

Interactive / complex / 3

query	Interactive / complex / 3																																		
title	Friends and friends of friends that have been to given countries																																		
pattern	<p>The diagram illustrates the query pattern. It starts with a person: Person entity (orange box) with a filter <code>id = \$personId</code>. This person is connected via a <code>knows*1..2</code> relationship to an otherPerson: Person entity (orange box) with filters <code>id</code>, <code>firstName</code>, and <code>lastName</code>. The otherPerson is connected via <code>hasCreator</code> relationships to two Message entities (purple boxes). Each Message entity has filters <code>\$startDate ≤ creationDate</code> and <code>≤ \$startDate + \$durationDays</code>. The Message entities are connected via <code>isLocatedIn</code> relationships to CountryX: Country and CountryY: Country entities (yellow boxes). Both CountryX and CountryY have filters <code>name = \$countryXName</code> and <code>name = \$countryYName</code> respectively. Additionally, there are City entities (teal boxes) connected to the Message entities via <code>isLocatedIn</code> relationships. The City entities have filters <code>«neg» isPartOf</code> and <code>«neg» isPartOf</code> pointing to CountryX and CountryY respectively. The Message entities also have aggregate filters <code>xCount = count</code> and <code>yCount = count</code>.</p>																																		
desc.	Given a start Person, find Persons that are their friends and friends of friends (excluding start Person) that have made Posts / Comments in both of the given Countries, CountryX and CountryY, within a given period. Only Persons that are foreign to Countries CountryX and CountryY are considered, that is Persons whose location is neither CountryX nor CountryY.																																		
params	<table><tr><td>1</td><td>personId</td><td>ID</td><td></td></tr><tr><td>2</td><td>countryXName</td><td>String</td><td></td></tr><tr><td>3</td><td>countryYName</td><td>String</td><td></td></tr><tr><td>4</td><td>startDate</td><td>Date</td><td>Beginning of requested period</td></tr><tr><td>5</td><td>durationDays</td><td>32-bit Integer</td><td>Duration of requested period, in days. The interval [startDate, startDate + durationDays) is closed-open</td></tr></table>					1	personId	ID		2	countryXName	String		3	countryYName	String		4	startDate	Date	Beginning of requested period	5	durationDays	32-bit Integer	Duration of requested period, in days. The interval [startDate, startDate + durationDays) is closed-open										
1	personId	ID																																	
2	countryXName	String																																	
3	countryYName	String																																	
4	startDate	Date	Beginning of requested period																																
5	durationDays	32-bit Integer	Duration of requested period, in days. The interval [startDate, startDate + durationDays) is closed-open																																
result	<table><tr><td>1</td><td>otherPerson.id</td><td>ID</td><td>R</td><td></td></tr><tr><td>2</td><td>otherPerson.firstName</td><td>String</td><td>R</td><td></td></tr><tr><td>3</td><td>otherPerson.lastName</td><td>String</td><td>R</td><td></td></tr><tr><td>4</td><td>xCount</td><td>32-bit Integer</td><td>A</td><td>Number of Messages from Country CountryX created by the Person within the given time</td></tr><tr><td>5</td><td>yCount</td><td>32-bit Integer</td><td>A</td><td>Number of Messages from Country CountryY created by the Person within the given time</td></tr><tr><td>6</td><td>count</td><td>32-bit Integer</td><td>A</td><td>count = xCount + yCount</td></tr></table>					1	otherPerson.id	ID	R		2	otherPerson.firstName	String	R		3	otherPerson.lastName	String	R		4	xCount	32-bit Integer	A	Number of Messages from Country CountryX created by the Person within the given time	5	yCount	32-bit Integer	A	Number of Messages from Country CountryY created by the Person within the given time	6	count	32-bit Integer	A	count = xCount + yCount
1	otherPerson.id	ID	R																																
2	otherPerson.firstName	String	R																																
3	otherPerson.lastName	String	R																																
4	xCount	32-bit Integer	A	Number of Messages from Country CountryX created by the Person within the given time																															
5	yCount	32-bit Integer	A	Number of Messages from Country CountryY created by the Person within the given time																															
6	count	32-bit Integer	A	count = xCount + yCount																															
sort	<table><tr><td>1</td><td>xCount</td><td>↓</td><td></td></tr><tr><td>2</td><td>otherPerson.id</td><td>↑</td><td></td></tr></table>					1	xCount	↓		2	otherPerson.id	↑																							
1	xCount	↓																																	
2	otherPerson.id	↑																																	
limit	20																																		
CPs	2.1, 3.1, 5.1, 8.2, 8.5																																		
relevance	<p>This query looks for paths of length two and three, starting from a Person, going to friends or friends of friends, and then moving to Messages. This query tests the ability of the query optimizer to select the most efficient join ordering, which will depend on the cardinalities of the intermediate results. Many friends of friends can be duplicate, then it is expected to eliminate duplicates and those people prior to access the Post and Comments, as well as eliminate those friends from Countries CountryX and CountryY, as the size of the intermediate results can be severely affected. A possible structural optimization could be to materialize the number of Posts and Comments created by a Person, and progressively filter those people that could not even fall in the top 20 even having all their posts in the Countries CountryX and CountryY.</p>																																		