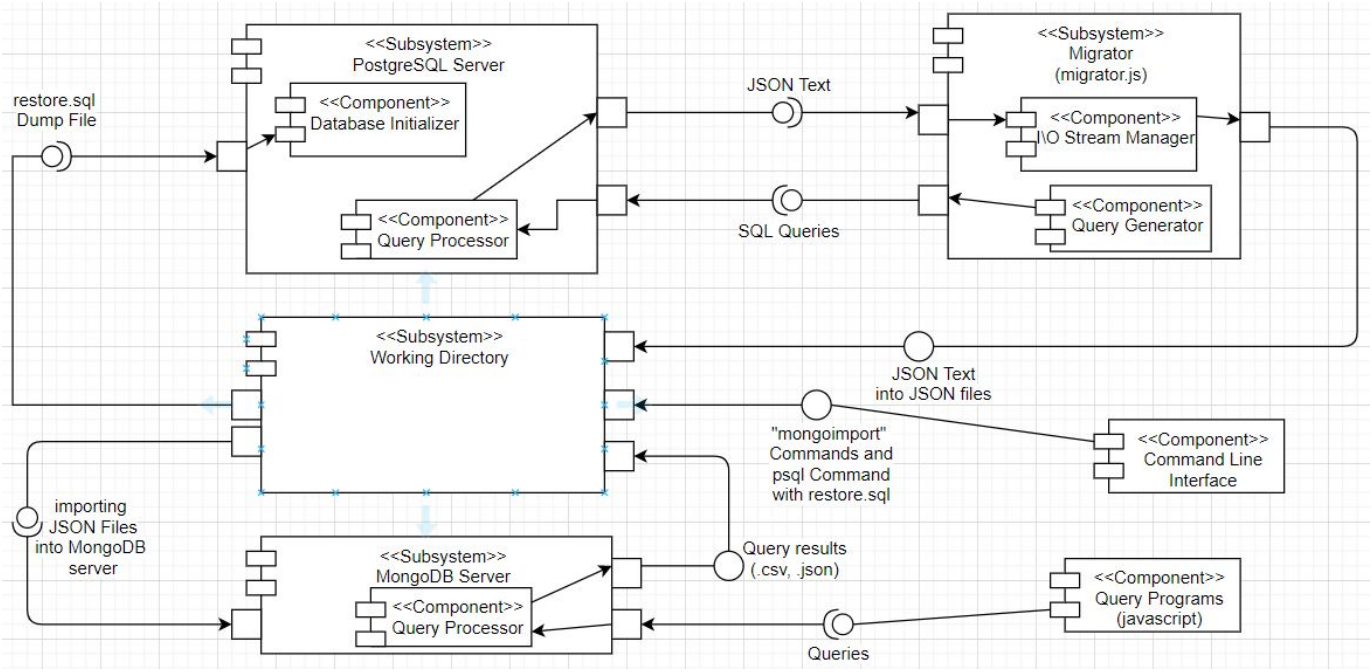
DMD 2 Assignment № 1

Danat Ayazbayev, BS-18-03 March 24, 2020

Component Diagram



The Process of Moving the Database

1. I created and filled PSQL **dvdrental** database by manual given in **README.txt**.

2. I wrote a script **migrator.js** which generated nested SQL queries for each relation(table) and exported query results into .json files (e.g. see **film.json**). SQL queries generated nested JSON text.

3. I write to command line **mongoimport** commands for every generated .json file. For example:
mongoimport –db dvdrental –collection film –file film.json –jsonArray
By such commands I imported all previously generated .json files into mongoDB collections.

4. I wrote five queries (e.g. **query1.js**, etc.) which somehow queried mongoDB and retrieved and put data into other .json or .csv files (e.g. see **query1.json** or **query2.csv**).

Adjustments for new database

1. Analogue of table connections are .json files with nested JSON text which are imported in mongoDB.

2. Dates are converted to pure strings, which are to be processed in query scripts.

Performance

Migrator: node migrator.js

Execution time (from battery): 42 seconds
Execution time (from circuit): 12.5 seconds

Query 1: node query1.js

Execution time (from battery): 6.5 seconds
Execution time (from circuit): 1.8 seconds

Query 2: node query2.js

Execution time (from battery): 2 seconds
Execution time (from circuit): 0.6 seconds

Query 3: node query3.js

Execution time (from battery): 5.1 seconds
Execution time (from circuit): 1.5 seconds

Query 4: node query4.js

Execution time (from battery): 5.2 seconds
Execution time (from circuit): 1.5 seconds

Query 5: node query5.js

Execution time (from battery): 1.6 seconds
Execution time (from circuit): 0.4 seconds

Comments

During work of laptop from battery, hardware components try to limit power consumption. According to observations, queries' execution time mostly depends on hardware load, because when battery is connected to the circuit, queries' execution time becomes around 3-4 times less.
Another thing to add is that queries' execution times could be optimized in case of real functioning software but it will need more lines of code. Also optimization can be reached by including additional collections to the database. For example, from collection **rental** we can find **film_id**. But according to **rental.json**, it is triply-nested JSON to find **film_id**. So we can create additional collection, whose correspondent .json file will be just one level JSON.