AIM 5001 Module 13 Assignment

PostgreSQL to Neo4j Data Migration

***** You may work in small groups of no more than three (3) people for this Assignment *****

Your task for the **Module 13 Assignment** is to migrate content from a PostgreSQL database to a Neo4j database, and then retrieve specific components of that data from Neo4j via Neo4j's **Cypher** query language. Specifically, you will export the entire content of the **Artist, Album, Track, Genre** and **MediaType** tables from the PostgreSQL **Chinook** database we've worked with in Modules 1 & 2, load that data into a Neo4j database and then subsequently construct and execute queries to retrieve specific Chinook content from your new Neo4j graph database. Note that **all of your Neo4j/Cypher work for this Assignment should be performed using Neo4j Desktop / Neo4j Browser software**.

You will start by exporting the required data from Postgres to CSV files. Next, you will define an appropriate Neo4j graph database schema for effectively organizing/managing the Chinook data within a Neo4j database. Then, you will construct the Cypher commands needed to implement your proposed graph database schema as you load the Chinook data from the CSV files obtained from Postgres into a new Neo4j database. Then, using your Neo4j data retrieval expertise, you will construct a set of Cypher queries to retrieve specific Chinook content from within your new Neo4j graph database.

Your data migration and retrieval solutions need to be 100% reproducible. The specific tasks you are required to implement for this assignment are explained below.

Your deliverable for this Assignment is a Jupyter Notebook. It should contain a combination of PostgreSQL commands/queries, Cypher commands/queries and explanatory narratives contained within **properly formatted Markdown cells**. The Notebook should contain (at a minimum) the following sections:

1) **Introduction (5 Points)**: Summarize the problem + explain the steps you plan to take to address the problem

2) **PostgreSQL to Neo4j Migration (55 Points)**: The steps required for migrating content from our Chinook Postgres database to a graph database are as follows:

   First, construct and execute the PostgreSQL commands necessary to export the content of each of the **Artist, Album, Track, Genre** and **MediaType** tables from Postgres to a separate CSV file (i.e., one CSV file for the content of each table). This tutorial explains one way in which to export data from a Postgres database table to a CSV file: https://www.postgresqltutorial.com/postgresql-tutorial/export-postgresql-table-to-csv-file/. Be sure to provide a short written narrative explaining your approach to extracting the required data from your Postgres database.

   Next, **define an appropriate graph database schema** to be applied to the data you have exported from Postgres, i.e., specify any **nodes, labels, and relationships** you believe are required for purposes of enabling the efficient retrieval of data from your new graph database (**NOTE**: The M13 content within canvas related to migrating content from an SQL database to a graph database may prove to be helpful, e.g., what type of relational database entity is analogous to a node? To a relationship? To a property?, etc. And think carefully about the graph relationships that will be necessary for purposes of enabling the same types of searches that were possible within the original relational database). When developing this schema, be sure to include the specification of any **indexes** or **constraints** that will be required to maintain graph database performance and data integrity. These tutorials (provided within the M13 assigned readings) may prove to be helpful for this task:

- https://neo4j.com/developer/relational-to-graph-modeling/
- https://neo4j.com/developer/guide-importing-data-and-etl/#northwind-graph-model

Your written narrative for this section should include **a clear, concise explanation of your schema**.

Next, within **Neo4j Desktop**, create a new project named **AIM 5001**. Within that Project add a new "Local DBMS" named **M13 Assignment**. Then, **start** the M13 Assignment database from within your Neo4j Desktop interface. If needed, a user manual for Neo4j is provided here: https://neo4j.com/developer/neo4j-desktop/.

Next, from within **Neo4j Desktop**, activate the **Neo4j Browser** application. (see https://neo4j.com/developer/neo4j-browser/ if needed).

Then, you will **construct and execute the Cypher commands needed to implement your proposed graph database schema** while simultaneously loading the Chinook data from the CSV files obtained from Postgres into your new **M13 Assignment** database. The following tutorials provide examples of how to use Cypher's LOAD CSV command to instantiate new nodes and their associated properties using the content of a CSV file as input:

- https://neo4j.com/developer/desktop-csv-import/ (see the section titled "Cypher")
- https://neo4j.com/docs/cypher-manual/current/clauses/load-csv/#load-csv-import-data-from-a-remote-csv-file

You should **construct/execute a separate LOAD CSV command from within Neo4j Browser** for each of the CSV files you've obtained from Postgres, with the syntax/content of each LOAD CSV command being tailored to match the content of the CSV file you are loading into Neo4j.

When constructing your LOAD CSV commands, be sure to pay close attention to the required syntax utilized within the Cypher Merge and Create commands shown within the tutorials cited above: As you can see, new nodes are being created from each line of a CSV file, and the Cypher queries are constructed so as to use specific fields from each line as specific node property values while also assigning an appropriate **label** to each node.

Next, construct and execute the Cypher commands needed to implement the **relationships** you specified within your graph database schema. Remember, when specifying relationships within a graph database, we need to consider the directionality of the relationship as well as whether there are any properties that we want to include as part of any given relationship. Furthermore, the relationships should make sense intuitively, e.g., within the context of the Chinook data, think about how artists, albums, tracks, genres, and media types are related to one another.

After establishing the relationships, you can use Neo4j Browser to visualize and interactively navigate through your new graph database. **Comment on whether the visual graph navigation behaves the way you anticipated it would**. Also, provide a screenshot of the visualized representation of your new graph.

This section should include an appropriate explanatory narrative that explains your approach to each of these tasks, including how you have specified that the Chinook data be stored within Neo4j, e.g., a description of each type of node you are creating, the properties pertaining to each type of node, any labels you are applying to the nodes, a description of each relationship you are creating (including their directionality), and the properties (if any) that you are associating with any relationships you are creating. This section should also include the Cypher commands you've used to load the Chinook data from the CSV files into your new Neo4j graph database. Your narrative should also clearly explain the indexes and constraints you are implementing within your new graph database for purposes of maintaining graph query performance and data integrity.

3) **Using Cypher to Retrieve Data from Neo4j (40 Points)**: Using Neo4j Desktop and Neo4j Browser, construct and execute Cypher queries that provide the required graph database output as per the query specifications cited below. Using formatted Markdown cells within your Jupyter Notebook, for each of the required queries listed below you should provide the Cypher command you have used to retrieve the requested data, a copy of the results of the query, AND provide a brief explanatory narrative describing why the query you've specified is able to retrieve the required data (i.e., explain the logic behind the structure of your graph database query).

   The required graph database queries are as follows:

   a) Write and execute a Cypher query that returns all Tracks from the 'Jazz' genre composed by 'Miles Davis'

   b) Write and execute a Cypher query that returns all Artists that have any Tracks available in the 'AAC audio file' media type.

   c) Write and execute a Cypher query that returns the Artist associated with the album 'Bongo Fury'.

   d) Write and execute a Cypher query that returns all Tracks from the album 'Coda' by the artist 'Led Zeppelin'.

   e) Write and execute a Cypher query that returns all Albums that contain Tracks composed by 'Alanis Morissette & Glenn Ballard'

   f) Write and execute a Cypher query that returns the names of all Albums containing Tracks for which no Composer has been specified.

Your Jupyter Notebook deliverable should be similar to that of a publication-quality / professional caliber document and should include clearly labeled graphics, high-quality formatting, clearly defined section and sub-section headers, and be free of spelling and grammar errors.

Upload / submit your Jupyter Notebook within the provided M13 Assignment Canvas submission portal. Be sure to save your Notebook using the following nomenclature: **first initial_last name_M13_assn**" (e.g., J_Smith_M13_assn). ***Small groups should identity all group members at the start of the Jupyter Notebook and each team member should submit their own copy of the team's work within Canvas.***