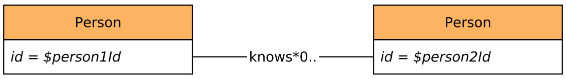
Interactive / complex / 14

<table>
<tr><td>query</td><td>Interactive / complex / 14</td></tr>
<tr><td>title</td><td>Weighted/unweighted paths</td></tr>
<tr><td>pattern</td><td>
</td></tr>
<tr><td>desc.</td><td>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 such that every reply (by one of the Persons) to a Post (by the other Person) contributes 1.0, and every reply (by ones of the Persons) to a Comment (by the other Person) contributes 0.5. Return all the paths with shortest length, and their weights. Do not return any rows if there is now path between the two Persons.</td></tr>
<tr><td>params</td><td><table>
<tr><td>1</td><td>person1.id</td><td>ID</td><td></td></tr>
<tr><td>2</td><td>person2.id</td><td>ID</td><td></td></tr>
</table></td></tr>
<tr><td>result</td><td><table>
<tr><td>1</td><td>[Person.id]</td><td>[ID]</td><td>C</td><td>Identifiers representing an ordered sequence of the Persons in the path</td></tr>
<tr><td>2</td><td>weight</td><td>64-bit Float</td><td>C</td><td></td></tr>
</table></td></tr>
<tr><td>sort</td><td><table>
<tr><td>1</td><td>weight</td><td>↓</td><td>The order of paths with the same weight is unspecified</td></tr>
</table></td></tr>
<tr><td>CPs</td><td>3.3, 7.2, 7.3</td></tr>
<tr><td>relevance</td><td>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 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.</td></tr>
</table>