# Final Project Report

## Databases - CSC 3320 121

### Hudson Arney

### 29 April 2024

Data Model

Collections:

* actor
  + \_id
  + Actor\_ID
  + Name
  + Recording\_ID
* associated\_with
  + \_id
  + Recording\_ID
  + Actor\_ID
* category
  + \_id
  + Name
  + Category\_ID
* director
  + \_id
  + Name
  + Director\_ID
* rating
  + \_id
  + Name
  + Rating\_ID
* recording
  + \_id
  + Recording\_ID
  + Director
  + Title
  + Category
  + Image\_Name
  + Duration
  + Rating
  + Year\_Released
  + Price
  + Stock\_Count

The tables I created in addition to the given data were Director, Rating, and Associated With. I added these in lab 3 as they looked to reduce repeated finite information (each movie had a rating and director with a specific value, associated\_with is for looking at what actors appear in what movies). Each collection represents a different entity in the system, such as actors, directors, categories, ratings, and recordings. The fields within each document provide information specific to that entity, allowing for relationships to be established between different entities.

After adding the data in .json format to the MongoDB project, it automatically adds a \_id for each value, this gets used for lookups of the data.

Nothing has really changed about the data since lab 3 part 2. Although not every feature is used in the questions asked, I kept all information to keep consistency between the labs.

Results:

When opening my UI you should see this page after connecting to my database:

A screen shot of a computer

Description automatically generated

Question 1: List the number of videos for each video category.

A screen shot of a computer

Description automatically generated

Question 2: List the number of videos for each video category where the inventory is non-zero.

A screen shot of a computer

Description automatically generated

Question 3: For each actor, list the video categories that actor has appeared in.

A screen shot of a computer

Description automatically generated

…

A black screen with colorful lines

Description automatically generated

Question 4: Which actors have appeared in movies in different video categories?

A screen shot of a computer

Description automatically generated

A black background with colorful lines

Description automatically generated

Question 5: Which actors have not appeared in a comedy?

A screen shot of a computer

Description automatically generated

…

A screenshot of a computer screen

Description automatically generated

Question 6: Which actors have appeared in both a comedy and an action adventure movie?

A screen shot of a computer

Description automatically generated

Question 7: Which actors have appeared in both multiple ratings and categories?

A screen shot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated

Bonus – Question 8: Which directors have also been actors?

A screen shot of a computer

Description automatically generated

Analysis

Benefits of MongoDB:

1. Cloud Storage – You don’t have to store the data on your local device. For this lab it was not as much of a problem, but if you were dealing with gigabytes to terabytes of data, hosting the data on you local machine starts to make less sense.
2. Speed of Access – This applies to both the physical side of coding and the information access. Once you get all the necessary code running, the queries are very fast to run. Also much of the code can be repeated/reused when looking at each query by making a pipeline. I wrote my code in python which is what I am most comfortable and familiar (more than SQL).

Drawbacks of MongoDB:

1. Data Consistency – MongoDB prioritizes availability and partition tolerance over strict consistency, which makes it harder to follower stricter ACID compliant structures. This could lead to issues with querying in a more complex database (this lab wasn’t as bad).
2. Learning Curve – Transitioning from SQL databases to MongoDB took awhile to get the hang of. I was having many weird issues at the beginning that I couldn’t explain. Once you get the necessary setup however (mostly by following the instructions in Lab 7) it made connecting to the database pretty easy.
   1. I also had to “translate” the SQL queries I had to fit the MongoDB structure which was a new part to learn and took a bit of time. Thankfully I have done work with json files in python before.
3. Still more to learn – In this lab I just uploaded my json formatted data to the UI online. I never used the mongosh to insert or write data. That would take me a bit more time to get the hang of and understand. Not sure if it is similar to command line usage in MySQL.

Code can be found in the final\_script\_databases.py in the zip file.