PARAGON
IT Professionals

Do not use others information, do it yourself. They have had multiple duplicates of tests. This is the first interview so see it as such. Information on format:

1. The test should not be in a zip if it is they will be automatically cut.

2. They need to make sure to follow all the questions stated in the test and give examples and screenshots. The more complete the test the better.

3. They are very picky the test can be right it is more about how they do the test (clean, organized and coded in the manner stated) (SQL for database)

Extra information: filename represents an ID assigned to the weather station collecting that data. Consider each weather data file to be a weather station to run the statistics on - no need to break them out by state, as that information is not included.

Overview

This coding exercise will help us understand how you approach some of the common problems we see in data engineering. Ask questions if things are unclear, use best practices and common software patterns, and feel free to go the extra mile to show off your skills. Imagine you are handing off your completed project to someone else to maintain -- it should be clear to another developer how things work and your reasoning behind your design decisions.

You will be asked to ingest some weather and crop yield data (provided), design a database schema for it, and expose the data through a REST API. You may use whatever software tools you would like to answer the problems below, but keep in mind the skills required for the position you are applying for and how best to demonstrate them. Read through all the problems before beginning, as later problems may inform your approach to earlier problems.

You can retrieve the data required for this exercise by cloning this repository:

https://github.com/corteva/code-challenge-template

Weather Data Description

The wx_data directory has files containing weather data records from 1985-01-01 to 2014-12-31. Each file corresponds to a particular weather station from Nebraska, Iowa, Illinois, Indiana, or Ohio.

Each line in the file contains 4 records separated by tabs:

1. The date (YYYYMMDD format)
2. The maximum temperature for that day (in tenths of a degree Celsius)
3. The minimum temperature for that day (in tenths of a degree Celsius)
4. The amount of precipitation for that day (in tenths of a millimeter)

Missing values are indicated by the value -9999.

Problem 1 - Data Modeling

Choose a database to use for this coding exercise (SQLite, Postgres, etc.). Design a data model to represent the weather data records. If you use an ORM, your answer should be in the form of that ORM's data definition format. If you use pure SQL, your answer should be in the form of DDL statements.

Problem 2 - Ingestion

Write code to ingest the weather data from the raw text files supplied into your database, using the model you designed. Check for duplicates: if your code is run twice, you should not end up with multiple rows with the same data in your database. Your code should also produce log output indicating start and end times and number of records ingested.

Problem 3 - Data Analysis

For every year, for every weather station, calculate:

* Average maximum temperature (in degrees Celsius)

* Average minimum temperature (in degrees Celsius)

* Total accumulated precipitation (in centimeters)

Ignore missing data when calculating these statistics.

Design a new data model to store the results. Use NULL for statistics that cannot be calculated.

Your answer should include the new model definition as well as the code used to calculate the new values and store them in the database.

Problem 4 - REST API

Choose a web framework (e.g. Flask, Django REST Framework). Create a REST API with the following GET endpoints:

/api/weather

/api/weather/stats

Both endpoints should return a JSON-formatted response with a representation of the ingested/calculated data in your database. Allow clients to filter the response by date and station ID (where present) using the query string. Data should be paginated.

Include a Swagger/OpenAPI endpoint that provides automatic documentation of your API.

Your answer should include all files necessary to run your API locally, along with any unit tests.

PARAGON
IT Professionals

Extra Credit - Deployment

(Optional.) Assume you are asked to get your code running in the cloud using AWS. What tools and AWS services would you use to deploy the API, database, and a scheduled version of your data ingestion code? Write up a description of your approach.

Submitting Your Answers

Consider using a linter, code formatter, and including tests and code comments. Anything that helps us understand your thought process is helpful!

Please provide us with a link to your Git repository, hosted on GitHub/GitLab, containing your answers and code.