

Big Data Management and Analytics
WS 2016/17

Tutorial 3: NoSQL

Assignment 3-1 *Cassandra*

In this assignment you are going to become familiar with the column based NoSQL database Cassandra.

- (a) Compare RDBMS like MySQL to the column database Cassandra: What are the counterparts of a database, a table, a primary key, a column name and a column value in Cassandra?
- (b) Download and install Cassandra. The download source can be found here: <http://cassandra.apache.org/doc/latest/>
An installation manual and tutorial can be found on <http://cassandra.apache.org/doc/latest/>
- (c) Create a keyspace named *mycompany*.
- (d) Create a column family named *employees*. It has the following column keys: *emp_id*, *emp_name*, *emp_city*, *emp_phone*, *emp_sal*.
- (e) Add the following employees to the *employees* column family:

emp_id	emp_name	emp_city	emp_phone	emp_sal
0	Santa Claus	Northpole	12345	90000
1	Dr. Strange			20000
2	James T. Kirk	USS Enterprise		
3	Ada Lovelace	London	10121815	45000

- (f) Return all employees which have a salary above 30000.
Hint: for this purpose you'll probably use a WHERE clause. Can you use WHERE clauses the way you'd use them in RDBMS?

Assignment 3-2 Neo4j

In this assignment you are going to become a bit more familiar with the graph based NoSQL database Neo4j

- (a) What are the main building blocks of the Neo4j data model? What are the corresponding counterparts for a database, a table, a row, a column and join in RDBMS?
- (b) Download and install Neo4j. An installation manual and tutorial can be found on <https://neo4j.com/docs/developer-manual/current/>
- (c) In this task you'll become a bit more familiar with the cypher query language (CQL). Based on the table below, create employee nodes with the properties emp_id, emp_name, emp_city, emp_phone and emp_sal. The label of all nodes is *Employee*.

node name	emp_id	emp_name	emp_city	emp_phone	emp_sal
santa	0	Santa Claus	Northpole	12345	90000
strange	1	Dr. Strange			20000
kirk	2	James T. Kirk	USS Enterprise		
ada	3	Ada Lovelace	London	10121815	45000

- (d) Write a query (by using a MATCH-RETURN statement) which returns all the employee names.
- (e) Write a query (by using a MATCH-WHERE-RETURN statement) which returns the names of all employees which have a salary above 40000.
- (f) Create the following node with the label *Customer*:

node name	cust_name	cust_city	cust_complain
thor	Thor	Asgard	Hammer not working

- (g) Santa Claus delivers Thor a new working hammer. Create a relation between Santa and Thor with the type *delivers_to* and the label *DELIVERS* (using the MATCH-CREATE statement). Use the Neo4j graph view in order to see your current graph.