Database Technology Evolution III: Knowledge Graphs and Linked Data

MALCOLM CROWE, FRITZ LAUX

IARIA CONGRESS 2024

Malcolm Crowe

University of the West of Scotland
Email: malcolm.crowe@uws.ac.uk

- Malcolm Crowe is an Emeritus Professor at the University of the West of Scotland, where he worked from 1972 (when it was Paisley College of Technology) until 2018.
- He gained a D.Phil. in Mathematics at the University of Oxford in 1979.
- He was appointed head of the Department of Computing in 1985. His funded research projects before 2001 were on Programming Languages and Cooperative Work.
- Since 2001 he has worked steadily on PyrrhoDBMS to explore optimistic technologies for relational databases and this work led to involvement in DBTech, and a series of papers and other contributions at IARIA conferences with Fritz Laux, Martti Laiho, and others.
- Prof. Crowe has recently been appointed an IARIA Fellow.

Prof. Dr. Fritz Laux

(Retired), Reutlingen University
Email: fritz.laux@reutlingen-university.de

- Prof. Dr. Fritz Laux was professor (now emeritus) for Database and Information Systems at Reutlingen University from 1986 - 2015. He holds an MSc (Diplom) and PhD (Dr. rer. nat.) in Mathematics.
- His current research interests include
  - Information modeling and data integration
  - Transaction management and optimistic concurrency control
  - Business intelligence and knowledge discovery
- He contributed papers to DBKDA and PATTERNS conferences that received DBKDA 2009 and DBKDA 2010 Best Paper Awards. He is a panellist, keynote speaker, and member of the DBKDA advisory board.
- Prof. Laux is a founding member of DBTech.net (http://www.dbtechnet.org/), an initiative of European universities and IT-companies to set up a transnational collaboration scheme for Database teaching. Together with colleagues from 5 European countries he has conducted projects supported by the European Union on state-of-the-art database teaching.
- He is a member of the ACM and the German Computer Society (Gesellschaft für Informatik).

Linked Data and Databases

- Everyone's data today is stored online
- Using vast databases: Google, Meta etc
  - The "Semantic Web" [1]
- Your PC also has its own/your local data
- Data can be accessed using database queries
- Results can be text, tables, images etc
- And links to more data, other databases
- All of this gives us today's world wide web of information systems and knowledge [2]

Knowledge and Links

- Information links to more information
- “China’s lunar probe has landed”
- Web pages such as news feeds, Wikipedia have clickable links
- For navigating through the knowledge web
- Or you can look up things using search
- Eventually it involves database technology

Database technology

- Supports durable storage of data
- Data can be shared subject to security
- Data is organised for efficient retrieval
- In tables and indexes (since 1970s)
- Or knowledge graphs [3]
  - Nodes are knowledge items such as words or objects
  - Edges link nodes (e.g., is, has,..)
- Or both? [4]

Database Management System

- There are thousands of DBMS today
- All shapes and sizes: company accounts, catalogues, health data, address books, shopping lists
- Few people buy a DBMS directly
  - Usually part of a larger service
- Many information workers simply use the systems
  - Applications access DBMS behind the scenes
- But someone has chosen the DBMS

Choosing a DBMS

- Data organised in tables (relations)
  - Start with creating tables
    - Homogeneous data structures
    - Oracle: declare everything before use
- Data coming from complex structures
  - Start with documents, indexed somehow
    - MongoDB: no schema
- Knowledge graphs and linked data
  - Focus on links between knowledge items
    - Neo4J: easy to get started

Industry standards

- Good practice: what people can expect
- Interoperability: enables links between data systems
- Reliability: keep things simple
- International Standards Organization
  - Internet Society, W3C, etc
- ISO9075: Database Language SQL
- ISO39075: Database Language GQL

Efficiency and structure

- Trade-offs
  - Structure vs content
    - Greater precision assists indexing
    - Many ways of indexing..
  - Precision vs clarity
    - Sharing across countries, cultures
    - Metadata to assist sharing
  - Accessibility vs security
  - Transactions vs inconsistency
- How up-to-date? Archiving old stuff?

Research areas

- Keen to find the best way of ..
  - Designing usable systems
  - Keeping ideas as clear as possible
  - Covering many use cases
- New application areas, use cases
  - Can we find more useful models
- This talk is part of a series looking at
  - Underlying principles
  - Practical implementation
  - Evolution of new standard: GQL

- Is SQL + GQL possible?

IARIA

Relational vs Graph databases

- Seem different in concept and scale
- Linked data mean joins of relations
  - Graph DBMS are better at following links
- Chains of links important in business
  - Transfers, logistics, components, supply chain, timelines, epidemics
- Last year we reported on extending SQL to handle graphs, we continue this today
- But GQL is new and still under discussion
  - No implementations yet of the GQL standard
  - Changes to GQL are already under discussion

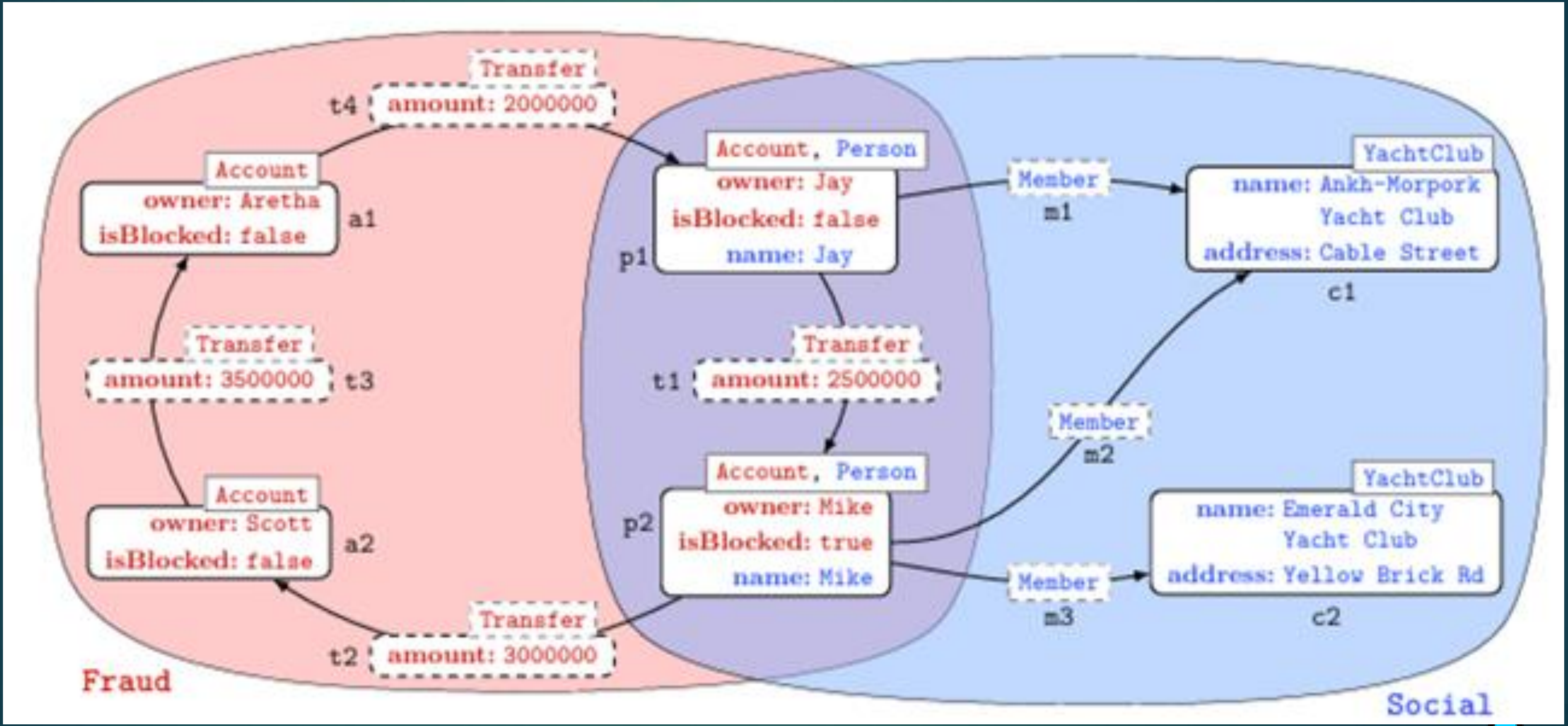
SQL+GQL can work

- ISO39075 leaves this possibility open
  - Syntax is designed to be compatible
  - Though reserved words etc are different
- Last year we showed graph creation and graph queries in PyrrhoDBMS [5]
  - Files are append-only transaction logs
  - Objects defining positions do not change
    - Position can be used for indexing, replaces keys
  - Optimistic concurrency control validation
  - Mixing of schema and data changes
  - Full range of SQL features: triggers, types etc

- The Yacht Club example

The Yacht Club example [6]

- Motivated by combination of 2 graphs



Building this example

- Can be done in just a few steps
  - 3 statements for schema and graph type
  - 1 statement to build all the nodes and edges
- But it is not completely GQL compliant
- GQL has “closed” and “open” graphs ☹
  - Closed = predefined, unchangeable
  - Open = typeless, like Mongo docs
- GQL nodes can have a set of labels
  - But cannot be in more than one graph
  - (We could put everything in one graph..)

- The graph types

Graph types in GQL

```
create schema /yc;

create graph type /yc/Social {node Person {name string},
    node YachtClub {name string,address string},
    directed edge "Member" connecting (Person->YachtClub)};
create graph /yc/Fraud ANY;
```

- This shows a closed graph for Social and an open graph for Fraud
- This allows us to add undeclared types Account, Transfer

- The insert statement

Inserting the yacht club nodes

```
insert (a2 :Account{owner:'Scott',isBlocked:false})
-[:Transfer{amount:350000}]->
(:Account{owner:'Aretha',isBlocked:false})
-[:Transfer{amount:2000000}]->
(p1 :Person&Account{owner:'Jay',name:'Jay',isBlocked:false})
-[:"Member"]->(:YachtClub {name:'Ankh-Morpork Yacht
Club',address: 'Cable Street'})
<-[:"Member"]-(p2 :Person&Account {owner:'Mike',name:'Mike',
isBlocked:true})
-[:"Member"]->(:YachtClub{name:'Emerald City Yacht
Club',address:'Yellow Brick Road'}),
(p1)-[:Transfer{amount:2500000}]->
(p2)-[:Transfer{amount:3000000}]->(a2);
```

A tiny knowledge graph [7]

t1 = (:John :masterFrom :DauphineUni),

t2 = (:John :phdFrom :DauphineUni),

t3 = (:masterFrom $<_{sp}$:degreeFrom),

t4 = (:phdFrom $<_{sp}$:degreeFrom)

- This paper is about “implies” relationships between edge types and saturation
- If GQL had INSERT SCHEMA this could be

```
INSERT (:John)-[:masterFrom]->(:DauphineUni);

INSERT (:John)-[:phdFrom]->(:DauphineUni);
INSERT SCHEMA [:masterFrom=>:DegreeFrom];
INSERT SCHEMA [:phdFrom=>:DegreeFrom];
```

GQL and linked data

- Combining graphs from different sources
  - Big use case, as in Social+Fraud above
  - We looked at view-mediated big data [8]
    - Query remote system, don’t import data
  - No concept yet of views in GQL
    - Viewed graph? Virtual links?
- Directory paths in CREATE statements do not extend to network links/HTTP
  - But they look promising
- More work is needed on sharing of data between graphs

Current state and conclusion

- Pyrrho DBMS demonstrates that GQL features can be added to an SQL implementation
- Lightweight and efficient for links
- Can it become GQL-compliant?
- GQL is designed for several business cases
- How will it be extended for knowledge graphs and linked data?

References

1. T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” Scientific American, pp. 34-43, May 2001.
2. M. Bennett, and K. Baclawski, “The role of ontologies in linked data, big data and semantic web applications”, Applied Ontology 12.3-4, pp. 189-194, 2017.
3. J. F. Sowa, “Conceptual graphs as a universal knowledge representation”, Computers & Mathematics with Applications 23.2-5, pp. 75-93, 1992.
4. H. Kaindl, S. Kramer, and L. M. Afonso, “Combining structure search and content search for the World-Wide Web”, Proceedings of the Ninth ACM conference on Hypertext and hypermedia: links, objects, time and space, 1998.
5. M. K. Crowe, The Pyrrho Database Management System. In PyrrhoV7alpha folder of https://github.com/MalcolmCrowe/ShareableDataStructures
6. N. Francis et al., “A Researcher’s Digest of GQL”, The 26th International Conference on Database Theory, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.
7. K. Belhajjame and M.-Y. Mejri, “Online maintenance of evolving knowledge graphs with RDFS-based saturation and why-provenance support”, Web Semantics: Science, Services and Agents on the World Wide Web 78 (2023) 100796
8. M. Crowe, C. Begg, F. Laux, and M. Laiho, "Data validation for big live data”, DBKDA 2017, The Ninth International Conference on Advances in Database, Knowledge, and Data Applications, ISSN 2308-4332, pp 30-36, 2017