

BI / read / 1

BI 1	query	BI / read / 1			
BI 2	title	Posting summary			
BI 3	pattern				
BI 4		<div>message: Message</div>			
BI 5		<div>creationDate < \$date</div>			
BI 6	desc.				
BI 7		<div>length</div>			
BI 8		<div>year(creationDate)</div>			
BI 9	desc.	<p>Given a date, find all Messages created before that date. Group them by a 3-level grouping:</p> <ol style="list-style-type: none"> by year of creation for each year, group into Message types: is Comment or not for each year-type group, split into four groups based on length of their content <ul style="list-style-type: none"> 0: 0 <= length < 40 (short) 1: 40 <= length < 80 (one liner) 2: 80 <= length < 160 (tweet) 3: 160 <= length (long) 			
BI 10					
BI 11					
BI 12					
BI 13					
BI 14					
BI 15					
BI 16	params				
BI 17		1	date	Date	
BI 18	result	1	year	32-bit Integer	R
BI 19		2	isComment	Boolean	M
BI 20		3	lengthCategory	String	C
BI 21		4	messageCount	32-bit Integer	A
BI 22		5	averageMessageLength	32-bit Integer	A
BI 23		6	sumMessageLength	32-bit Integer	A
BI 24		7	percentageOfMessages	32-bit Float	A
BI 25	sort	1	year	↓	
		2	isComment	↑	false < true, i.e. the ordering puts Posts first, and Comments second
		3	lengthCategory	↑	order based on the length of the category, 0 (short), 1 (one liner), etc.
	CPs	1.2, 3.2, 4.1, 8.5			

BI / read / 2

query	BI / read / 2																																		
title	Top tags for country, age, gender, time																																		
pattern	<div><div><div>country: Country</div><div><div>name = \$country1 or name = \$country2</div><div>name</div></div></div><div><div>City</div><div>isPartOf</div><div>country: Country</div></div><div><div>person: Person</div><div><div>gender birthday</div></div></div><div><div>isLocatedIn</div><div>City</div></div><div><div>hasCreator</div><div>person: Person</div></div><div><div>message: Message</div><div><div>creationDate >= \$date1 and creationDate <= \$date2</div><div>month(creationDate)</div></div></div><div><div>hasTag</div><div>message: Message</div></div><div><div>tag: Tag</div><div>name</div></div><div><div>messageCount = count</div><div>message: Message</div></div><div><div>messageCount > 100</div></div></div>																																		
desc.	<p>Select all Messages created in the range of [startDate, endDate] by Persons located in country1 or country2. Select the creator Persons and the Tags of these Messages. Split these Persons, Tags and Messages into a 5-level grouping:</p> <div><div>1. name of country of Person,</div><div>2. month the Message was created,</div><div>3. gender of Person,</div><div>4. age group of Person, defined as years between person’s birthday and end of simulation (2013-01-01), divided by 5, rounded down (partial years do not count),</div><div>5. name of tag attached to Message.</div></div> <p>Consider only those groups where number of Messages is greater than 100.</p>																																		
params	<table><tr><td>1</td><td>startDate</td><td>Date</td><td></td></tr><tr><td>2</td><td>endDate</td><td>Date</td><td></td></tr><tr><td>3</td><td>country1</td><td>String</td><td></td></tr><tr><td>4</td><td>country2</td><td>String</td><td></td></tr></table>					1	startDate	Date		2	endDate	Date		3	country1	String		4	country2	String															
1	startDate	Date																																	
2	endDate	Date																																	
3	country1	String																																	
4	country2	String																																	
result	<table><tr><td>1</td><td>country.name</td><td>String</td><td>R</td><td></td></tr><tr><td>2</td><td>messageMonth</td><td>32-bit Integer</td><td>R</td><td>1–12</td></tr><tr><td>3</td><td>person.gender</td><td>String</td><td>R</td><td>male or female</td></tr><tr><td>4</td><td>ageGroup</td><td>32-bit Integer</td><td>C</td><td></td></tr><tr><td>5</td><td>tag.name</td><td>String</td><td>R</td><td></td></tr><tr><td>6</td><td>messageCount</td><td>64-bit Integer</td><td>A</td><td>The number of messages in the group</td></tr></table>					1	country.name	String	R		2	messageMonth	32-bit Integer	R	1–12	3	person.gender	String	R	male or female	4	ageGroup	32-bit Integer	C		5	tag.name	String	R		6	messageCount	64-bit Integer	A	The number of messages in the group
1	country.name	String	R																																
2	messageMonth	32-bit Integer	R	1–12																															
3	person.gender	String	R	male or female																															
4	ageGroup	32-bit Integer	C																																
5	tag.name	String	R																																
6	messageCount	64-bit Integer	A	The number of messages in the group																															
sort	<table><tr><td>1</td><td>messageCount</td><td>↓</td><td></td></tr><tr><td>2</td><td>tag.name</td><td>↑</td><td></td></tr><tr><td>3</td><td>ageGroup</td><td>↑</td><td></td></tr><tr><td>4</td><td>person.gender</td><td>↑</td><td></td></tr><tr><td>5</td><td>messageMonth</td><td>↑</td><td></td></tr><tr><td>6</td><td>country.name</td><td>↑</td><td></td></tr></table>					1	messageCount	↓		2	tag.name	↑		3	ageGroup	↑		4	person.gender	↑		5	messageMonth	↑		6	country.name	↑							
1	messageCount	↓																																	
2	tag.name	↑																																	
3	ageGroup	↑																																	
4	person.gender	↑																																	
5	messageMonth	↑																																	
6	country.name	↑																																	
limit	100																																		
CPs	1.1, 1.2, 1.3, 2.1, 2.3, 3.1, 3.2, 8.2, 8.5																																		

BI / read / 3

BI 1	query	BI / read / 3			
BI 2	title	Tag evolution			
BI 3	pattern				
BI 4					
BI 5					
BI 6					
BI 7					
BI 8					
BI 9					
BI 10	desc.	<p>Find the Tags that were used in Messages during the given month of the given year and the Tags that were used during the next month.</p> <p>For the Tags and for both months, compute the count of Messages.</p>			
BI 11					
BI 12					
BI 13					
BI 14					
BI 15	params	1	year	32-bit Integer	
BI 16		2	month	32-bit Integer	
BI 17					
BI 18	result	1	tag.name	String	R
BI 19		2	countMonth1	32-bit Integer	A
BI 20		3	countMonth2	32-bit Integer	A
BI 21		4	diff	32-bit Integer	A
BI 22					
BI 23	sort	1	diff	↓	
BI 24		2	tag.name	↑	
BI 25	limit	100			
	CPs	2.4, 3.1, 3.2, 4.1, 4.3, 5.3, 6.1, 8.2, 8.5			

BI / read / 4

BI 1	query	BI / read / 4			
BI 2	title	Popular topics in a country			
BI 3	pattern				
BI 15	desc.	<p>Given a TagClass and a Country, find all the Forums created in the given Country, containing at least one Post with Tags belonging directly to the given TagClass.</p> <p>The location of a Forum is identified by the location of the Forum's moderator.</p>			
BI 18	params	1	tagClass	String	
BI 19		2	country	String	
BI 21	result	1	forum.id	64-bit Integer	R
BI 22		2	forum.title	String	R
BI 23		3	forum.creationDate	DateTime	R
BI 24		4	person.id	64-bit Integer	R
BI 25		5	postCount	32-bit Integer	A
	sort	1	postCount	↓	
		2	forum.id	↑	
	limit	20			
	CPs	1.1, 1.2, 1.3, 2.1, 2.2, 2.4, 3.3, 8.2			

BI / read / 5

BI 1	query	BI / read / 5		
BI 2	title	Top posters in a country		
BI 3	pattern			
BI 5				
BI 6				
BI 7				
BI 8				
BI 9				
BI 10				
BI 11				
BI 12				
BI 13				
BI 14				
BI 15	desc.	Find the most popular Forums for a given Country, where the popularity of a Forum is measured by the number of members that Forum has from the given Country.		
BI 16		Calculate the top 100 most popular Forums. In case of a tie, the forum(s) with the smaller id value(s) should be selected.		
BI 17		For each member Person of the 100 most popular Forums, count the number of Posts (postCount) they made in any of those (most popular) Forums. Also include those member Persons who have not posted any messages (have a postCount of 0).		
BI 18				
BI 19				
BI 20				
BI 21				
BI 22	params	1	country	String
BI 23				
BI 24				
BI 25				
result	1	person.id	64-bit Integer	R
	2	person.firstName	String	R
	3	person.lastName	String	R
	4	person.creationDate	DateTime	R
	5	postCount	32-bit Integer	A
sort	1	postCount	↓	
	2	person.id	↑	
limit	100			
CPs	1.2, 1.3, 2.1, 2.2, 2.3, 2.4, 3.3, 5.3, 6.1, 8.2, 8.4			

BI / read / 6

query	BI / read / 6																									
title	Most active posters of a given topic																									
pattern	<div><div>score = 1*messageCount + 2*replyCount + 10*likeCount</div><div><div><div>Person</div><div>likes</div><div>likeCount = count</div><div>Message</div></div><div><div>Tag</div><div>name = \$tag</div><div>hasTag</div><div>messageCount = count</div></div><div><div>comment: Comment</div><div>replyOf</div><div>replyCount = count</div></div><div><div>hasCreator</div><div>person: Person</div><div>id</div></div></div></div>																									
desc.	<p>Get each Person (person) who has created a Message (message) with a given Tag (direct relation, not transitive). Considering only these messages, for each Person node:</p> <ul style="list-style-type: none">Count its messages (messageCount).Count likes (likeCount) to its messages.Count Comments (replyCount) in reply to it messages. <p>The score is calculated according to the following formula: 1 * messageCount + 2 * replyCount + 10 * likeCount.</p>																									
params	<div><div>1</div><div>tag</div><div>String</div><div></div></div>																									
result	<table><tr><td>1</td><td>person.id</td><td>64-bit Integer</td><td>R</td><td></td></tr><tr><td>2</td><td>replyCount</td><td>32-bit Integer</td><td>A</td><td></td></tr><tr><td>3</td><td>likeCount</td><td>32-bit Integer</td><td>A</td><td></td></tr><tr><td>4</td><td>messageCount</td><td>32-bit Integer</td><td>A</td><td></td></tr><tr><td>5</td><td>score</td><td>32-bit Integer</td><td>A</td><td></td></tr></table>	1	person.id	64-bit Integer	R		2	replyCount	32-bit Integer	A		3	likeCount	32-bit Integer	A		4	messageCount	32-bit Integer	A		5	score	32-bit Integer	A	
1	person.id	64-bit Integer	R																							
2	replyCount	32-bit Integer	A																							
3	likeCount	32-bit Integer	A																							
4	messageCount	32-bit Integer	A																							
5	score	32-bit Integer	A																							
sort	<table><tr><td>1</td><td>score</td><td>↓</td><td></td></tr><tr><td>2</td><td>person.id</td><td>↑</td><td></td></tr></table>	1	score	↓		2	person.id	↑																		
1	score	↓																								
2	person.id	↑																								
limit	100																									
CPs	1.2, 2.3, 8.2																									

BI / read / 7

BI 1	query	BI / read / 7			
BI 2	title	Most authoritative users on a given topic			
BI 3	pattern				
BI 4					
BI 5					
BI 6					
BI 7					
BI 8					
BI 9					
BI 10					
BI 11					
BI 12					
BI 13					
BI 14					
BI 15					
BI 16					
BI 17	desc.	<p>Given a Tag, find all Persons (person) that ever created a Message (message1) with the given Tag. For each of these Persons (person) compute their “authority score” as follows:</p> <ul style="list-style-type: none"> The “authority score” is the sum of “popularity scores” of the Persons (person2) that liked any of that Person’s Messages (message2) with the given Tag. A Person’s (person2) “popularity score” is defined as the total number of likes on all of their Messages (message3). 			
BI 18					
BI 19					
BI 20					
BI 21					
BI 22					
BI 23					
BI 24					
BI 25					
	params	1	tag	String	
	result	1	person.id	64-bit Integer	R
		2	authorityScore	32-bit Integer	A
	sort	1	authorityScore	↓	
		2	person1.id	↑	
	limit	100			
	CPs	1.2, 2.3, 3.2, 3.3, 6.1, 8.2			

BI / read / 8

query	BI / read / 8			
title	Related topics			
pattern	<pre> graph LR Tag[tag: Tag name = \$tag] Message[Message] Comment[comment: Comment] relatedTag[relatedTag: Tag name != \$tag name] Message -- hasTag --> Tag Comment -- replyOf --> Message Comment -- hasTag --> relatedTag Comment -- count --> Count[count] </pre>			
desc.	Find all Messages that have a given Tag. Find the related Tags attached to (direct) reply Comments of these Messages, but only of those reply Comments that do not have the given Tag. Group the Tags by name, and get the count of replies in each group.			
params	1	tag	String	
result	1	relatedTag.name	String	R
	2	count	32-bit Integer	A
sort	1	count	↓	
	2	relatedTag.name	↑	
limit	100			
CPs	1.4, 3.3, 5.2, 8.1			

BI / read / 9

query	BI / read / 9				
title	Forum with related tags				
pattern	<pre>graph TD Forum[Forum] -- containerOf --> Post1[Post] Forum -- containerOf --> Post2[Post] Post1 -- hasTag --> Tag1[Tag] Post2 -- hasTag --> Tag2[Tag] Tag1 -- hasType --> TagClass1[TagClass] Tag2 -- hasType --> TagClass2[TagClass] Forum -- hasMember --> Person[Person] count1 = count1 count2 = count2 threshold < members</pre>				
desc.	<p>Given two TagClasses (tagClass1 and tagClass2), find Forums that contain</p> <ul style="list-style-type: none">at least one Post (post1) with a Tag with a (direct) type of tagClass1 andat least one Post (post2) with a Tag with a (direct) type of tagClass2. <p>The post1 and post2 nodes may be the same Post.</p> <p>Consider the Forums with a number of members greater than a given threshold. For every such Forum, count the number of post1 nodes (count1) and the number of post2 nodes (count2).</p>				
params	1	tagClass1	String		
	2	tagClass2	String		
	3	threshold	32-bit Integer		
result	1	forum.id	64-bit Integer	R	
	2	count1	32-bit Integer	A	Number of post1 nodes
	3	count2	32-bit Integer	A	Number of post2 nodes
sort	1	abs(count2 - count1)		↓	
	2	forum.id		↑	
limit	100				
CPs	1.2, 1.3, 2.1, 2.3, 2.4, 8.2				

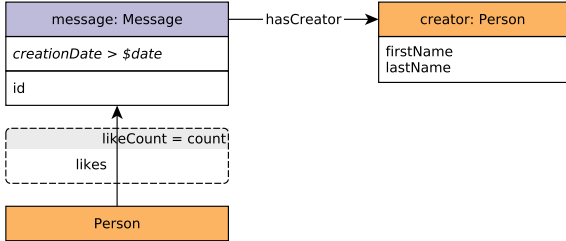
BI / read / 10

BI 1	query	BI / read / 10			
BI 2	title	Central person for a tag			
BI 3	pattern				
BI 4					
BI 5					
BI 6					
BI 7					
BI 8					
BI 9					
BI 10					
BI 11					
BI 12					
BI 13	desc.	<p>Given a Tag, find all Persons that are interested in the Tag and/or have written a Message (Post or Comment) with a <code>creationDate</code> after a given date and that has a given Tag. For each Person, compute the <code>score</code> as the sum of the following two aspects:</p> <ul style="list-style-type: none"> • 100, if the Person has this Tag as their interest, or 0 otherwise • number of Messages by this Person with the given Tag <p>Also, for each Person, compute the sum of the score of the Person's friends (<code>friendsScore</code>).</p>			
BI 14					
BI 15					
BI 16					
BI 17					
BI 18					
BI 19					
BI 20					
BI 21	params	1	tag	String	
BI 22		2	date	Date	
BI 23	result	1	person.id	64-bit Integer	R
BI 24		2	score	32-bit Integer	A
BI 25		3	friendsScore	32-bit Integer	A
	sort	1	score + friendsScore	↓	
		2	person.id	↑	
	limit	100			
	CPs	1.2, 2.1, 2.3, 3.2, 8.2, 8.4, 8.5			

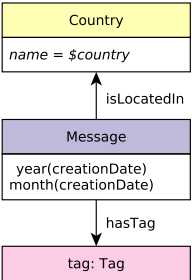
BI / read / 11

query	BI / read / 11																								
title	Unrelated replies																								
pattern																									
desc.	<p>Find those Persons of a given Country that replied to any Message, such that the reply does not have any Tag in common with the Message (only direct replies are considered, transitive ones are not). Consider only those replies that do no contain any word from a given <code>blacklist</code>. For each Person and valid reply, retrieve the Tags associated with the reply, and retrieve the number of likes on the reply.</p> <p>The detailed conditions for checking blacklisted words are currently as follows. Words do not have to stand separately, i.e. if the word “Green” is blacklisted, “South-Greenland” cannot be included in the results. Also, comparison should be done in a case-sensitive way. These conditions are preliminary and might be changed in later versions of the benchmark.</p>																								
params	<table><tr><td>1</td><td>country</td><td>String</td><td></td></tr><tr><td>2</td><td>blacklist</td><td>String[]</td><td></td></tr></table>					1	country	String		2	blacklist	String[]													
1	country	String																							
2	blacklist	String[]																							
result	<table><tr><td>1</td><td>person.id</td><td>64-bit Integer</td><td>R</td><td></td></tr><tr><td>2</td><td>tag.name</td><td>String</td><td>R</td><td></td></tr><tr><td>3</td><td>likeCount</td><td>32-bit Integer</td><td>A</td><td>The count of likes to replies with that Tag</td></tr><tr><td>4</td><td>replyCount</td><td>32-bit Integer</td><td>A</td><td>The count of replies with that Tag</td></tr></table>					1	person.id	64-bit Integer	R		2	tag.name	String	R		3	likeCount	32-bit Integer	A	The count of likes to replies with that Tag	4	replyCount	32-bit Integer	A	The count of replies with that Tag
1	person.id	64-bit Integer	R																						
2	tag.name	String	R																						
3	likeCount	32-bit Integer	A	The count of likes to replies with that Tag																					
4	replyCount	32-bit Integer	A	The count of replies with that Tag																					
sort	<table><tr><td>1</td><td>likeCount</td><td>↓</td><td></td></tr><tr><td>2</td><td>person.id</td><td>↑</td><td></td></tr><tr><td>3</td><td>tag.name</td><td>↑</td><td></td></tr></table>					1	likeCount	↓		2	person.id	↑		3	tag.name	↑									
1	likeCount	↓																							
2	person.id	↑																							
3	tag.name	↑																							
limit	100																								
CPs	1.1, 2.1, 2.2, 2.3, 3.1, 3.2, 6.1, 8.1, 8.3																								

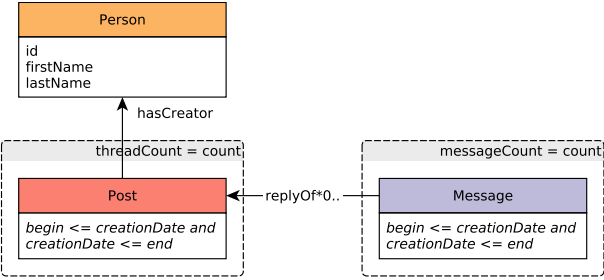
BI / read / 12

BI 1	query	BI / read / 12			
BI 2	title	Trending posts			
BI 3	pattern	 <pre> graph TD subgraph Message [message: Message] direction TB CDate[creationDate > \$date] ID[id] Likes[likes] end subgraph Person [creator: Person] direction TB FName[firstName] LName[lastName] end Message -- hasCreator --> Person Likes -- likeCount = count --> Count[] Person --> Count </pre>			
BI 11	desc.	Find all Messages created after a given date (exclusive), that received more than a given number of likes (likeThreshold).			
BI 13	params	1	date	Date	
BI 14		2	likeThreshold	32-bit Integer	
BI 16	result	1	message.id	64-bit Integer	R
BI 17		2	message.creationDate	DateTime	R
BI 18		3	creator.firstName	String	R
BI 19		4	creator.lastName	String	R
BI 20		5	likeCount	32-bit Integer	A
BI 21					
BI 22	sort	1	likeCount	↓	
BI 23		2	message.id	↑	
BI 24	limit	100			
BI 25	CPs	1.2, 2.2, 3.1, 6.1, 8.5			

BI / read / 13

BI 1	query	BI / read / 13			
BI 2	title	Popular tags per month in a country			
BI 3	pattern	 <pre> graph TD Country[Country name = \$country] Message[Message year(creationDate) month(creationDate)] Tag[Tag tag: Tag] Message -- isLocatedIn --> Country Message -- hasTag --> Tag </pre>			
BI 12	desc.	<p>Find all Messages in a given Country, as well as their Tags.</p> <p>Group Messages by creation year and month. For each group, find the 5 most popular Tags, where popularity is the number of Messages (from within the same group) where the Tag appears.</p> <p>Note: even if there are no Tags for Messages in a given year and month, the result should include the year and month with an empty popularTags list.</p>			
BI 17	params	1	country	String	
BI 19	result	1	year	32-bit Integer	C
BI 20		2	month	32-bit Integer	C
BI 21		3	popularTags	TagPairs	C
BI 22					(tag.name [String], popularity [32-bit Integer])
BI 23					pairs, sorted descending by popularity, then
BI 24					ascending by tag.name
BI 25	sort	1	year	↓	
		2	month	↑	
	limit	100			
	CPs	1.2, 2.2, 2.3, 3.2, 6.1, 8.3, 8.5			

BI / read / 14

BI 1	query	BI / read / 14			
BI 2	title	Top thread initiators			
BI 3	pattern				
BI 12	desc.	<p>For each Person, count the number of Posts they created in the time interval [startDate, endDate] (equivalent to the number of threads they initiated) and the number of Messages in each of their (transitive) reply trees, including the root Post of each tree. When calculating Message counts only consider messages created within the given time interval.</p> <p>Return each Person, number of Posts they created, and the count of all Messages that appeared in the reply trees (including the Post at the root of tree) they created.</p>			
BI 18	params	1	startDate	Date	
BI 19		2	endDate	Date	
BI 21	result	1	person.id	64-bit Integer	R
BI 22		2	person.firstName	String	R
BI 23		3	person.lastName	String	R
BI 24		4	threadCount	32-bit Integer	A
BI 25		5	messageCount	32-bit Integer	A
					The number of Posts created by that Person (the number of threads initiated)
					The number of Messages created in all the threads this Person initiated
	sort	1	messageCount	↓	
		2	person.id	↑	
	limit	100			
	CPs	1.2, 2.2, 2.3, 3.2, 7.2, 7.3, 7.4, 8.1, 8.5			

BI / read / 15

BI 1	query	BI / read / 15			
BI 2	title	Social normals			
BI 3	pattern				
BI 4					
BI 5					
BI 6					
BI 7					
BI 8					
BI 9					
BI 10					
BI 11					
BI 12					
BI 13					
BI 14					
BI 15					
BI 16					
BI 17					
BI 18					
BI 19					
BI 20	desc.	<p>Given a Country <code>country</code>, determine the “social normal”, i.e. the floor of average number of friends that Persons of <code>country</code> have in <code>country</code>.</p> <p>Then, find all Persons in <code>country</code>, whose number of friends in <code>country</code> equals the social normal value.</p>			
BI 21	params	1	country	String	
BI 22					
BI 23	result	1	person.id	64-bit Integer	R
BI 24		2	count	32-bit Integer	A
BI 25	sort	1	person.id	↑	
	limit	100			
	CPs	1.2, 2.3, 3.2, 3.3, 5.3, 6.1, 8.2, 8.4			

BI / read / 16

query	BI / read / 16																							
title	Experts in social circle																							
pattern																								
desc.	<p>Given a Person, find all other Persons that live in a given Country and are connected to given Person by a transitive trail with length in range $[\text{minPathDistance}, \text{maxPathDistance}]$ through the knows relation.</p> <p>In the trail, an edge can be only traversed once while nodes can be traversed multiple times (as opposed to a path which allows repetitions of both nodes and edges).</p> <p>For each of these Persons, retrieve all of their Messages that contain at least one Tag belonging to a given TagClass (direct relation not transitive). For each Message, retrieve all of its Tags.</p> <p>Group the results by Persons and Tags, then count the Messages by a certain Person having a certain Tag.</p> <p>(Note: it is not yet decided whether a Person connected to the start Person on a trail with a length smaller than minPathDistance, but also on a trail with the length in $[\text{minPathDistance}, \text{maxPathDistance}]$ should be included. The current reference implementations allow such Persons, but this might be subject to change in the future.)</p>																							
params	<table><tr><td>1</td><td>personId</td><td>64-bit Integer</td><td></td></tr><tr><td>2</td><td>country</td><td>String</td><td></td></tr><tr><td>3</td><td>tagClass</td><td>String</td><td></td></tr><tr><td>4</td><td>minPathDistance</td><td>32-bit Integer</td><td></td></tr><tr><td>5</td><td>maxPathDistance</td><td>32-bit Integer</td><td></td></tr></table>				1	personId	64-bit Integer		2	country	String		3	tagClass	String		4	minPathDistance	32-bit Integer		5	maxPathDistance	32-bit Integer	
1	personId	64-bit Integer																						
2	country	String																						
3	tagClass	String																						
4	minPathDistance	32-bit Integer																						
5	maxPathDistance	32-bit Integer																						
result	<table><tr><td>1</td><td>person.id</td><td>64-bit Integer</td><td>R</td><td></td></tr><tr><td>2</td><td>tag.name</td><td>String</td><td>R</td><td></td></tr><tr><td>3</td><td>messageCount</td><td>32-bit Integer</td><td>A</td><td>Number of Messages created by that Person containing that Tag</td></tr></table>				1	person.id	64-bit Integer	R		2	tag.name	String	R		3	messageCount	32-bit Integer	A	Number of Messages created by that Person containing that Tag					
1	person.id	64-bit Integer	R																					
2	tag.name	String	R																					
3	messageCount	32-bit Integer	A	Number of Messages created by that Person containing that Tag																				
sort	<table><tr><td>1</td><td>messageCount</td><td>↓</td><td></td></tr><tr><td>2</td><td>tag.name</td><td>↑</td><td></td></tr><tr><td>3</td><td>person.id</td><td>↑</td><td></td></tr></table>				1	messageCount	↓		2	tag.name	↑		3	person.id	↑									
1	messageCount	↓																						
2	tag.name	↑																						
3	person.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																							

BI / read / 17

BI 1	query	BI / read / 17		
BI 2	title	Friend triangles		
BI 3	pattern			
BI 4				
BI 5				
BI 6				
BI 7				
BI 8				
BI 9				
BI 10	desc.	<p>For a given <code>country</code>, count all the distinct triples of Persons such that:</p> <ul style="list-style-type: none"> • <code>a</code> is friend of <code>b</code>, • <code>b</code> is friend of <code>c</code>, • <code>c</code> is friend of <code>a</code>. <p>Distinct means that given a triple t_1 in the result set R of all qualified triples, there is no triple t_2 in R such that t_1 and t_2 have the same set of elements.</p>		
BI 11				
BI 12				
BI 13				
BI 14				
BI 15				
BI 16				
BI 17				
BI 18	params	1	country	String
BI 19				
BI 20				
BI 21	result	1	count	32-bit Integer
BI 22				A
BI 23				
BI 24	CPs	1.1, 2.3		
BI 25				

BI / read / 18

query	BI / read / 18				
title	How many persons have a given number of messages				
pattern	<div><div><div>2. personCount = count</div><div><div>Person</div></div><div>aggregate Persons having the same messageCount value</div></div><div><div>1. messageCount = count</div><div><div>Message</div><div>content not empty & length < \$lengthThreshold & creationDate > \$date</div></div><div>replyOf*0..→<div><div>Post</div><div>language in \$languages</div></div></div><div>←hasCreator</div></div></div>				
desc.	<p>For each Person, count the number of Messages they made (messageCount). Only count Messages with the following attributes:</p> <ul style="list-style-type: none">• Its content is not empty (and consequently, imageFile empty for Posts).• Its length is below the lengthThreshold (exclusive, equality is not allowed).• Its creationDate is after date (exclusive, equality is not allowed).• It is written in any of the given languages. <p>– The language of a Post is defined by its language attribute.</p> <p>– The language of a Comment is that of the Post that initiates the thread where the Comment replies to.</p> <p>The Post and Comments in the reply tree’s path (from the Message to the Post) do not have to satisfy the constraints for content, length and creationDate.</p> <p>For each messageCount value, count the number of Persons with exactly messageCount Messages (with the required attributes).</p>				
params	<div><div>1</div><div>date</div></div>	<div><div>Date</div></div>			
	<div><div>2</div><div>lengthThreshold</div></div>	<div><div>32-bit Integer</div></div>			
	<div><div>3</div><div>languages</div></div>	<div><div>String[]</div></div>			
result	<div><div>1</div><div>messageCount</div></div>	<div><div>32-bit Integer</div></div>	<div><div>A</div></div>	<div>Number of Messages created</div>	
	<div><div>2</div><div>personCount</div></div>	<div><div>32-bit Integer</div></div>	<div><div>A</div></div>	<div>Number of Persons with messageCount messages</div>	
sort	<div><div>1</div><div>personCount</div></div>	<div><div>↓</div></div>			
	<div><div>2</div><div>messageCount</div></div>	<div><div>↓</div></div>			
CPs	1.1, 1.2, 1.4, 3.2, 4.2, 4.3, 8.1, 8.2, 8.3, 8.4, 8.5				

BI / read / 19

query	BI / read / 19																			
title	Stranger's interaction																			
pattern																				
desc.	<p>For all the Persons (person) born after a certain date, find all the strangers they interacted with, where strangers are Persons that do not know person. There is no restriction on the date that strangers were born. (Of course, person and stranger are required to be two different Persons.) Consider only strangers that are</p> <ul style="list-style-type: none">• members of Forums tagged with a Tag with a (direct) type of tagClass1 and• members of Forums tagged with a Tag with a (direct) type of tagClass2. <p>The Tags may be attached to the same Forum or they may be attached to different Forums. Interaction is defined as follows: the person has replied to a Message by the stranger B (the reply might be a transitive one). For each person, count the number of strangers they interacted with (strangerCount) and total number of times they interacted with them (interactionCount).</p>																			
params	<table><tr><td>1</td><td>date</td><td>Date</td><td></td></tr><tr><td>2</td><td>tagClass1</td><td>String</td><td></td></tr><tr><td>3</td><td>tagClass2</td><td>String</td><td></td></tr></table>					1	date	Date		2	tagClass1	String		3	tagClass2	String				
1	date	Date																		
2	tagClass1	String																		
3	tagClass2	String																		
result	<table><tr><td>1</td><td>person.id</td><td>64-bit Integer</td><td>R</td><td></td></tr><tr><td>2</td><td>strangerCount</td><td>32-bit Integer</td><td>A</td><td></td></tr><tr><td>3</td><td>interactionCount</td><td>32-bit Integer</td><td>A</td><td></td></tr></table>					1	person.id	64-bit Integer	R		2	strangerCount	32-bit Integer	A		3	interactionCount	32-bit Integer	A	
1	person.id	64-bit Integer	R																	
2	strangerCount	32-bit Integer	A																	
3	interactionCount	32-bit Integer	A																	
sort	<table><tr><td>1</td><td>interactionCount</td><td>↓</td><td></td></tr><tr><td>2</td><td>person.id</td><td>↑</td><td></td></tr></table>					1	interactionCount	↓		2	person.id	↑								
1	interactionCount	↓																		
2	person.id	↑																		
limit	100																			
CPs	1.1, 1.3, 2.1, 2.3, 2.4, 3.3, 5.1, 7.3, 7.4, 8.1, 8.5																			

BI / read / 20

BI 1	query	BI / read / 20			
BI 2	title	High-level topics			
BI 3	pattern	<p><i>UNWIND \$tagClasses AS \$tagClassName</i></p>			
BI 4					
BI 5					
BI 6					
BI 7					
BI 8					
BI 9					
BI 10					
BI 11					
BI 12					
BI 13					
BI 14					
BI 15					
BI 16	desc.	For all given TagClasses, count number of Messages that have a Tag that belongs to that TagClass or any of its children (all descendants through a transitive relation).			
BI 17	params	1	tagClasses	String[]	
BI 18					
BI 19	result	1	tagClass.name	String	R The TagClass of the root
BI 20		2	messageCount	32-bit Integer	A
BI 21					
BI 22	sort	1	messageCount	↓	
BI 23		2	tagClass.name	↑	
BI 24	limit	100			
BI 25	CPs	1.4, 2.1, 6.1, 8.1			

BI / read / 21

query	BI / read / 21				
title	Zombies in a country				
pattern	<div><div>1. zombies = collect(zombie)</div><div><div><div>Country</div><div>name = \$country</div></div><div><div>City</div><div></div></div><div><div>zombie: Person</div><div>creationDate < \$endDate & messageCount / months < 1</div></div><div><div>message: Message</div><div>creationDate < \$endDate</div></div><div>City -- isPartOf --> Country</div><div>City -- isLocatedIn --> zombie</div><div>zombie -- hasCreator --> message</div></div><div>2. for each zombie, calculate: zombieScore = zombieLikeCount / totalLikeCount</div><div><div><div>zombie: Person</div><div>IN zombies</div></div><div><div>likerPerson: Person</div><div>creationDate < \$endDate</div></div><div><div>Message</div><div></div></div><div><div>likerZombie: Person</div><div>creationDate < \$endDate & likerZombie IN zombies</div></div><div>likerPerson -- likes --> Message</div><div>likerZombie -- likes --> Message</div><div>zombie -- hasCreator --> Message</div></div><div>totalLikeCount = count(likerPerson)</div><div>zombieLikeCount = count(likerZombie)</div></div>				
desc.	<p>Find zombies within the given country, and return their zombie scores. A zombie is a Person created before the given endDate, which has created an average of $[0, 1)$ Messages per month, during the time range between profile's creationDate and the given endDate. The number of months spans the time range from the creationDate of the profile to the endDate with partial months on both end counting as one month (e.g. a creationDate of Jan 31 and an endDate of Mar 1 result in 3 months).</p> <p>For each zombie, calculate the following:</p> <ul style="list-style-type: none">• zombieLikeCount: the number of likes received from other zombies.• totalLikeCount: the total number of likes received.• zombieScore: zombieLikeCount / totalLikeCount. If the value of totalLikeCount is 0, the zombieScore of the zombie should be 0.0. <p>For both zombieLikeCount and totalLikeCount, only consider likes received from profiles that were created before the given endDate.</p>				
params	1	country	String		
	2	endDate	Date		
result	1	zombie.id	64-bit Integer	R	
	2	zombieLikeCount	32-bit Integer	A	
	3	totalLikeCount	32-bit Integer	A	
	4	zombieScore	64-bit Float	A	zombieLikeCount / totalLikeCount
sort	1	zombieScore	↓		
	2	zombie.id	↑		
limit	100				
CPs	1.2, 2.1, 2.3, 2.4, 3.2, 3.3, 5.1, 5.3, 8.2, 8.4, 8.5				

BI / read / 22

query	BI / read / 22				
title	International dialog				
pattern	<div><div>For each pair, calculate the cost as a sum of cases #1-5. Cases that has at least one match, add to the final score with the specified value. Each case only counts once, multiple matches do not increase to the score.</div><div><div><div>Country</div><div>name = \$country1</div></div><div>←isPartOf</div><div><div>city1: City</div><div>name</div></div><div>←isLocatedIn</div><div><div>person1: Person</div><div>id</div></div></div><div><div><div>Country</div><div>name = \$country2</div></div><div>←isPartOf</div><div><div>City</div></div><div>←isLocatedIn</div><div><div>person2: Person</div><div>id</div></div></div></div> <div><div><div>case 1: score += 4</div><div><div><div>person1: Person</div><div>hasCreator</div><div>Comment</div></div><div>replyOf</div><div><div>person2: Person</div><div>hasCreator</div><div>Message</div></div></div></div><div><div><div>case 2: score += 1</div><div><div><div>person1: Person</div><div>hasCreator</div><div>Message</div></div><div>replyOf</div><div><div>person2: Person</div><div>hasCreator</div><div>Comment</div></div></div></div></div><div><div><div>case 3: score += 15</div><div><div><div>person1: Person</div><div>knows</div><div>person2: Person</div></div></div></div><div><div><div>case 4: score += 10</div><div><div><div>person1: Person</div><div>likes</div><div>Message</div></div><div>hasCreator</div><div><div>person2: Person</div></div></div></div><div><div><div>case 5: score += 1</div><div><div><div>person1: Person</div></div><div>hasCreator</div><div><div>person2: Person</div><div>likes</div><div>Message</div></div></div></div></div></div></div></div>				
desc.	<p>Consider all pairs of people (person1, person2) such that one is located in a City of Country country1 and the other is located in a City of Country country2. For each City of Country country1, return the highest scoring pair. The score of a pair is defined as the sum of the subscores awarded for the following kinds of interaction. The initial value is score = 0.</p> <ol style="list-style-type: none">1. person1 has created a reply Comment to at least one Message by person2: score += 42. person1 has created at least one Message that person2 has created a reply Comment to: score += 13. person1 and person2 know each other: score += 154. person1 liked at least one Message by person2: score += 105. person1 has created at least one Message that was liked by person2: score += 1 <p>Consequently, the maximum score a pair can obtain is: 4 + 1 + 15 + 10 + 1 = 31. To break ties, order by (1) person1.id ascending and (2) person2.id ascending.</p>				
params	1	country1	String		
	2	country2	String		
result	1	person1.id	64-bit Integer	R	
	2	person2.id	64-bit Integer	R	
	3	city1.name	String	R	
	4	score	32-bit Integer	C	
sort	1	score	↓		
	2	person1.id	↑		
	3	person2.id	↑		
CPs	1.3, 1.4, 2.1, 3.1, 3.3, 5.1, 5.2, 5.3, 8.3, 8.4				

BI / read / 23

query	BI / read / 23				
title	Holiday destinations				
pattern					
desc.	<p>Count the Messages of all residents of a given Country (home), where the message was written abroad. Group the messages by month and destination.</p> <p>A Message was written abroad if it is located in a Country (destination) different than home.</p>				
params	1	country	String		
result	1	messageCount	32-bit Integer	A	The number of Messages in each group
	2	destination.name	String	R	The name of the destination Country
	3	month	32-bit Integer	C	month(message.creationDate)
sort	1	messageCount	↓		
	2	desination.name	↑		
	3	month	↑		
limit	100				
CPs	1.4, 2.3, 2.4, 3.3, 4.3, 8.5				

BI / read / 24

BI 1	query	BI / read / 24			
BI 2	title	Messages by topic and continent			
BI 3	pattern	<pre> graph TD TagClass[TagClass] -- hasType --> Tag[Tag] Tag -- hasTag --> Message[message: Message] Message -- isLocatedIn --> Country[Country] Country -- isPartOf --> Continent[continent: Continent] Person[Person] -- likes --> Message Message -- messageCount = count --> Tag Message -- likeCount = count --> Person TagClass -- name = \$tagClass --> TagClass Message -- year(creationDate) --> Message Message -- month(creationDate) --> Message Country -- continent --> Continent Continent -- name --> Continent </pre>			
BI 4					
BI 5					
BI 6					
BI 7					
BI 8					
BI 9					
BI 10					
BI 11					
BI 12					
BI 13					
BI 14					
BI 15					
BI 16					
BI 17					
BI 18	desc.	Find all Messages tagged with a Tag that has the (direct) type of the given tagClass. Count all Messages and their likes grouped by Continent, year, and month.			
BI 19	params	1	tagClass	String	
BI 20					
BI 21	result	1	messageCount	32-bit Integer	A
BI 22		2	likeCount	32-bit Integer	A
BI 23		3	year	32-bit Integer	C
BI 24		4	month	32-bit Integer	C
BI 25		5	continent.name	String	R
	sort	1	year	↑	
		2	month	↑	
		3	continent.name	↓	
	limit	100			
	CPs	1.4, 2.1, 2.3, 2.4, 3.2, 4.3, 8.5			

BI / read / 25

query	BI / read / 25				
title	Trusted connection paths through forums created in a given timeframe				
pattern	<div><div>Enumerate all shortest paths on knows edges from person1 to person2.</div><div><div>person1: Person</div><div>id = \$person1Id</div></div><div>knows*</div><div><div>person2: Person</div><div>id = \$person2Id</div></div></div> <div><div>For each edge on the path, calculate a weight based on interactions between the pair of Persons of the edge, are calculated as a sum of cases #1 and #2 for the Persons (both ways), and the sum of these weights determine the total weight of each path.</div><div>p1 — knows — pX — knows — pY — ... — pW — knows — p2</div></div> <div><div>case 1: Replies on Posts, weight += 1.0 * count(c)</div><div><div>personA: Person</div><div>hasCreator ↑</div><div>c: Comment</div><div>knows</div><div>personB: Person</div><div>hasCreator ↑</div><div>post: Post</div><div>replyOf →</div><div>containerOf ↑</div><div>forum: Forum</div><div>\$start <= creationDate & creationDate <= \$end</div></div></div> <div><div>case 2: Replies on Comments, weight += 0.5 * count(c1)</div><div><div>personA: Person</div><div>hasCreator ↑</div><div>c1: Comment</div><div>knows</div><div>personB: Person</div><div>hasCreator ↑</div><div>c2: Comment</div><div>replyOf →</div><div>replyOf* ↓</div><div>Post</div><div>containerOf ↑</div><div>forum: Forum</div><div>\$start <= creationDate & creationDate <= \$end</div></div></div>				
desc.	<p>Given two Persons, find all (unweighted) shortest paths between these two Persons, in the sub-graph induced by the knows relationship.</p> <p>Then, for each path calculate a weight. The nodes in the path are Persons, and the weight of a path is the sum of weights between every pair of consecutive Person nodes in the path.</p> <p>The weight for a pair of Persons is calculated based on their interactions:</p> <ul style="list-style-type: none">• Every direct reply (by one of the Persons) to a Post (by the other Person) contributes 1.0.• Every direct reply (by one of the Persons) to a Comment (by the other Person) contributes 0.5. <p>Only consider Messages that were created in a Forum that was created within the timeframe [startDate, endDate]. Note that for Comments, the containing Forum is that of the Post that the comment (transitively) replies to.</p> <p>Return all paths with the Person ids ordered by their weights descending.</p>				
params	1	person1Id	64-bit Integer		
	2	person2Id	64-bit Integer		
	3	startDate	Date		
	4	endDate	Date		
result	1	person.id	64-bit Integer[]	R	Identifiers representing an ordered sequence of the Persons in the path weight
	2	weight	64-bit Double	R	
sort	1	weight	↓	The order of paths with the same weight is unspecified	
	2	personIds	↑	The ids in the paths are used for lexicographical sorting	
CPs	1.2, 2.1, 2.2, 2.4, 3.3, 5.1, 5.3, 7.2, 7.3, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6				