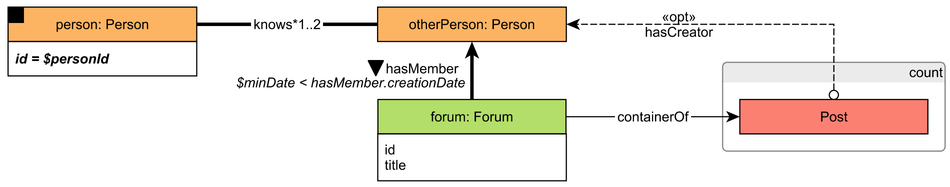
IC 1
IC 2
IC 3
IC 4
IC 5
IC 6
IC 7
IC 8
IC 9
IC 10
IC 11
IC 12
IC 13
IC 14v1
IC 14v2

Interactive / complex / 5

query	Interactive / complex / 5
title	New groups
pattern	 person: Person id = $personId knows*1..2 otherPerson: Person «opt» hasCreator hasMember $minDate < hasMember.creationDate count forum: Forum id title containerOf Post
description	Given a start Person with ID `$personId`, denote their friends and friends of friends (excluding the start Person) as `otherPerson`. Find Forums that any Person `otherPerson` became a member of after a given date (`$minDate`). For each of those Forums, count the number of Posts that were created by the Person `otherPerson`.

params

1	`$personId`	ID	
2	`$minDate`	Date	

result

1	`forum.title`	Long String	R	
2	`postCount`	32-bit Integer	A	Number of Posts made in `forum` that were created by the Person `otherPerson`

sort

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 indices, 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 indices will be highly beneficial.