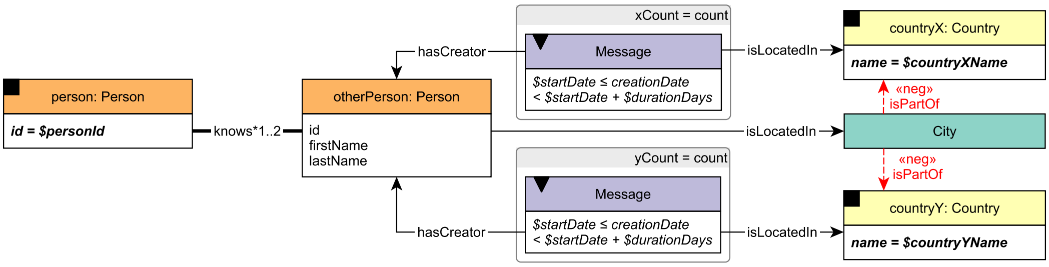
Interactive / complex / 3

IC 1
IC 2
IC 3
IC 4
IC 5
IC 6
IC 7
IC 8
IC 9
IC 10
IC 11
IC 12
IC 13
IC 14v1
IC 14v2

<table>
<tr><td>query</td><td>Interactive / complex / 3</td></tr>
<tr><td>title</td><td>Friends and friends of friends that have been to given countries</td></tr>
<tr><td>pattern</td><td>
</td></tr>
<tr><td>description</td><td>Given a start Person with ID $personId, find Persons that are their friends and friends of friends (excluding the start Person) that have made Posts / Comments in both of the given Countries (named $countryXName and $countryYName), within [$startDate, $startDate + $durationDays) (closed-open interval). Only Persons that are foreign to these Countries are considered, that is Persons whose location Country is neither named $countryXName nor $countryYName.</td></tr>
<tr><td>params</td><td>
<table>
<tr><td>1</td><td>$personId</td><td>ID</td><td></td></tr>
<tr><td>2</td><td>$countryXName</td><td>String</td><td>In SNB Interactive v2, this query has two variants:
(a) Correlated Countries
(b) Anti-correlated Countries</td></tr>
<tr><td>3</td><td>$countryYName</td><td>String</td><td></td></tr>
<tr><td>4</td><td>$startDate</td><td>Date</td><td>Beginning of requested period</td></tr>
<tr><td>5</td><td>$durationDays</td><td>32-bit Integer</td><td>Duration of requested period, in days. The interval [$startDate, $startDate + $durationDays) is closed-open</td></tr>
</table>
</td></tr>
<tr><td>result</td><td>
<table>
<tr><td>1</td><td>otherPerson.id</td><td>ID</td><td>R</td><td></td></tr>
<tr><td>2</td><td>otherPerson.firstName</td><td>String</td><td>R</td><td></td></tr>
<tr><td>3</td><td>otherPerson.lastName</td><td>String</td><td>R</td><td></td></tr>
<tr><td>4</td><td>xCount</td><td>32-bit Integer</td><td>A</td><td>Number of Messages from Country named $countryXName created by the Person within the given time</td></tr>
<tr><td>5</td><td>yCount</td><td>32-bit Integer</td><td>A</td><td>Number of Messages from Country named $countryYName created by the Person within the given time</td></tr>
<tr><td>6</td><td>count</td><td>32-bit Integer</td><td>A</td><td>count = xCount + yCount</td></tr>
</table>
</td></tr>
<tr><td>sort</td><td>
<table>
<tr><td>1</td><td>count</td><td>↓</td><td></td></tr>
<tr><td>2</td><td>otherPerson.id</td><td>↑</td><td></td></tr>
</table>
</td></tr>
<tr><td>limit</td><td>20</td></tr>
<tr><td>CPs</td><td>2.1, 3.1, 5.1, 8.2, 8.5</td></tr>
<tr><td>relevance</td><td>This query looks for paths of length two and three, starting from a Person, going to friends or friends of friends, and then moving to Messages. This query tests the ability of the query optimizer to select the most efficient join ordering, which will depend on the cardinalities of the intermediate results. Many friends of friends can be duplicate, then it is expected to eliminate duplicates and those people prior to access the Post and Comments, as well as eliminate those friends from Countries named $countryXName and $countryYName, as the size of the intermediate results can be severely affected. A possible structural optimization could be to materialize the number of Posts and Comments created by a Person, and progressively filter those people that could not even fall in the top 20 even having all their posts in the Countries named $countryXName and $countryYName.</td></tr>
</table>