Hands-on MongoDB Development

Contents

[Introduction 2](#_Toc477082975)

[1 Advanced Query – and a comparison between MongoDB Find and Aggregate vs. (Oracle) SQL 3](#_Toc477082976)

[2. MongoDB Compass 5](#_Toc477082977)

[3. Development with NodeJS and MongoDB 6](#_Toc477082978)

[4. Development with Java and MongoDB 7](#_Toc477082979)

## Introduction

In this hands-on, you will be working with MongoDB – in the MongoDB shell on advanced query functionality, with Node.JS and Java to interact from those technologies with a MongoDB database and through MongoDB Compass – a GUI on top of MongoDB.

Each of the sections is optional and do not depend on each other.

The assumption is that you have a running MongoDB server in your workshop environment and that you can connect to that instance through the Mongo Shell.

Additional requirements apply to each of the sections – a Node.JS and npm installation, a Java and Maven installation and a Compass installation.

All sources, instructions and the presentations can be found in this GitHub Repository: <https://github.com/lucasjellema/sig-nosql-mongodb> . You can fetch the contents of this repository as a zip file (<https://github.com/lucasjellema/sig-nosql-mongodb/archive/master.zip> ) or use a git client to clone the repository to your workshop environment.

## 1 Advanced Query – and a comparison between MongoDB Find and Aggregate vs. (Oracle) SQL

In this section, you will perform a number of query actions against a MongoDB database that contains two collections that are similar to data sets frequently used in samples and tutorials for relational databases – especially Oracle Database: Departments and Employees. You will perform a number of MongoDB search operations against these collections – and you can compare what you can do with MongoDB put next to (Oracle) SQL.

A presentation is available that presents 30 queries against MongoDB and Oracle SQL side by side: <https://github.com/lucasjellema/sig-nosql-mongodb/blob/master/mongodb-find-agg-vs-oracle-sql.pptx> - you can use this presentation for reference.

1. Load Data Set from CSV files into MongoDB  
     
   Directory hr-queries contains the data sets in two CSV files: export\_dept.csv and export\_emp.csv. Two helper files - empFields.txt and deptFields.txt – specify how the mongoimport utility should map the fields in each record to document properties in the MongoDB collection.  
     
   Using the following commands – running the monogimport tool that is installed in the bin directory of your MongoDB installation that also holds the server runtime and the shell – you import the datasets into collections emp and dept in database called hr. The database and the collections will be dropped (if they already exist) and (re)created through these commands.

mongoimport --host 127.0.0.1:27017 --db hr --collection emp --drop --file export\_emp.csv --type csv --fieldFile empFields.txt

mongoimport --host 127.0.0.1:27017 --db hr --collection dept --drop --file export\_dept.csv --type csv --fieldFile deptFields.txt

1. Run MongoDB Shell and Inspect Collections  
     
   Start the MongoDB shell as you have done before. Switch to database hr:

use hr

Verify whether the collections were created:

show collections

Check the statistics for collection emp:

db.emp.stats()

Check the contents for both collections:

db.emp.find()

db.dept.find()

1. Perform queries

To get started, let’s find the names of all managers:  
  
db.emp.find({"JOB":"MANAGER"},{ENAME:1})

And – find all salesmen, listed by salary descending; only show top 2  
  
db.emp.find({"JOB":"SALESMAN"},{ENAME:1, SAL:1}).sort({'SAL':-1}).limit(2)

1. The file sample-hr-queries.txt contains a number of queries, of increasing complexity, that you can try out to see what MongoDB can do in terms of retrieving information.   
     
   You will come across:  
   \* aggregation  
   \* join & lookup  
   \* nested collections  
   \* geo spatial search (cities closest to…)  
   \* text search  
   \* materialized views  
   \* stored procedures  
   \* facet search

Note: You can compare the MongoDB search commands with their SQL counterparts in the presentation: <https://www.slideshare.net/lucasjellema/comparing-30-mongodb-operations-with-oracle-sql-statements> .

## 2. MongoDB Compass

aaa

## 3. Development with NodeJS and MongoDB

aaa

## 4. Development with Java and MongoDB

aaa