

LDBC GQL Implementation Work Charter

Working Group Name

LDBC GQL Implementation Working Group

Contents

Lead	2
Initial members	2
Work group	2
Charter version date	2
Charter version	2
Date charter version agreed by Members Policy Council	2
Mission	2
Motivation	2
Scope of work	3
Intended output documents	3
Mode of work, cadence and intended timescales	4
Related Task Forces or Working Groups	5
References to relevant documents, standards, etc.	5
Appendix 1: Initial membership	7

Lead

Michael Burbidge

Initial members

There are 11 member organizations of LDBC represented, and 22 individual associate members. The organization member list is in [Appendix 1. Initial membership](#).

Work group

Ad-Hoc (approximate lifetime of 18 months)

Charter version date

14 November 2023

Charter version

1.1

Date charter version agreed by Members Policy Council

21 November 2023

Mission

Create tooling and documentation to assist in the implementation and adoption of GQL.

Motivation

The GQL specification is a complex, highly technical document. It defines the syntax and behavior of a GQL Implementation in a very precise, yet somewhat unapproachable language.

The specification defines the GQL syntax using BNF. But the grammar productions are organized to facilitate the association of behavior rules with grammar productions and not to facilitate the implementation of a GQL parser. The productions cannot be used directly to implement a parser using industry standard tooling such as ANTLR or even a hand-written recursive descent parser.

GQL is a large, complex language when compared to modern programming languages such as [Go](#). Many developers will find it challenging to interpret and reorganize the set of BNF productions defined by the specification into a set of productions that can be used to implement a parser, using industry standard tooling such as ANTLR.

While the specification does define the associated behavior of a GQL program, there is no tooling to help developers validate an implementation against the specification.

To address these issues, and to facilitate and accelerate the adoption of GQL, as the industry standard graph query language, the LDBC GQL Implementation Working Group will produce additional tooling and documentation to assist developers and early adopters in the implementation, testing and adoption of GQL.

Scope of work

This work will produce tools and documentation that make it easier for developers to implement a GQL compliant implementation. In addition, the documentation will make it easier for early adopters to use a GQL compliant implementation.

An earlier version of this charter stated that the produced tools and documentation would address SQL/PGQ as well as GQL. After further discussion among members of the working group it was decided that the possible implication that the group would address SQL/PGQ with equal vigor and completeness to GQL was unintended. This could result in an overwhelming set of requirements and an unmanageable amount of work. As a result of this discussion, SQL/PGQ was removed from the title of this charter.

The LDBC GQL Implementation Working Group's work will focus on GQL implementation tooling and documentation: our consideration of SQL/PGQ will be limited to documentary commentary on the likely usage patterns, the overlaps and the differences between the two standards, and the consequent motivation for their parallel existence.

These twin standards have been created in a very deliberate and structured coexistence (centering on the common property graph data model and the Graph Pattern Matching sub-language of both), but the software tooling outputs of this Working Group will consider only GQL, and not SQL/PGQ, while taking it as read that the currently undocumented norm that SQL/PGQ and GQL share the same GPML (or at least, will never contain contradictory GPMLs) applies and will continue to apply, as a matter of settled policy within SC32 and its WG3."

At some future date, after successful delivery of tooling and documentation focused on GQL, this charter could be expanded to adapt that tooling and documentation to address SQL/PGQ also.

Intended output documents

The LDBC GQL Implementation Working Group intends to produce the following outputs, which address tooling, testing and additional documentation.

ANTLR grammar

ANTLR grammar derived from the GQL Specification grammar digital artifact. i.e. the XML representation of the BNF grammar.

How the ANTLR grammar is created and kept up to date, will be one of the initial tasks of the working group. Research may show that an automated process can be created. Alternatively the ANTLR grammar may be created and kept in sync by hand.

Grammar railroad diagrams

Grammar tooling from openCypher applied to GQL to produce and publish a set of railroad diagrams for the GQL grammar.

Technology Compatibility Kit

Technology Compatibility Kit modeled on the [openCypher TCK infrastructure and tests](#). Parts of openCypher and GQL are very similar, it may be possible to use the openCypher TCK tests as a starting point.

LDBC Technical Report

Free LDBC Technical Report which describes the GQL specification contents in a more informal fashion with many examples. This should cross-ref and explicate the actual specification. The audience for this report is implementers, early adopters.

The report will be similar to the [Cypher 9 Guide](#) by Petra Selmer and/or the [SQL++ Tutorial for SQL Users](#) by Don Chamberlain.

The report can and probably will document the relationship between GQL and SQL/PGQ. In particular it may define how the two overlap and differ, as well as why and how the two specifications coexist. It would not document SQL/PGQ itself.

Specification feedback mechanism

Specification feedback mechanism directed or channeled to W3C.

Mode of work, cadence and intended timescales

It is anticipated that all work will be remote, and coordinated via the tools available on basecamp and one or two video meetings each month.

The working group will be segmented into three subgroups one focused on each of the following major outputs: ANTLR grammar, Test Compatibility Kit and the LDBC Technical Report.

The working group will leverage the existing LDBC github infrastructure, subscription and support as the place where the outputs are developed and published from.

Timescales

The following table provides estimates of the first versions of outputs (and provide for continual, rolling updates). The outputs are prioritized according to when they might be needed by consumers.

Output	Timeframe
ANTLR grammar	May 2024
Railroad diagrams	May 2024
Technology Compatibility Kit	December 2024

Output	Timeframe
Specification feedback mechanism	February 2024
Technical report	First draft June 2024, second December 2024

Phased development

As documented earlier, the scope of the working group is focused primarily on GQL. However since GQL and SQL/PGQ share a common Graph Pattern Matching Language (GPML), some subset of the outputs of the working group could be beneficial to those implementing SQL/PGQ. It would be ideal if those subsets are available early in our development cycle, rather than later.

As a result, the working group will phase the development of the TCK. In the first phase we will focus on the GQL variant of GPML. In later phases we will expand our work to address other parts of GQL not covered in phase 1. e.g. Data Manipulation Language (DML) and Data Definition Language (DDL).

This specific, documented, phasing of work is only applicable to the TCK. For the other outputs, the working group can stage the work in any order it decides.

Vendor Bias

The outputs of the working group will not be biased towards any particular vendor.

There are two ways in which the GQL Specification allows for flexibility in vendor implementations. Without care in how these are dealt with the outputs of this working group might be biased towards one or more vendors.

The first axis of flexibility is optional features. The specification defines a number of optional features, which a given implementation may or may not implement and still be in conformance. To avoid vendor bias, the outputs of the working group will address the superset of all optional features.

The second axis of flexibility is implementation-defined and implementation-dependent elements of the specification. The working group will develop a plan for how each output will deal with implementation defined and dependent elements, so as to avoid vendor bias.

Related Task Forces or Working Groups

None

References to relevant documents, standards, etc.

[August 2023 GQL Draft Specification](#)

[ANTLR Parser Generator](#)

[openCypher](#)

[Cypher Query Language Reference Version 9](#)

SQL++ Tutorial for Users

openCyber TCK

Appendix 1: Initial membership

Member Name	Organization	Email
Marko Budiselič	Memgraph	
Michael Burbidge		mburbidg@gmail.com
Pengwei Chen	Beijing Volcano Engine (ByteDance)	
Malcom Crowe	PyrrhoDB	Malcolm.Crowe@uws.ac.uk
Alin Deutsch	TigerGraph	
Sanjay Dixit		
Thomas Frisendahl		
Alastair Green		
Cole Greer	Improving, Apache Tinkerpop	
Longbin Lai	Alibaba Damo Academy	
Kaiwei Li	Ant Group	
Yunkai Lou	Alibaba Damo Academy	
Keith Hare	JCC Consulting, Inc.	
Bei Li		
Stefan Plantikow	Neo4j	
Lincheng Ge	Ant Group	
Samuel Schwebel		
Petra Selmer		
Dominik Tomaszuk	University of Bialystok	
Hannes Voigt	Neo4j	
Yang Xia	Improving, Apache Tinkerpop	
Nick Yakovets	TU Eindhoven, AvantGraph	n.yakovets@tue.nl
Yan Zhou	CreateLink	zhouyan@chuanglintech.com
Diwen Zhu	Alibaba Damo Academy	