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

Advanced Topics in Databases
Database and Software Engineering Working Group
Exercise Sheet 8

Magdeburg, June 7th 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 1 The Case for Document Stores? **7** (1 + 2 + 3 + 1) **Points**

Document stores manage denormalized records that must not typically match a predefined schema.

1. Name and describe at least two application scenarios that imply or require semi-structured data **[Group 6]**!
2. Imagine an application scenario where a schema can be defined up-front and does not change too much afterwards. Assume further that many entities have relationships to each other. Discuss the claim: “*A semi-structured data model is a reasonable choice for this use case*” **[Group 6]**!
3. Denormalization lead to the risk of low data integrity due to *data anomalies* resulting from data redundancy and update, insertion or deletion operations that do not affect each copy of the data. Discuss *update*, *insertion*, and *deletion anomalies* in context of semi-structured data **[Group 7]**!
4. List at least two use cases that benefit from semi-structured data despite the fact of data anomalies, and give one example each **[Group 7]**!

Task 2 JSON and the Document Database Model **3** (1 + 1 + 1) **Points**

The data model of document stores centers around records that are JSON-like.

1. Explain the terms *document* and *collection* (resp. database)! For documents, include statements on the schema, data normalization, object identification, nesting and referencing in your explanation. For collections, include statements on schema across records contained in the collection **[Group 8]**.

2. State whether the following statements are *true* or *false*! Give an explanation **[Group 8]**!

    a. JSON is a human-readable markup language similar to XML.

    b. JSON is designed for applications having unspecific knowledge of their data.

    c. JSON is not a general serialization format and language independent.

    d. JSON represents primitive, and structured data types.

    e. The string `[[{answer:"no"}]]` is a valid JSON file according to the latest specification.

    f. JSON is self-describing; there's no mechanism in JSON for schema verification.

3. JSON Pointers is a concept to enable references to specific value within a JSON document. Construct a minimal example for a JSON file for which the JSON pointer `/x/1/y/0/4` evaluates to a numeric value of `42` **[Group 8]**!

Task 3 Hands on Document Stores **4** (1 + 3) **Points**

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.

1. MongoDB and CouchDB are two prominent implementation for document stores. Compare both systems with respect to their storage engine design! Include the following in your comparison **[Group 9]**:

    a. Which concurrency control mechanism is used?

    b. What is about consistency, availability, and/or partition tolerance (cf. CAP theorem)?

    c. How is the support for mapreduce, filter operations, and further aggregation queries?

    Afterwards, explain the master-master architecture of CouchDB and the sharding architecture of MongoDB!

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. Give the value for the key `"html_url"` for the research paper having the property `"name": "Python"` by executing a database query! Additionally, give your query statement!

Note: Task 3.2 will be continued with another sub task (b) once we continued the lecture with MapReduce

[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
```