Cypher schema constraints proposal

CIP2016-12-16 “Constraints syntax”, Mats Rydberg

A presentation summarizing this proposal is appended

“Schema and constraints”, Mats Rydberg

First openCypher Implementers Meeting (oCIM 1) - 8 February 2017
SAP Walldorf, Germany
LDBC Open Access to External Papers

In this series, Linked Data Benchmark Council makes papers published originally for a restricted audience available for open access.

Such papers are of interest to our members and the public, and are concerned with topics that relate to the work of LDBC. They are published with the permission of their copyright holders, which may have been given by a licence grant.

This article and accompanying presentation are relevant for the work of the LEX (LDLC Extended GQL Schema) Working Group.

These documents have the character of technical reports: they have not been submitted to or accepted via peer review by an established scholarly publication.

Copyright © 2019-23 Neo4j Inc.

Linked Data Benchmark Council by the terms of our LDBC membership agreement and the IP policies contained therein hereby licences these documents Attribution 4.0 International (CC BY 4.0), in accordance with our Byelaws.
A constraint is removed by referring to its name. Subsequently the handle with which a user may refer to the constraint, for example when dropping it. In the case where a constraint command will always return exactly one record, if successful. Note that also constraint name. In order for a user to be able to drop such a constraint, the system-generated name is therefore returned since constraints always are named, but user-defined names are optional, the system must sometimes generate a.

This CIP specifies the general syntax for constraint definition (and constraint removal), and provides several examples of possible uses of the language defined by that syntax. None of the examples provided are to be.

Alternative syntaxes have been discussed: 2.5. Alternatives 4.Benefits to this proposal 2.4. Interaction with existing features 3. above. The domain of a node key constraint is thus.

A wider definition is not necessary for this type to satisfy the requirements of the. Some constraints may prove challenging to enforce in a system seeking to implement the contents of this CIP, as these.

The contents of this field are left unspecified, to be used for implementation-specific messages and/or details. In order for a user to be able to drop such a constraint, the system-generated name is therefore returned since constraints always are named, but user-defined names are optional, the system must sometimes generate a.

Alternative syntaxes have been discussed:
Schema and Constraints

Mats Rydberg
mats@neotechnology.com
Schema in Cypher

- Cypher is *schema-optional*
  - Fits well with heterogenous data
  - Makes typing and query planning harder
  - Does not fit well with many existing table-based engines
- Constraints are the only tools to enforce structure in the data

Schema in Cypher is a point where we expect there to be major developments as more actors get involved.
New constraint syntax

- Consistent syntax for all types of constraints (CIP)
- Re-use as much as possible from the rest of the language
- Allow for a large set of future constraints, some of which are vendor-specific
  - This allows vendors to use more strict schema when necessary

```
CREATE CONSTRAINT <name>
FOR <simple pattern>
REQUIRE <constraint expression>
```
Constraints, examples

- Property uniqueness constraint
  
  ```cypher
  CREATE CONSTRAINT unique_person_names
  FOR (p:Person)
  REQUIRE UNIQUE p.firstName, p.lastName
  ```

- Property existence constraint
  
  ```cypher
  CREATE CONSTRAINT person_must_have_firstName
  FOR (p:Person)
  REQUIRE exists(p.firstName)
  ```
Constraints, examples

- **Property value constraint**
  
  ```
  CREATE CONSTRAINT roads_must_have_positive_finite_length
  FOR ()-[r:ROAD]-()
  REQUIRE 0 < r.distance < infinity
  ```

- **Property type constraint**
  
  ```
  CREATE CONSTRAINT people_schema
  FOR (p:Person)
  REQUIRE p.email IS STRING
  REQUIRE p.name IS STRING?
  REQUIRE p.age IS INTEGER?
  ```
Constraints, examples

- **Cardinality constraints**
  - `CREATE CONSTRAINT spread_the_love
    FOR (p:Person)
    REQUIRE size((p)-[:LOVES]->()) > 3`

- **Endpoint constraints**
  - `CREATE CONSTRAINT can_only_own_things
    FOR ()-[:OWNS]->(t)
    REQUIRE (t:Vehicle) OR (t:Building) OR (t:Object)`

- **Label coexistence constraints**
  - `CREATE CONSTRAINT programmers_are_people_too
    FOR (p:Programmer)
    REQUIRE p:Person`
That's it!

Questions?