

Interactive / complex / 1

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 14

query	Interactive / complex / 1																																																																					
title	Friends with certain name																																																																					
pattern	<pre>graph LR Person1[Person] -- knows*1..3 --> Person2[person: Person] Person2 -- isLocatedIn --> City1[City] Person2 -- workAt --> Company[Company] Person2 -- studyAt --> University[University] City1 -- isLocatedIn --> Country[Country] University -- isLocatedIn --> City2[City]</pre> <p>The diagram illustrates the query pattern. It starts with a Person entity (orange box) with attribute <code>id = \$id</code>. This person is connected via a <code>knows*1..3</code> relationship to another person: Person entity (orange box) with attribute <code>firstName = \$firstName</code>. This second person is then connected to three other entities: City (teal box) via <code>isLocatedIn</code>, Company (green box) via <code>workAt</code>, and University (yellow box) via <code>studyAt</code>. The City entity has attribute <code>name</code>. The Company entity has attribute <code>name</code>. The University entity has attribute <code>name</code>. The Country entity (yellow box) has attribute <code>name</code> and is connected to the City entity via <code>isLocatedIn</code>. The City entity connected to the University has attribute <code>name</code>.</p>																																																																					
desc.	Given a start Person, find Persons with a given first name (<code>firstName</code>) that the start Person is connected to (excluding start Person) by at most 3 steps via the knows relationships. Return Persons, including the distance (1..3), summaries of the Persons workplaces and places of study.																																																																					
params	<table><tr><td>1</td><td>Person.id</td><td>ID</td><td>personId</td></tr><tr><td>2</td><td>Person.firstName</td><td>String</td><td>firstName</td></tr></table>					1	Person.id	ID	personId	2	Person.firstName	String	firstName																																																									
1	Person.id	ID	personId																																																																			
2	Person.firstName	String	firstName																																																																			
result	<table><tr><td>1</td><td>Person.id</td><td>ID</td><td>R</td><td>friendId</td></tr><tr><td>2</td><td>Person.lastName</td><td>String</td><td>R</td><td>friendLastName</td></tr><tr><td>3</td><td>distanceFromPerson</td><td>32-bit Integer</td><td>C</td><td>distanceFromPerson</td></tr><tr><td>4</td><td>Person.birthday</td><td>Date</td><td>R</td><td>friendBirthday</td></tr><tr><td>5</td><td>Person.creationDate</td><td>DateTime</td><td>R</td><td>friendCreationDate</td></tr><tr><td>6</td><td>Person.gender</td><td>String</td><td>R</td><td>friendGender</td></tr><tr><td>7</td><td>Person.browserUsed</td><td>String</td><td>R</td><td>friendBrowserUsed</td></tr><tr><td>8</td><td>Person.locationIP</td><td>String</td><td>R</td><td>friendLocationIp</td></tr><tr><td>9</td><td>{Person.email}</td><td>{String}</td><td>R</td><td>friendEmails</td></tr><tr><td>10</td><td>{Person.speaks}</td><td>{String}</td><td>R</td><td>friendLanguages</td></tr><tr><td>11</td><td>Person-isLocatedIn->City.name</td><td>String</td><td>R</td><td>friendCityName</td></tr><tr><td>12</td><td>{Person-studyAt->University.name, Person-studyAt->.classYear, Person-studyAt->University-isLocatedIn->City.name}</td><td>{<String, 32-bit Integer, String>}</td><td>A</td><td>friendUniversities</td></tr><tr><td>13</td><td>{Person-workAt->Company.name, Person-workAt->.workFrom, Person-workAt->Company-isLocatedIn->Country.name}</td><td>{<String, 32-bit Integer, String>}</td><td>A</td><td>friendCompanies</td></tr></table>					1	Person.id	ID	R	friendId	2	Person.lastName	String	R	friendLastName	3	distanceFromPerson	32-bit Integer	C	distanceFromPerson	4	Person.birthday	Date	R	friendBirthday	5	Person.creationDate	DateTime	R	friendCreationDate	6	Person.gender	String	R	friendGender	7	Person.browserUsed	String	R	friendBrowserUsed	8	Person.locationIP	String	R	friendLocationIp	9	{Person.email}	{String}	R	friendEmails	10	{Person.speaks}	{String}	R	friendLanguages	11	Person-isLocatedIn->City.name	String	R	friendCityName	12	{Person-studyAt->University.name, Person-studyAt->.classYear, Person-studyAt->University-isLocatedIn->City.name}	{<String, 32-bit Integer, String>}	A	friendUniversities	13	{Person-workAt->Company.name, Person-workAt->.workFrom, Person-workAt->Company-isLocatedIn->Country.name}	{<String, 32-bit Integer, String>}	A	friendCompanies
1	Person.id	ID	R	friendId																																																																		
2	Person.lastName	String	R	friendLastName																																																																		
3	distanceFromPerson	32-bit Integer	C	distanceFromPerson																																																																		
4	Person.birthday	Date	R	friendBirthday																																																																		
5	Person.creationDate	DateTime	R	friendCreationDate																																																																		
6	Person.gender	String	R	friendGender																																																																		
7	Person.browserUsed	String	R	friendBrowserUsed																																																																		
8	Person.locationIP	String	R	friendLocationIp																																																																		
9	{Person.email}	{String}	R	friendEmails																																																																		
10	{Person.speaks}	{String}	R	friendLanguages																																																																		
11	Person-isLocatedIn->City.name	String	R	friendCityName																																																																		
12	{Person-studyAt->University.name, Person-studyAt->.classYear, Person-studyAt->University-isLocatedIn->City.name}	{<String, 32-bit Integer, String>}	A	friendUniversities																																																																		
13	{Person-workAt->Company.name, Person-workAt->.workFrom, Person-workAt->Company-isLocatedIn->Country.name}	{<String, 32-bit Integer, String>}	A	friendCompanies																																																																		
sort	<table><tr><td>1</td><td>distanceFromPerson</td><td>↑</td><td></td></tr><tr><td>2</td><td>Person.lastName</td><td>↑</td><td></td></tr><tr><td>3</td><td>Person.id</td><td>↑</td><td></td></tr></table>					1	distanceFromPerson	↑		2	Person.lastName	↑		3	Person.id	↑																																																						
1	distanceFromPerson	↑																																																																				
2	Person.lastName	↑																																																																				
3	Person.id	↑																																																																				
limit	20																																																																					
CPs	2.1, 5.3, 8.2																																																																					
relevance	<p>This query is a representative of a simple navigational query. It looks for paths of length 1..3 through the knows relation, starting from a given Person and ending at a Person with a given first name. It is interesting for several aspects. (1) It requires for a complex aggregation for returning the concatenation of universities, companies, languages and email information of the Person. (2) It tests the ability of the optimizer to move the evaluation of sub-queries functionally dependant on the Person, after the evaluation of the top-k. (3) Its performance is highly sensitive to properly estimating the cardinalities in each transitive path, and paying attention not to explore already visited Persons.</p>																																																																					