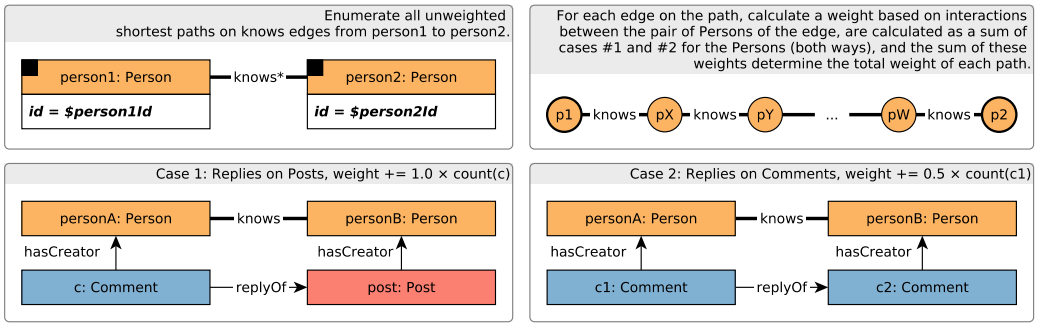
Interactive / complex / 14

query	Interactive / complex / 14
title	Trusted connection paths
pattern	
desc.	Given two Persons, find all (unweighted) shortest paths between these two Persons, in the subgraph induced by the knows relationship. Then, for each path calculate a weight. The nodes in the path are Persons, and the weight of a path is the sum of weights between every pair of consecutive Person nodes in the path. The weight for a pair of Persons is calculated based on their interactions: • Every direct reply (by one of the Persons) to a Post (by the other Person) contributes 1.0. • Every direct reply (by one of the Persons) to a Comment (by the other Person) contributes 0.5. Note that interactions are counted both ways (e.g. if Alice writes 2 Post replies and 1 Comment reply to Bob, while Bob writes 3 Post replies and 4 Comment replies to Alice, their interaction score is $2 \times 1.0 + 1 \times 0.5 + 3 \times 1.0 + 4 \times 0.5 = 7.5$). Return all the paths with shortest length, and their weights. Do not return any rows if there is no path between the two Persons.
params	1 person1Id ID 2 person2Id ID
result	1 personIdsInPath [ID] C identifiers representing an ordered sequence of the Persons in the path 2 pathWeight 64-bit Float C
sort	1 pathWeight ↓ The order of paths with the same weight is unspecified
CPs	3.3, 5.3, 7.2, 7.3, 7.5, 7.7, 8.1, 8.2, 8.3, 8.6
relevance	This query looks for a variable length path, starting at a given Person and finishing at an another given Person. This is a more complex query as it not only requires computing the path length, but returning it and computing a weight. To compute this weight one must look for smaller sub-queries with paths of length three, formed by the two Persons at each step, a Post and a Comment.