

## BI / read / 6

BI 1	query	BI / read / 6			
BI 2	title	Most authoritative users on a given topic			
BI 3	pattern	<pre> graph TD     Tag[Tag] -- hasTag --&gt; message1[message1: Message]     person1[person: Person] -- hasCreator --&gt; message1     message1 -- "«opt» likes" --&gt; p2[p2: Person]     subgraph Compute_p2_popularityScore         p2 -- hasCreator --&gt; message2[message2: Message]         message2 -- "«opt» likes" --&gt; p3[p3: Person]         p2 -- "p2.popularityScore = count(p3)" --&gt; p3     end     p2 -- "person.authorityScore = sum(p2.popularityScore)" --&gt; person1         </pre>			
BI 6	desc.	<p>Given a Tag (tag), find all Persons (person) that ever created a Message with the Tag. For each of these Persons (person) compute their “authority score” as follows:</p> <ul style="list-style-type: none"> <li>The “authority score” is the sum of “popularity scores” of the Persons (p2) that liked any of that Person’s Messages with the given Tag (same criterion as for message1).</li> <li>A Person’s (p2) “popularity score” is defined as the total number of likes on all of their Messages (message2).</li> </ul>			
BI 10	params	1	tag	Long String	Tags with a similar amount of Messages are selected. To avoid caching, different Tags should be used than the ones in Q5 and Q7.
BI 12	result	1	person.id	ID	R
BI 13		2	authorityScore	32-bit Integer	A
BI 14	sort	1	authorityScore	↓	
BI 15		2	person1.id	↑	
BI 16	limit	100			
BI 17	CPs	1.2, 2.3, 3.3, 6.1, 8.2			
BI 18	relevance	Computing the authority scores might involve computing the popularity score for the same Person multiple times. Implementations are advised to avoid such redundant computations.			