## Task 1 (parser)

There is a text file containing a database dump of the following entities:

* ExchangeOffice - currency exchange office.
* Rate - currency pair. The rate remains constant until changed over time.
* Exchange - currency exchange transaction. Ask is the price at which the client buys the currency in the exchanger. Bid value (the price at which the client sells the currency in the exchanger) is missing in our data but can be calculated by extracting the rate data.
* Country (name and code).

Here’s an example of the file structure:

```

exchange-offices

exchange-office

id = 1

name = Exchanger 1

country = UKR

exchanges

exchange

from = EUR

to = USD

ask = 110

date = 2023-04-24 22:55:33

exchange

from = USD

to = UAH

ask = 400

date = 2023-04-24 22:55:33

rates

rate

from = EUR

to = USD

in = 1.1

out = 1

reserve = 120000

date = 2023-04-24 22:55:33

rate

from = USD

to = UAH

in = 1

out = 40

reserve = 150000

date = 2023-04-24 22:55:33

exchange-office

id = 2

name = Exchanger 2

country = UKR

rates

rate

from = AUD

to = CAD

in = 1

out = 2

reserve = 150000

date = 2023-04-24 22:55:33

countries

country

code = UKR

name = Ukraine

```

As you can see, it uses 2 spaces for indenting nested objects / props.

You need to create a NodeJS application that imports this file into the PostgreSQL database.

## Task 2 (API request)

Get the program you created to accept incoming API requests. In response, it must output the top 3 currency exchangers ordered by profit value over the last month for each of the three countries with the highest profit. Profit for the country is the total profit of all exchangers belonging to it. The profit of the exchanger is the total profit throughout all currency pairs for a certain period, expressed in the base currency (USD) at the rate that was relevant at the moment of the exchange.

For example, on April 24 at 21:00 in exchanger 1 there was a transaction EUR -> UAH with 4000 UAH ask, and an hour later 50 EUR was changed in the opposite direction. The bid value is unknown yet, but it can be calculated by finding the current exchange rate for EUR->UAH at that moment (e.g. 40:1) and dividing the ask by it (e.g. 4000 UAH / (40 / 1) = 100 EUR). Now that you calculated the bids you find out that the exchanger sold 4000 UAH for 100 EUR and bought 50 EUR for 1900 UAH from 21:00 till 22:00 on April 24. Converting 4000 UAH, 100 EUR, 50 EUR and 1900 UAH into USD at rates that were at the time of the exchanges (respectively) you’ll get something like 112 USD -> 110 USD, 61 USD -> 60 USD, and this $3 of the difference make up the currency pair profit during the specified time.

Note: You may assume the existence of a B->A pair in the database if there is an A->B pair, as well as the existence of X->USD and USD->X pairs for any currency X. All calculations must be performed by a single query to the database.

## Questions

// Write down your answers in the README.md file at the root of your project folder.

1. How to change the code to support different file format versions?
2. How will the import system change if in the future we need to get this data from a web API?
3. If in the future it will be necessary to do the calculations using the national bank rate, how could this be added to the system?
4. How would it be possible to speed up the execution of requests if the task allowed you to update market data once a day or even less frequently? Please explain all possible solutions you could think of.