MI-PDB, MIE-PDB: **Advanced Database Systems**

Practical Class 5:

Neo4j – Cypher

12. 5. 2017

Martin Svoboda
svoboda@ksi.mff.cuni.cz

http://www.ksi.mff.cuni.cz/~svoboda/courses/2016-2-MIE-PDB/

Data Model

- Graph
  - **Nodes** = graph vertices
    - Set of labels
    - Set of properties
  - **Relationships** = directed graph edges
    - Type
    - Set of properties
- **Properties**
  - Key-value pairs

Cypher Expressions

- **Literals**
  - Integers and doubles: 13, -40000, 3.14, 6.022E23
  - Hexadecimal/octal integers: 0xFC3A9, -0x66eff, 01372
  - Strings: "Hello", 'World'
    - Standard escaping sequences using \
  - Boolean: TRUE, FALSE
- **Identifiers**
  - n, node, `A weird name!`
- **Properties**
  - node.prop, relationship.prop, n["prop"]

Cypher Expressions

- **Collections**
  - ["a", "b"], [1,2,3], ["a", 2, n.property], [ ]
- **Function calls**
  - length(p), nodes(p), …, avg(x.prop), count(*)
- **Path patterns**
  - (a)-->()<--(b)
- Operator applications, comparisons, …
  - 1 + 2, 3 < 4, a.prop = "Hello", length(p) > 10
  - …

Cypher Clauses

- Reading clauses
  - **MATCH**
    - Description of sub-graph patterns we are interested in
    - `MATCH (d:Label)-[:Type]->(e)`
    - OPTIONAL MATCH
  - **WHERE**
    - Additional (filtering) constraints to graph patterns
    - `WHERE (e.prop > 100) AND (e.name)`

Cypher Clauses

- General clauses
  - **RETURN**
    - Definition of columns to be included in the query result
    - `RETURN e.prop, COUNT(*) AS count`
  - **WITH**
    - Intermediate result to be pipelined to further clauses
  - **ORDER BY**
    - `ORDER BY e.prop ASC, count DESC`
  - **SKIP + LIMIT**
    - Number of rows to be skipped and included in the result
    - `SKIP 30 LIMIT 10`

Cypher Functions

- Scalar functions
  - **SIZE**(collection), **SIZE**(pattern)
  - **LENGTH**(path), **LENGTH**(string)
  - **TYPE**(relationship), **ID**(node), **ID**(relationship)
  - **HEAD**(collection), **LAST**(collection)
  - **STARTNODE**(relationship), **ENDNODE**(relationship)
- Collection functions
  - **NODES**(path), **RELATIONSHIPS**(path)
  - **LABELS**(node)
  - **TAIL**(collection)

Assignment 1

- Express the following Cypher query
  - **Return personal numbers and full (concatenated) names of all employees**
  - Include only employees with salaries greater than *2200* and e-mail addresses from domain *co.org*

e.number	name
E4	Peter Brown
E2	Arthur Taylor
E3	Martin Smith

Assignment 2

- Express the following Cypher query
  - **Return personal numbers of all employees together with ids and names of departments they work in**
  - Include only employees that were employed on *2013-01-01* or later

e.number	d.id	d.name
E4	D1.2	Engineering
E3	D1.1	Development

Assignment 3

- Express the following Cypher query
  - **Return identifiers and names of all departments**
  - **Exclude departments with no direct employees**
  - Order these departments according to their names
  - Omit the first *2* of these departments and return at most the *3* following

d.id	d.name
~~D1.1~~	~~Development~~
~~D1.2~~	~~Engineering~~
D1	Production

Assignment 4

- Express the following Cypher query
  - **Return unique last names of all employees that work (even indirectly) in a particular department (e.g. *D1*)**

e.lastName
Smith
Brown
Taylor

Assignment 5

- Express the following Cypher query
  - **Compute the average salary of all employees (direct only) for each department**
  - Exclude departments without names
  - Include departments with at least *2* employees only (direct)
  - Order the output using these salary averages (*DESC*) and then using department names (*ASC*)

id	avgSalary
D1.2	2750

Assignment 6

- Express the following Cypher query
  - **Return identifiers of all departments and for each one of them provide the following information**
    - Full name of their direct manager (only when specified)
    - Collection of identifiers of all sub-departments they contain (even indirectly), ordered using these identifiers

id	subdepartments	manager
D1	[D1.1, D1.2, D1.2.1]	John Smith
D1.1	[]	*NULL*
D1.2	[D1.2.1]	Arthur Taylor
…		