From: **Alastair Green** alastair.green@neo4j.com
Subject: Copy of submission of Position Statement sent 17 December
Date: 23 January 2019 at 14:22
To: W3C Data Workshop Program Committee group-data-ws-pc@w3.org

The submitter, Mats Rydberg of Neo4j, is out of the office, and it appears that his submission on 17 December got caught in a mailing list moderation process that was not completed.

The text/HTM: version is as follows:

Multiple Graphs in Cypher for Apache Spark

Authors: Mats Rydberg (Neo4j), Max Kiessling (Neo4j), Hannes Voigt (Neo4j)

Cypher is a graph querying language, well-known in its domain. It has been developed as a part of the transactional graph database Neo4j as a sort of graph sibling of SQL, and is now governed by the openCypher project [1]. Cypher as a language operates similarly to SQL in that it always binds values in a tabular format, the difference being that the main input is a graph, not a set of tables. In order for Cypher to become closed over its domain, it needs to be able to output also graphs. This is being addressed in the upcoming release of Cypher, version 10, in a language extension called Multiple Graphs.

In Multiple Graphs Cypher, it is possible to reference several input graphs, construct new graphs, combine graphs, output graphs, and manage a catalog of graphs and graph views. Key goals have been to enable Cypher queries to become composable and federative.

Cypher for Apache Spark (CAPS) is an open-source project [2] that brings Cypher to the distributed dataflow system Apache Spark. It is the first project to implement support for Multiple Graphs Cypher. We implement Cypher on top of Spark SQL, which presents the challenges of translating Cypher into relational algebra, as well as taking into account the characteristics of a distributed dataflow system. An overview of the implemented parts of the language extension may be studied in the Morpheus User Guide [3]. (Morpheus is the name of a commercial product built around CAPS by Neo4j.)

Due to the schema-strict nature of Spark SQL, we have also made innovations in graph schema modelling, through a language called Graph DDL. In Graph DDL, graphs may be defined by their constituent parts, node and relationship types, and abstract graph types may be extracted and shared between graph instances. Graph instances may be populated through mapping to SQL tables. A language overview may be studied in the Morpheus User Guide [3].

[1]: http://www.opencypher.org/

[2]: https://github.com/opencypher/cypher-for-apache-spark

[3]: https://neo4j.com/docs/morpheus-user-guide/preview/mg-cypher/

Alastair Green
Lead, Query Languages Standards and Research
Product Manager, Cypher OSS/Neo4j Morpheus for Apache Spark

Neo4j Inc
8th Floor, Friars Bridge Court
41-45 Blackfriars Road
London SE1 8NZ

+44 795 841 2107
alastair.green@neo4j.com