LDBC
The graph & RDF benchmark reference

Social Network Benchmark: Business Intelligence workload

Gábor Szárnyas
szarnyas@db.bme.hu

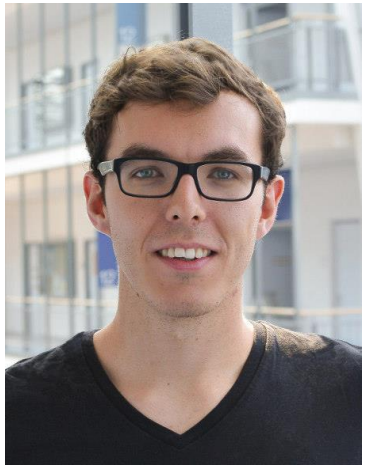
10th LDBC TUC meeting
1 September, Munich

The SNB task force

LDBC
The graph & RDF benchmark reference

Arnau Prat
Sparsity / DAMA-UPC
(Task Force Leader)

Marcus Paradies
SAP



Moritz Kaufmann
TUM / Tableau

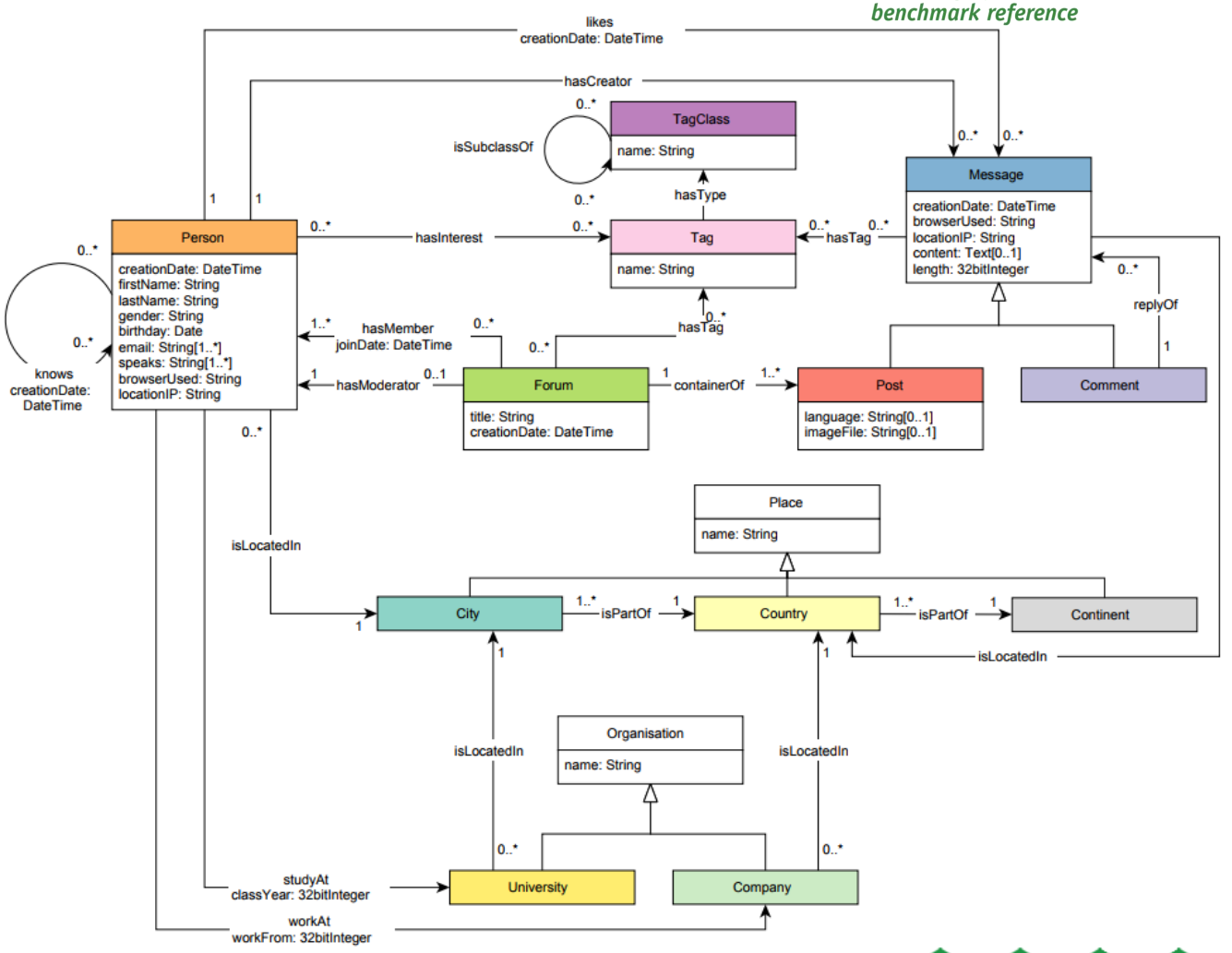
Alex Averbuch
Neo4j

Gábor Szárnyas
BME / McGill

Social Network Benchmark

LDBC
The graph & RDF benchmark reference

- Social network graph
- Scalable generator (DATAGEN)
- Analytical workload
  - Graphalytics (VLDB 2016)
- Query evaluation workloads
  - Interactive (SIGMOD 2015)
  - Business Intelligence
- Choke-point driven design



LDBC
The graph & RDF benchmark reference

Challenges

Queries difficult to comprehend

LDBC
The graph & RDF benchmark reference

Complex Reads Query Descriptions

Notes:

- Some queries require returning the content of a post. As stated in the schema, posts have content or imageFile, but not both. An empty string in content represents the post not having content, therefore, it must have a non empty string in imageFile and the other way around.

1. Friends with certain name

   - **Description:** Given a start Person, find Persons with a given first name that the start Person is connected to (excluding start Person) by at most 3 steps via Knows relationships. Return Persons, including summaries of the Persons workplaces and places of study.
   - **Parameters:**

Person.id	ID
Person.firstName	String

   - **Results:**

Person.id	ID
Person.lastName	String

Outdated docs / inconsistencies

LDBC
The graph & RDF benchmark reference

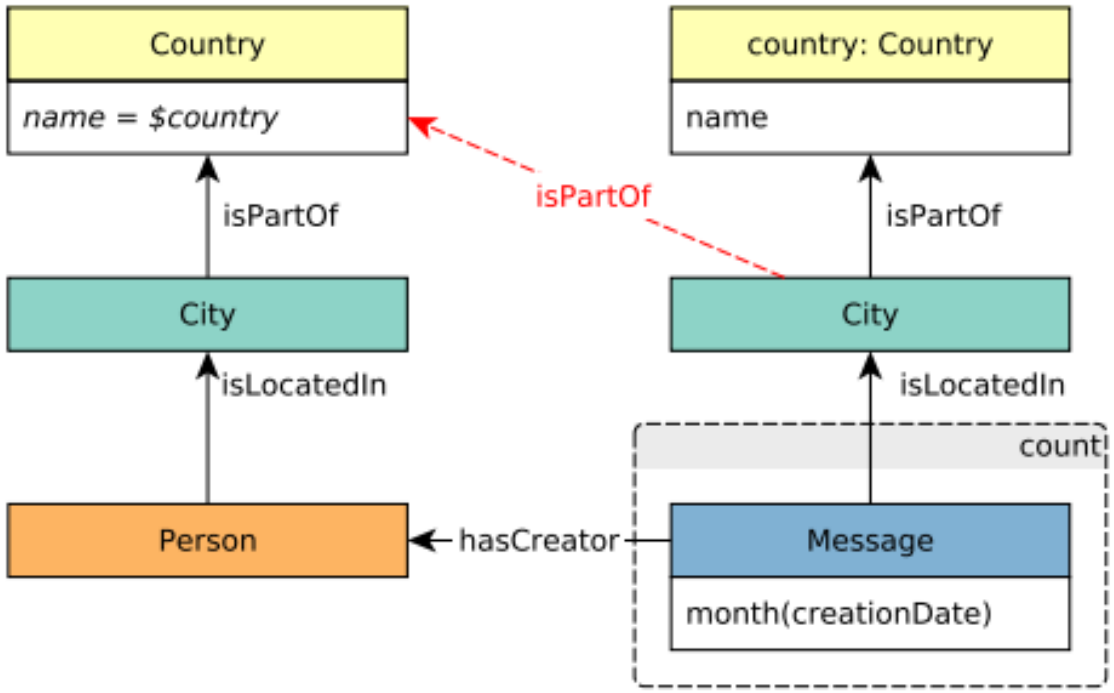
APD
POST

3 Post.id	ID
4 Post.imageFile	String
5 Post.creationDate	DateTime
6 Post.locationIp	String
7 Post.browserUsed	String
8 Post.language	String
9 Post.content	Text
10 Post.length	32-bit Integer
11 Post-hasCreator->Person.id	ID
12 Forum-containerOf->Post.id	ID
13 Post-isLocatedIn->Country.id	ID
14 {Post-hasTag->Tag.id}	{ID}

Forum.id

Graphical notation

LDBC
The graph & RDF benchmark reference



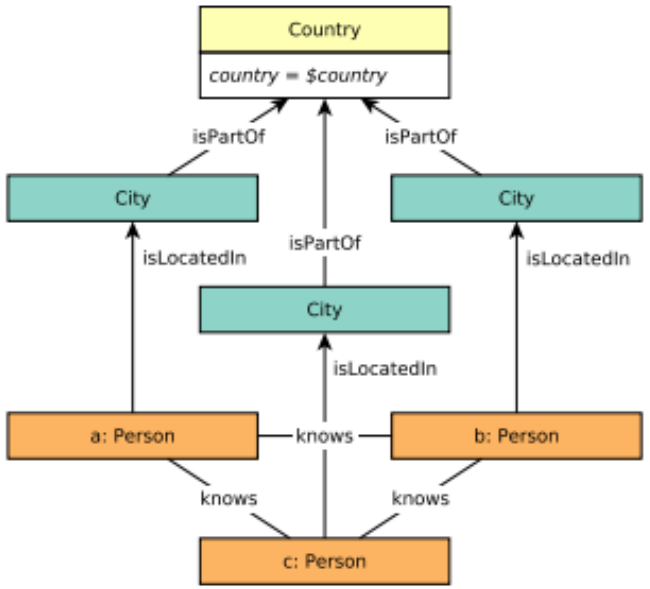
Query specification

LDBC
The graph & RDF benchmark reference

- YAML files as a single source of truth
  - Generating query cards (TeX)
  - Generating wiki entries (Markdown) is also possible
- Python/Jinja2 for defining templates

```yaml
workload: BI
operation: read
number: 17
title: Friend triangles
description: |
  For a given country, count all the distinct triples of persons such that `a` is friend of `b`, `b` is friend of `c`,
  and `c` is friend of `a`.

  Distinct means that given a triple $t_1$ in the result set $R$ of all qualified triples,
  there is not a triple $t_2$ in $R$ such that $t_1$ and $t_2$ have the same set of elements.
parameters:
  - name: country
    type: String
result:
  - name: count
    type: 32-bit Integer
    category: aggregated
choke_points: [1.1, 2.3]
```

query	BI / read / 17
title	Friend triangles
pattern	
desc.	For a given country, count all the distinct triples of persons such that `a` is friend of `b`, `b` is friend of `c`, and `c` is friend of `a`. Distinct means that given a triple t_1 in the result set R of all qualified triples, there is not a triple t_2 in R such that t_1 and t_2 have the same set of elements.
params	1 `country` String
result	1 `count` 32-bit Integer R
CPs	1.1, 2.3

LDBC
The graph & RDF benchmark reference

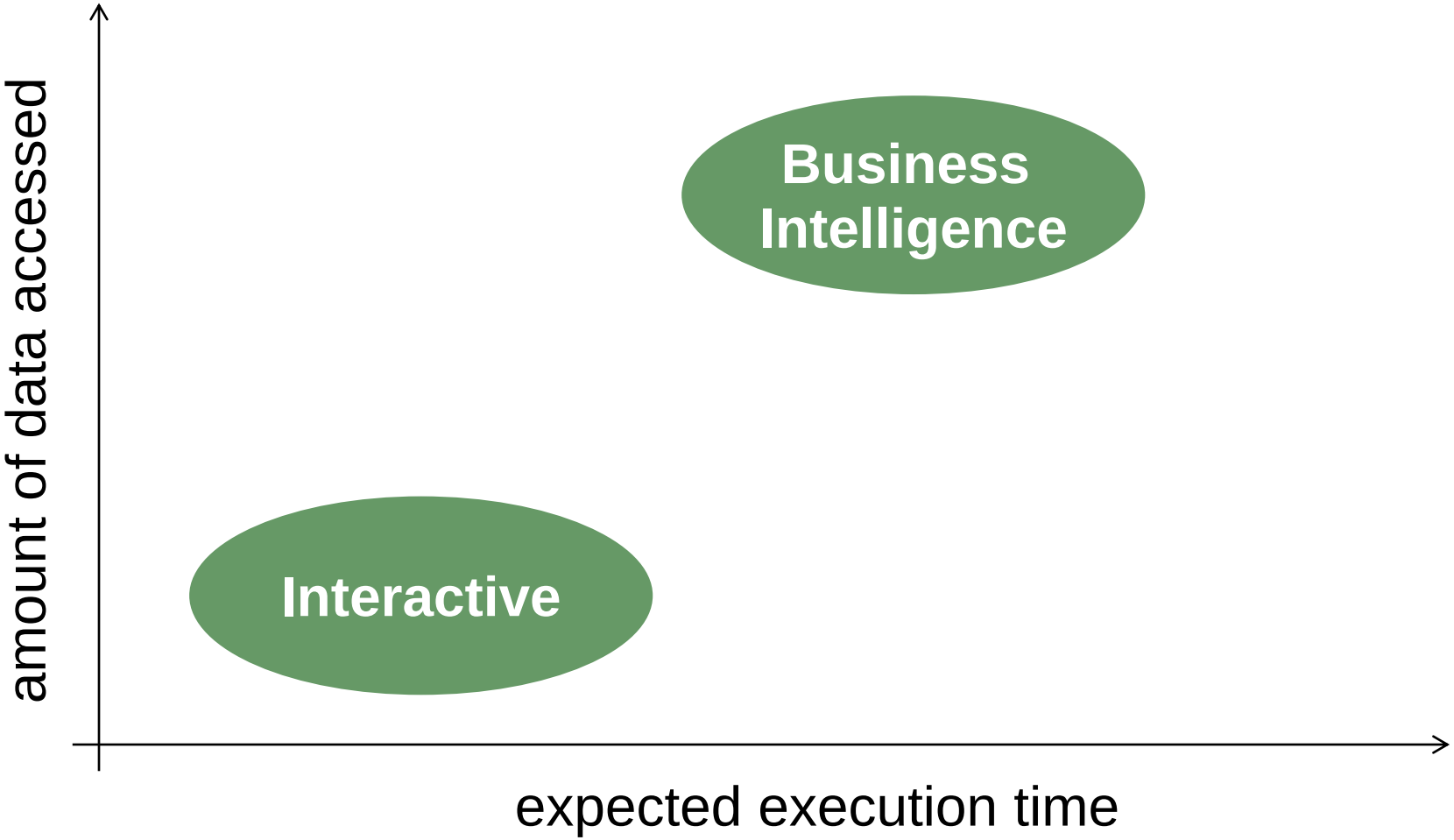
Note: In case of ambiguity, the textual description takes precedence

LDBC
The graph & RDF benchmark reference

Query evaluation workloads

Query evaluation workloads

LDBC
The graph & RDF benchmark reference



Interactive workload

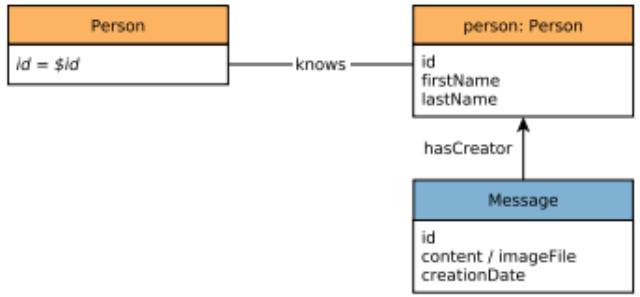
LDBC
The graph & RDF benchmark reference

- 14 complex read queries
- 8 short read queries
- 7 update queries

- **Queries explore the graph around a given node**

The LDBC Social Network Benchmark: Interactive Workload

O. Erling (Openlink), A. Averbuch (Neo), J.L. LarribaPey (UPC), Hassan Chafi (Oracle Labs), Andrey Gubichev (TU Munich), Arnau Prat (DAMA-UPC), Minh-Duc Pham (VU Amsterdam), Peter Boncz (CWI).
The LDBC Social Network Benchmark: Interactive Workload.
Proceedings of SIGMOD 2015, Melbourne

query	Interactive / complex / 2
title	Recent posts and comments by your friends
pattern	
desc.	Given a start Person, find (most recent) Messages from all of that Person's friends, that were created before (and including) a given date.

params

1	Person.id	ID	
2	date	DateTime	

result

1	Message-hasCreator->Person.id	ID	R	
2	Message-hasCreator->Person.firstName	String	R	
3	Message-hasCreator->Person.lastName	String	R	
4	Message.id	ID	R	
5	Message.content or Post.imageFile	String	R	
6	Message.creationDate	DateTime	R	

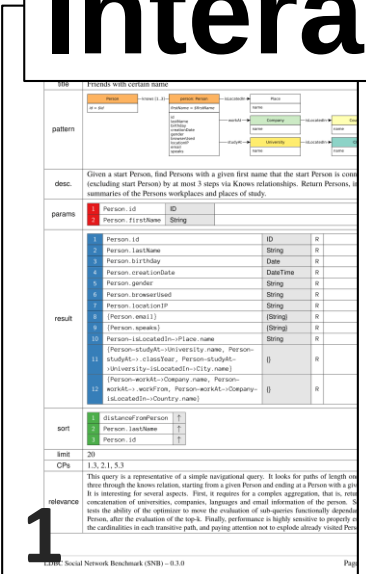
sort

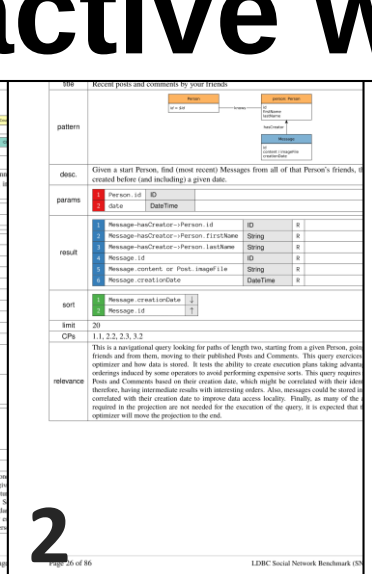
1	Message.creationDate	↓
2	Message.id	↑

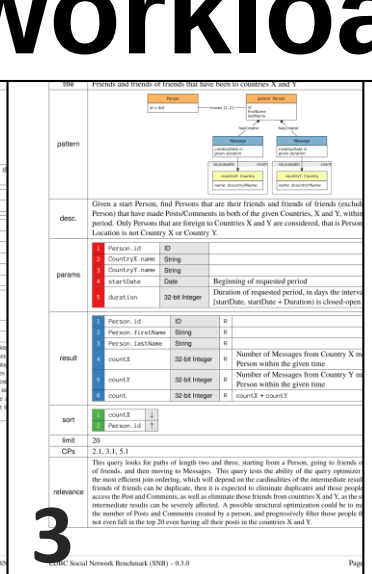
limit	20
CPs	1.1, 2.2, 2.3, 3.2

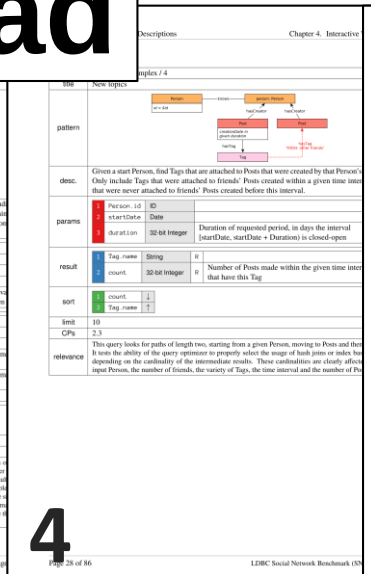
LDBC
The graph & RDF benchmark reference

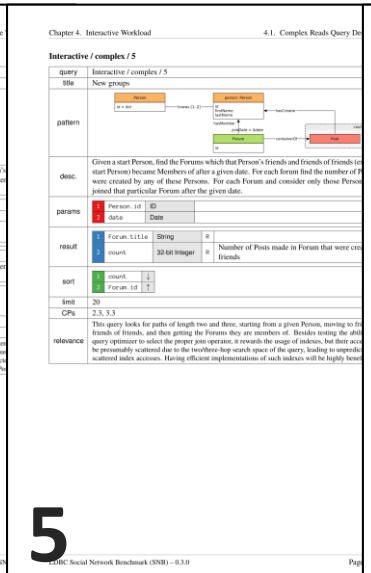
Interactive workload

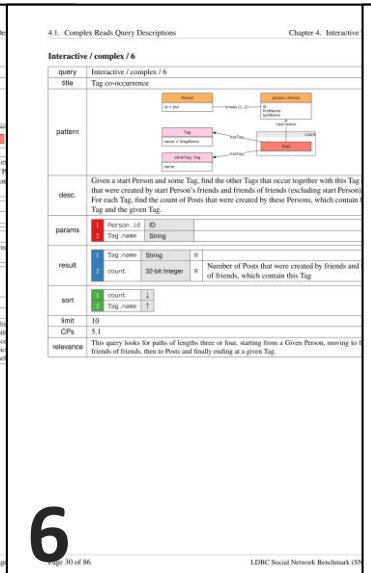

1

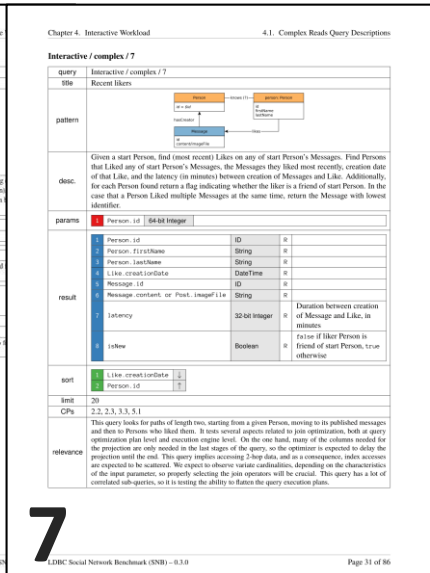

2

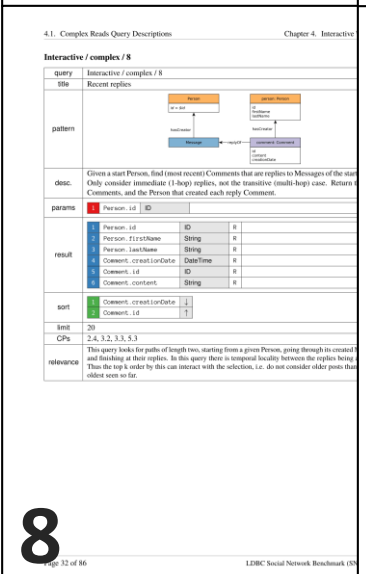

3

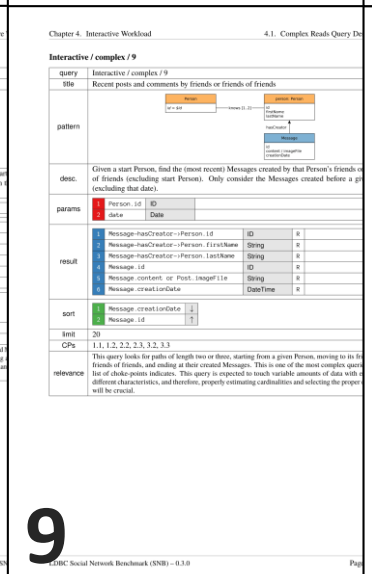

4

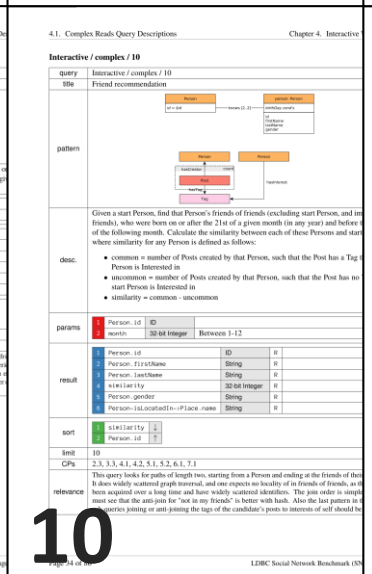

5

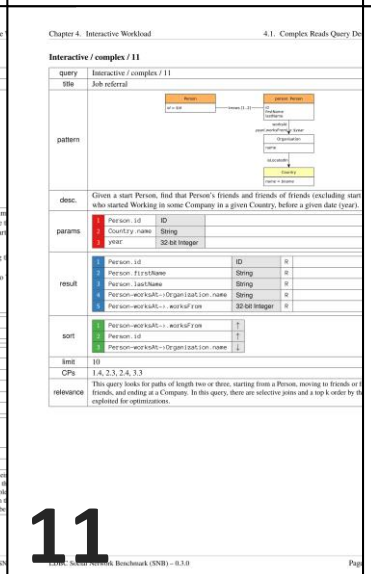

6

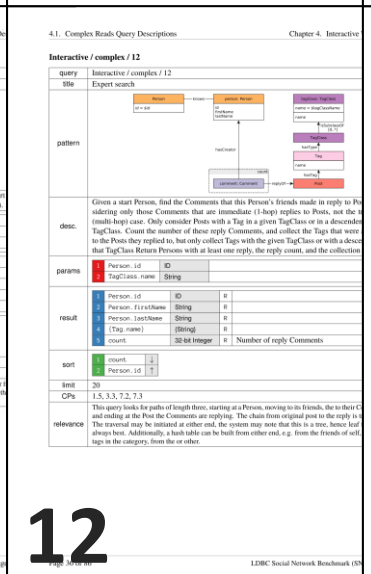

7

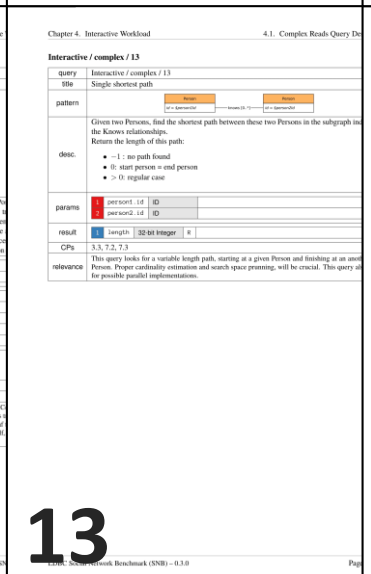

8

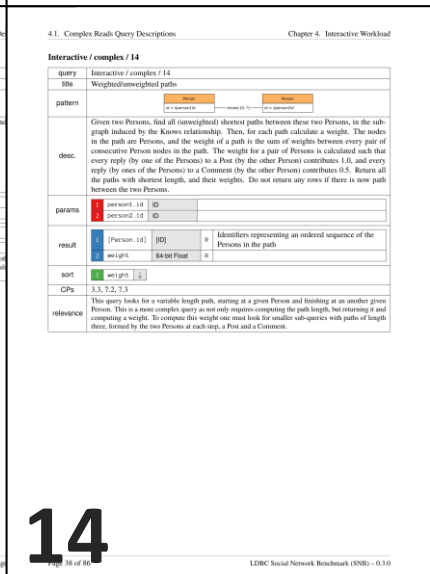

9


10


11


12


13

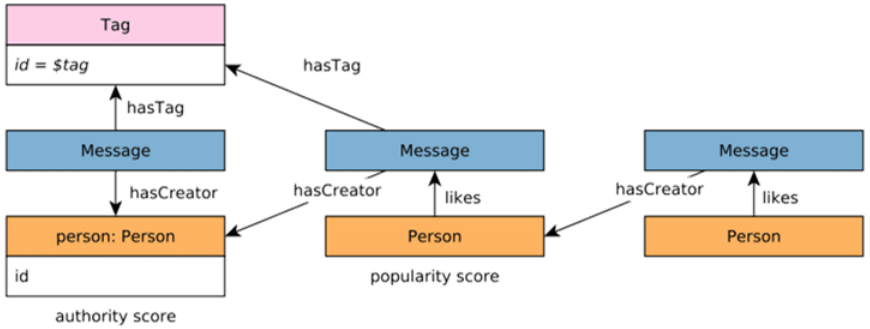

14

Business Intelligence workload

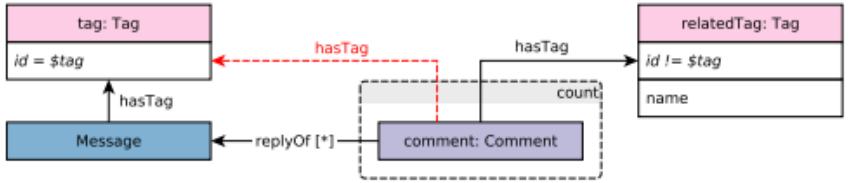
LDBC
The graph & RDF benchmark reference

- 25 read queries
- Batch updates (not yet defined)

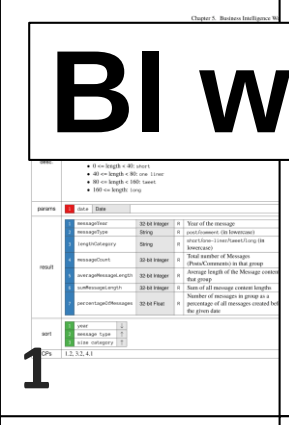
- **Queries explore vast portions of the graph**

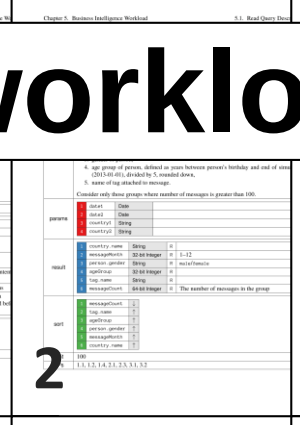
query	BI / read / 7
title	Most authoritative users on a given topic
pattern	
desc.	Given a Tag, find all Persons that ever created a Message with the given Tag. For each of these Persons compute their "authority score" as follows: • The "authority score" is the sum of "popularity scores" of the Persons that liked any of that Person's Messages with the given Tag. • A Person's "popularity score" is defined as the total number of likes on all of their Messages.
params	1 tag 32-bit Integer
result	1 person1.id 64-bit Integer R 2 authorityScore 32-bit Integer R
sort	1 authorityScore ↓ 2 person1.id ↑
limit	100
CPs	1.2, 2.3, 3.2, 3.3, 6.1

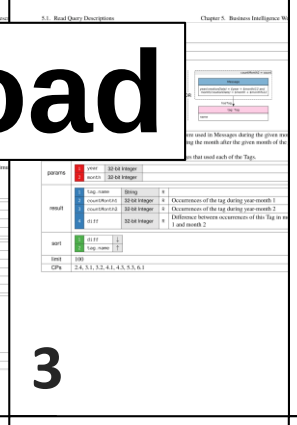
LDBC
The graph & RDF benchmark reference

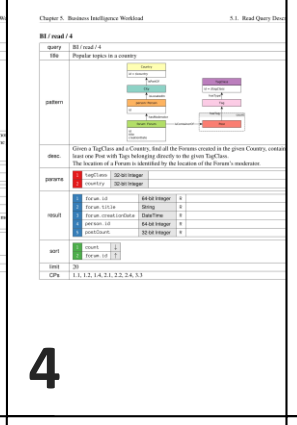
query	BI / read / 8
title	Related topics
pattern	
desc.	Find all Messages that have a given Tag. Find the related Tags attached to replies of these Messages (direct relation not transitive). but only of those replies that do not have the given Tag. Group the Tags by name, and get the count of replies in each group.
params	1 | tag | 32-bit Integer |
result	1 | relatedTag.name | String | R | 2 | count | 32-bit Integer | R |
sort	1 | count | ↓ 2 | relatedTag.name | ↑
limit	100
CPs	1.6, 3.3, 5.2

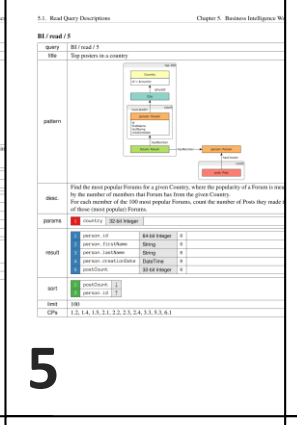
BI workload

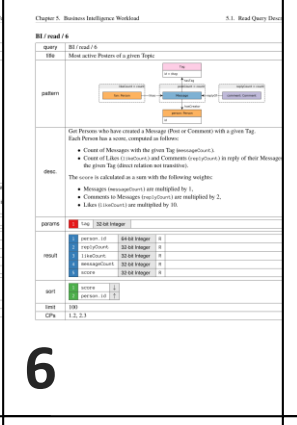

1

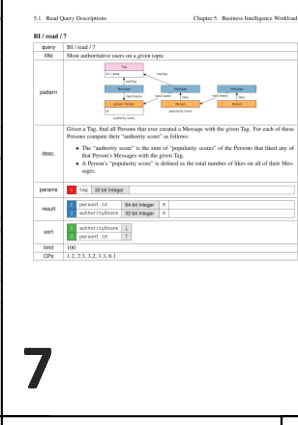

2


3


4

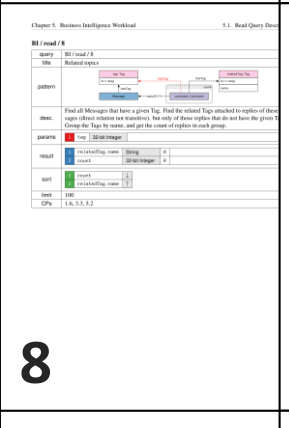

5

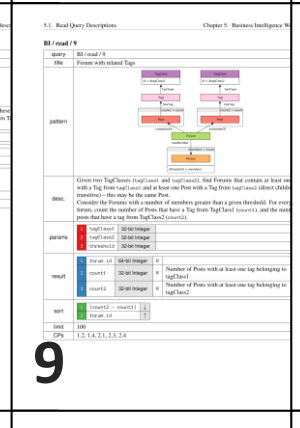

6

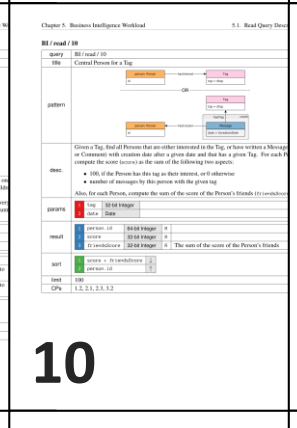

7

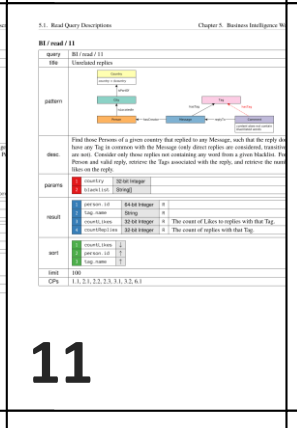
LDBC

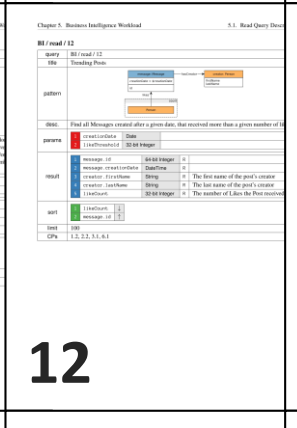
The graph & RDF benchmark reference

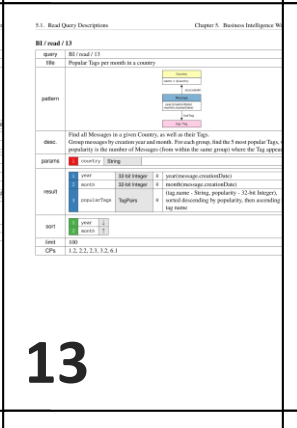

8

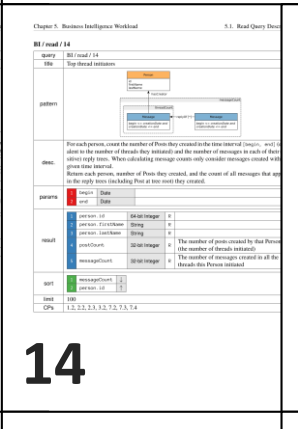

9

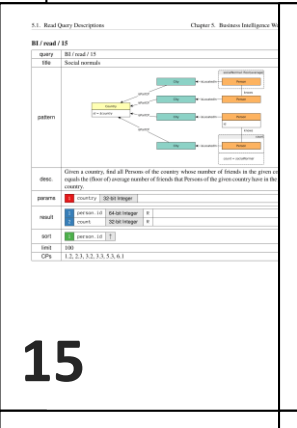

10

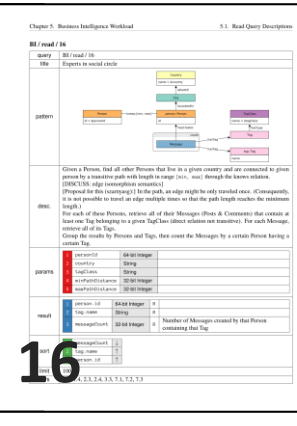

11

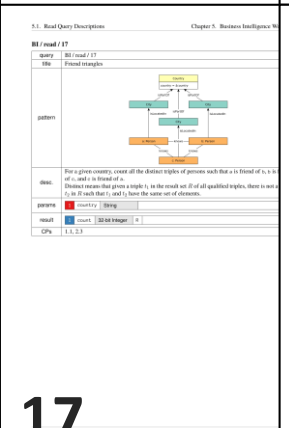

12

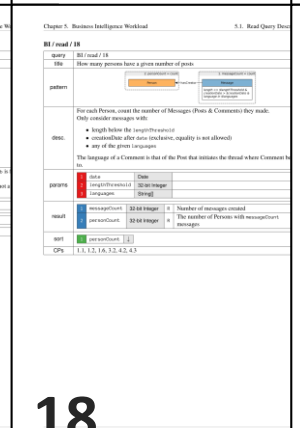

13

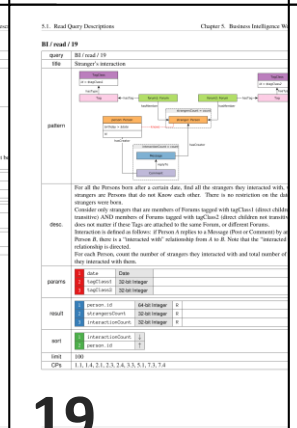

14

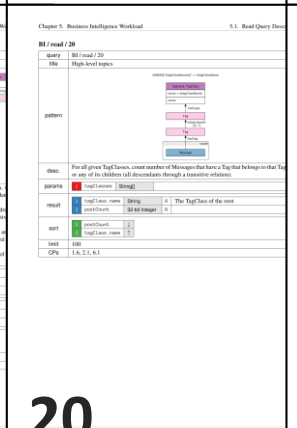

15

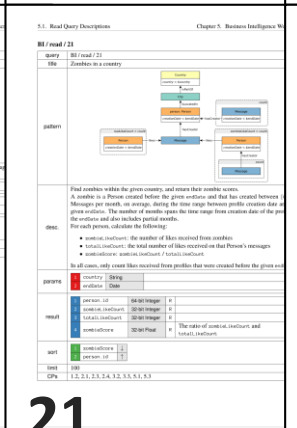

16

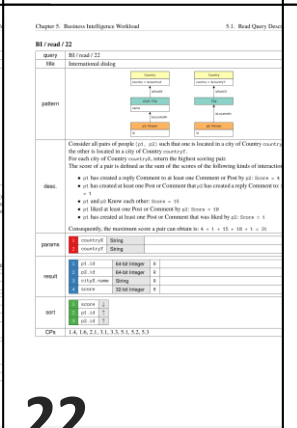

17

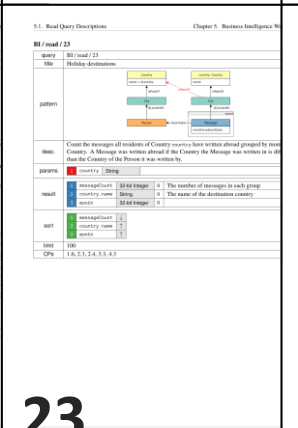

18

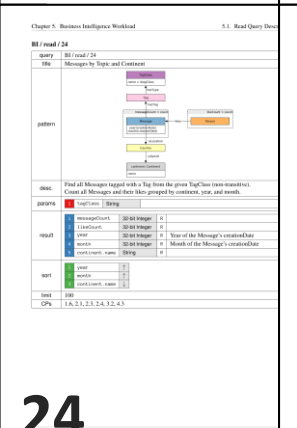

19

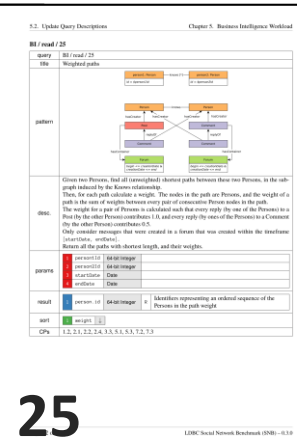

20


21


22


23


24


25

LDBC
The graph & RDF benchmark reference

Choke points

A.2 Join Performance

CP-2.1: [QOPT] Rich join order optimization

TPC-H 2.3

This choke-point tests the ability of the query optimizer to find optimal join orders. A graph can be traversed in different ways. In the relational model, this is equivalent as different join orders. The execution time of these orders may differ by orders of magnitude. Therefore, finding an efficient join (traversal) order is important, which in general, requires enumeration of all the possibilities. The enumeration is complicated by operators that are not freely re-orderable like semi-, anti-, and outer-joins. Because of this difficulty most join enumeration algorithms do not enumerate all possible plans, and therefore can miss the optimal join order. Therefore, these chokepoint tests the ability of the query optimizer to find optimal join (traversal) orders.

Queries. BI 2 BI 4 BI 5 BI 9 BI 10 BI 11 BI 19 BI 20 BI 21 BI 22 BI 24 BI 25 Interactive 1 Interactive 3

Peter Boncz, Thomas Neumann, Orri Erling.
TPC-H Analyzed: Hidden Messages and Lessons Learned from an Influential Benchmark.
TPCTC 2013

	1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3	2.4	3.1	3.2	3.3	4.1	4.2	4.3	5.1	5.2	5.3	6.1	7.1	7.2	7.3	7.4
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						●	●													●				
BI 21		●					●		●	●		●	●				●		●					
BI 22				●		●	●				●		●				●	●	●					
BI 23						●			●	●			●			●								
BI 24						●	●		●	●		●				●								
BI 25		●					●	●		●			●				●		●			●	●	

LDBC
The graph & RDF benchmark reference

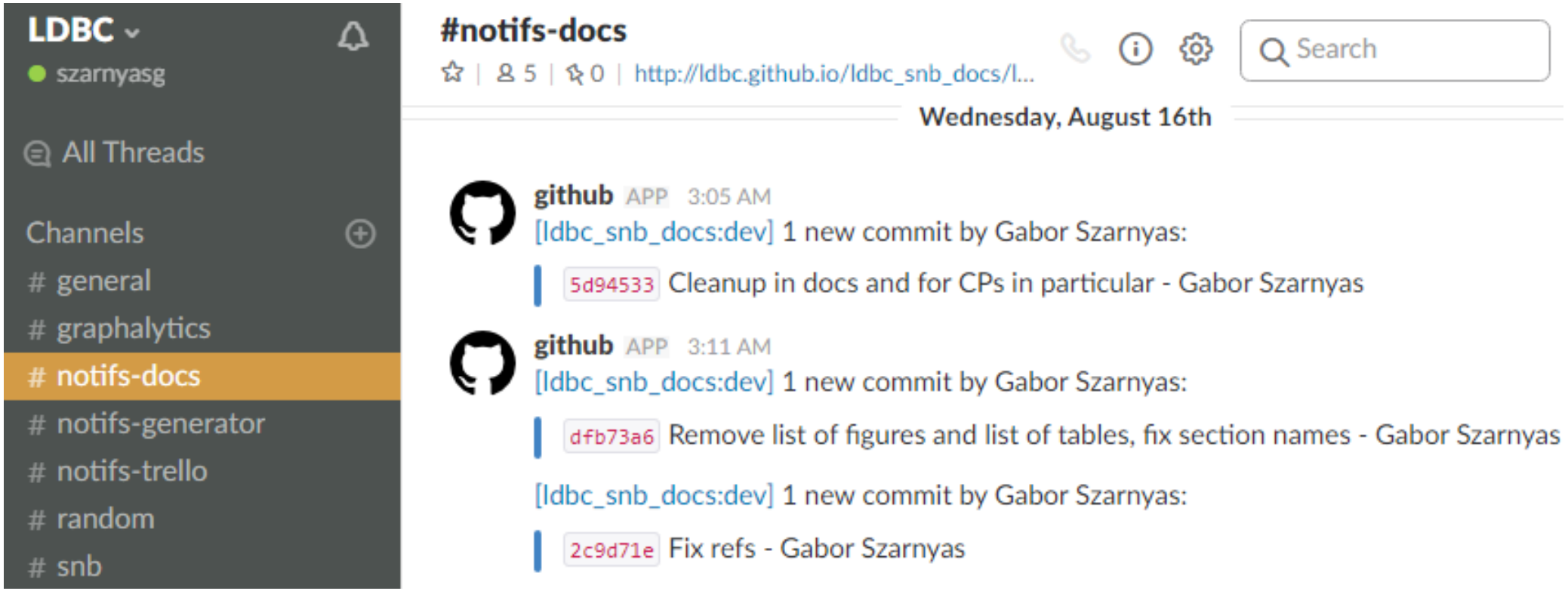
Continuous integration

LDBC
The graph & RDF benchmark reference

- Use Travis CI to
  - build DATAGEN
  - generate the technical report
- https://github.com/ldbc/ldbc_snb_docs
- https://github.com/ldbc/ldbc_snb_datagen

LDBC Slack team

https://ldbcouncil.slack.com/



Progress

LDBC
The graph & RDF benchmark reference

- 54 Trello cards
- Documentation
  - 180+ commits
  - 14 issues
  - +12 LaTeX packages
- DATAGEN
  - 40+ commits
  - Talk by Arnau at 13:30

Roadmap

LDBC
The graph & RDF benchmark reference

- Implement & validate for Neo4j, PostgreSQL and Sparksee
- Publish a subset of the benchmark in a workshop
  - GraphQ @ EDBT (late Nov)
  - GRADES @ SIGMOD (late March)
- Gather feedback & refine
- Define update operations

- We are recruiting!