Course: INFO 579: SQL/NoSQL Databases for Data and Information Sciences
Project: 3
Topic: NoSQL Database Implementation and Querying

Instructions

In this project, you will work on migrating to MongoDB part of the database you designed in Project 1 and designing commands to retrieve information from it. You must create an aggregate boundary and denormalize your data accordingly. You must also produce the MongoDB operations required to create the collection of documents based on the resulting aggregates. In the second part of this project, you will have to design well motivated MongDB queries to retrieve information from the database.

You must complete the following tasks:

1. Design an aggregate boundary that involves at least 3 related tables from your database. Denormalize the tables accordingly to obtain a single collection that contains at least 5 documents with their corresponding fields and embedded documents. All documents and subdocuments must include all fields from the original database, including primary keys but discarding foreign keys. You must store the resulting collection as a valid BSON object into **collection.json**. Additionally, provide a short paragraph (**collection.pdf**) to motivate the aggregate boundary explaining its possible application within your business plan.

2. Develop the MongoDB statements required to create the collection of documents. You must present the statements in a single javascript script named **collection_creation.js**.

3. Develop a variety of MongoDB queries using the find method following the requirements for find queries on the next page. You must present the queries in a single javascript script named **find_queries.js**.

4. Develop a variety of MongoDB aggregate pipelines following the requirements for aggregate pipelines on the next page. You must present the pipelines in a single javascript script named **aggregate_pipelines.js**.

Every command should make sense within the logic of your application and must be preceded by a comment explaining its purpose.

Compress all the deliverables into a single zip file. Upload the file to the assignment of project 1 Dropbox in D2L. The name file must follow the format: **project03_groupcode.zip**

Achieving the requirements in each of the 4 tasks provides up to 22.5% of the final score, 90% in total. The remaining 10% can be obtained by including additional queries or pipelines based on their complexity. In task 1, the additional 10% can also be obtained by including additional entities or tables as embedded documents.

NOTE: Late submission will receive at most 70% of the maximum grade.

<u>Requirements for find queries</u>

1. Use the find method to implement at least 5 queries that include both a filter and a projection.
   a. At least one of the queries must use the $regex operator.
   b. At least one of the queries must use the $elemMatch operator.
   c. At least one of the queries must be on embedded documents.
   d. At least one of the queries must use the $ symbol to return the matching element of an array.
   e. At least one of the queries must use the $nor operator.

<u>Requirements for aggregate pipelines</u>

2. Implement at least 5 aggregate pipelines with at least 2 stages each (not counting $limit, $skip and $sort).
   a. At least one of the pipelines must use an $unwind stage.
   b. At least one of the pipelines must use a $group stage with accumulators.
   c. At least one of the pipelines must use a $group stage with the $push accumulators.
   d. At least one of the pipelines must use a $project stage that includes a $filter on an array.
   e. At least one of the pipelines must use a $project stage that reshapes the output document.