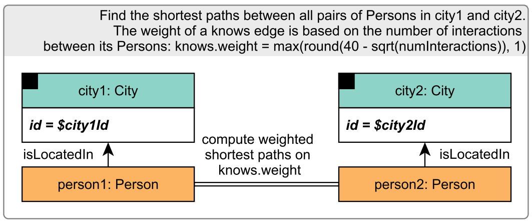
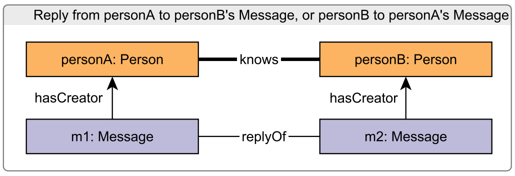
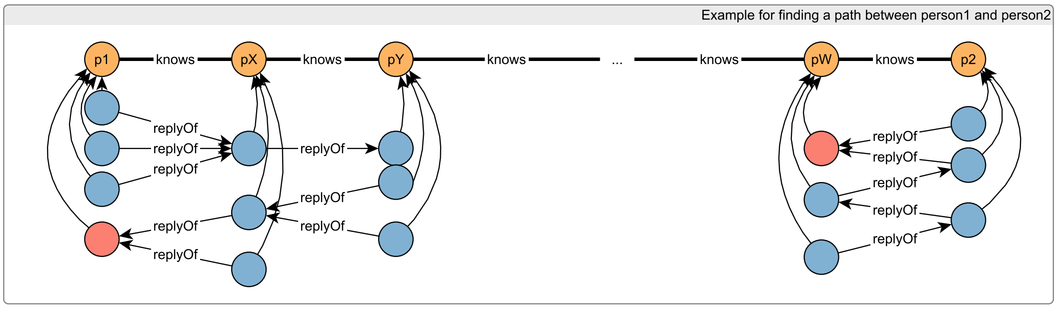
BI / read / 19

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

<table>
<tr><td>query</td><td>BI / read / 19</td></tr>
<tr><td>title</td><td>Interaction path between cities</td></tr>
<tr><td>pattern</td><td>

Find the shortest paths between all pairs of Persons in city1 and city2.
The weight of a knows edge is based on the number of interactions
between its Persons: knows.weight = max(round(40 - sqrt(numInteractions)), 1)
city1: City
id = $city1Id
isLocatedIn
person1: Person
compute weighted
shortest paths on
knows.weight
city2: City
id = $city2Id
isLocatedIn
person2: Person




</td></tr>
<tr><td>description</td><td>Given two Cities with IDs $city1Id, $city2Id, find Persons person1, person2 living in these Cities (respectively) with the <i>cheapest</i> interaction path between them.
The cheapest path is equivalent to the <i>weighted shortest</i> path. It is computed on a subgraph of the Person-knows-Person graph with the edge weights based on the number of interactions. An <i>interaction</i> is a direct reply Comments from one Person to Messages by the other Person. Only knows edges with at least one interaction between their endpoint Persons are considered. For these, the weight of a knows edge is defined as: $\max(\mathrm{round}(40 - \sqrt{numInteractions}), 1)$
If there are multiple pairs of people with cheapest paths that have the same total weight, return all of them.
<i>Note:</i> Interactions are counted both ways, e.g. if Alice knows Bob, Alice writes 2 reply Comments to Bob's Messages and Bob writes 3 reply Comments to Alice's Messages, their total number of interactions is 5 and the weight of the knows edge is 38.
<i>Remark:</i> Determinism is ensured by using square root followed by rounding. For all integers between 1 and 100 000, the square root's fractional part is more than 10e-5 from 0.5, where the rounding could be non-determinstic based on floating point inaccuracies. As 10e-5 is significantly larger than the machine epsilon of IEEE 754 floats (both 32- and 64-bit), the floating point inaccuracies have no chance to affect the derived integer edge weights.</td></tr>
<tr><td>params</td><td>
<table>
<tr><td>1</td><td>$city1Id</td><td>ID</td><td>(a) Small Cities within the same Country
(b) Larger Cities from different Countries</td></tr>
<tr><td>2</td><td>$city2Id</td><td>ID</td><td></td></tr>
</table>
</td></tr>
<tr><td>result</td><td>
<table>
<tr><td>1</td><td>person1.id</td><td>ID</td><td>R</td><td></td></tr>
<tr><td>2</td><td>person2.id</td><td>ID</td><td>R</td><td></td></tr>
<tr><td>3</td><td>totalWeight</td><td>32-bit Integer</td><td>C</td><td></td></tr>
</table>
</td></tr>
<tr><td>sort</td><td>
<table>
<tr><td>1</td><td>person1.id</td><td>↑</td><td></td></tr>
<tr><td>2</td><td>person2.id</td><td>↑</td><td></td></tr>
</table>
</td></tr>
<tr><td>limit</td><td>n/a</td></tr>
<tr><td>CPs</td><td>3.3, 7.6, 7.7, 8.4, 8.6</td></tr>
<tr><td>relevance</td><td>To find the weighted shortest paths efficiently, the system can use e.g. a bidirectional Dijkstra algorithm. As the edge weights do not depend on any parameter, systems can pre-compute them (if they do not interleave reads and writes).</td></tr>
</table>