Project MongoDB Demo

Description

In this project, you will learn to use MongoDB as an example of a document-oriented NOSQL system, and see how data is stored and queried in such a system. You will also learn about the difference between storing data in a flat (relational) format versus in a document (complex object) JSON or XML format.

The input to your program will be data files in flat relational format (text files in .csv format comma separated values) for the COMPANY database from the textbook. The schemas for this data are the same as for the COMPANY database in the textbook in chapters 5 and 6 for the tables DEPARTMENT, EMPLOYEE, PROJECT, and WORKS_ON. You will need to design three document collections (complex objects) corresponding to this data and store each as a document collection in MongoDB:

1. The **PROJECTS document collection will store a collection of PROJECT documents. Each PROJECT document will include the following data about each PROJECT object** (document): PNAME, PNUMBER, DNAME (for the controlling DEPARTMENT), and a collection of the workers (EMPLOYEES) who work on the project. This will be nested within the PROJECT object (document) and will include for each worker: EMP_LNAME, EMP_FNAME, HOURS.
2. The **EMPLOYEES document collection will store a collection of EMPLOYEE documents. Each EMPLOYEE document will include the following data about each EMPLOYEE object** (document): EMP_LNAME, EMP_FNAME, DNAME (department where the employee works), and a collection of the projects that the employee works on. This will be nested within the EMPLOYEE object (document) and will include for each project: PNAME, PNUMBER, HOURS.
3. The **DEPARTMENTS document collection will store a collection of DEPARTMENT documents. Each DEPARTMENT document will include the following data about each DEPARTMENT object** (document): DNAME, MANAGER_LNAME (the last name of the employee who manages the department), MGR_START_DATE, and a collection of the employees who work for that department. This will be nested within the DEPARTMENT object (document) and will include for each employee: E_LNAME, E_FNAME, SALARY.

Your tasks for this project are as follows:

1. Install MongoDB on your computer.
2. Write programs to extract the data needed for the three document types above (PROJECTS, EMPLOYEES, and DEPARTMENTS) from the relational data files, and load these documents into the MongoDB system. Notice that the data from each document will be the result of a join on more than one file.

3. **Hints:** There are several ways to do this project. Here are some options (other options are also possible). **Option 1:** Load the data in a relational DBMS such as MySQL. Then write three different queries to extract the data needed for each document collection in relational format, and then convert the query result from the relational format into JSON for loading on MongoDB. But **you must create the nested structure within each object**. You can **use ORDER BY to order the result of the relational query by the nesting (root) attributes**, then you can create the JSON document in a single scan algorithm on the result of each query. **Option 2:** Load the data into MongoDB as a relational normalized data. Then write three different queries to extract the data needed for each document collection that creates new document collections to convert the query result from the normalized format into JSON for loading the nested structures in MongoDB. But **you must create the nested structure within each object**. **Option 3:** You can design your own method to create the nested document collections.

4. **Describe clearly which option you used to get the document collections of nested documents (objects).**

5. Write some MongoDB queries to retrieve some of the nested stored documents.