A Data Engineering Practicum Project Report

Submitted March of 2022 to the

Anderson College of Business and Computing

Regis University

in partial fulfillment of the requirements for a

Master of Science in Data Science with a Specialization in Data Engineering

by

Lacey Conrad

Practicum Project Advisor

Bob Mason, Ph.D., M.B.A.

**Table of Contents**

Table of Contents……………………………………………….…………………………………ii

Chapter 1-Introduction…………………………………………………………………………….2

Chapter 2 – Application Components………………………….……….…………………………3

Chapter 3 – Results………………………….……….……………………………………………4

Chapter 4 – Conclusions……………………………………….……….…………………………4

**Chapter 1-Introduction**

**Overview**

The MSDE692 and MSDE696 – Data Engineering (DE) Practicum courses satisfy the Graduate Final Project requirement for graduation from the Master of Science in Data Science with a Specialization in Data Engineering. A total of 36 credit hours of course work is required to graduate from RU with a Data Science degree. These courses are taken as the last courses within the program and each student is required to participate in the Data Engineering Practicum for two eight-week terms to earn the Specialization in DE along with the other four DE courses. The goal of the Data Engineering Practicum project is to facilitate a real-life hands-on learning experience via the development of data infrastructure for use by Data Scientists. Each student selects their own DE project to customize. The Database Practicum Project allows students to demonstrate their design and coding skills that they have acquired throughout their course work within the MS in DS - Specialization in Data Engineering.

**Research Statement**

The purpose of this research is to create a data pipeline that uploads genomic datasets either from a user’s computer or the internet into Neo4J. A specific type of genomic data will be focused on (to be determined during research), and will initially be modeled into a graph structure after identifying nodes and vertices. This model will then be used to create a Neo4J database. Additional insight will be provided on the feasibility of database changes including adding or deleting data, and changing the structure to reflect new information. The primary goal of this research will be to create a Neo4J database comprised of genomic data that can be queried in a manner reflecting potential research needs.

**Background**

Much data in the biological sciences are interrelated. In particular, genomics data represents a multitude of connections, interactions, patterns, and classifications. In short, it is data not suited to traditional relational databases, where queries typically require numerous join operations. Graph databases however more accurately reflect the interrelated nature of data in genomics. Nodes, in this case representing genes and their associated functions, are stored with their observed relationships to other genes. This could include any relationships that exist between the genes, such as when the expression of one gene influences another gene or two genes are expressed together. While the potential for using graph databases to this end has been noted, graph databases have not been readily accepted by the genomics community, largely due to a lack of database experience. Therefore it makes sense to create an easy-to-use pipeline that collects data from local computers and, when, necessary, websites storing genome data and stores it in a graph database. Making this process easy while also promoting the use of a graph database where it could be highly beneficial will genomics researchers to answer more complex questions quicker than by a traditional relational database.

**Deliverables Statement**

The deliverable will be a model representing how genomic data could be represented in a graph database. Additionally, a part or whole of a pipeline allowing for data upload to Neo4J will be included.

**Chapter 2 – Technical Components**

The NoSQL graph database Neo4J will be used for data storage. R and possibly Python will be used to develop parts of the pipeline when needed. Other languages may be used as research progresses.

Research Plan:

1. Understand gene expression/population genetics dataset: variables and relationships.
   1. Download csv file and explore/clean in R.
   2. Convert file to csv if needed .
      1. Future: make sure code can convert any common file formats to csv (or JSON, depending on data).
   3. Save the resulting csv file for uploading into Neo4J.
2. Document relationships between variables in R and create a model of what a gene expression network (or population genetics model) would look like.
   1. Use csv file in R to understand what the nodes/relationships/properties would be
      1. Future: Do this for Python.
   2. Create a hypothetical graph model similar to what would be created in Neo4J
      1. Are there multiple ways this system can be modeled?
3. Learn how to upload and recreate the model on Neo4J.
   1. Learn Cypher
      1. Try uploading a simplistic dataset first, possibly following a tutorial.
      2. Perform several queries using Cypher
   2. Upload genome expression data (1-2 animals? 1-2 genes? I need to figure out the scale for the beginning upload(s))
4. Determine how well it represents the system.
   1. Does my model need to be adjusted?
   2. Is important information missing in the model that may change the outcome of the analysis?
5. Investigate adding more data to the database.
   1. Learn how to incorporate additional data into the established database
      1. Adding more nodes.
      2. Adding more/new relationships.
      3. Adding/changing properties.
6. Use NCBI website API to GET data into R or Python
   1. Get a version of gene data into R or Python for data cleaning and to structure for upload into Neo4J.

**Chapter 3 –Results**

TBD

**Chapter 4 – Conclusions**

TBD