Practice Lab on
Graph Databases with Neo4j and DuckDB

G. Raschia

Last update: November 11, 2025

Preamble

- Group in pairs of students.
- From your own laptop with Eduroam Wifi network, it requires to set up proxy in the Browser: `http://proxy-etu.polytech.univ-nantes.prive:3128/`.
- Resources and help:
  - official CYPHER V5 documentation
  - CYPHER Cheat Sheet

Warm Up

(i) Open Neo4j AuraDB Browser: `https://console.neo4j.io/` ⟩ log-in. Alternatively, open Neo4j Desktop 2 application on your laptop or Neo4j Desktop on School workstations.

(ii) Create a fresh new DB instance (delete the previous one if any) and Open in Query Editor.

(iii) Import the movie database: process the CYPHER script from the Movie Graph GitHub repository.

Try out query:

```
neo4j$ MATCH (n)
       WITH COUNT(n) AS numVertices
       MATCH ()-[e]->()
       RETURN numVertices, COUNT(e) AS numEdges
```

Expected result:

numVertices	numEdges
171	253

(iv) Try out query:

```
neo4j$ MATCH (n) RETURN n
```

(v) Show meta-graph (schema of the movie database):

```
neo4j$ CALL db.schema.visualization()
```

Tips

- Save Neo4j queries in Favorites (bookmark symbol) to export all at once in the end
- Save utility scripts like the one to create the database
- Write Neo4j query on multiple lines: `Shift+Enter`

Add Data

1. Add movie, actors (three main characters), director as *vertices* and `ACTED_IN`/`DIRECTED` *edges* for the movie "The Bridges of Madison County" (see IMDb entry).

   Follow the actor's links to get info. *Annie Corlay* was born in 1960.

   Do not insert vertices that already exist in the database!!

   In case of emergency, one could blank the database and restart from scratch:

   ```
   neo4j$ MATCH (n) DETACH DELETE n;
   ```

   and re-import the movie database.

Graph Patterns

2. Find all actors that directed a movie they also acted in and return actor and movie nodes.
3. Find the name of the actor who played *DeDe* in the movie "Joe Versus the Volcano".
4. Find the director's names of movies where *Madonna* and *Tom Hanks* were co-actors.
5. Find all actors that acted in a movie together after 2009 (release date, 2009 included) and return the list of actor names along with the movie node.

   Hint: `COLLECT(.)` aggregate function

6. By extending the previous query, find all movies where the cast of the movies found before also acted in. For instance, if $C(m) = \{a_1, a_2\}$ is the list of actors (the cast) of a movie m in the previous answer, then, one seeks for movies m' such that the cast $C(m')$ contains $C(m)$.

Path Expressions

7. Match pairs of reviewers and those they follow directly or via a third reviewer.
8. Count the number of more-than-70-years-old actors reachable in at most 4 hops starting from *Clint Eastwood* and following the `ACTED_IN` relationship through movies. For instance, a valid pattern is

   $$\text{clint} \rightarrow m_1 \leftarrow p1 \rightarrow m_2 \leftarrow p_2$$

   with m_i the movies and p_i the elders. p_1 is 1-hop from clint, whereas p_2 is 2-hops. Each p_i of the chain is born before 1954. This is "the elder network".

Adjacency Lists and Distribution

9. Return the whole graph as a simple adjacency list of vertex ids (`id(.)` scalar function) ordered by decreasing vertex degree. Don't forget nodes without outgoing edge.

Centrality

10. Find actors with top 10 – sort of – KATZ centrality along `ACTED_IN` edges (through movies)

$$\text{Centrality}(v) = \sum_{i=1}^{\infty} p_i \cdot w_i(v)$$

where v is a vertex, w_i is the number of walks of length i starting from v, and p_i is a distance penalty reciprocal of path length (e.g. 3-hop neighbor gets a penalty of 1/3).

Return actor vertex and KATZ centrality.

SQL

In the following, you will set up and query a DuckDB database, either with the DuckDB CLI and its awesome `-ui` option, or with your favorite SQL Client/IDE, or even with a temporary DuckDB Shell.

You may refer to the official DuckDB documentation if needed.

11. Find a way to create and populate a new DuckDB database from the Neo4j movie graph database.

    Hint: *export* feature of CYPHER query answers.

    Hint: `startNode(.)` and `endNode(.)` scalar functions.

    Hint: DuckDB allows for considering CSV files as regular tables in SQL queries!

    Provide CYPHER queries, DuckDB commands, CSV files and any other required material to achieve the task.

12. Redraw in SQL the series of questions, except Q1 (add data) and Q7 (not applicable).