**INF0 6210 Data Management and Database Design**

**NoSQL MongoDB Assignment 3**

Ashwin Lakshman 001353233

1. **Domain & Design Approach**:

In this assignment, I have continued to use the Movie data scraped from the IMDB website and Twitter API. Since the generated CSV files are Normalized and fragmented, first, the tables are de-normalized and combined. It is then converted into JSON objects which are then inserted into the MongoDB movies Database.

1. **Sample Data:**

The starting database consists of a total of 8 CSV Files:

* 1. **Movies CSV File:**

|  |
| --- |
|  |

* 1. **Producers CSV File:**

|  |
| --- |
|  |

* 1. **Movies-Producer Linking CSV File:**

|  |
| --- |
|  |

* 1. **Genres CSV File:**

|  |
| --- |
|  |

* 1. **Movies-Genres Linking CSV File:**

|  |
| --- |
|  |

* 1. **Twitter Users CSV File:**

|  |
| --- |
|  |

* 1. **Tweets CSV File:**

|  |
| --- |
|  |

* 1. **Tweet-Movie Linking CSV File:**

|  |
| --- |
|  |

1. **De-Normalized Data**

|  |
| --- |
| * 1. **Movies Data** |
|  |

|  |
| --- |
| * 1. **Tweets Data** |
|  |

1. **Implementation:**

|  |
| --- |
| * 1. **The Normalized CSV files are added into Dataframes** |
|  |

|  |
| --- |
| * 1. **The DataFrames are added into the SQL Database** |
|  |

|  |
| --- |
| * 1. **The Tables are De-Normalized and combined using LEFT JOIN** |
|  |

|  |
| --- |
| * 1. **Converting the Resultant tables into CSV Files** |
|  |

|  |
| --- |
| * 1. **Reading the CSV Files into the DataFrame** |
|  |
|  |
| * 1. **Converting the DataFrames to JSON** |
|  |
|  |

|  |
| --- |
| * 1. **Inserting the JSON objects into the respective MongoDB Collections** |
|  |
|  |

1. **QUESTIONS ANSWERED**
   1. **What are tags are associated with a person, place or thing?**

**Interpretation: Which users have most tags for a specific movie?**

|  |
| --- |
|  |

The output displays the USER\_ID followed by the Count of tags for the movie “Mission Impossible”

* 1. **What social media users are like other social media users in your domain?**

**Interpretation: To find all User\_ids with similar Comments(Using REGEX)**

|  |
| --- |
|  |

The output displays all the User\_Ids which have commented about “Mission Impossible”

* 1. **What people, places or things are popular in your domain?**

**Interpretation: Which movie has the most number of tweets?**

|  |
| --- |
|  |

The output displays the list of movies with the most number of tweets.

* 1. **What people, places or things are trending in your domain?**

**Interpretation: Which Production Company has the most number of movies in Recent Years**

|  |
| --- |
|  |

The output displays the production companies making the most movies in recent years.

1. **USE CASES:**

|  |
| --- |
| * 1. **Who Directed this Movie?** |
|  |

|  |
| --- |
| * 1. **What are the top rated movies?** |
|  |

|  |
| --- |
| * 1. **Which are the most active User\_ids?** |
|  |

1. **References:**
   1. <https://api.mongodb.com/python/current/>
   2. <https://github.com/mongodb/mongo-python-driver>
   3. <https://studio3t.com/knowledge-base/articles/sql-to-mongodb-migration/>
   4. <https://stackoverflow.com/questions/52427002/remove-n-from-values-fields-results-in-mongo-db>