

ANSI INCITS DM32.2 2018-00128r1
Discussion Paper

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

Alastair Green
Neo4j Query Languages Standards and Research Team
13 July 2018, r1 14 July 2018

This paper was prepared for Meeting 255 of DM32.2.

It explains and motivates

[\[DM32.2 2018-00127\]](#) **A proposal to the database industry. Not three but one: GQL**

[\[DM32.2 2018-00126\]](#) **The GQL Manifesto**

[\[DM32.2 2018-00087\]](#) **GQL Project Proposal,**

responds to some points made in

[\[DM32.2 2018-00141\]](#) **Response to DM32.2 2018-00087**

and makes an interim proposal for forwarding work on a property Graph Query Language (GQL).

The revision 2018-00128r1

Amends the references to 2018-00144 and 2018-00145 to use the correct paper numbers

Contains slide 14 which was presented by screen share to Meeting 255 but not contained in the original paper

Contains this slide 2 which explains the relationship to this paper to other papers, as discussed at the meeting

References (1)

[\[DM32.2 2018-000144\]](#)

Creating an Open Industry Standard for a Declarative Property Graph Query Language Alastair Green, July 2016

[\[DM32.2 2017-00026\]](#)

(Property) Graphs & SQL/Graph

Jan Michels, Oskar Van Rest, Sungpack Hong, Jinha Kim, Matthew Perry, Zhe Wu, Hassan Chafi, January 2017

[\[DM32.2 2017-00058\]](#)

Standardizing Graph Database Functionality: An Invitation to Collaborate

Jan Michels, Keith Hare, Jim Melton, February 2017

[\[DM32.2 2018-000145\]](#)

The Cypher Language 2017: Presentation to the LDBC Query Language Task Force Neo Technology Cypher Language Group, July 2017

[\[sql-pg-2018-0014\]](#)

Consensus scope and roadmap for SQL/PG, Fred Zemke, with co-authors from Oracle, Neo4j, JCC, SAP, IBM and Microsoft, May 2017

References (2)

[\[DM32.2 2018-00127\]](#)

A proposal to the database industry. Not three but one: GQL
Alastair Green, originally published 11 May 2018

[\[DM32.2 2018-00126\]](#)

The GQL Manifesto, Alastair Green, originally published 13 May 2018

[\[DM32.2 2018-00087\]](#)

GQL Project Proposal (WG3 YTZ-031), Alastair Green, 14 May 2018

[\[DM32.2 2018-00085r1\]](#)

An overview of the recent history of Graph Query Languages
(WG3 YTZ-029r1) Tobias Lindaaker, 14 May 2018

[\[DM32.2 2018-00086r1\]](#)

Summary Chart of Cypher, PGQL and G-Core (WG3 YTZ-030r1)
Stefan Plantikow, 14 May 2018

[\[G-CORE Authors for GQL\]](#)

G-CORE Authors Support GQL Manifesto, [gql.today](#), May 2018

[\[DM32.2 2018-000141\]](#)

Response to DM32.2 2018-00087 Jan Michels, July 2018

July 2016 Neo Technology proposal for a standard PGQL

Taking initial inputs like Neo's Cypher, Oracle's PGQL, and other proposals or existing language specifications, work in the standard committee to create three main outputs:

A proposal for a cross-company initiative

Creating an Open Industry Standard for a Declarative Property Graph Query Language

Copyright © 2016, Neo Technology Inc.

Neo Technology Inc.

[previous name of Neo4j Inc.]

to **Oracle**, **Microsoft** and **SAP**

1. **Grammar and Semantic Specification** of a standard property graph query language to support read and write queries and typing/schema.
2. A proposal for submission to the ISO SQL standard process for a simple **SQL/graph query language Composition** based on making graph queries operate as SQL sub-queries
3. A **Reference Implementation (RI)** and **Technology Compatibility Kit (TCK)** which reflects all of the features of the standard and the proposed standard for SQL composition.

No feature of the property graph query language specification or SQL/graph query language composition proposal will be adopted unless it is reflected in the RI/TCK.

Cypher's syntax and semantics will form the core of the standard graph query language, given Cypher's very widespread industrial use today. This does not imply that Cypher is "finished" or immutable, but the output query language should avoid stylistic or "elegant" variation on existing features, and should have at least the power and scope of Cypher today.

Adding support to the language for Conjunctive Regular Path Queries, return of graphs from queries and an optional strong typing/schema will be considered in scope for the work of the Committee.

January 2017 Cape Town WG3 initiative

In January 2017 several Oracle authors [\[DM32.2 2017-00026\]](#) proposed a twin-track approach to property graph querying:

“ ... rather than reinventing the wheel, the **SQL-embedded language** should reuse as much as possible from the **yet to-be standardized stand-alone graph query language**. Therefore, it seems imperative to develop these two approaches as closely together as possible”

This perspective was endorsed at the Cape Town meeting of WG3 that month

February 2017 WG 3 invitation to collaborate

In February 2017 key WG 3 members created a presentation inviting LDBC to collaborate

Standardizing Graph Database Functionality

An Invitation to Collaborate

Jan Michels, Keith Hare, Jim Melton
ISO/IEC JTC 1/SC 32/WG 3 Members

February 9, 2017

- WG 3 is interested in property graph technology
 - **Would like to work with LDBC (Graph QL Task Force) to develop formal standards**
- At the June 2017 SC 32 Plenary, WG 3 will consider:
 - Applying for a project split for SQL/Graph
 - Applying for a new work item/give notice of a preliminary new work item for a “Property Graph Query* Language”

ORACLE

* “Query” does not necessarily mean read-only retrieval operations only, but can include general DML/DDI operations.

April 2017 SQL PG Ad Hoc chartered

Active working group chaired by Jan Michels

Oracle, Neo4j and Microsoft have made most frequent written contributions

~6 organizations typically represented regularly

Frequent meetings to discuss

- ❑ **DDL** to declare “graphs over tables”
- ❑ **GRAPH_TABLE** function
- ❑ **MATCH** <graph patterns>

Title: **SQL-Graph Ad Hoc status report #9**

Author: Jan Michels
Source: Ad Hoc chair
Status: Status report
Date: July 12, 2018

Abstract

This paper provides a brief summary of the meetings of the DM32.2 Ad Hoc on SQL extensions for property graphs that took place since the last status report [Status report #8].

References

- [sql-pg-2018-0003r2] Neo4j SQL working group, “Property Graph Data Model for SQL”, sql-pg-2018-0003r2
- [sql-pg-2018-0006] Neo4j SQL working group, “Aligning Cypher and SQL Graph Pattern Syntax”, sql-pg-2018-0006
- [sql-pg-2018-0007] Craig Freedman, *et. al.*, “Microsoft Response to GRAPH_TABLE Proposal”, sql-pg-2018-0007
- [sql-pg-2018-0009r1] Fred Zemke, “Regular expression syntax for graph queries”, sql-pg-2018-0009r1
- [sql-pg-2018-0015] Fred Zemke, “Response to [sql-pg-2018-0007] on GRAPH_TABLE”, sql-pg-2018-0015
- [sql-pg-2018-0016] Neo4j SQL working group, “On the Semantics of Alternation, a response to sql-pg-2018-0009”, sql-pg-2018-0016

May 2018 Consensus on SQL:2020 PGQ scope

Consensus scope document was agreed on 9 May

[\[sql-pg-2018-0014\]](#)

Data model and DDL

API

Graph pattern language

Read-only

No graph projection

No DML for insert, update, delete

Title: **Consensus scope and roadmap for SQL/
PG**

Author: Fred Zemke (Oracle)
Cosigners: Zhen Hua Liu, Jan Michels, Oskar van Rest (Oracle);
Neo4j SQL Working Group (Tobias Lindaaker, Peter Furniss, Petra Selmer, Stefan Plantikow, Alastair Green) with a caveat noted in Section 1.1.4 “Supplementary statement from Neo4j SQL Working Group”;
Calisto Zuzarte (IBM);
Keith Hare (JCC);
Romans Kasperovics (SAP);
Shreya Verma, Craig Freedman, Umachandar Jayachandran (Microsoft) with a caveat noted in Section 1.1.5 “Supplementary statement from Microsoft cosigners”

GQL Manifesto

PGQL

- READ Only
- RPQs
- No GRAPH CONSTRUCT/PROJECT:
- Not COMPOSABLE YET

ORACLE PGX

G.CORE
ADVISES

- CREATE - READ
- RPQs
- GRAPH CONSTRUCT/PROJECT:
- COMPOSABLE

NO IMPLEMENTATIONS YET

Cypher

- CREATE - READ - UPDATE - DELETE
- No RPQs
- GRAPH CONSTRUCT/PROJECT:
- COMPOSABLE

• Neo4j DB • Cypher for SPARK/Gremlin
• Agens Graph • Memgraph
• Redis Graph • inGraph
• SAP HANA Graph • Cypher.PL

NEW FUSED

GQL

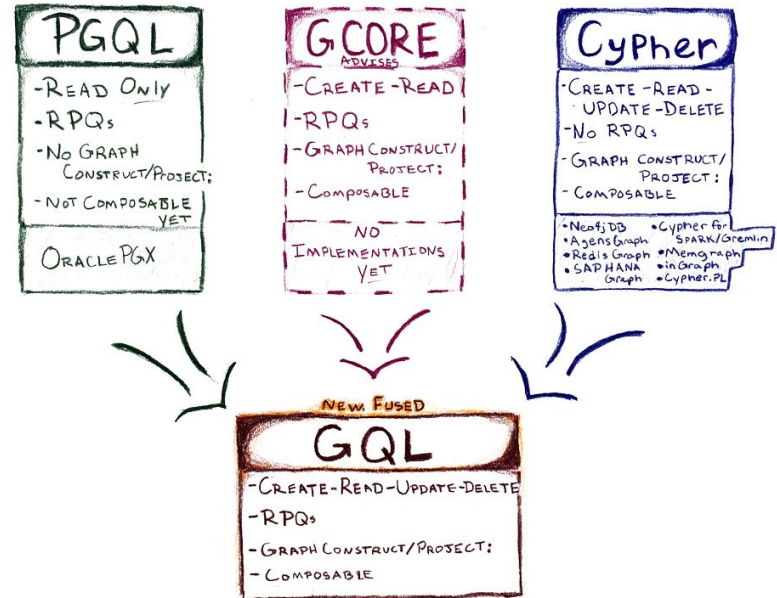
- CREATE - READ - UPDATE - DELETE
- RPQs
- GRAPH CONSTRUCT/PROJECT:
- COMPOSABLE

Proposed Inputs

[DM32.2 2018-00086r1]

The ASCII Art pattern-matching language family is a good candidate for standardization because there is a common core of design concepts, syntax and agreed semantics overlaid with useful variations, which should be subsumed into a single standard (and also by pointless variations which can be eliminated by a single standard).

The two industrial languages, Cypher and PGQL, have a high number of existing users, so we feel it's important to work carefully with these closely-related “dialects” to enable a high degree of backward compatibility and/or careful deprecation with respect to the inputs.



The [\[DM32.2 2018-00087\]](#) proposal

“That a project be started for the definition of a new standard database query language, GQL, which is specific to the property graph data model, and is distinct from, but complementary to, the SQL project, as motivated in [\[The GQL Manifesto\]](#) and in the [\[Open Letter\]](#), and whose inputs shall (with the agreement of the respective authors and/or copyright holders) include (but not be limited to) Oracle’s PGQL, the openCypher community’s Cypher language, and the LDBC research language G-CORE, and whose outputs and timescales shall be defined by the project, if established.”

[\[The GQL Manifesto\]](#) → [\[DM32.2 2018-00126\]](#)

[\[Open Letter\]](#) → [\[DM32.2 2018-00127\]](#)

Relationship to SQL Property Graph Query

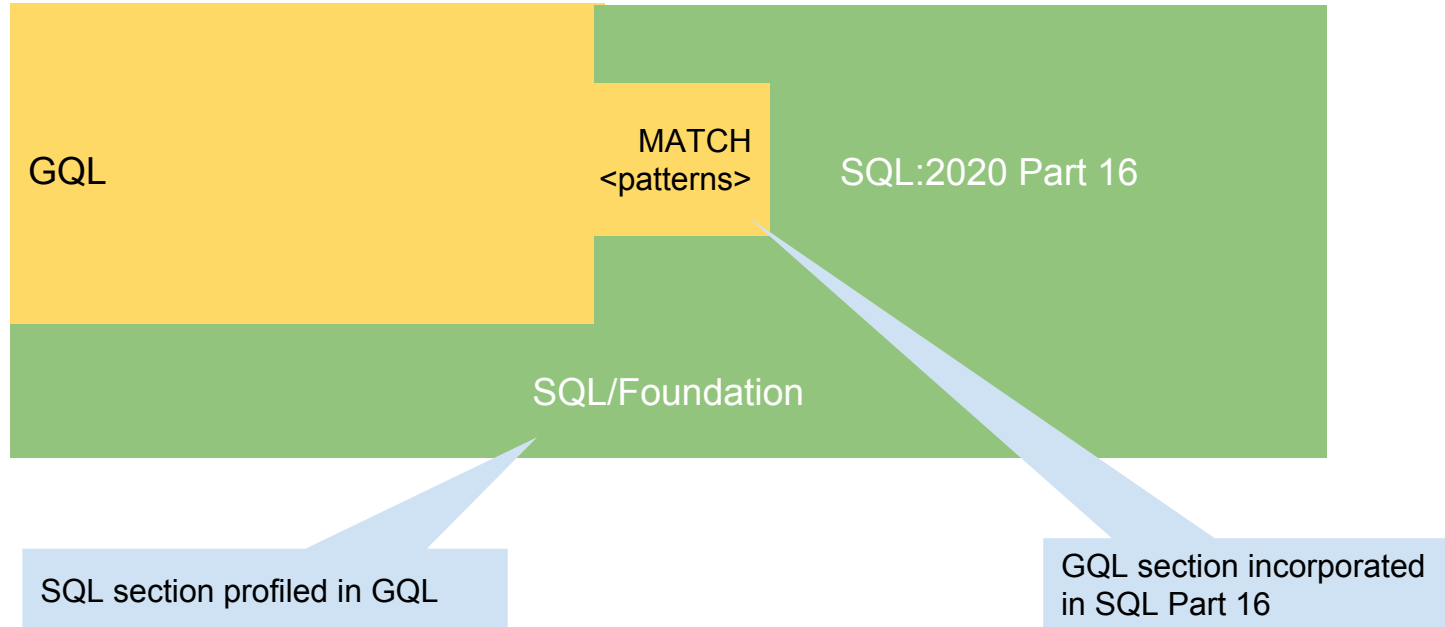
“In the public Manifesto I did not dwell on SQL PGQ as a potential input for two reasons: first, the inheritance from PGQL and Cypher has been explicit in the work of the Ad Hoc; second because the work of the Ad Hoc and of DM32.2 is not public, whereas the three languages openCypher, PGQL and G-CORE are documented in public open-source materials.

It is therefore worth emphasizing three additional points:

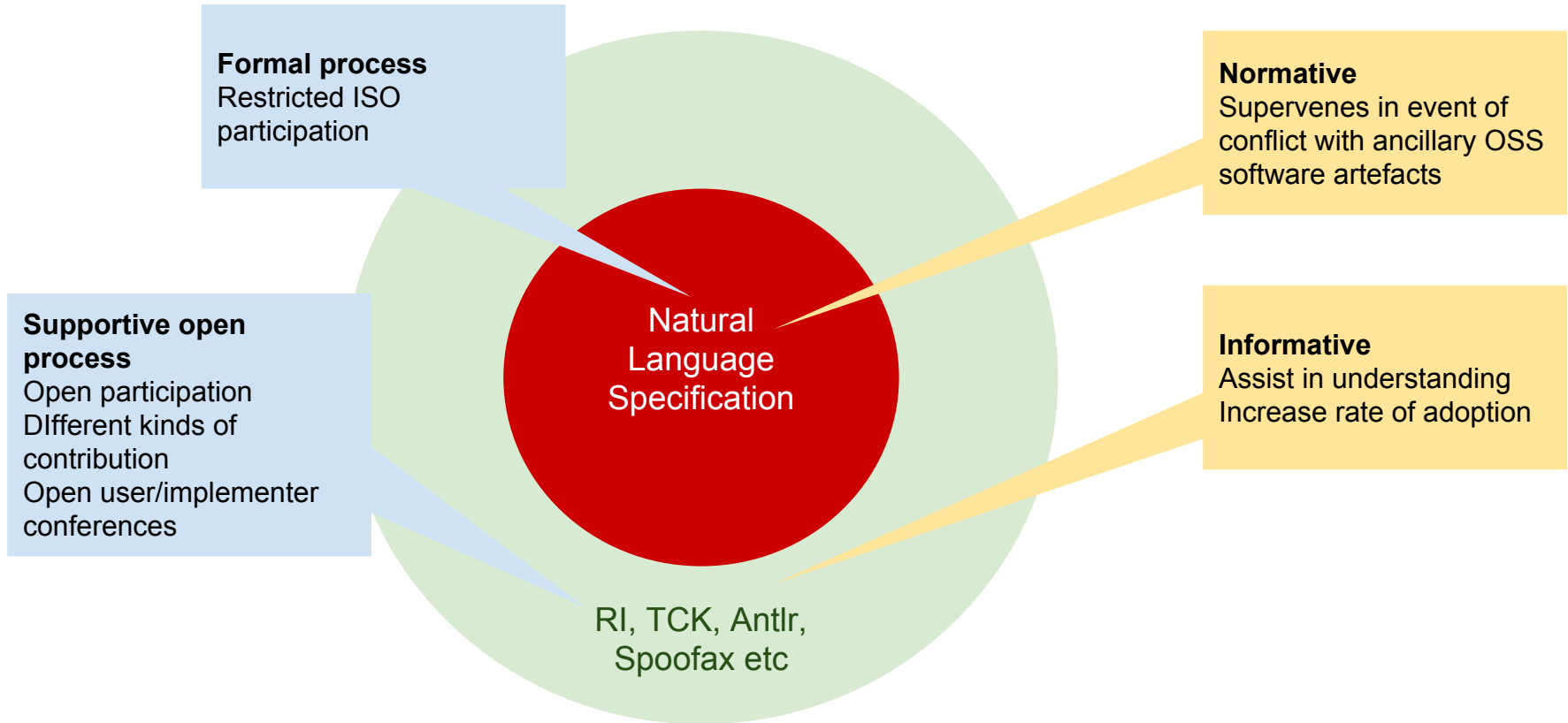
1. SQL PGQ innovations arising from the work of the Ad Hoc are a “fourth natural input” into the proposed GQL language.
2. GQL should incorporate a relevant profile of SQL/Foundation with respect to basic datatypes, expressions, ternary logic etc., to maximize adoption and interoperability.
3. SQL PGQ pure graph querying features should be viewed as a subset of GQL (whereas features that interface SQL to graph queries are better viewed as a part of SQL, and not of a stand-alone language).”

[\[DM32.2 2018-00126\]](#)

Potential for cross-references between GQL and SQL



Bridging Open Source and international standards



Interim proposal

In the spirit of these statements in Jan Michel's document [\[DM32.2 2018-141\]](#)

“... it does not necessarily mean that SQL/PGQ has to be completed for work to start on a stand-alone property graph language, either. They can run in parallel as long as priority is given to completing SQL/PGQ first and the stand-alone property graph query language does not divert from SQL/PQG where they overlap.

In addition to the weekly Ad Hoc calls concerning the SQL extensions, *a monthly Ad Hoc call for matters that concern only the stand-alone property graph query language where it does not have any overlap with the SQL extensions could be added [emphasis added]*. Where functionality between SQL/PGQ and the stand-alone property graph query language overlap (syntax and/or semantics), this would be handled by the existing weekly Ad Hoc call.”,

to request the Chair of DM32.2 to extend the charter of the SQL PG Ad Hoc

To work, in additional monthly meetings, on the GQL proposal for a stand-alone property graph query language