

Interactive / complex / 1

IC 1	query	Interactive / complex / 1						
IC 2	title	Transitive friends with a certain name						
IC 3	pattern							
IC 4	description	Given a start Person with ID \$personId, find Persons with a given first name (\$firstName) 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.						
IC 5	params	1	\$personId	ID				
IC 6		2	\$firstName	String				
IC 7	result	1	otherPerson.id	ID	R			
IC 8		2	otherPerson.lastName	String	R			
IC 9		3	distanceFromPerson	32-bit Integer	C			
IC 10		4	otherPerson.birthday	Date	R			
IC 11		5	otherPerson.creationDate	DateTime	R			
IC 12		6	otherPerson.gender	String	R			
IC 13		7	otherPerson.browserUsed	String	R			
IC 14v1		8	otherPerson.locationIP	String	R			
IC 14v2		9	otherPerson.email	{Long String}	R			
		10	otherPerson.speaks	{String}	R			
		11	locationCity.name	String	R			
		12	universities	{<String, 32-bit Integer, String>}	A	{<university.name, studyAt.classYear, universityCity.name>}		
		13	companies	{<String, 32-bit Integer, String>}	A	{<company.name, workAt.workFrom, companyCountry.name>}		
	sort	1	distanceFromPerson	↑				
		2	otherPerson.lastName	↑				
		3	otherPerson.id	↑				
	limit	20						
	CPs	2.1, 5.3, 8.2						
	relevance	This query is a representative of a simple navigational query. 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.						