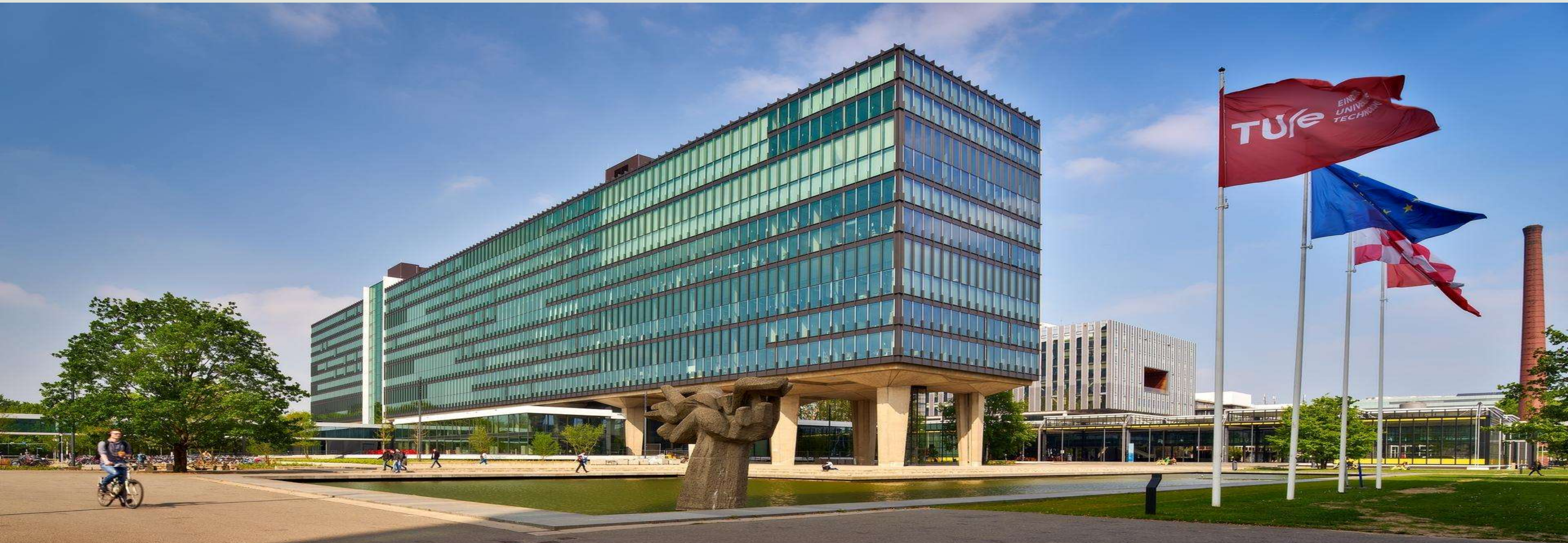
EINDHOVEN UNIVERSITY OF TECHNOLOGY



Linked Data Council's Business Intelligence and Interactive Benchmark from GSQL to GQL

Ryan Meghoe -Web Science

TigerGraph

2017

AIM : Fast DBM for graphs

Language : GSQL

SCOPE

Bi workload (20)

Interactive workload (26)

GQL parser: pattern match evaluation test

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

LDBC

Business Intelligence

- OLAP
- multi-hop/path/subgraph queries
- inserts

Interactive Workload

- OLTP
- 2 hop/3hop queries
- inserts and deletes

Property Graphs & GQL

- Property Graphs terminology
- GQL

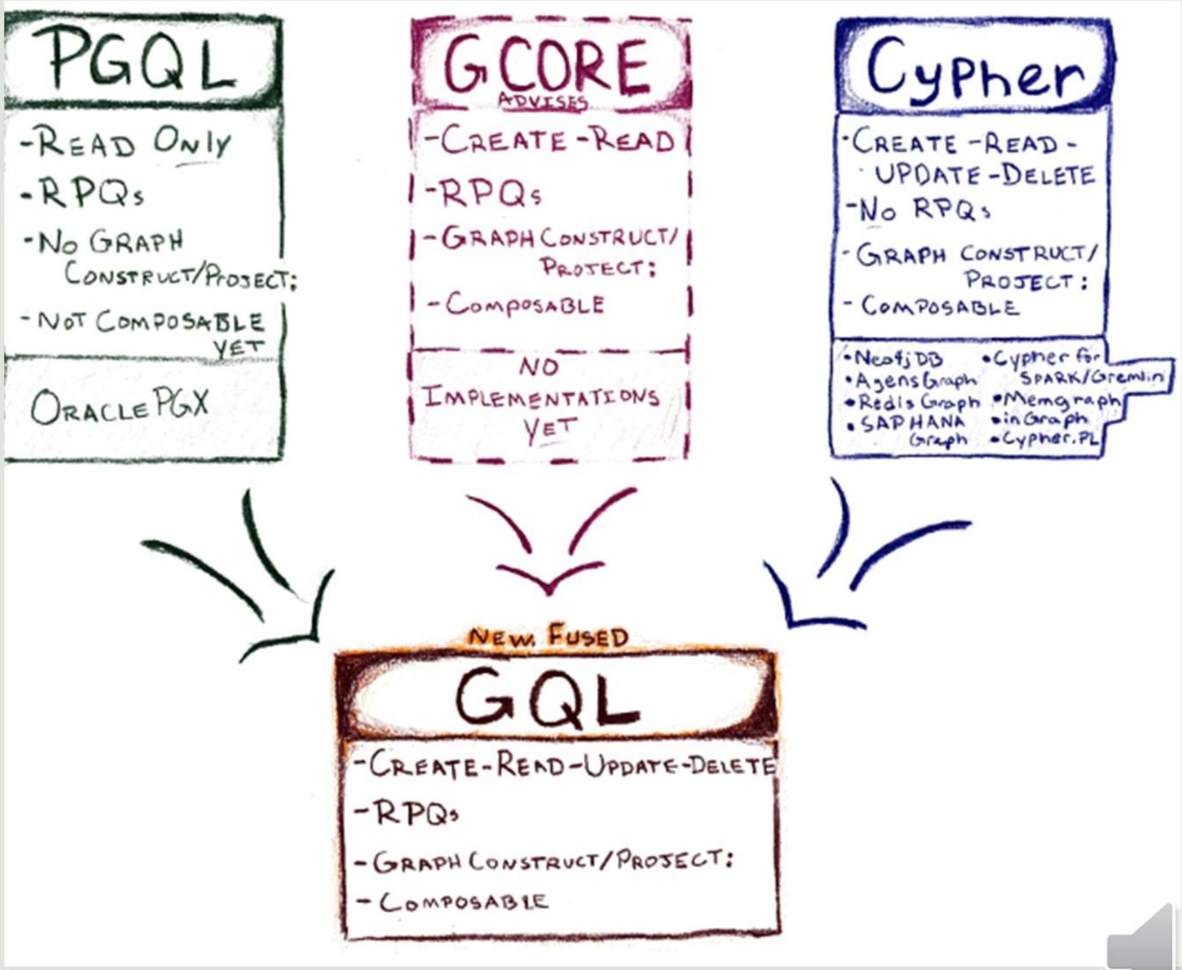
GQL

Standard

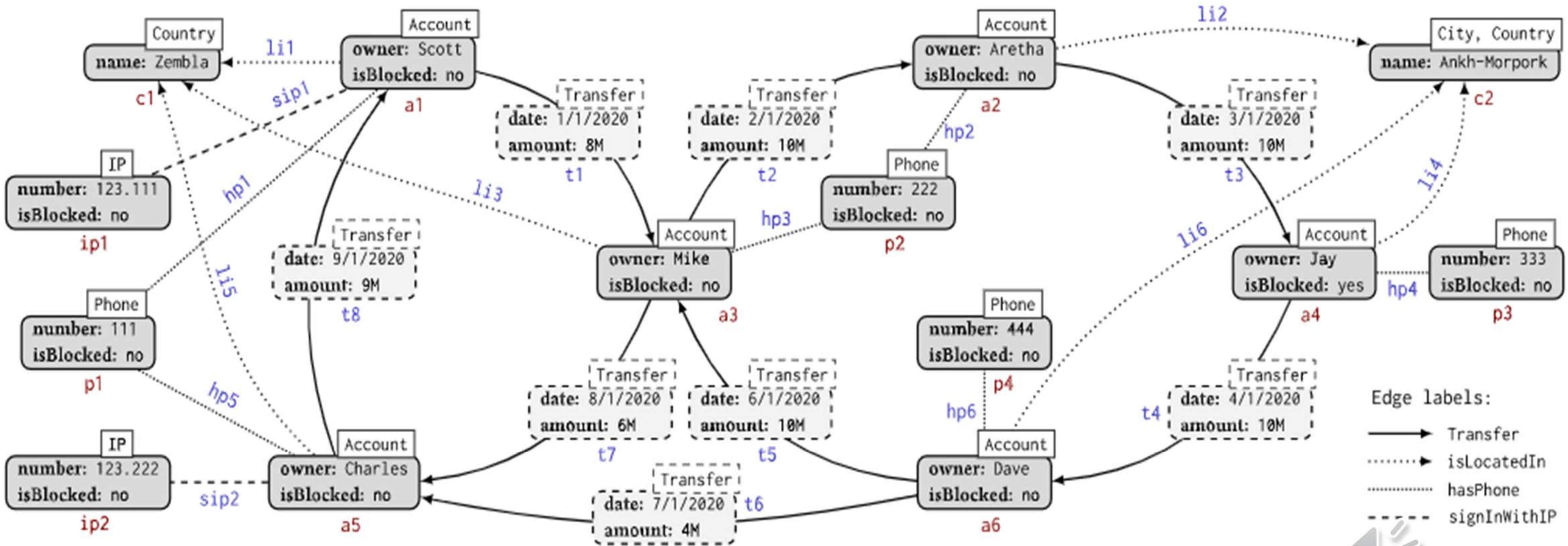
Property graphs

RPQ

ISO Standard :Final Delivery -> 2024



Property Graphs



Property Graphs

$$G = (N, E, rho, \lambda, \sigma)$$

1. N , E
2. ρ : E ->$(N \times N)$
3 . **λ** :$(N \cup E)$ SET+(L)
4. σ : $(N \cup E) \times P$ SET+(V)

Node Pattern

χ and is a triple of (a,L,P) :

- $a \in A \cup \{nil\}$ is an optional name.
- $L \subset \emptyset \cup \{L1,...,Ln\}$, which is finite
- P can be nil or (m,n) where $m,n \in N \cup \{nil\}$

Relationship Pattern

$\rho = (d,a,L,P,I)$

direction: d ∈ {→,→,−}

- name: a ∈ A ∪ {nil}
- empty finite set : L ⊂ ∅ ∪ {L1,...,Ln}
- P can be an empty finite set of key-value pairs, in the form (k,v);

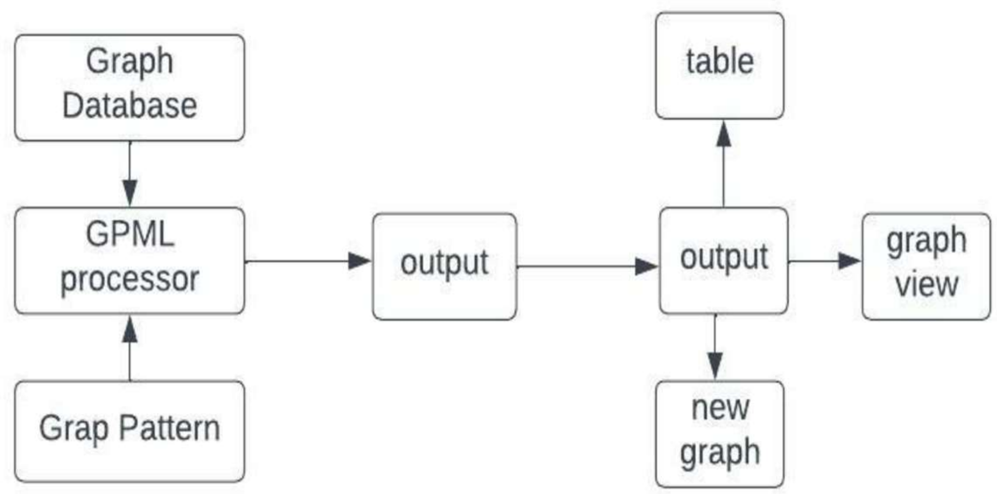
k,∈ κ, v,∈, ν

- I can be nil or (m,n) where m,n ∈ N ∪ {nil}

Data-Model

GQL values

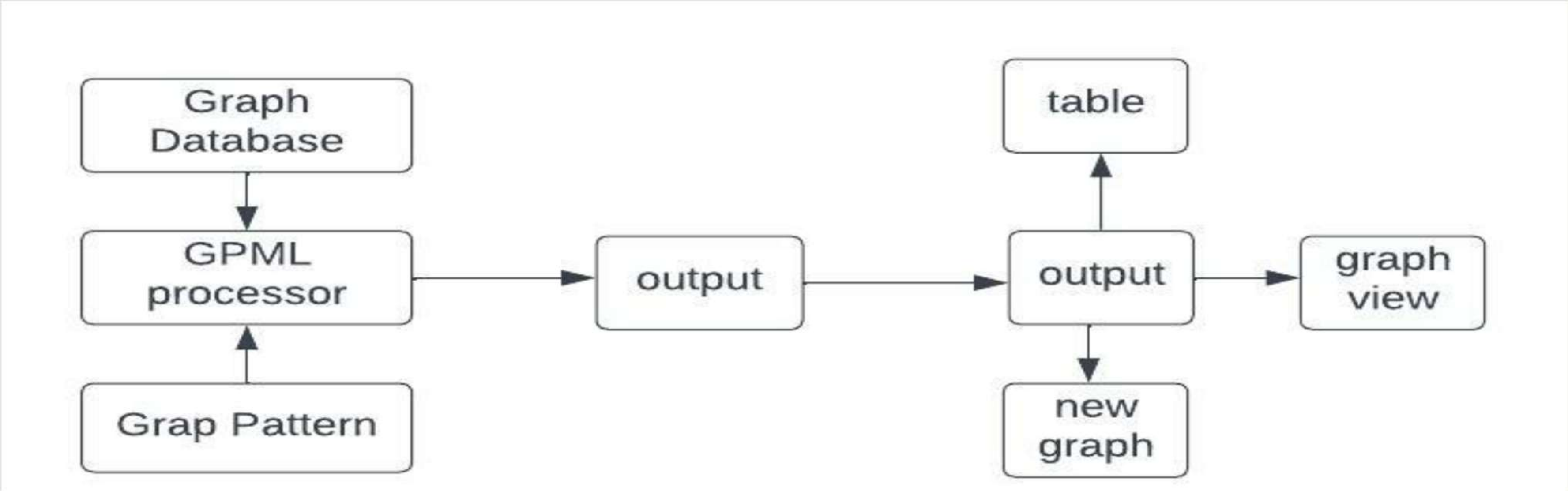
- Base types
- Node and Edge indentifiers
- (Multiset)Set(Vn...Vm)
- Map()
- Paths



Pattern Matching

- Match Pattern Relationship
- Path Pattern Union & Multiset Alternation
- Restrictors & Selectors

Pattern Matching for GPML



Relationship Pattern

MATCH *pattern*

<-[specification]- left directed

<~[specification]~ left undirected

-[specification]-> right directed

~[specification]~> right undirected

~-[specification]~ undirected

Path patterns
UNION and Multiset Alternation

MATCH *pattern_1* | pattern 2

MATCH *pattern_1* |+| pattern 2

Restrictors

TRAIL ACYCLIC SIMPLE

Selectors

ANY vs ANY K

ANY SHORTEST vs ALL SHORTEST vs SHORTEST K vs SHORTEST K GROUP

Quantifiers

{m,n} between m and n repetitions

{m,}

*

+

Ensuring path Termination:

MATCH Type-selector Type-restrictor
......remaining query..

- GQL Syntax
- Demo

GQL features and Syntax

$$\textbf{query} ::= \text{query expr (query conjunction query expr)*}$$

$$\textbf{query conjunction} ::= \text{set operator} \mid \text{OTHERWISE}$$

$$\text{query expr} ::= \text{focused query expr} \mid \text{ambient query expr}$$

$$\textbf{focused query expr} ::= \text{(FROM a match clause+)+ return statement a} \in A$$

$$ambientqueryexpr ::= matchclause + returnstatement$$

$$\textbf{return statement} ::= RETURN setquantifier?(*|returnlist)$$

$$\textbf{set operator} ::= unionoperator|othersetoperator$$

$$\textbf{union operator} ::= UNION(setquantifier|MAX)?$$

$$\textbf{other set operator} ::= (EXCEPT|INTERSECT)setquantifier?$$

$$\textbf{set quantifier} ::= DISTINCT|ALL$$

$$\textbf{return list} ::= returnitem(,returnitem)*$$

GSQL	GQL
CREATE OR REPLACE DISTRIBUTED QUERY bi7(STRING tag) SYNTAX v2 {	-
TYPEDEF TUPLE <STRING relatedTagName, UINT replyCount> RESULT;	-
HeapAccum<RESULT>(100, replyCount DESC, relatedTagName ASC) @@result;	-GROUP BY ORDER BY
SumAccum<UINT> @count;	- COUNT

GSQL breakdown to GSQL

```
 tagWithName = SELECT t FROM Tag:t
WHERE t.name == tag;
```

Pattern match 1

```
 replies = SELECT c FROM tagWithName -
(<HAS_TAG.<REPLY_OF)- Comment:c;
```

Pattern match 2

```
repliesWithTag =
   SELECT r
   FROM tagWithName -(<HAS_TAG)- replies:r;
```

Pattern match 3

```
  PRINT @@result as result;
}
```

-

GSQL	GQL
repliesWithoutTag = replies MINUS repliesWithTag;	- EXCEPT ALL
tmp = SELECT t FROM repliesWithoutTag:r -(HAS_TAG>)- Tag:t ACCUM t.@count += 1 POST-ACCUM @@result += RESULT(t.name, t.@count);	- - Return

Final GQL QUERY

```
FROM BI
MATCH  (tag:Tag {name:$tagname} )<-[:HAS_TAG]-(m:Message),
          (m)<-[:REPLY_OF]-(c:Comment)-[:HAS_TAG]->(rTag:Tag)
RETURN rTag.name AS  rTag_name,
     count(*) AS count

EXCEPT ALL

FROM BI
MATCH  (c1:Comment)~[:HAS_TAG]~(tag1:Tag)
Return rTag_name
```

- DEMO
- Limitation(s)
- Future Research

Limitations

VIEWS -> Tables

Graph-> Binding Table -> Subgraph projection -> View-Table

Functions , Predicates

Brief Demo

Future Work

- TLP
- Views and Subgraphs
- Extension of the GQL parser to support those
- Path termination

Thank You

Question : r.a.meghoe@student.tue.nl

Bibliography

[1]:Alin Deutsch, Yu Xu, Mingxi Wu, and Victor E. Lee. Aggregation support for modern graph analytics in tigergraph. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, SIGMOD ’20, page 377392, New York, NY, USA, 2020. Association for Computing Machinery

[2]:N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, M. Schuster, P. Selmer, and others. Formal semantics of the language cypher. *arXiv preprint arXiv:1802.09984*, 2018.

[3]:Alastair Green, Paolo Guagliardo, and Leonid Libkin. Property graphs and paths in gql: Mathematical definitions. Technical Reports TR-2021-01, Linked Data Benchmark Council (LDBC), Oct 2021