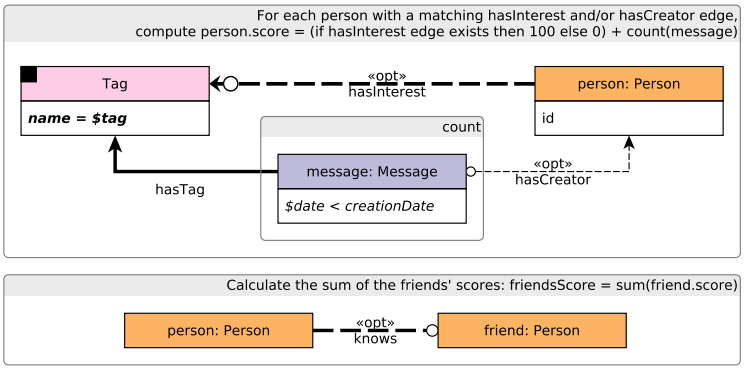
BI / read / 8

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 / 8
title	Central person for a tag
pattern	
desc.	Given a Tag, find all Persons that are interested in the Tag and/or have written a Message (Post or Comment) with a `creationDate` after a given `date` and that has a given Tag. For each Person, compute the `score` as the sum of the following two aspects: • 100, if the Person has this Tag as their interest, or 0 otherwise • number of Messages by this Person with the given Tag Also, for each Person, compute the sum of the score of the Person's friends (`friendsScore`).

params

1	tag	Long String	Tags with a similar amount of Messages are selected
2	date	Date	Dates from around the same day are selected. (TODO - how exactly? what distribution?)

result

1	`person.id`	ID	R	
2	`score`	32-bit Integer	A	
3	`friendsScore`	32-bit Integer	A	The sum of the score of the `person`'s friends

sort

1	`score + friendsScore`	↓
2	`person.id`	↑

limit	100
CPs	1.2, 2.1, 2.3, 3.2, 5.3, 8.2, 8.4, 8.5
relevance	Similarly to BI 16, there are two major ways to compute this query: (1) creating an induced subgraph of the interested Persons and their friends and performing the scoring on this graph or (2) performing the scoring without creating an induced subgraph and scoring the friends of a Person on-the-fly. The first approach is more efficient as it avoids redundant computations, however, specifying it needs support for composable graph queries.