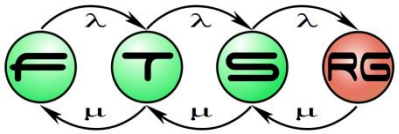
Hatékony gráflekérdezési technikák

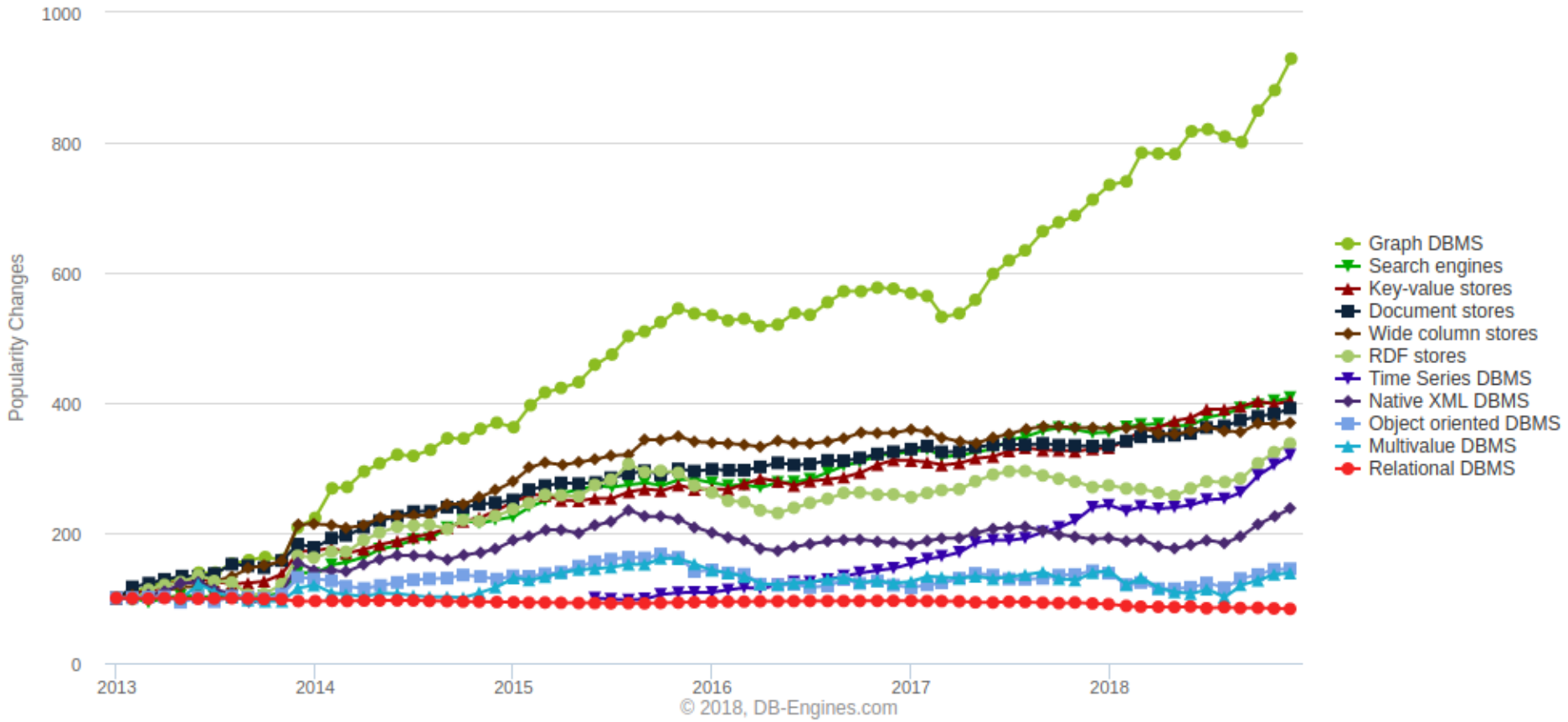
Antal János Benjamin

Konzulens: Szárnyas Gábor

Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport



Motiváció

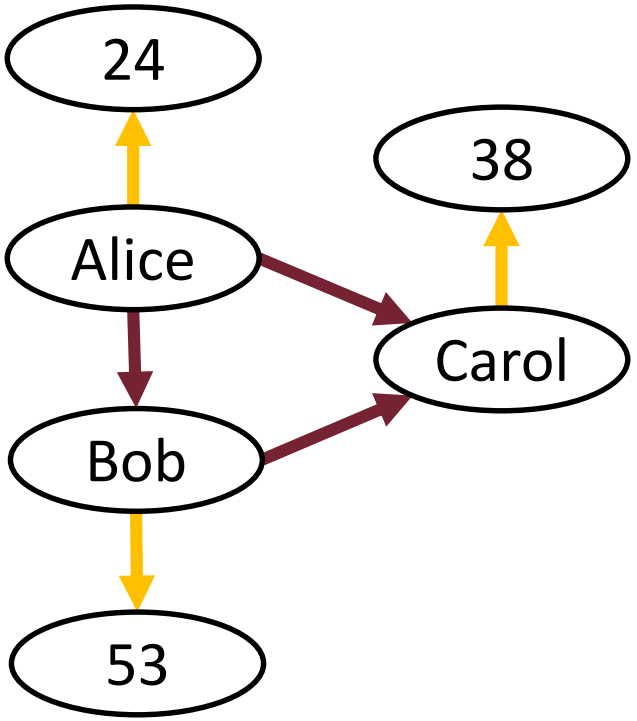

1000
800
600
400
200
0
Popularity Changes
2013
2014
2015
2016
2017
2018
© 2018, DB-Engines.com
Graph DBMS
Search engines
Key-value stores
Document stores
Wide column stores
RDF stores
Time Series DBMS
Native XML DBMS
Object oriented DBMS
Multivalue DBMS
Relational DBMS

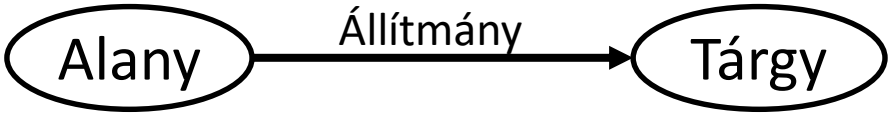
Motiváció

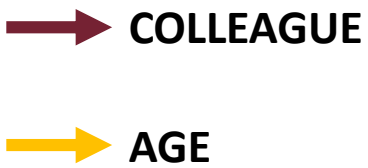
- Gráfadatbázisok előnyei
  - Intuitív adatmodell
  - Olvashatóság
  - Tömörség
  - Gyors prototipizálás
- Hatékony megoldások?
- Optimalizációk?

Szemantikus gráf

- Egy gráfban van a meta- és példánymodell
- Resource Description Framework
- Alany, állítmány, tárgy hármasok
- Csúcs- és élcímkézett gráfok

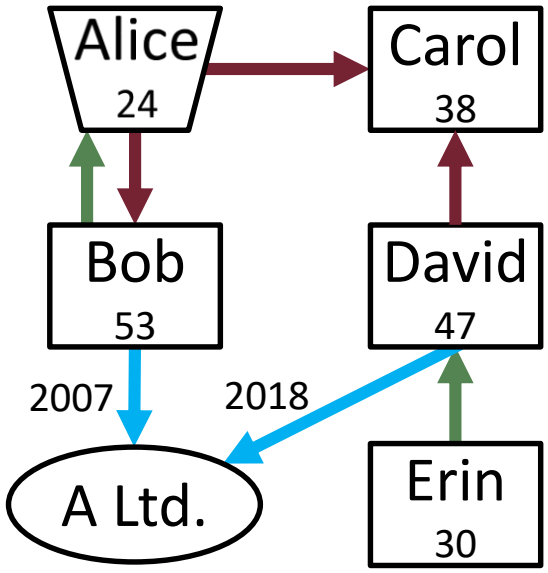


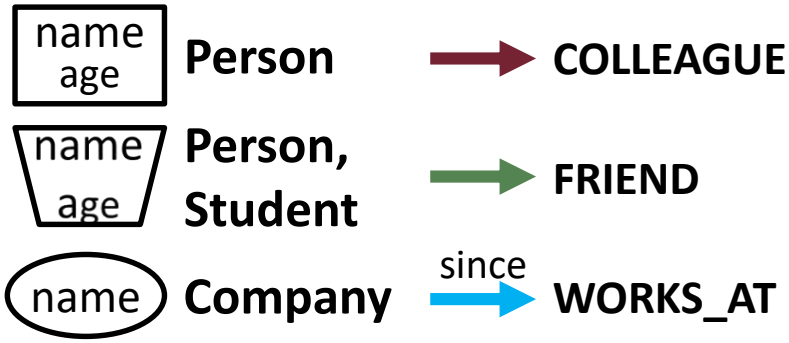




Tulajdonsággráf

- Leggyakoribb gráf-alapú adatmodell
- Tankönyvi gráf: $G = (V, E)$
- Kiterjesztve:
  - Címkézett csúcsok
  - Típusos élek
  - Csúcs- és éltulajdonságok
- Implicit séma
  - Adatok felépítése adja
  - Bármikor bővíthető

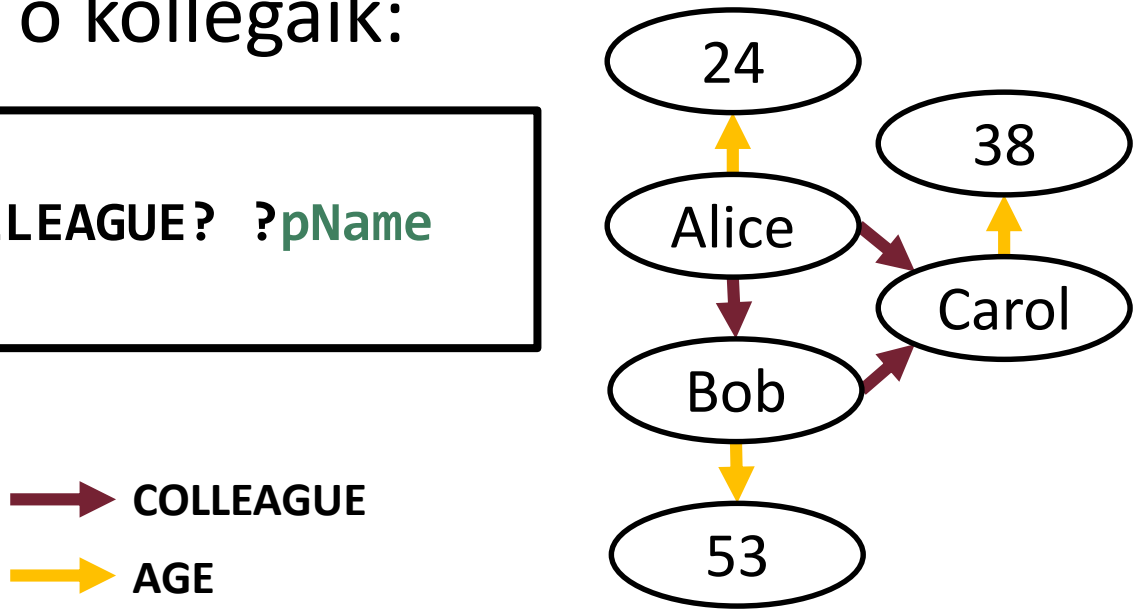




SPARQL

- Szemantikus gráfokhoz használható
- $H \leftarrow B$ alakú lekérdezés
  - B: RDF alapú komplex gráfminta (`OPTIONAL`, `UNION` stb.)
  - H: eredmény összeállítása (projekció, szelekció)
- Alice kollégái és az ő kollégáik:

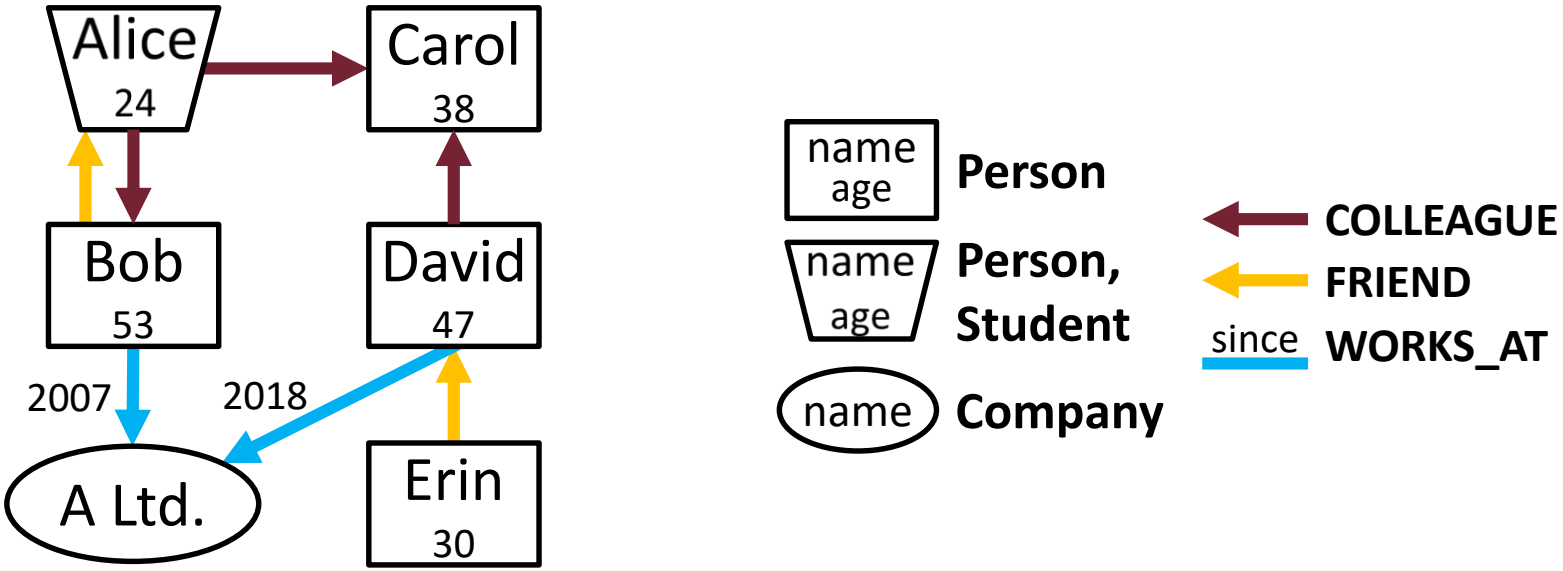
```
SELECT ?pName WHERE {
  'Alice' COLLEAGUE/COLLEAGUE? ?pName
}
```



Cypher

- Tulajdonsággráfokhoz használható
- Legelterjedtebb gráflekérdező nyelv
- Alice kollégái és az ő kollégáik:

```
MATCH (p1:Person {name: 'Alice'})-[c:COLL*1..2]-(p2:Person)
RETURN p2.name
```



TELJESÍTMÉNYMÉRÉS

Teljesítménymérési keretrendszer

- Linked Data Benchmark Council 
  - Teljesítménymérési keretrendszerek és munkafolyamatok definiálása
  - Gráf alapú adatbáziskezelőkhöz
  - Auditált eredmények publikálása
- Social Network Benchmark – LDBC SNB
  - Szintetizált közösségi háló gráf alapú adathalmaz
  - Lekérdezések specifikációja
    - *Business Intelligence* terhelési profil
    - *Interactive* terhelési profil
  - Méréshez szükséges szoftveres keretrendszer

LDBC SNB bővítése

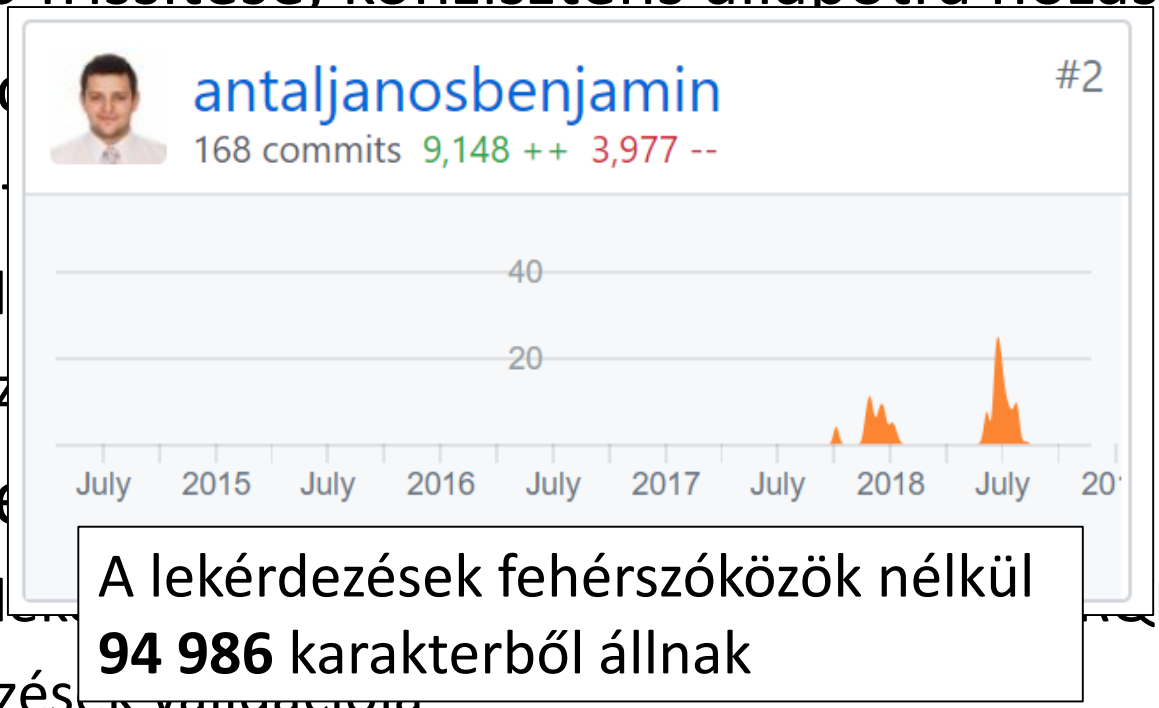
- Szükséges modulok implementálása több adatbázis-kezelőhöz
- Specifikáció frissítése, konzisztens állapotra hozása
- Hibajavítások
- Business Intelligence terhelési profil
  - 24 darab lekérdezés implementálása SPARQL nyelven
  - A lekérdezések validációja
- Interactive terhelési profil
  - 29 darab lekérdezés implementálása Cypher és SPARQL nyelven
  - A lekérdezések validációja
- Teljesítménymérés

LDBC SNB bővítése

- Szükséges modulok implementálása több adatbázis-kezelőhöz
- Specifikáció frissítése, konzisztens állapotra hozása
- Hibajavítás
- Business In
  - 24 darab l
  - A lekérdez
- Interactive
  - 29 darab l L nyelven
  - A lekérdezések validációja
- Teljesítménymérés

LDBC SNB bővítése

- Szükséges modulok implementálása több adatbázis-kezelőhöz
- Specifikáció frissítése, konzisztens állapotra hozása
- Hibajavítás
- Business In
  - 24 darab l
  - A lekérdez
- Interactive
  - 29 darab l… …QL nyelven
  - A lekérdezések validációja
- Teljesítménymérés



A lekérdezések fehérszóközök nélkül **94 986** karakterből állnak

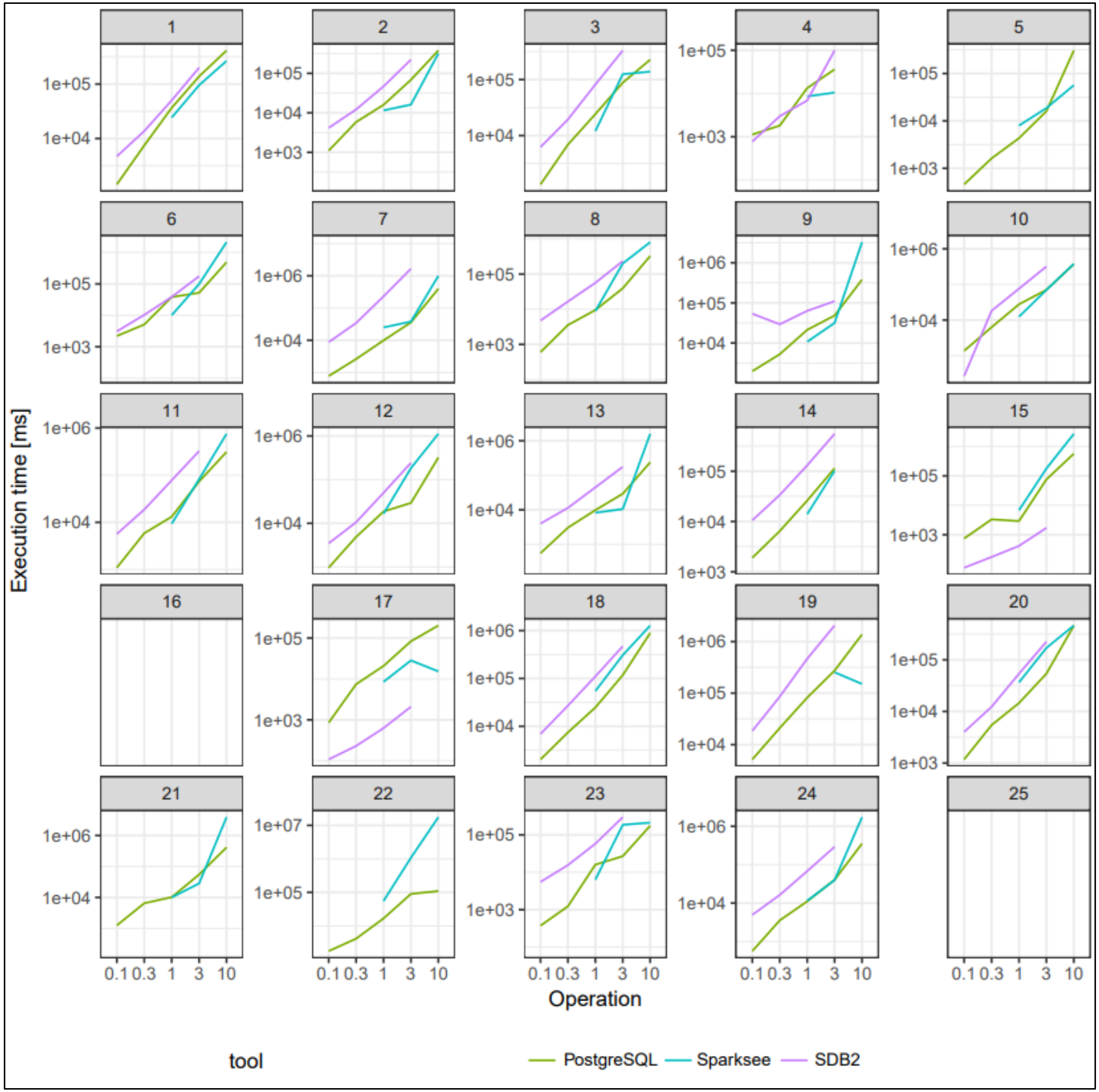
Teljesítménymérés

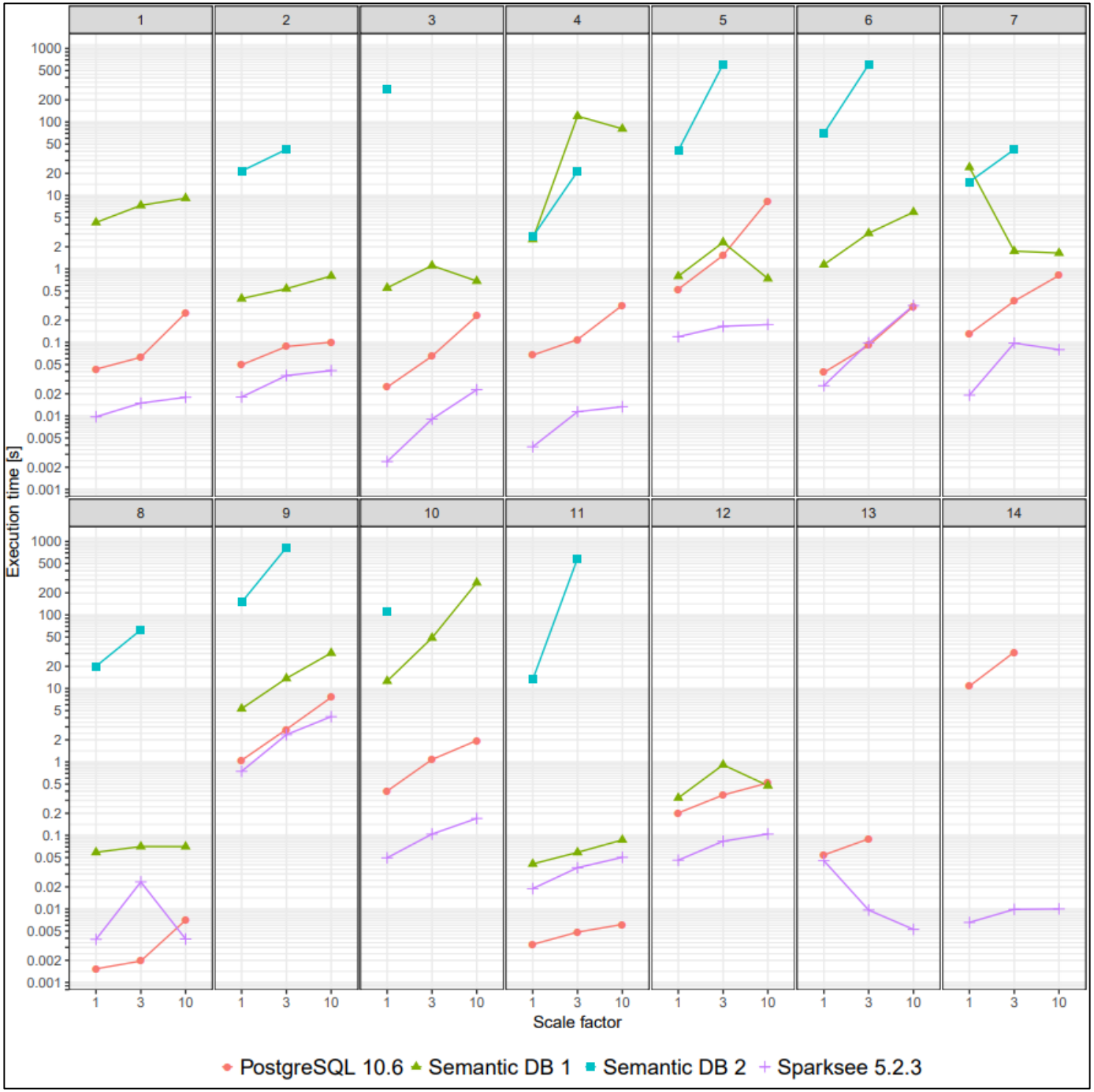
- Eszközök
  - Sparksee (referencia implementáció)
  - PostgreSQL (már meglévő implementáció)
  - SDB1: szemantikus adatbázis-kezelő
  - SDB2: szemantikus adatbázis-kezelő
- Lekérdezésenként legalább 20 db különböző behelyettesítési paraméterrel

Teljesítménymérés

- Eszközök
  - Spa
  - PostgreSQL (n neglévő implementáció)
  - SDB1: szemantikus adatbázis-kezelő
  - SDB2: szemantikus adatbázis-kezelő
- Lekérdezésenként legalább 20 db különböző behelyettesítési paraméterrel

Anonimizált eredmények, mert nem auditáltak


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
Execution time [ms]
Operation
0.1 0.3 1 3 10
tool
PostgreSQL
Sparksee
SDB2

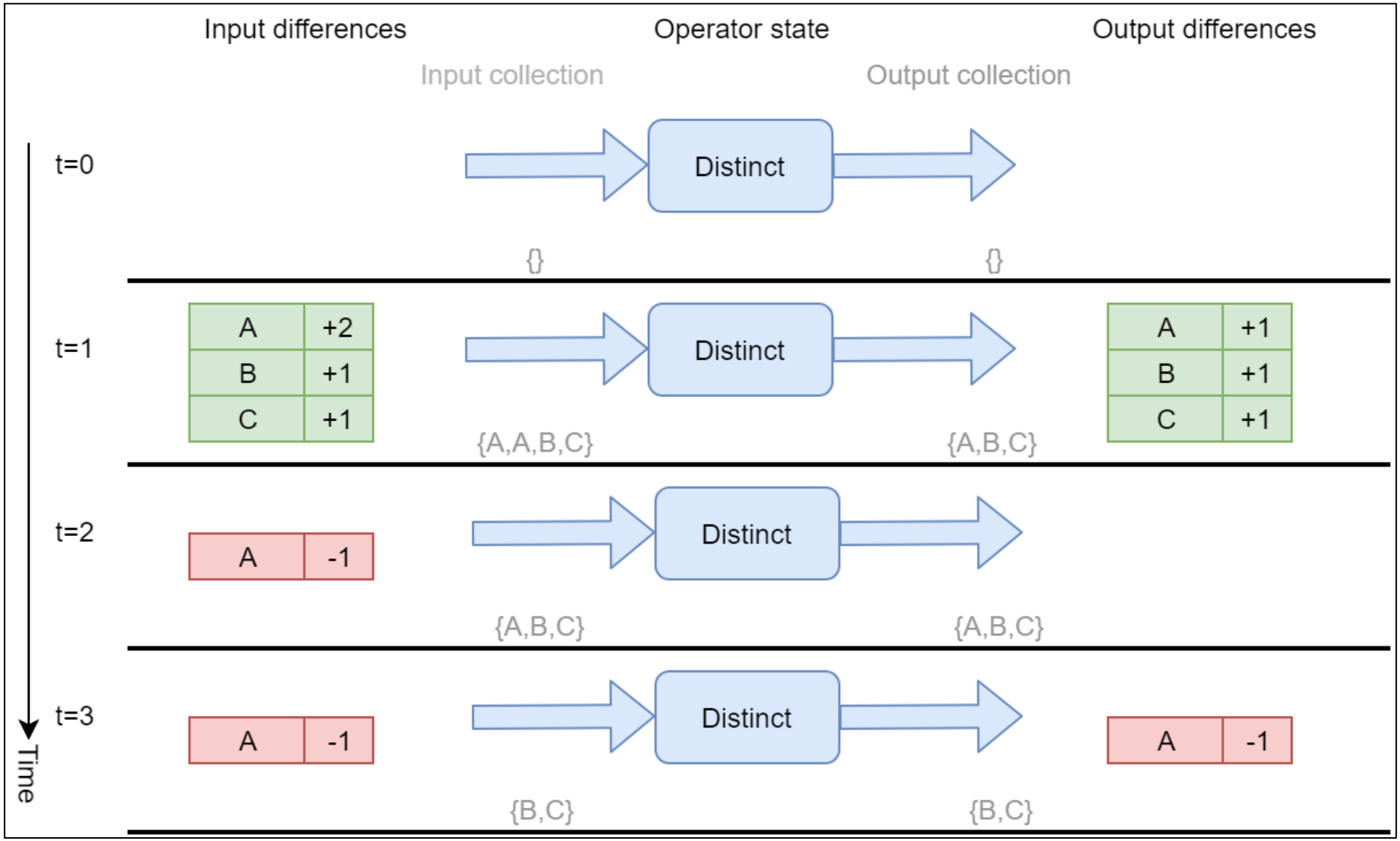

1
2
3
4
5
6
7
8
9
10
11
12
13
14
Execution time [s]
1000
500
200
100
50
20
10
5
2
1
0.5
0.2
0.1
0.05
0.02
0.01
0.005
0.002
0.001
1
3
10
Scale factor
PostgreSQL 10.6
Semantic DB 1
Semantic DB 2
Sparksee 5.2.3

INKREMENTÁLIS NÉZETKARBANTARTÁS

Adatfolyam alapú számítási modellek

- Időzített adatfolyam
  - Az adatfolyamban minden adatrekordhoz egy virtuális időbélyeget rendelünk
  - Nagy áteresztőképességű, elosztott számítási modell
  - Támogat iteratív számításokat is
- Differenciális adatfolyam
  - Időzített adafolyamon felett implementált
  - Inkrementális nézetkarbantartáshoz
  - Csak a változás továbbítódik az adatfolyamban
  - Hagyományos operátorok: **SELECT**, **JOIN**, **DISTINCT**, **COUNT**
  - **FIXEDPOINT**, **ENTERLOOP**

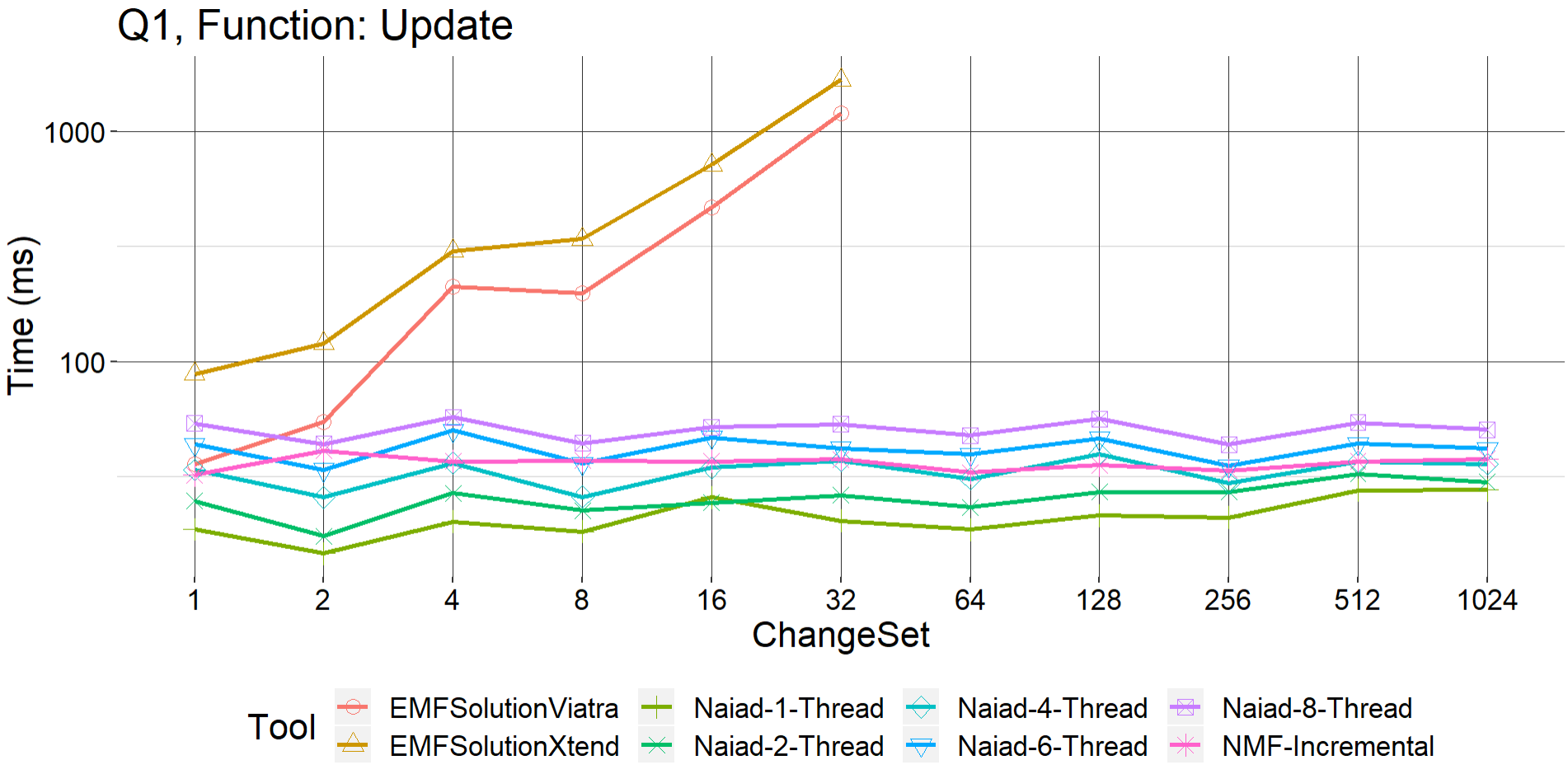
Distinct operátor működése



Transformation Tool Contest

- Modelltranszformációs eszközök összehasonlítása
  - kifejezőerő, használhatóság és teljesítmény
- 2018-as feladat az LDBC SNB egyszerűsített sémája feletti inkrementális nézetkarbantartás
- Kettő konkrét lekérdezés megoldása volt a feladat
- Több, már meglévő megoldás
  - Xtend
  - Viatra
  - .NET Modelling Framework (NMF)

TTC 2018 Q1 mérési eredmények



Összefoglalás

- Gráf-alapú adatmodellek és lekérdezőnyelvek
  - Szemantikus gráf, tulajdonsággráf, SPARQL , Cypher
- LDBC SNB bővítése
  - Business Intelligence: SPARQL implementáció (**32 417** karakter)
  - Interactive: SPARQL és Cypher implementáció (**62 569** karakter)
  - Szükséges szoftvermodulok
  - Teljesítménymérés (25 + 14 darab lekérdezés)
- Inkrementális nézetkarbantartás
  - Időzített és differenciális adatfolyamok megismerése
  - TTC 2018 megoldása differenciális adatfolyammal
  - Teljesítménymérés

Bírálói kérdések 1.

- A háttérismeretek fejezetben szerepel, hogy „[Az inkrementális nézetkarbantartás] két csoportját különböztetjük meg: az algebrai és a procedurális megközelítéseket”. A differenciális adatfolyamokat használó technikát hova sorolja, és miért?
- Válasz:
  - A kategorizálást alapvetően deklaratív lekérdezőnyelvekhez találták ki
  - Általánosítani lehet
  - **Procedurális**

Bírálói kérdések 2.

- A Cypher lekérdezés Gremlinre fordított változata szuboptimális és több technikai probléma is felmerült. Van-e valós gyakorlati igény a Cypher-Gremlin transzformációra?
- Válasz:
  - Microsoft CosmosDB támogatja a Gremlin lekérdezőnyelvet
    - 367 szavazat a Cypher támogatásra is [1]
  - Az motiváció a több gráfadatbázis teljesítménymérése a lekérdezések újabb implementációja nélkül („low hanging fruit”)

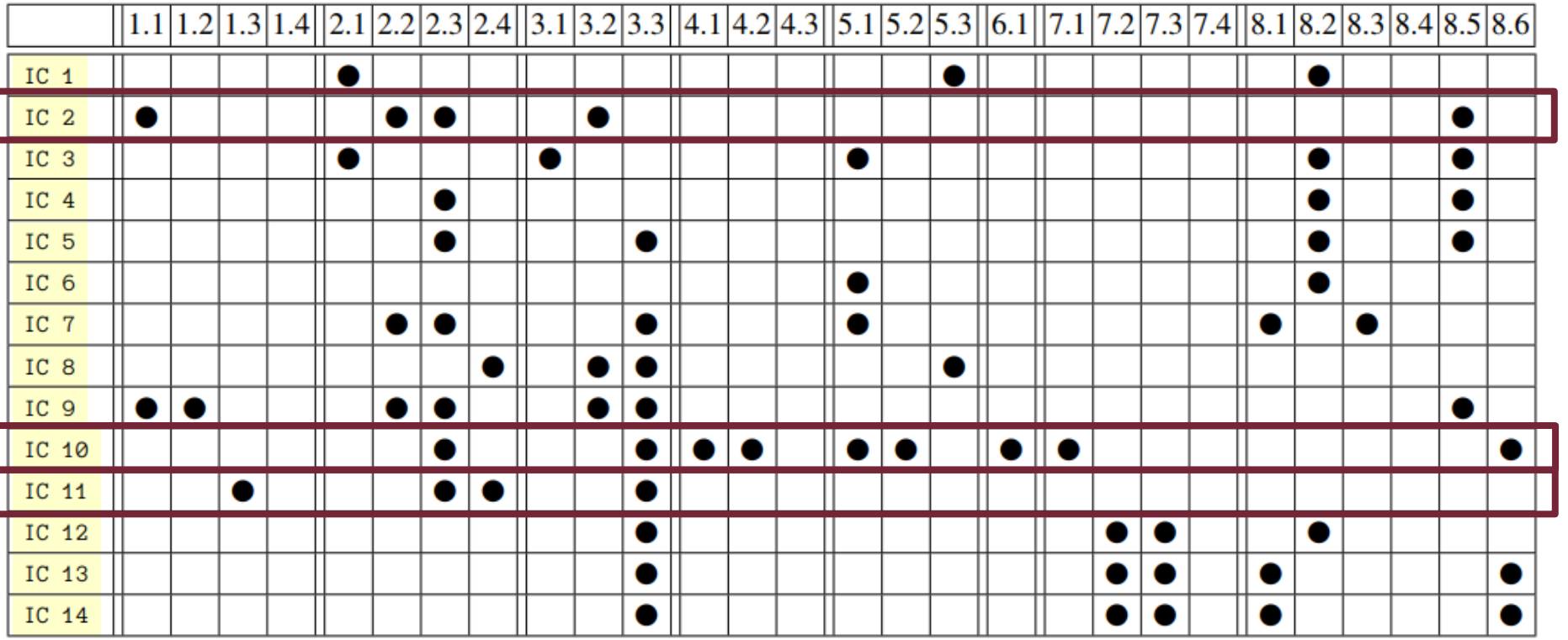
[1] https://feedback.azure.com/forums/263030-azure-cosmos-db/suggestions/19275547-support-cypher-as-a-query-language-for-graph-data

Bírálói kérdések 3./I

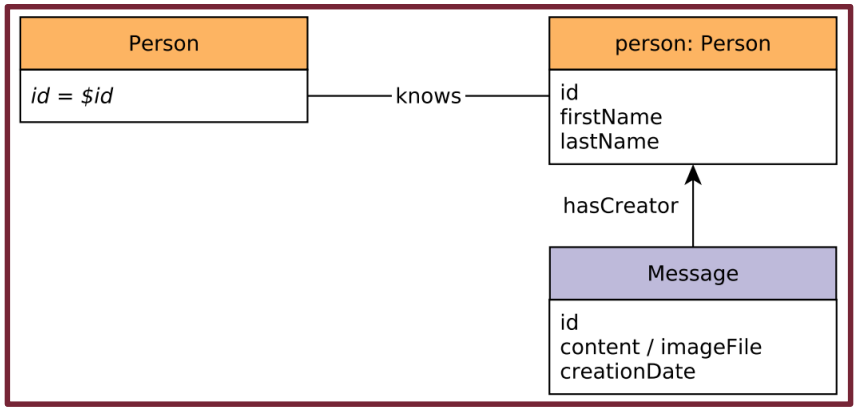
- A 2. és 10. (és esetleg még a 11.) lekérdezések jól látható teljesítménybeli sorrendet állítanak fel az egyes rendszerek között. Ki tud-e valami közös jellemzőt emelni ezen lekérdezések esetén, amik ezt a jelenséget okozhatják?

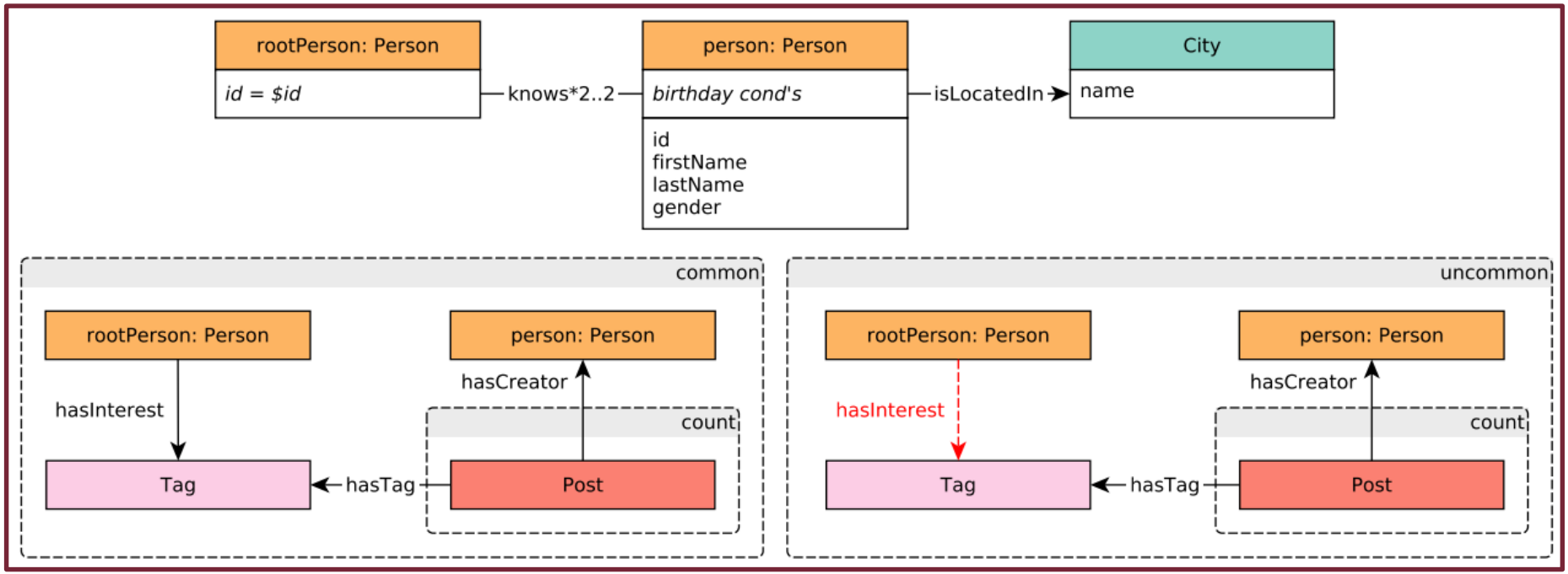
Bírálói kérdések 3./I

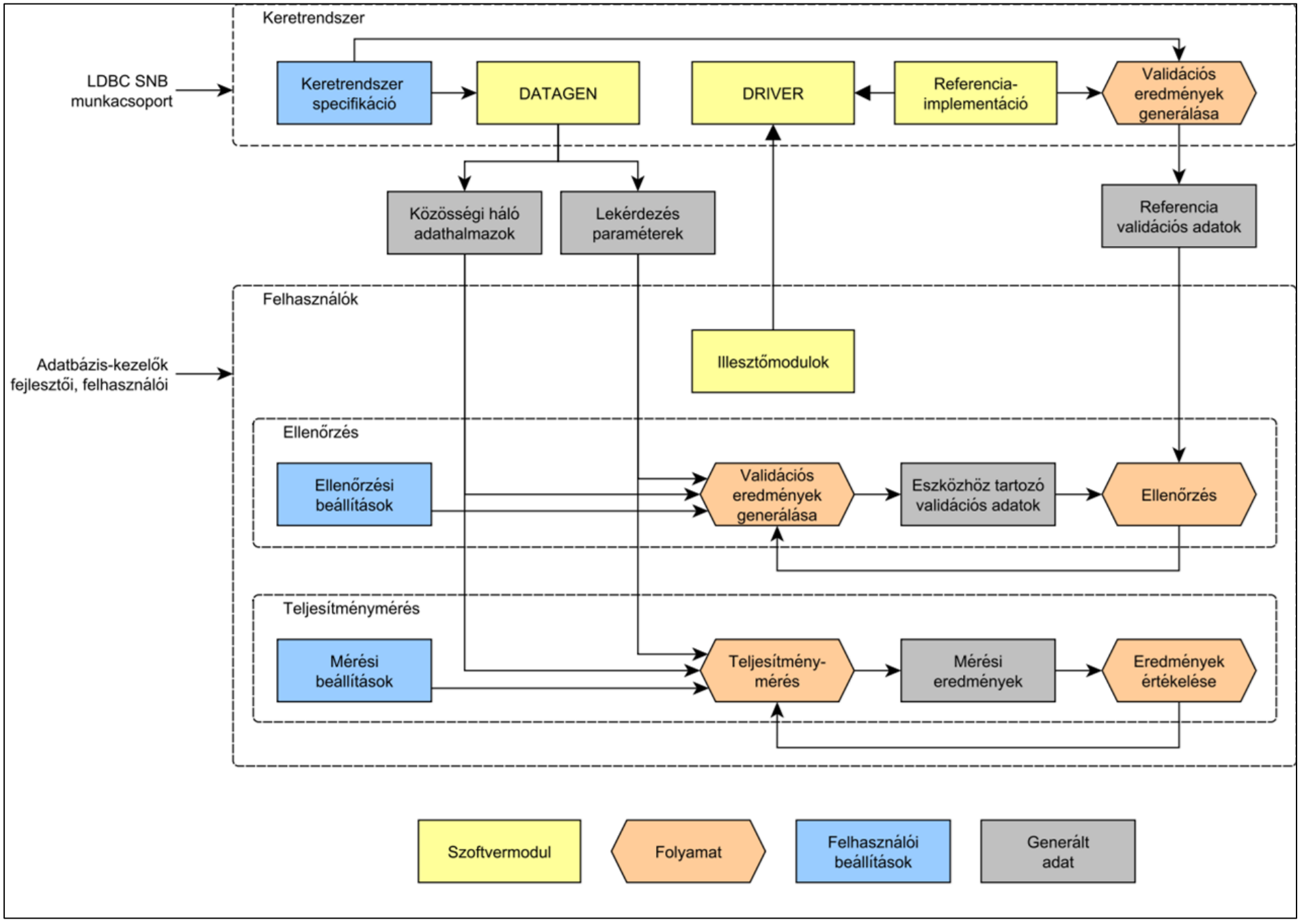
- A 2. és 10. (és esetleg még a 11.) lekérdezések jól látható teljesítménybeli sorrendet állítanak fel az egyes rendszerek között. Ki tud-e valami közös jellemzőt emelni ezen lekérdezések esetén, amik



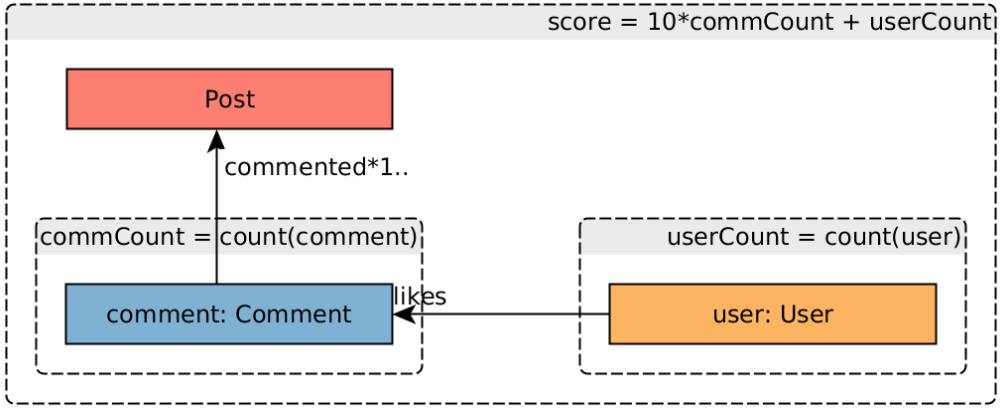
Bírálói kérdések 3./II

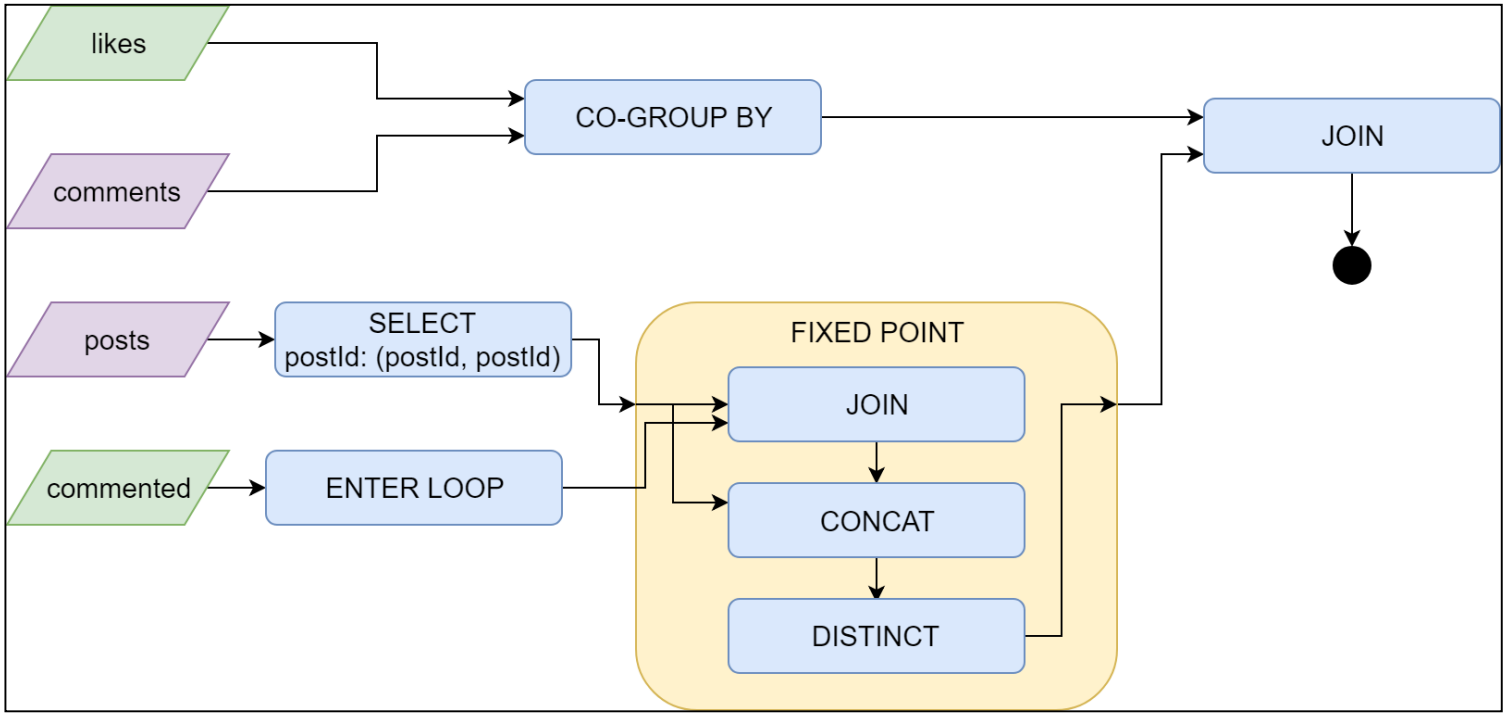





Keretrendszer
LDBC SNB munkacsoport
Keretrendszer specifikáció
DATAGEN
DRIVER
Referencia-implementáció
Validációs eredmények generálása
Közösségi háló adathalmazok
Lekérdezés paraméterek
Referencia validációs adatok
Felhasználók
Adatbázis-kezelők fejlesztői, felhasználói
Illesztőmodulok
Ellenőrzés
Ellenőrzési beállítások
Validációs eredmények generálása
Eszközhöz tartozó validációs adatok
Ellenőrzés
Teljesítménymérés
Mérési beállítások
Teljesítmény-mérés
Mérési eredmények
Eredmények értékelése
Szoftvermodul
Folyamat
Felhasználói beállítások
Generált adat

TTC 2018 Q1


score = 10*commCount + userCount
Post
commented*1..
commCount = count(comment)
userCount = count(user)
comment: Comment
likes
user: User


likes
comments
posts
commented
CO-GROUP BY
SELECT
postId: (postId, postId)
ENTER LOOP
FIXED POINT
JOIN
CONCAT
DISTINCT
JOIN