## **BI / read / 10**

BI 1	query	BI / read / 10
BI 2	title	Experts in social circle
BI 3		Country
BI 4		name = \$country
BI 6		isPartOf
BI 7		City
BI 8		isLocatedIn
BI 9	pattern	startParson: Parson knows* expertCandidateParson: Parson
BI 10	•	### sminPathDistance \$maxPathDistance id id ################################
BI 11		hasCreator
BI 12		count for each (tag, person) hasType
BI 13		tag: Tag hasTag hasTag Tag
BI 14		name
BI 15		
BI 16		Given a Person (startPerson), find all other Persons (expertCandidatePerson) that live in a given
BI 17		Country and are connected to given Person by a <i>shortest path</i> with length in range [minPathDis-
BI 18		tance, maxPathDistance] through the knows relation.
BI 19	desc.	For each of these expertCandidatePerson nodes, retrieve all of their Messages that contain at least
BI 20		one Tag belonging to a given TagClass (direct relation not transitive). For each Message, retrieve all of its Tags.
		Group the results by Persons and Tags, then count the Messages by a certain Person having a certain
		Tag.
		rag.
		The ID of the startPerson. Persons with a similar
	l	degree of knows edges are selected
		Country String Countries with a similar number of Persons are
		selected
	params	TagClasses with a similar degree of hasType edges are
		selected selected
		4 minPathDistance 32-bit Integer 1 or 2
		5 maxPathDistance 32-bit Integer 2 or 3
-		
		1 expertCandidatePerson.id ID R
	1.	2 tag.name Long String R
	result	Number of Messages created by that
		3 messageCount 32-bit Integer A Person containing that Tag
		1 messageCount ↓
	sort	2 tag.name
		3 expertCandidatePerson.id ↑
	limit	100
	CPs	1.2, 1.3, 2.3, 2.4, 3.3, 5.3, 7.1, 7.2, 7.3, 8.1, 8.6
L		, , , , , , , , , , , , , , , , , , , ,