**ISTM 622**

**Reflection Paper**

**Praveen Gadugin**

**Texas A&M University**

**Project Description**

For this project, I was tasked with making an information system using a large amount of data. I had to use the same data for two different backends. I had looked at some of the data files provided on eCampus, but I didnot find any that I was comfortable with. Instead, I surfed the web in search of interesting .CSV files to use for our data. I found multiple interesting options, but the one that I ultimately chose was a flight cancellation database. The data included: flight names, cancellation times, airport information, etc. Using this data I was able to query the top flight routs with the most cancellations, which airports had the biggest effect from those cancellations, and how that affects those airports. Initially, I did this using MariaDB Galera and I coded the queries using SQL commands. All of the group members had experience in SQL in the past, so the queries were very familiar to the group. The second half of the project required us to use a NoSQL backend. I, as well as most of the class, chose to use a document database for our backend. The reasoning behind choosing to use a document database is explained later in this report. Only 1 group member had experience with NoSQL queries, but not with document database queries. For this half, I had to learn the queries using the textbook, in-class notes, Google searches, and Youtube videos. This took more time, but was very satisfying when I finally got the code right. For the front-end for both projects, I used Node.js and even manipulated our code to allow runtime user inputs.

**Project Accomplishments**

The time taken to load data in MongoDB was much lower than the time taken to load data in MariaDB, as the focus was primarily on loading a blob of data and not worrying about ensuring referential integrity as I did in case of MariaDB. Moreover, MongoDB performs asynchronous inserts and updates, and does not insert data into the database as soon as the query is processed. In other words, it implements a “fire and forget” operation where the database does not receive an immediate reply of the status of operation. All these characteristics of a non-transactional database like MongoDB makes write operation really fast as compared to writing to a transactional database like MariaDB.

**Project Challenges**

Accomplishing a group project can be challenging because it requires the effort and participation of every member in the group. I encountered several challenges while accomplishing the project goals.

The major technical difficulties I encountered in the project in part one was configuring the cluster and in part two was working with MongoDB which required extensive research owing to the technical skill gap. Cluster creation firstly required optimal RAM, processing power and hard disk which I struggled with owing to the system infrastructure limitations. Moreover, I debugged several networking errors while setting up the cluster and linking the nodes which took up a considerable amount of time. Transitioning from a SQL mindset to a NoSQL way of approaching databases was a major technology shift which required additional effort of the team. For us, constructing SQL queries was much faster than constructing NoSQL queries which required significant time investment in syntax research and query design. Nevertheless, we got an opportunity to learn MongoDB’s query language to design similar queries. In summary, the main challenges were as follows:

* Communication & Coordination
* Project rework
* Project debugging & fixes
* Conflict resolution

What went well was my ability to persevere through difficulties when I was not as knowledgeable on the syntax of the querying different results. I had no prior experience using Mongodb, but I did research on the subject and was able to figure it out.. By doing this, I was able to troubleshoot our problems and ensure that the queries worked correctly.

One thing that I would recommend changing for future classes would be to add more options for the data files on eCampus or suggesting data sources with different industry database. Another thing that I would change would be to provide additional resources for the front-end coding.

There were differences between using SQL and NoSQL databases. We enjoyed SQL better for this application since I was more experienced on that database. For us, constructing SQL queries were much faster than constructing NoSQL queries which required significant time investment in syntax research and query design. Nevertheless, we got an opportunity to learn MongoDB’s query language to design similar queries.

For NoSQL, as mentioned earlier, I used Mongodb as our backend. I believe we chose the right NoSQL backend because loading was extremely fast compared to mariaDB. As the database had no additional tasks of ensuring referential integrity I just had to load a plethora of data. There is also the preference for availability over consistency. MongoDB performs asynchronous inserts and updates wherein when an insert or update operation is performed, it doesn’t insert data into DB as soon as the query is processed. In other words, it is a “fire and forget” operation where database does not reply with the status of operation. All the characteristics of a non transactional database like MongoDB makes it really fast when compared to transactional database like Maria DB.

**Works Cited**

“Cursor.explain()¶.” *Cursor.explain() - MongoDB Manual*, 2018,

docs.mongodb.com/manual/reference/method/cursor.explain/#cursor.explain.