ANSI INCITS DM32.2 2018-00144

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

Title	Creating an Open Industry Standard for a Declarative Property Graph Query Language
Author	Alastair Green, Neo4j Inc.
Status	Informational Paper
Date	Date of original publication, 12 July 2016
	Date of submission to DM32.2, 13 July 2018

The attached document was circulated to Oracle, SAP and Microsoft on **12 July 2016**.

A proposal for a cross-company initiative

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

Document version	**Date**	**Status**	**Author(s)**
0.1	8 July 2016	Draft	Alastair Green, Neo Technology Inc.
0.2	11 July 2016	Draft	Alastair Green, Neo Technology Inc., post review
0.3	12 July 2016	Public Draft	Alastair Green, Neo Technology Inc., post CLG review

This document is made available by Neo Technology solely for the purpose of confidential discussions to investigate the possibility of a collaborative process for the formation of an open industry standard for a declarative property graph query language which involves Neo Technology as an active participant.

Recipients of this document, whose copyright resides with Neo Technology, are permitted to reproduce and distribute copies for that purpose and no other.

Summary: An Open Standard for a Declarative Property Graph Query Language

A proposal to create a technical Committee in an existing industry standardization consortium such as OASIS or W3C to define a standard declarative property graph query language.

The work of the Committee to be based on open, equal participation via contributions by all members, and adoption by agreement of all participating full members of the consortium, starting in October 2016 with the goal of initial adoption by end-2017.

This process should start with a phase of private collaboration among a limited number of Initiating Participants (vendor or service provider companies with weight in the database industry generally and/or the nascent category of property graph databases, query/processing engines or associated tooling, or major end-user organizations using the property graph data model for data processing or analytics).

In this phase the Initiating Participants will agree to a charter for the Committee to include scope and IP policy. Neo Technology volunteers to organize and facilitate this first phase.

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:

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.

SQL composition will be as simple as possible (graph subqueries) and no other extensions to SQL will be considered in the work of the Committee. An SQL Working Group (SWG) of the Committee will be formed with its own chair and proposal editor to work on this key output.

The Initiating Participants will make public moral commitments to independent implementations of the standard as the work of the Committee and SWG commences.

The Demand for an Open Standard

A number of discussions held by Neo Technology with other vendors of database and allied technologies, and with some of our partners, customers and prospective customers, reveal a widespread desire for a standardized declarative language for Property Graph data model-based technologies.

Such a standard, analogous to SQL, would help the adoption of property graph databases and query engines in data processing and analysis applications, increasing the overall size of the market for software and services that aid the use of this data model.

Frequent announcements and releases of new planned and actual property graph databases, engines and algorithmic libraries, and active research work in this area, evidence a still nascent but increasingly solid market category with a proliferation of independent suppliers.

The lack of a standard query language can therefore rapidly become an obstacle to further growth in this market.

Encouragement of Competition and Technical Innovation

Concomitantly, the creation of a standard would increase the degree of competition and resulting innovation, allowing vendors to focus on the features and quality of their implementations rather than the surface programming interface, and assisting the growth of the tools and professional services markets.

Application and service owners would feel more confident about using the property graph model, as the fear of vendor lock-in would decline, and the development of expertise and a single, common skill-set would be facilitated.

Advantages of a Declarative Language like SQL

Neo Technology has played a major role in developing the category of property graph databases, and has a very large user base for its Neo4j product.

The Cypher language has hugely helped Neo's growth as a company through increasing the understanding of tens of thousands of developers of the underlying data model, and sharply reducing the footprint of code required to program against a database that natively supports that model.

Other vendors have also been working to produce a declarative query language, for example Oracle Corporation's Property Graph Query Language (PGQL).

Others, for example Datastax, have oriented to a different style of database programming interface, developed under the aegis of the JVM-centric Apache Tinkerpop project, including the Gremlin executable DSL.

Neo Technology welcomes competition between different types of language, but we favour an attempt to create a single, standard language for each type or style.

In the past Neo has played a leading role in the evolution of Tinkerpop (which our Neo4j product supports), but we have found it to be less than ideal for easy adoption by a large and growing user community.

The proposal in this document is not competitive to Apache Tinkerpop as such: it simply addresses a different language “niche”. As a company Neo has decided to focus its language development and implementation effort on creating a declarative language for property graph data, which will play the role that SQL does for tabular data (and be presented for processing in a way that is similar to SQL).

The SQL declarative query language has the following useful features, which we wish to carry over to the standard declarative property graph query language.

- ❏ It is familiar to an enormous number of analysts, designers, data modellers and developers. Creating an analogous language will leverage this vast pool of skills.
- ❏ For the domains of schema definition, data querying and data insertion and update, it states the functional behaviours required by a database or application designer, but it does not concern itself with the manner in which those behaviours are achieved.
- ❏ The language is expressed in groups of statements that can be recorded in free-standing source modules, or as embedded extensions to other languages for compilation or precompilation, or can be successively inserted into database drivers using a variety of application programming interfaces (saliently ODBC and JDBC) for communication to remote processors, including through the medium of interactive tools.
- ❏ The language is therefore independent from general purpose programming languages and other domain-specific languages. It can be fed into interpreters, compilers and execution engines of any kind executing in any operating system environment.

Creating an analogue to SQL (occupying the same niche in the database ecosystem) does not imply that the standard query language for graph data will necessarily resemble SQL in its syntax.

Nature of the Proposed Standard

The history of the software industry shows many ways in which de facto standards can emerge. Different communities, in different eras, have chosen distinct approaches to collaboration and to achieving clarity on the definition of an effective standard.

However, the criteria for successful standards that are implemented in products used in widely varying technical and business environments (like databases) tend to be fairly constant.

The standard must be

- ❏ proven in running code, in multiple independent implementations
- ❏ described in a way that is independent of any particular implementation.

Implementation tends to limit a standard to the genuinely useful set of features that are required to achieve the goals of the domain user, and helps to radically improve its quality and unambiguous character.

On the other hand significant source code bases are often impenetrable to the reader, even to expert programmers and domain specialists. The audience and reach of natural language prose specifications is vastly wider. It is also easier to express the normative (minimal and mandatory) requirements of a standard in the abstraction of a written specification.

This is not to say that mathematically formal semantic specifications cannot also be used to achieve further precision, but such formal specifications (which are rare) tend to supplement and follow natural language specifications, again for reasons of accessibility.

Contents of an Open Standard

We propose that the **Outputs of an open standardization process** should be

> a) a **Grammar in Extended BNF with a normative prose semantic Specification in English for a declarative property graph query language**
>
> b) an accompanying **Reference Implementation (RI) and Technology Compatibility Kit (TCK)** to assist multiple implementations and to give confidence in conformance and resulting application portability.

In addition we would propose that this standardization process should include the task of

> c) proposing to the ISO SQL committee a simple **SQL/graph query language composition**, which enables a graph query language expression to act as an SQL sub-query.

Arriving at an Effective Standard

Given the industry's long history of using de jure standards for database languages and drivers, we propose that an official, open-participation process in an established industry standards consortium such as W3C or OASIS would be the most appropriate way of defining and achieving adoption of a standard comprising the three Outputs listed above.

The likely co-existence and composition of the graph query language with SQL also militates in favour of a de jure standard for a declarative property graph query language. The fact that property graph implementations are beginning to come on stream from major historical database vendors also points in this direction.

Neo Technology therefore proposes that interested companies who are active in exploiting property graph technology work together to initiate a formal standards process in an established standards venue.

This would imply the formation of a technical Committee or working group under the standard rules of the venue to accomplish the three tasks/outputs.

The formulation of a proposal for SQL composition, which is a distinct but closely related activity (task c) in the list of Outputs above) could usefully be conducted in an SQL Working Group, subordinate to the main Committee, with its own chair and editor.

The sections that follow in this document are a starting point for discussion on the details of how this initial collaboration would work, and on key aspects of the charter and underpinning commitments of effort for the formal standardization process that would ensue.

Schedule

We at Neo Technology would like to aim for **a 1.0 standard to achieve the first (lowest) rank of official adoption by December 2017.**

Our goal would be to see the **Committee starting work officially at the beginning of October 2016**.

This would imply an initial phase of **collaboration from July to September 2016** to launch the open standardization process.

Initiating Collaboration

In a first phase of collaboration we envisage a small number (something in the order of two to five) Initiating Participants working together to agree a charter for a technical standard Committee or other working group. The venue, governance and Intellectual Property policies and other ground rules of collaboration would be agreed in this phase.

Neo Technology suggests that OASIS might be the most suitable venue, given the (relative!) lack of constraints on the approach and process imposed by OASIS, but we would certainly be open to other proposals.

In addition, to the extent possible within the rules of the agreed standards venue, the Initiating Participants would agree to support commonly agreed candidates for the initial holders of primary roles such as chair, editor of the specification and maintainer/lead committer and initial committer group for the RI/TCK, as well as the chair and editor of the working group for SQL/graph query language composition. Once the standard committee is constituted any changes in the holders of these key roles will be determined by the standard processes of the standards venue.

The content of agreed common publicity and the timing of announcements of the initiative would also be determined in this phase.

This initial phase of collaboration would occur privately, and the fact of the collaboration and any news of its development would only be publicized by unanimous agreement of the participants.

At all times the Initiating Participants would use all reasonable efforts to ensure that the initial collaboration and the proposed standardization approach do not violate any antitrust or other legal restrictions on anti-competitive behaviour stipulated by market-significant jurisdictions such as the OECD countries (including the United States), People’s Republic of China, Brazil, Republic of South Africa, India, and Russia, or supranational treaty associations such as the European Union/European Economic Association, or the World Trade Organisation.

Neo Technology volunteers to organize and facilitate this initial phase, which, if successful, would end with the establishment of a formal, open standard Committee in the agreed venue.

Open Equal Participation and Collaboration

We at Neo Technology have some initial thoughts about a few simple principles to achieve effective collaboration and mutual trust.

The rules and conventions of the venue, and the standards committee charter should combine to ensure completely equal rights to propose and discuss the technical content of the standard for any member adhering to the charter and joining the committee, whatever their status (individual, commercial company, non-profit, government body, university researcher etc).

Contributions from any participant should be eligible for inclusion in the outputs of the committee and its working groups.

The editors of the graph query language specification and of the SQL/graph query language composition proposal, and the maintainer/committer group of the RI/TCK, would have the right to exercise reasonable and customary quality control on the outputs of the committee's work as contributions are proposed and absorbed.

No participant would have a right of veto or other privileges with respect to the content of the Specification and the SQL Composition Proposal or the required functionality of the RI/TCK, with the following proviso: that voting on formal adoption of the standard should in some reasonable and customary way be restricted to reflect the level of involvement in the standards formation process, and the size and/or influence of participants in this market category.

The corollary of this is that the Initiating Participants would always endeavour to maximize consensus, and to minimize the perception or reality of exclusion of smaller or less influential participants throughout the process leading to adoption of the standard.

The OASIS model of tiered membership may be helpful in drawing this line in an objective way using established processes. For example, an individual (unaffiliated) contributor to an OASIS Technical Committee cannot vote in a ballot required to adopt a Committee Specification.

Agreement on Initial Inputs and a Scope of Outputs is vital to achieving a viable committee charter. A successful process requires limits that enable cordial and productive collaboration and induce time bounds leading to definitive, usable work product.

Initial Inputs

Neo Technology proposes the following **Initial Inputs**, subject to the condition that all Initial Inputs should be licensed for use, inter alia, as an input for the proposed standardization process under the provisions of any agreed Intellectual Property policy for the proposed standard committee:

1) The Neo4j Cypher language, as documented in the Cypher Language Reference Card and other documentation accompanying Neo4j 3.0, and specifically the subset of that language formulated under the aegis of the existing openCypher initiative, including the EBNF grammar and Test Compatibility Kit (which is shortly to be released at the time of writing). These artefacts define syntax and implied semantics relating to graph querying (read activity) and to graph creation and modification (write activity).

2) The Oracle Property Graph Query Language (PGQL) as codified in the PGQL 0.9 specification and Github repository of supporting parser and Abstract Syntax Tree class libraries.

3) Any other specification or other rendition of a pre-existing purely declarative language developed by one of the Initiating Participants involved in the initiation of the standardization process relating to property graph querying, graph creation and modification, or graph schema and typing.

4) Any other partial or whole specification or other rendition of a pre-existing or proposed purely declarative language developed by a person other than one of the Initiating Participants relating to property graph querying, graph creation and modification, or graph schema and typing, which is agreed to be valuable as an input by all of Initiating Participants and whose authors agree to its use for that purpose. The work of the LDBC graph query language task force in the past year might be a case in point.

5) Any written proposal for change, development or elaboration of one of the Initial Inputs listed in 1) to 4) above.

Clearly, once the Committee’s work is under way it is likely that other inputs may be submitted and the list above is not intended to be exhaustive or to discourage other contributions.

Scope of Outputs

Neo Technology proposes the following **Scope of Outputs** which, once revised and agreed and incorporated into the charter of the standard Committee, should not easily be modified[1]:

1. The outputs of the standardization process will be limited to language features that support property graph querying (read activity), creation and modification (write activity), application of graph algorithms, and incorporation of procedures, functions or other subroutines in the specified language or other languages, the definition of optional types and schema or ontological constraints that enable fixity in the structure, content and meaning of graphs or sub-graphs within graphs, the creation of profiles or vocabularies that delimit sub-languages, and a means of composing graph queries as subqueries of the SQL language.

2. The Grammar and semantic Specification of the property graph query language that results from the standardization process will be based primarily upon the grammar and semantics of the Cypher language as described in Initial Input 1) above, and will not materially reduce its scope and power. This primacy reflects the widespread prior use in hundreds of production applications using the Cypher language as part of paid subscription usage of Neo4j (and unpaid community usage is likely to be one order of magnitude greater).

   Key elements of the grammar of Cypher that will be preserved in the outputs of the process include the pattern matching syntax that describes nodes (vertices) and their relationships (edges), including the direction of relationships, and the naming and cardinality of node and relationship types or labels, the retention of a simple core mandatory type system for property values, the MATCH … RETURN/WITH syntax, the pipelining or chaining of query statements, and the CREATE, MERGE and SET syntax.

3. The grammar and semantic specification of the property graph query language that results from the standardization process can be extended to include Conjunctive Regular Path Queries, and the ability to return new or sub-graphs, and significantly enhanced (optionally applied) typing and schema or ontological constraints as described in 1) above.

4. Mechanisms for clearly identifiable vendor or user extensions of the property graph query language must be defined.

5. A proposal for SQL use of graph queries as subqueries of SQL that causes only minimal change to the SQL standard must be created suitable for submission to the ISO SQL standard change process, and no other proposal for changes to the SQL language will be discussed in the work of the standard committee or its working groups.

[1] For example, an OASIS Technical Committee has special processes for modifying a charter, which make it very difficult to expand the scope of the intended outputs once the process is underway.

6. No feature in the property graph query language standard or the SQL composition proposal can be adopted by the committee if it has not been reflected in the RI and TCK by the time of an adoption meeting or ballot.

Morally Binding Commitments of Effort

In order to ensure that the standard is based on running code, we also propose that Initiating Participants (see the section headed “Initiating Collaboration” above) should all commit themselves to at least an independent research implementation of the graph query language Specification as a whole, as it develops in the work of the committee.

Such an implementation could leverage elements of the Reference Implementation for language parsing and AST formation, and would be tested for conformance by use of the TCK, but would have to contain original query planning/execution and storage components to validate the exactness of the semantic Specification.

At least two Initiating Participants would commit to implementing read-only query features in a product or service intended for generally available use in the open market. Neo Technology would make that commitment with respect to its Neo4j product.

At least one Initiating Participant would commit to implementing insert/update features in a product or service destined for the market. Again, Neo would make that commitment for Neo4j.

At least two Initiating Participants would commit to independently implementing the SQL composition proposal. At least one of these implementations will be in a product or service destined for the market.

In addition each Initiating Participant would enable at least one member of their qualified technical staff to actively participate in the formation of all the written outputs of the Committee and SQL Working Group, and at least one other to contribute to the development of the RI/TCK, preferably to the level of acting as committers.

Such staff would be readily available at all points in the process leading to adoption of the standard to actively participate in formulating and processing change proposals and associated coding, testing, conformance proving and documentation tasks.

These commitments of effort would not be legally binding, but would have to engage the reputation and credibility of the participants, and would therefore be made public at the point of initiation of the formal standard Committee.