

Interactive / short / 1

IS 1	query	Interactive / short / 1			
IS 2	title	Profile of a person			
IS 3	pattern	<pre> graph LR subgraph pattern [Pattern] direction TB subgraph person [person: Person] id[\$id = \$personId] fName[firstName] lName[lastName] bDay[birthday] locIP[locationIP] browser[browserUsed] gender[gender] createDate[creationDate] end city[city: City] id -- isLocatedIn --> city end </pre>			
IS 4	description	Given a start Person with ID \$personId, retrieve their first name, last name, birthday, IP address, browser, and city of residence.			
IS 5	params	1	\$personId	ID	
IS 6	result	1	person.firstName	String	R
IS 7		2	person.lastName	String	R
		3	person.birthday	Date	R
		4	person.locationIP	String	R
		5	person.browserUsed	String	R
		6	city.id	ID	R
		7	person.gender	String	R
		8	person.creationDate	DateTime	R

Interactive / short / 2

IS 1	query	Interactive / short / 2															
IS 2	title	Recent messages of a person															
IS 3	pattern	<pre> classDiagram class Person { id: String } class Message { id: String content: String / imageFile creationDate: Date } class Post { id: String } class originalPoster { id: String firstName: String lastName: String } Person "1" --> "1" Message : hasCreator Message "1" --> "1" originalPoster : hasCreator Message "*" --> "0..1" Post : replyOf originalPoster "1" --> "1" Post : hasCreator </pre>															
IS 4																	
IS 5																	
IS 6																	
IS 7																	
description																	
Given a start Person with ID \$personId, retrieve the last 10 Messages created by that user. For each Message, return that Message, the original Post in its conversation (post), and the author of that Post (originalPoster). If any of the Messages is a Post, then the original Post (post) will be the same Message, i.e. that Message will appear twice in that result.																	
params																	
result	1	\$personId	ID														
	1	message.id	ID	R													
	2	message.content or message.imageFile (for photos)	Text	R													
	3	message.creationDate	DateTime	R													
	4	post.id	ID	R													
	5	originalPoster.id	ID	R													
	6	originalPoster.firstName	String	R													
sort																	
<table border="1"> <tr> <td>1</td> <td>message.creationDate</td> <td>↓</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>message.id</td> <td>↓</td> <td></td> <td></td> <td></td> </tr> </table>						1	message.creationDate	↓				2	message.id	↓			
1	message.creationDate	↓															
2	message.id	↓															
limit																	

Interactive / short / 3

IS 1	query	Interactive / short / 3			
IS 2	title	Friends of a person			
IS 3					
IS 4					
IS 5					
IS 6					
IS 7					
	pattern	<p>The diagram shows two nodes: 'person: Person' and 'friend: Person'. The 'person' node has an attribute 'id = \$personId'. An edge labeled 'knows' with a role 'creationDate' connects it to the 'friend' node. The 'friend' node has attributes 'id', 'firstName', and 'lastName'.</p>			
	description	Given a start Person with ID \$personId, retrieve all of their friends, and the date at which they became friends.			
	params	1	\$personId	ID	
	result	1	friend.id	ID	R
		2	friend.firstName	String	R
		3	friend.lastName	String	R
		4	knows.creationDate	DateTime	R
	sort	1	knows.creationDate	↓	
		2	friend.id	↑	

Interactive / short / 4

IS 1	query	Interactive / short / 4			
IS 2	title	Content of a message			
IS 3					
IS 4					
IS 5					
IS 6					
IS 7					
	pattern	<p>The diagram shows a single node 'message: Message' with an attribute 'id = \$messageId'. It has two outgoing edges: one labeled 'creationDate' and another labeled 'content / imageFile'.</p>			
	description	Given a Message with ID \$messageId, retrieve its content and creation date.			
	params	1	\$messageId	ID	
	result	1	message.creationDate	DateTime	R messageCreationDate
		2	message.content or message.imageFile (for photos)	Text	R messageContent

Interactive / short / 5

IS 1	query	Interactive / short / 5			
IS 2	title	Creator of a message			
IS 3	pattern	<pre> classDiagram class message { <<Message>> id : String } class person { <<Person>> id : String firstName : String lastName : String } message "1" --> "1" person : hasCreator </pre>			
IS 4	description	Given a Message with ID \$messageId, retrieve its author.			
IS 5	params	1	\$messageId	ID	
IS 6					
IS 7	result	1	person.id	ID	R
		2	person.firstName	String	R
		3	person.lastName	String	R

Interactive / short / 6

IS 1	query	Interactive / short / 6			
IS 2	title	Forum of a message			
IS 3	pattern	<pre> classDiagram class message { <<Message>> id : String } class post { <<Post>> } class forum { <<Forum>> id : String title : String } class moderator { <<Person>> id : String firstName : String lastName : String } message "1" --> "0..*" post : replyOf post <--> "1" forum : containerOf forum --> "1" moderator : hasModerator </pre>			
IS 4	description	Given a Message with ID \$messageId, retrieve the Forum that contains it and the Person that moderates that Forum. Since Comments are not directly contained in Forums, for Comments, return the Forum containing the original Post in the thread which the Comment is replying to.			
IS 5	params	1	\$messageId	ID	
IS 6					
IS 7	result	1	forum.id	ID	R
		2	forum.title	Long String	R
		3	moderator.id	ID	R
		4	moderator.firstName	String	R
		5	moderator.lastName	String	R

Interactive / short / 7

IS 1	query	Interactive / short / 7							
IS 2	title	Replies of a message							
IS 3	pattern	<pre> classDiagram class message { id = \$messageId } class comment { id content creationDate } class Person { messageAuthor replyAuthor } message "1" --> "1" comment : replyOf message "1" --> "1" Person : hasCreator comment "1" --> "1" Person : hasCreator Person "1" --> "1" Person : «opt» knows </pre>							
IS 4									
IS 5	description	<p>Given a Message with ID \$messageId, retrieve the (1-hop) Comments that reply to it. In addition, return a boolean flag knows indicating if the author of the reply (replyAuthor) knows the author of the original message (messageAuthor). If author is same as original author, return False for knows flag.</p>							
IS 6		<table border="1"> <tr> <td>1</td> <td>\$messageId</td> <td>ID</td> <td></td> <td></td> </tr> </table>				1	\$messageId	ID	
1	\$messageId	ID							
IS 7	params								
result	1	comment.id	ID	R					
	2	comment.content	Text	R					
	3	comment.creationDate	DateTime	R					
	4	replyAuthor.id	ID	R					
	5	replyAuthor.firstName	String	R					
	6	replyAuthor.lastName	String	R					
	7	knows	Boolean	C	True if the knows edge exists between the replyAuthor and the messageAuthor nodes, False otherwise (including the case when the two nodes are the same)				
sort	1	comment.creationDate	↓						
	2	replyAuthor.id	↑						