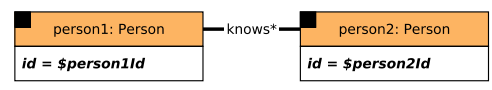
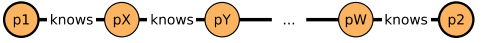
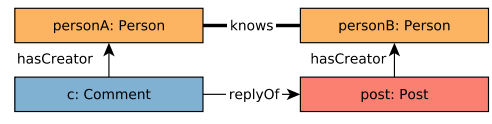
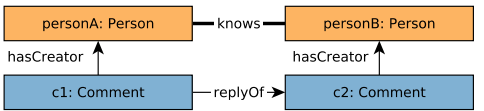


## Interactive / complex / 14

IC 1	query	Interactive / complex / 14		
IC 2	title	Trusted connection paths		
IC 3	pattern	<p>Enumerate all unweighted shortest paths on knows edges from person1 to person2.</p>  <p>For each edge on the path, calculate a weight based on interactions between the pair of Persons of the edge, are calculated as a sum of cases #1 and #2 for the Persons (both ways), and the sum of these weights determine the total weight of each path.</p>  <p>Case 1: Replies on Posts, weight += 1.0 × count(c)</p>  <p>Case 2: Replies on Comments, weight += 0.5 × count(c1)</p> 		
IC 4		<p>Given two Persons, find all (unweighted) shortest paths between these two Persons, in the subgraph induced by the knows relationship.</p> <p>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.</p> <p>The weight for a pair of Persons is calculated based on their interactions:</p> <ul style="list-style-type: none"> <li>• Every direct reply (by one of the Persons) to a Post (by the other Person) contributes 1.0.</li> <li>• Every direct reply (by one of the Persons) to a Comment (by the other Person) contributes 0.5.</li> </ul> <p>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 <math>2 \times 1.0 + 1 \times 0.5 + 3 \times 1.0 + 4 \times 0.5 = 7.5</math>).</p> <p>Return all the paths with shortest length, and their weights. Do not return any rows if there is no path between the two Persons.</p>		
IC 5				
IC 6				
IC 7				
IC 8				
IC 9				
IC 10	desc.	<p>1 person1Id ID</p> <p>2 person2Id ID</p>		
IC 11				
IC 12	result	1	personIdsInPath	[ID] C identifiers representing an ordered sequence of the Persons in the path
IC 13		2	pathWeight	64-bit Float C
IC 14	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.		