Graph queries, schemas and standards

Birkbeck Knowledge Lab, 10 March 2021

Alastair Green

Part-time Ph.D. student at Birkbeck [updateable property graph views], Alex/Peter
Vice-chair of Linked Data Benchmark Council
Author of The GQL Manifesto (Lead at Neo4j for query language/research group 2017-9)

A new International Standard, GQL (Graph Query Language)

September 2019 ISO/IEC Joint Technical Committee 1 [IT standards] agrees to start a new project for the SC 32 [Data Management] Working Group 3 [Database Languages]

WG3 is the SQL standard committee.

The new project is GQL.

A query and schema language for mutable **property graph** databases.

The read-only pattern matching part of GQL is also part of **SQL/PGQ**.

WG3 has not worked on any language other than SQL in the past 35 years

This could be a big deal. I think it's the tip of an iceberg

What is a property graph?

A graph where the nodes and edges are data records.

Sometimes, a field can be a tags, with no value (a “label”)

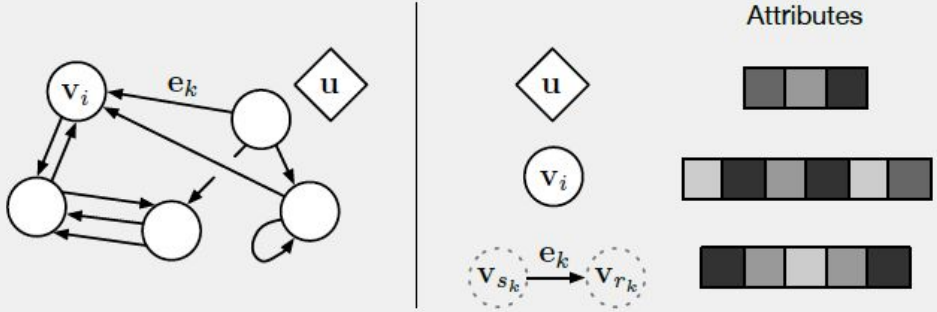
Sometimes, the graph can have its own attributes

Usually, the field values are not graphs or graph elements

Relational inductive biases, deep learning, and graph networks

Peter W. Battaglia[1]*, Jessica B. Hamrick[1], Victor Bapst[1],
Alvaro Sanchez-Gonzalez[1], Vinicius Zambaldi[1], Mateusz Malinowski[1],

Box 3: Our definition of “graph”



Here we use “graph” to mean a directed, attributed multi-graph with a global attribute. In our terminology, a node is denoted as $\mathbf{v}_i$, an edge as $\mathbf{e}_k$, and the global attributes as $\mathbf{u}$. We also use s_k and r_k to indicate the indices of the sender and receiver nodes (see below), respectively, for edge k. To be more precise, we define these terms as:

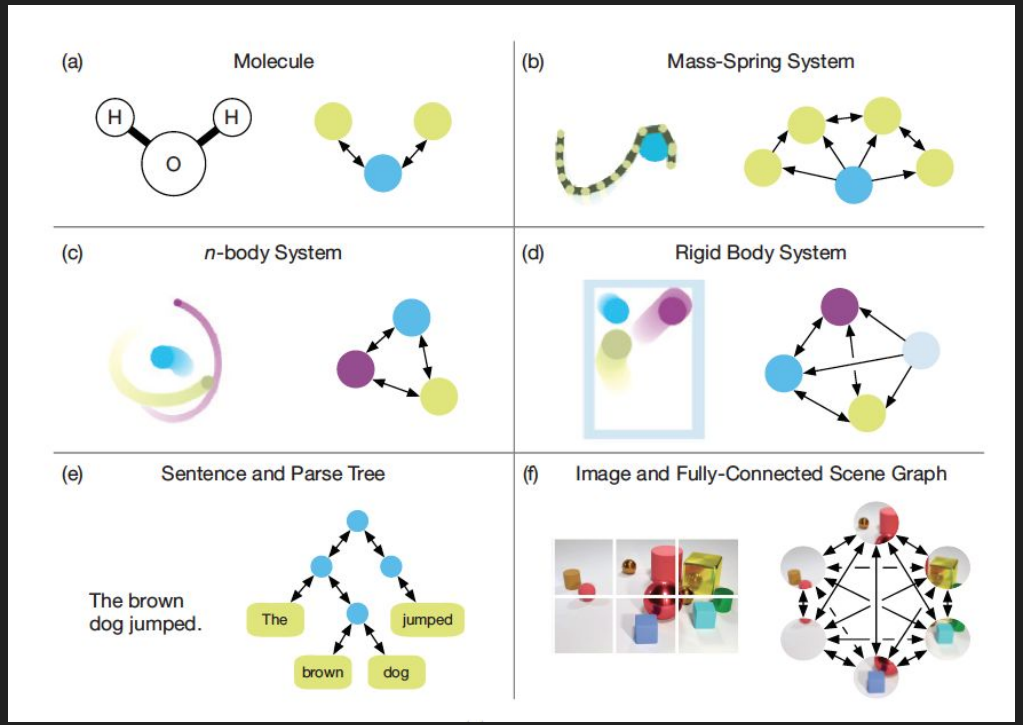
Directed : one-way edges, from a “sender” node to a “receiver” node.
Attribute : properties that can be encoded as a vector, set, or even another graph.
Attributed : edges and vertices have attributes associated with them.
Global attribute : a graph-level attribute.
Multi-graph : there can be more than one edge between vertices, including self-edges.

Is graph data important?

Late 2020 DeepMind's Alphafold: a solution to a 50-year-old grand challenge in biology

> *'A folded protein can be thought of as a "spatial graph".... [our] neural network system ... interpret[s] the structure of this graph, while reasoning over the implicit graph that it's building'.*

Late 2018, in the Battaglia et al. paper, Deep Mind and other scientists advocated graph networks as a paradigm of deep learning, using the property graph data model.



Are graph databases important?

The 2018 Seattle Report on database research directions does not mention graphs.

~50 papers and multiple sessions at VLDB 2020 focussed on graphs.

February 2021 a small graph database/analytics company called TigerGraph received third-round venture funding of $105m.

Databricks, a large SQL data/analytics company, received $1bn the same month

There are numerous other fundings and acquisitions (Bitnine in Korea first graph IPO)

Oracle are a big force behind SQL/PGQ

Ant Group has multiple graph databases

Amazon Neptune is the leading graph cloud service

WG3 and LDBC liaison

Birkbeck has just joined Linked Data Benchmark Council

Jan Hidders in the CS department co-leads an LDBC community **working group on PGS**

CWI/VU in Amsterdam, ENS-Paris, Edinburgh, Lyon, INRIA Lille, Warsaw, UPC ... lots of academic partners

Another LDBC project (Leonid Libkin ENS/Edinburgh) in is a GQL Formal Semantics working group

Cypher to GQL is a likely direction for future community efforts

Petra Selmer (Birkbeck alumna and visiting lecturer) at Neo4j heads a working group on Existing Languages analysis

Queries

Graph queries using the pattern-matching paradigm of Cypher (originally from Neo4j) are not schema-aware: they search for topological patterns (sub-graphs) and data structures and values.

Cypher 9

```
FROM languageGraph
MATCH (a:Engineer)-[:LIKES]->(l:Language)
WHERE (l)-[:SUPPORTS]->(:Feature {name: "Pattern Matching"})
RETURN a.name, l.name
```

PGQL 1.1

```
SELECT a.name, l.name FROM languageGraph
MATCH (a:Engineer)-[:LIKES]->(l:Language)
WHERE EXISTS (
  SELECT *
  MATCH (l)-[:SUPPORTS]->(:Feature {name: "Pattern Matching"})
)
```

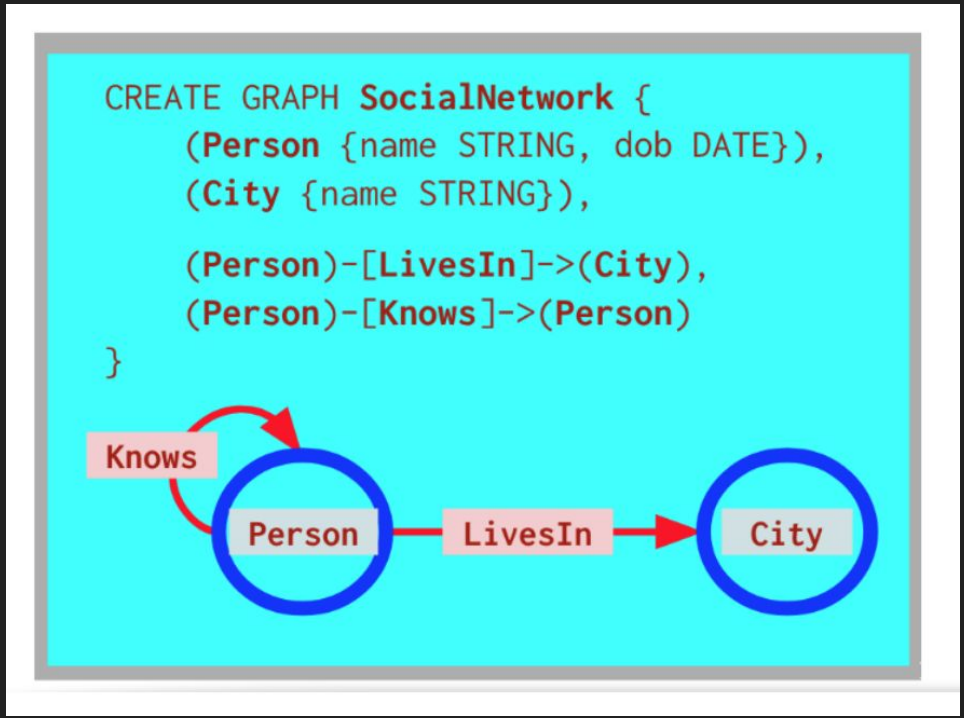
Queries and Schemas

SQL/PGQ is about adding (conjunctive) regular path queries to sophisticate this approach

You don't have to have a schema to have a graph database (unlike SQL)

But there is great interest in property graph schema: one of the motivations for GQL

Water to Ice



WG3 and LDBC liaison

Birkbeck has just joined Linked Data Benchmark Council

Jan Hidders in the CS department co-leads an LDBC community **working group on PGS**

CWI/VU in Amsterdam, ENS-Paris, Edinburgh, Lyon, INRIA Lille, Warsaw, UPC ... lots of academic partners

Another LDBC project (Leonid Libkin ENS/Edinburgh) in is a **GQL Formal Semantics** working group

Cypher to GQL is a likely direction for future community efforts

Petra Selmer (Birkbeck alumna and visiting lecturer) at Neo4j heads a working group on **Existing Languages** analysis

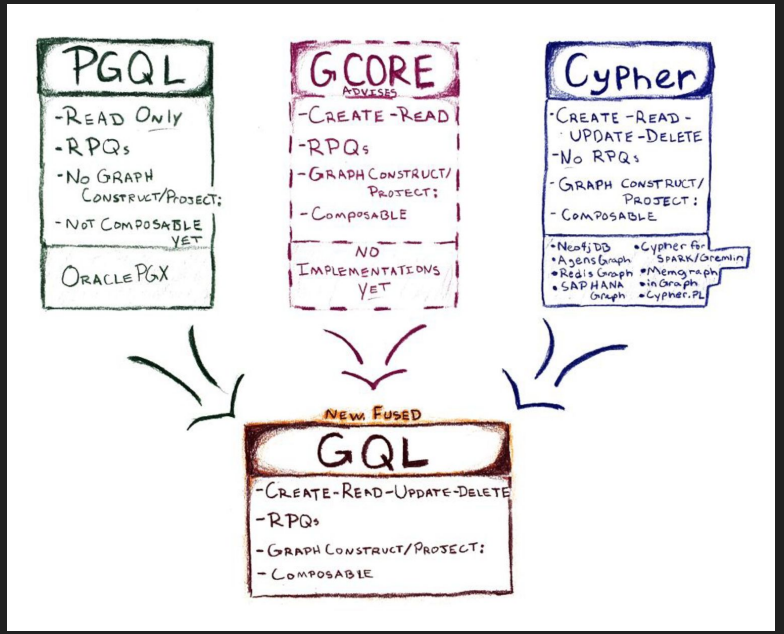
Input languages and output timescales

[Working towards a New Work Item for GQL, to complement SQL PGQ]

Cypher, PGQL, G-CORE, GSQL, SQL-PGQ, Cypher for Apache Spark, GXPath

US, UK, China, Sweden, NL, S. Korea, Finland, Denmark, Japan

SQL-PGQ CD Q4 2021
GQL CD Q3 2022



What's (not) coming when?

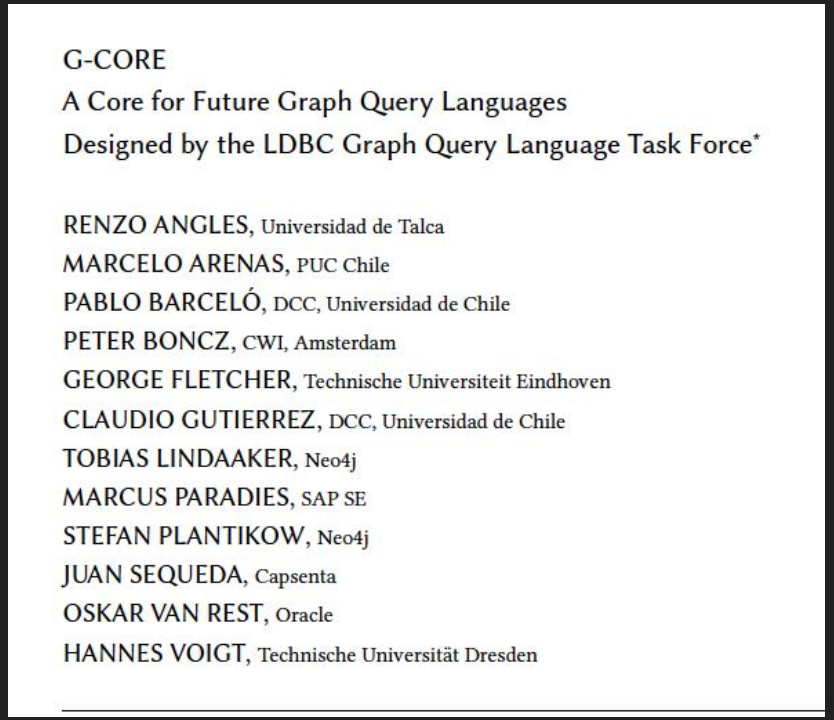
In 2022

DQL returns table from **path pattern match**
DML to **mutate graph** (including merge)
DDL to create **graph types and typed graphs**

Not yet (not 2022)

G-CORE
Graph query language closed over graphs

Extensible schema, keys, participation constraints



G-CORE
A Core for Future Graph Query Languages
Designed by the LDBC Graph Query Language Task Force*

RENZO ANGLES, Universidad de Talca
MARCELO ARENAS, PUC Chile
PABLO BARCELÓ, DCC, Universidad de Chile
PETER BONCZ, CWI, Amsterdam
GEORGE FLETCHER, Technische Universiteit Eindhoven
CLAUDIO GUTIERREZ, DCC, Universidad de Chile
TOBIAS LINDAAKER, Neo4j
MARCUS PARADIES, SAP SE
STEFAN PLANTIKOW, Neo4j
JUAN SEQUEDA, Capsenta
OSKAR VAN REST, Oracle
HANNES VOIGT, Technische Universität Dresden