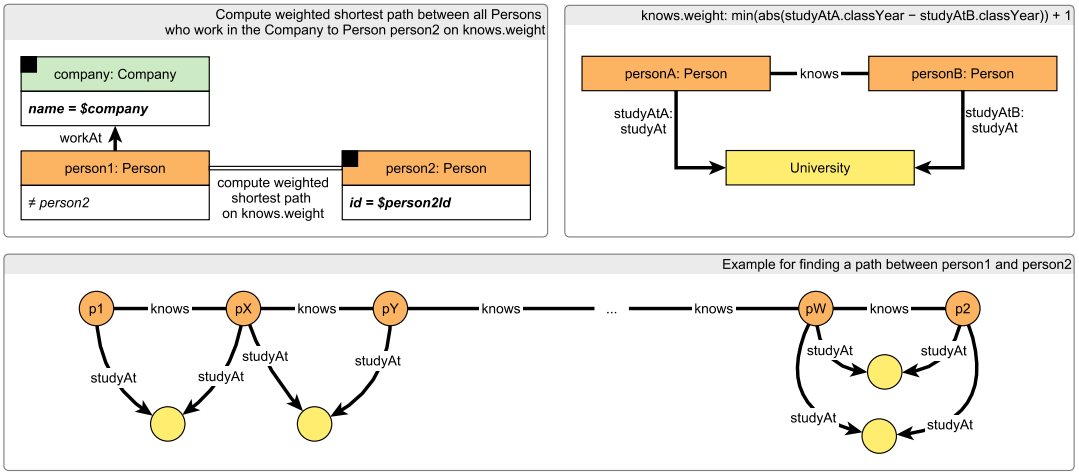
BI / read / 20

BI 1
BI 2
BI 3
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

query	BI / read / 20
title	Recruitment
pattern	
description	Consider knows edges where the endpoint Persons attended the same University and set the weight of the edge to the absolute difference between the year of enrolment plus 1. If the Persons attended multiple universities, we select the smallest (min) value. Formally: $$w = \min_{\text{studyAt}_A, \text{studyAt}_B} |\text{studyAt}_A.\text{classYear} - \text{studyAt}_B.\text{classYear}| + 1$$ Given a $company and a Person person2 with ID $person2Id (who is not working and has not worked at $company), find a different Person (person1) who works or at some point worked in $company and is reachable from person2 through people who have studied together through the shortest weighted path. If there are multiple Person person1 nodes with the same shortest path length, return all of them.

params

1	$company	Long String	Companies with a similar number of employees (former or current) are selected
2	$person2Id	ID	(a) There is guaranteed to be no path between any person1 working at company and person2 (b) There is guaranteed to be a 2-hop path between at least one person1 working at company and person

result

1	person1.id	ID	R
2	totalWeight	32-bit Integer	C

sort

1	totalWeight	↑
2	person1.id	↑

limit	20
CPs	3.3, 7.6, 7.7, 7.8, 8.4, 8.6
relevance	To find the weighted shortest paths efficiently, the system can use e.g. a bidirectional Dijkstra algorithm. As the edge weights do not depend on any parameter, systems can pre-compute them (if they do not interleave reads and writes).