AWS Lambda + NodeJS Skillset Assessment

Dear Interviewer,

Thank you for taking the time to complete our evaluation assignment for the **NodeJS Software Engineer (Remote)** position at Universium.

Before you jump in, let us share our mission.

Universium is a team of researchers, designers and engineers with a simple goal in mind: to invest our time and energy into building solutions that have the potential to steer the world we will live in tomorrow.

It takes a lot to keep us moving. Each Universium member must have a **great motivation** and **desire** to **learn & grow**. To build great things, we must be **independent**, **thoughtful** and **inventive** in each task we undertake. We must show leadership in every aspect of our work.

This is a big ask, yet we believe that you have the potential to become one of us on this journey. And this assignment is here to see if you are willing to take a step in this direction.

We expect you to take the pride of ownership in every choice you make while implementing this micro-project. Consider, what would it take to build a large-scale solution based on this scenario? What would be the necessary steps to make sure that the code you write could be easily maintained and expanded? How could you make the job easier for those who come to work on this project after you?

With this, we wish you the best of luck! This assignment should not introduce any significant challenges, yet should you have any questions, feel free to follow-up with your recruiter to ask us any questions.

We are looking forward to reviewing your implementation and hope you see you among us soon!

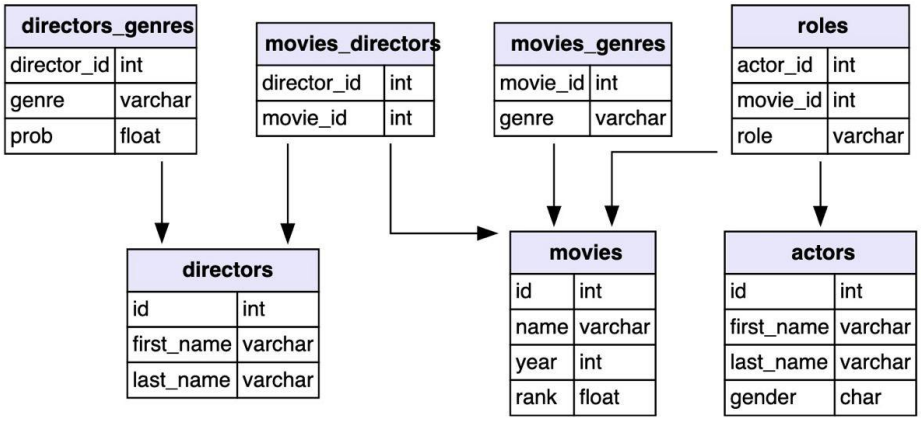
Best regards,
Universium Founders

You are given a simplified movie dataset called “IMDB” (use this link to download the dataset). Your objective is to use provided data to:

1. Implement SQL schema creation script (see schema diagram below);
2. Implement a tool to import raw data from a root subfolder **‘data’** to a database;
   Advice: this script should be launched from the **terminal** with an input parameter specifying path to directory with necessary files.
3. Create a new trial AWS account, or use an existing one
4. Implement an API usign AWS Lambda with endpoints capable to provide the following functionality

   - `GET /movies`: returns a collection of movies using the following parameters
     - `page`: page index number starting from 0
     - `size`: number of results per page max 100
     - `director`: filter movies by director
     - `genre`: filter movies by genre
   - `GET /actor_stats/{actor_id}`: consider & implement possible approaches to optimizing this endpoint performance. Should return the following data schema:
     - `id`
     - `name`
     - `top_genre`
     - `number_of_movies`
     - `number_of_movies_by_genre`: dictionary, key: genre_name, value: number
     - `most_frequent_partner`: find actor that acts in the same movies most frequently
       - `partner_actor_id`
       - `partner_actor_name`
       - `number_of_shared_movies`

Data Schema:



Requirements:

- Up to 2 days to deliver your implementation
- AWS SAM or Serverless Framework
- PostgreSQL or MySQL
- [*Optional*] Consideration of useful frameworks
- [*Optional*] Implementation of unit tests
- [*Optional*] Set up Swagger API definitions (will be a **HUGE** plus)

Delivery Instructions:

1. Create & publish your codebase to a **private** GitHub repository (*don't forget .gitignore!*);
2. Share access to this repo with: stepanmazokha, vladrudych, provide read permissions;
3. Notify your Universium recruiter about completion.