

Interactive / complex / 11

| | | | | | |
|-------|-----------|--|-----------------------|----------------|---|
| IC 1 | query | Interactive / complex / 11 | | | |
| IC 2 | title | Job referral | | | |
| IC 3 | pattern | <pre> graph TD P1[person: Person id = \$personId] -- knows*1..2 --> P2[otherPerson: Person id firstName lastName] P2 -- "workAt year(workFrom) < \$year" --> C[company: Company name] C -- isLocatedIn --> CO[country: Country name = \$name] </pre> | | | |
| IC 4 | | | | | |
| IC 5 | | | | | |
| IC 6 | | | | | |
| IC 7 | | | | | |
| IC 8 | | | | | |
| IC 9 | | | | | |
| IC 10 | | | | | |
| IC 11 | | | | | |
| IC 12 | | | | | |
| IC 13 | desc. | Given a start Person, find that Person's friends and friends of friends (excluding start Person) who started working in some Company in a given Country, before a given date (year). | | | |
| IC 14 | params | 1 | personId | ID | |
| | | 2 | countryName | String | |
| | | 3 | workFromYear | 32-bit Integer | |
| | result | 1 | otherPerson.id | ID | R |
| | | 2 | otherPerson.firstName | String | R |
| | | 3 | otherPerson.lastName | String | R |
| | | 4 | company.name | String | R |
| | | 5 | workAt.workFrom | 32-bit Integer | R |
| | sort | 1 | workAt.workFrom | ↑ | |
| | | 2 | otherPerson.id | ↑ | |
| | | 3 | company.name | ↓ | |
| | limit | 10 | | | |
| | CPs | 1.3, 2.3, 2.4, 3.3, 4.2 | | | |
| | relevance | This query looks for paths of length two or three, starting from a Person, moving to friends or friends of friends, and ending at a Company. In this query, there are selective joins and a top-k order by that can be exploited for optimizations. | | | |