Analytics Engineer Exercise

Overview:

Brightwheel combines data from a variety of data sources to enable searching for the deepest insights possible. Some of those data sources are internal, like our production database, and some are external. The goal of this exercise is to combine multiple data sources for California Child Care Providers into one for deeper and easier analysis. The exercise is meant to assess data pipeline coding ability.

Expected Time:

Please limit your time to **2 hours**. We know it's always tempting to do more, but we want to see what you can do in 2 hours, and we want every candidate to spend a similar amount of time on the exercise. Take your time to review the specifications below, make a plan, and then give yourself 2 hours to make as much progress on it as possible

Sources:

Internal API data
https://bw-interviews.herokuapp.com/data/providers

Attached CSV file
ca_omcc_providers.csv

Naccrra OMCC Provider Database
http://naccrrapps.naccrra.org/navy/directory/programs.php?program=omcc&state=CA

Specifications:

Create an ETL service that will aggregate data from the above sources. The resulting data set should include all data, and standardized formats for data types. Evaluate the data quality, in particular,

- The various sources will contain overlapping datasets
- Your output will be the source of truth
- Your output should normalize data types where possible

As part of the exercise, you do not need to set up a working schedule, but design the service in such a way that it could be run regularly, updating the output from all three sources on a nightly basis.

In addition to the ETL service, your submission should include SQL that answers the following questions:

- How many Family Child Care Home providers are there in the dataset?
- Which Zip code has the most providers?

Implementation Requirements:

This is a simple exercise, but organize, design, and test your code as if it were going into production.

Please include a README file in your repository with the following information:

- How to install/run your service
- Which language, framework and libraries you chose and why
- Tradeoffs you might have made, anything you left out, or what you might do differently if you were to spend additional time on the project
- Anything else you wish to include.

Provide us the link to your final product as a cloneable repo on github, gitlab, or bitbucket.

How we Review:

- Functionality: Does the ETL service meet the specification?
- Code Quality: Is the code easy to understand and maintainable? Is it well tested?
- Technical Choices: Do your choices of libraries, architecture etc seem appropriate for the application?