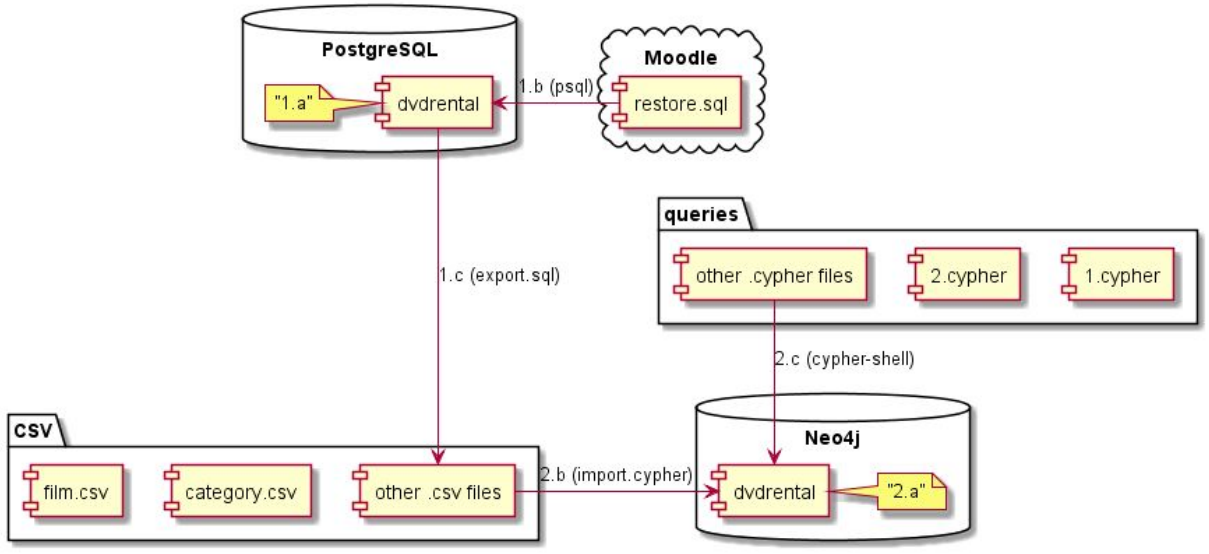
DMD2 S20 Assignment 1 Report

By Sergey Semushin (BS18-05)

For this assignment I have chosen Neo4j as a database to transfer to. Everything is tested only with `PostgreSQL 12.2`, `Neo4j Desktop 1.2.4`, `Neo4j Server 4.0.0 Enterprise` and `Cypher-Shell 4.0.1`

Moving the database

1. Working with postgresql
    a. Create postgre database `dvdrental`
    b. Importing SQL dump from moodle to database (change paths inside the file first)
    `psql -U postgres -h localhost -p 5432 -d dvdrental -f ".../restore.sql"`
    c. Export to CSV files using `export.sql` (change paths inside the file first)
    `psql -U postgres -h localhost -p 5432 -d dvdrental -f ".../export.sql"`
2. Working with neo4j
    a. Create a neo4j database `dvdrental` with password `password`
    In config file, change `dbms.directories.import` to the directory containing exported CSV files OR copy exported CSV files to default import folder (`<NEO4J_HOME>/import/`)
    b. Import to Neo4j using `import.cypher`
    `cypher-shell -u neo4j -p password -f ".../import.cypher"`
    c. Execute queries
    `cypher-shell -u neo4j -p password -f ".../queries/n.cypher"`



Adjustments that were necessary for the new database

1. Neo4j has no “enum” type, so I had to leave the `film.rating` field as a text
2. Neo4j can not store the binary data, so `staff.picture` field may be converted to unique id for each picture and pictures itself can be stored in file system
3. For fulltext search in Neo4j, corresponding index can be used, so field `film.fulltext` is not needed
4. Pure Neo4j does not support triggers, so `last_update` field would have to be handled in insert queries
5. Foriegn keys are represented as relations in Neo4j, so corresponding indexes are not needed