

## Interactive / complex / 13

|       |             |   |             |                    |   |
|-------|-------------|---|-------------|--------------------|---|
| IC 1  | query       | Interactive / complex / 13  |             |                    |   |
| IC 2  | title       | Single shortest path  |             |                    |   |
| IC 3  | pattern     | <pre> graph LR     P1[Person<br/>id = \$person1Id] -- knows*0.. --- P2[Person<br/>id = \$person2Id]         </pre>  |             |                    |   |
| IC 6  | description | <p>Given two Persons with IDs \$person1Id and \$person2Id, find the shortest path between these two Persons in the subgraph induced by the knows edges. Return the length of this path:</p> <ul style="list-style-type: none"> <li>• -1: no path found</li> <li>• 0: start person = end person</li> <li>• &gt; 0: path found (start person ≠ end person)</li> </ul> |             |                    |   |
| IC 7  |             | 1   | \$person1Id | ID                 | <p>In SNB Interactive v2, this query has two variants:</p> <p>(b) Guaranteed that there is no path between the two Persons</p> <p>(b) Guaranteed that there is a 4-hop path between the two Persons</p> |
| IC 8  |             |   | 2           | \$person2Id        |   |
| IC 9  |             | result  | 1           | shortestPathLength | 32-bit Integer  |
| IC 10 | CPs         | 3.3, 7.2, 7.3, 7.5, 7.8, 8.1, 8.6   |             |                    |   |
| IC 11 | relevance   | <p>This query looks for a variable length path, starting at a given Person and finishing at an another given Person. Proper cardinality estimation and search space pruning, will be crucial. This query also allows for possible parallel implementations.</p>   |             |                    |   |