Marcus Pinnecke, M.Sc.
Prof. Dr. Gunter Saake

Advanced Topics in Databases
Database and Software Engineering Working Group
Exercise Sheet 9

Magdeburg, June 14 2019

Summer Term 2019

A Gentle Introduction to Document Stores and Querying with the SQL/JSON Path Language

This is a one-week per-student sheet.

Prepare to present details of your solution during the tutorium.

Good Luck!

Task 3 **Hands on Document Stores** **4** (1 + 3) **Points**
(continued from last sheet)

Document stores are database system that store, retrieve and manage semi-structured data. This system class defines one of the main categories of modern NoSQL databases and is trending in popularity.

(…)

2. Download and setup *either* an instance of MongoDB and CouchDB on your system. Additionally, clone the *libcarbon* repository from GitHub[1], and checkout the branch `teaching/atdb/2019`. In this branch you will find the directory `ds/`, which contains excerpts of pre-processed datasets. The *GitHub Repository API Excerpt* dataset is the one you will work with.

   Create either a new database in CouchDB or a new collection in MongoDB for this dataset, and import the file `ds/github-repo-api/snapshot-excerpt.json`!

   Tip: For MongoDB look for a tool called `mongoimport` and use the flag `--jsonArray`

   Tip: For CouchDB you may want to import the dataset as a bulk, see http://docs.couchdb.org/en/2.3.1/api/database/bulk-api.html#inserting-documents-in-bulk for documentation. Further tip: the github dataset must be wrapped with some additional text to match CouchDBs importer syntax.

   Afterwards, implement the following queries in the database of your choice (either MongoDB or CouchDB):

   a. (…)

   b. Give the set of key names (1st level properties (no nested object), no duplicate key names) for all research papers stored in the database! You may use mapreduce for this purpose! Additionally, give your query statement.

[1] Type the following in your bash

```
$ git clone https://github.com/protolabs/libcarbon.git && cd libcarbon && git checkout -b
  teaching/atdb/2019 origin/teaching/atdb/2019
```

Task 4 Document Stores under the Hood **2** (1 + 1) **Points**

Despite differences in their data model, document stores share many conceptual ideas and internal structures with relational systems. However, due to a focus on scalability that heavily benefit from the denormalization of stored records, document stores typically have significant differences at storage level.

1. Consider the append-only storage of CouchDB, and the update-in-place storage of MongoDB. Explain how database modifications (inserts and updates) are handled, and discuss benefits and drawbacks of each approach **[Group 10]**.

2. Recap the default storage engine of MongoDB since version 3.2, WiredTiger. Explain where B+-tree structures are used **[Group 11]**.

Task 5 **Semi-Structured Data by Bits and Bytes** **9** (3 + 3 + 3) **Points**

Document records are physically organized by a variety of data formats, that result from a diversity of applications (such as fast parsability, understandability, or expressibility) .

1. Justify the statement "*There is no 'One-Size-Fits-All' format for representation of semi-structured data*" w.r.t. the diversity of application requirements, and discuss this statement for at least two formats not already mentioned in the lecture **[Group 12]**.

2. Clone the *libcarbon* repository from GitHub[2], and checkout the branch `teaching/atdb/2019`. In this branch you will find the directory `ds/`, which contains excerpts of pre-processed datasets. The *GitHub Repository API Excerpt* dataset is the one you will work with. Analyze the (disk) size requirements for the following formats on the GitHub Repository API dataset snapshot (that you must download from our FTP server) by converting the snapshot into each format, and compare them to the plain-text JSON format.

   (a) Universal Binary JSON (UBJSON)
   Tip: Find a library or tool under ubjson.org/libraries/ that matches your preferences

   (b) Binary JSON (BSON)
   Tip: Study the toolchain of MongoDB, which provides an export to BSON

   (c) Columnar Binary JSON (CARBON)
   Tip: Build `carbon-tool` from sources in the branch *teaching/atdb/2019* from our repository github.com/protolabs/libcarbon (see README.md), and run in your bash:

   ```
   $ build/carbon-tool convert --size-optimized --no-string-id-index github-repo-api.carbon
           ds/github-repo-api/snapshot-excerpt.json
   ```

   (conversion with `carbon-tool` may take XXX min for this dataset)

   Tip: Use a Linux distribution or macOS as operating system to match the building tools and tool chains.

3. Revisit your results from sub task 2 for each format (including the given plain-text JSON format). Speculate on the reason why your see differences in the file sizes for each format **[Group 15]**.

[2] Type the following in your bash
```
$ git clone https://github.com/protolabs/libcarbon.git && cd libcarbon && git checkout -b
  teaching/atdb/2019 origin/teaching/atdb/2019
```