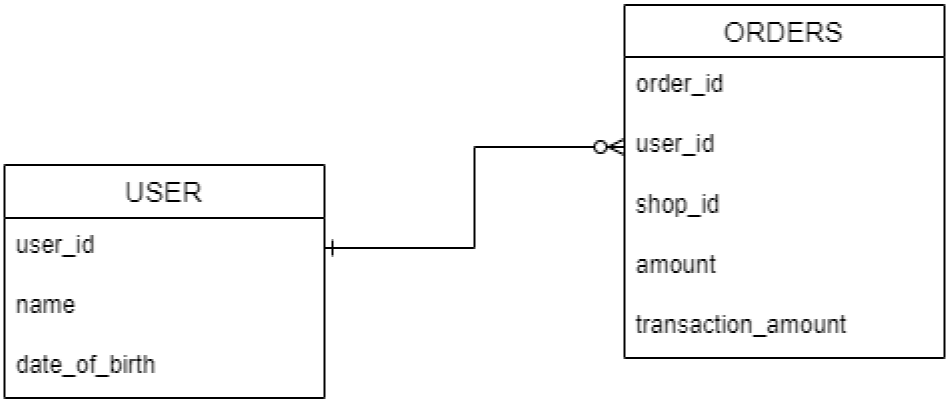
Data Engineer: Technical Test

The technical test consists of two tasks, the first is a SQL test and the second is a coding task.

You will be given up to a week to finish this test.

SQL test

We have a database with 2 tables



The user table has a one-to-many relationship to the orders table, and all orders must have a user, however not all users have orders.

Question 1: Write a SQL query to list ALL the users (including user_id and name) with the number of orders they have done including users who haven't ordered anything yet.

Question 2: Write a SQL query to list the users (including user_id and name) that haven't got any orders.

Question 3: Write a SQL query to list the users that have orders greater than £100. Include user_id, name, and the total number of orders they have done including those less than £100.

Coding task

The coding task consists of three parts:

1) Build a XML to CSV processor for the provided XML file using only Python's standard library
2) Build a Python client to extract data from an API
3) Draw a diagram showing how you would design a solution for extracting data in part 1 & 2 and making it available for other people in the data team

More details on the tasks will be available in the README.md file.

1 Build a XML to CSV processor

In Python and only using Python’s standard library, create a program to convert the XML file accountbalance20210101.xml in the files directory into a CSV.
Considerations:

- Expect the XML files to be sufficiently large that the whole file cannot be loaded into memory.

2 API Client

Using Python and any additional libraries create a client for extracting data from the API in app.py.
The API has four endpoints:

- /index: returns a welcome message
- /campaign_statistics: returns data about a marketing campaign performance, see files/campagin_statistics.json
- /campaigns.json: returns data about marketing campaigns, see campaigns.json
- /creatives.json: returns data about creatives for ads, see creatives.json

Considerations:

- The client has to send an api key in the header to authenticate
- The client should be able to paginate over responses

3 Architecture Diagrams

You should now have a program to process XML files to CSVs and to extract data from the marketing API.
The XML files you are receiving come from a third party Financial company and have to be reconciled with our own records. New files are dumped daily in either a FTP or object store and our own records reside in a SQL database (MySQL, Postgres, SQL Server, etc.).
Draw a pipeline showing the process for reconciling these files and specify the tools you would use.

The data team needs to report on marketing performance to optimize marketing spend. Draw an ETL(ELT) pipeline to extract data from the API in part 2 and specify the tools you would use. In addition, structure the data so it is efficient to query and easy to understand.