Data Engineer

Interview Exercise

Introduction

This exercise is designed to test your analytical skills as well as your data manipulation and coding skills.

Instructions

The exercise is divided into two unrelated sections. Please see below for instructions and additional information for each section. If you have clarifying questions about the exercise, feel free to reach out to the hiring team.

---

(1) Customer data modeling

Background:

You have been provided with two attachments: **customers.csv** and **orders.csv**.

- **customers.csv** contains the customer account information for 200 customers that have created accounts on our site
- **orders.csv** contains purchase information for all orders made by these customers, as well as guest checkouts

Instructions:

Using these two files as a starting point, create the following outputs using SQL:

- A customer list that contains a "master" customer_id, email address, first name, and last name for each customer
    - Note: customers that have an account can also place orders as guests. Guest checkouts made by customers that have accounts should be linked under the same customer_id. Use email as the master key to identify and join these records.
- An order list that assigns the master customer_id to each order
- A "customer facts" table that contains the following fields for each customer:
    - "Master" customer_id
    - First order date
    - Latest order date
    - Lifetime gross revenue
    - Lifetime discounts
    - Lifetime net revenue
    - Lifetime # orders

Please submit:

- Your final output tables (format of your choosing), as outlined above
- An overview of your approach to creating each file and all relevant SQL queries/scripts

NOTE: Email and name data is anonymized, but does align between files.

Data Engineer

Interview Exercise

(2) Data ingestion

Background:

OpenAQ collects air quality data from sources around the world, and makes that data available for free via their API. The API does not require a token or password, though it does limit the volume of requests.

Instructions:

Using the language of your choice, create a program/script that accomplishes the following:

- Using the *Sources* endpoint, return a count of active data sources by country (.csv format)
- Using the *Cities* endpoint, return a count of cities and a count of total locations by country (.csv format)
- Using the *Sources* endpoint, define a function that returns a list of source URLs for each country that the user specifies as an input; if no sources exist for a given country, the function should alert the user
    - Run your function and return the results (.csv format), with the following countries as input: Canada, Ireland, Japan, Morocco, and Norway

Please submit:

- Your final output files, as outlined above
- Your program/script (this should be "ready to run", e.g. requires no additional setup work)