Working Student, Data Engineer - Technical test

Attached to this technical test is the backup of a Postgres database (db_backup.sql.zip) with four tables containing fictional sales and customer data:

1. `orders`
   a. This table contains all orders from 2021 to the end of June 2023
2. `order_items`
   a. An order can contain multiple products, and each product in an order is recorded in the `order_items`
3. `products`
   a. This table contains all products along with their unit price
4. `customers`
   a. This table contains all customer info, such as addresses and contact information

There are also three views used for reporting and analysis:

1. `v_customer_orders_by_month`
   a. A monthly breakdown of orders by customer and product name, including quantity ordered and total sales value
2. `v_orders_by_country`
   a. A monthly breakdown of orders by country, including total sales value, and the average order size
3. `v_product_orders_by_month`
   a. A monthly breakdown of orders by product, including the number of orders each product is included in, the share of total orders that the product appears in, as well as the total sales value of that product

You need Python and PostgreSQL installed and running locally to complete this assessment.

To get started, create a database from the backup (e.g. by running the command below) and complete the tasks.

```
psql -d postgres -h localhost -U postgres  -f db_backup.sql
```

1) SQL

The users of the views are saying that they are taking a long to time execute.

- Help improve their experience by optimizing the query behind the `v_product_orders_by_month` view
- If there are other actions you can take to improve the performance of this view, take them as well

2) Python

Now that you've helped improve the performance of the views, the market analysis team wants to know more about our customers, specifically which industry they operate in.

- Write a Python script that fetches the industry for each of our customers by using this API endpoint (https://apolloio.github.io/apollo-api-docs/?python#organization-enrichment) and insert this data as a new column in the `customers` table

Once you are happy with your solutions, please return your answers as a zip file containing:

- A fully contained Python project/script, including your code and any relevant instructions for running the code
- A backup of your database (obtained by e.g. running the command below)
- A write-up of the steps you have taken to optimize the performance of the views, as well as any comments you have on enriching the customer data using the API

```
pg_dump -C -h localhost -U postgres -d postgres -f db_backup.sql
```