Property Graph Querying roadmap and initial scope

Title	Property Graph Querying roadmap and initial scope
Authors	Neo4j SQL working group[1]
Status	Outline partial draft of initial working document on SQL PGQ
Date	7 May 2018

Contents

1. References

[sql-pg-2017-0047r1]	Peter Furniss, Alastair Green, Petra Selmer, “SQL Graph Query Procedures” (with corrigenda), August/October 2017

[1] Current members of the Neo4j SQL working group are: Alastair Green, Peter Furniss, Petra Selmer, Stefan Plantikow, Tobias Lindaaker

2. Roadmap and Initial Scope of extensions to SQL for Property Graphs

Taking into account verbal comments by Fred Zemke (Oracle) at the Ad Hoc meeting on 2 May 2018, and the proposals made by Neo4j in [sql-pg-2017-0047r1], and the state of progress in the Ad Hoc to date, we'd like to suggest the following roadmap and initial scope.

This is a working proposal: there are edge cases and different ways of describing the potential progression.

> Text in **Bold** is In Scope for the Initial Working Draft to be discussed at WG3 meeting in Ilmenau, early October, with the assumption that the IWD will not change scope as it transmutes into a CD for inclusion as a new part of a presumed SQL:2020.
>
> Text in *Italic* is Potentially In Scope for the Initial Working Draft, but only if the In Scope work has been successfully drafted into an IWD.
>
> Text in Roman is Out of Scope for the Initial Working Draft, but indicates useful progressions in functionality supporting graph data processing.

3. SQL Property Graph Data Model and DDL

1. **Define the SQL version of the PGDM**
2. **Define how DDL will enable Graph objects to be created and dropped in the Information Schema, which conform to the PGDM**
3. **Define how DDL will enable named Graph objects to be optionally mapped to Tables in the Information Schema, including the lifecycle of that mapping (initial state, additions, amendments, partial deletions)**
4. Definition of Graph metadata and of Graph (reference) type to support graph existence searches and graph projections (construction).
5. Definition of Node, Edge and Path data types.

4. Graph functions and operations

Function	Operations
GRAPH_TABLE	**MATCH verb** **Path pattern match with simple node and edge type and property value predicates for fixed length paths, with presence, alternation and conjunction tests on node and edge types, and an entity identity function** **Same for variable length paths, with constraints on min and max length of paths** **Same for shortest and top *k* paths, including distinction and limitation** **Control over degrees of isomorphism, including homomorphism, edge isomorphism, isomorphism***, and extending to control for specified edge and node variables* Ability to express path pattern matches using regular expressions*, including patterns of sub patterns* *Ability to name a path pattern, for use as the operand of another pattern (pattern "macros")* **Support for property projection into a SQL table** Support for entity projection into a SQL table **Support for "foreign languages": ability to insert a table-projecting Cypher query, or a PGQL query, into the function to act as a SQL sub-query**
GRAPH_VALUE	*As for GRAPH_TABLE but for any scalar projection of MATCH results*
GRAPH_EXISTS	Return a boolean given graph metadata inputs which allow a Graph in the

	Information Schema to be uniquely identified. Return a table of graph references given partial matches on input graph metadata.
GRAPH_QUERY	GRAPH ENTITY INSERT GRAPH ENTITY UPDATE GRAPH ENTITY DELETE GRAPH PROJECT into the Information Schema as named object (CREATE GRAPH <foo> FROM GRAPH_QUERY), but without support for GRAPH reference datatype (no projection into outer query) Use of GRAPH PROJECT to project graphs into outer query, implying GRAPH reference type and graph metadata Support for “foreign languages”: ability to insert a graph-projecting Cypher query into the function to act as a SQL sub-query

5. An ITI and openCypher contribution from Neo4j Inc.

This contribution is a Deliverable under the terms of clause 2.2.1 of the Agreement for Membership in the InterNational Committee for Information Technology Standards ("INCITS"), a Division of the Information Technology Industry Council ("ITI") to which Neo4j Inc. is a party.

It is also a contribution to the openCypher community[2] and like all such contributions is:

This document is a contribution by Neo4j's SQL working group to the openCypher project and to the SQL standards formation process.

[2] https://www.opencypher.org/