer to select the proper join operator, it rewards the usage of indexes, but their accesses will be presumably scattered due to the two/three-hop search space of the query, leading to unpredictable and scattered index accesses. Having efficient implementations of such indexes will be highly beneficial.														