Tibber Data Engineering Task Assignment

We're excited to have you here and look forward to seeing how you approach this challenge. In this exercise, you'll fetch external data from an API, ingest CDC data, and bring it all together in a presentation layer.

The goal is to assess your ability to build reliable data ingestion pipelines, apply incremental and idempotent strategies, and maintain data consistency in a PostgreSQL environment.

Take your time, read through the tasks carefully, and if you have any questions, don’t hesitate to reach out—we’re happy to help.

Once done with the task please send it to niklas.nordansjo@tibber.com and we will book a meeting with two people from the data team where you will present your solution.

Good luck, and have fun! 🚀

Prerequisites: Connection Details

Here are the connection details to the PostgreSQL database

Parameter	Value
Host	pg-a13753b-tibber-4cd2.b.aivencloud.com
Port	24792
Database	aa12
Username	tibber_aa12
Password	empLYSIcAni

Task 1: Currency and Conversion Rate Ingestion

Objective:
Fetch currency data and conversion rates from the VATComply API and persist the information in your PostgreSQL database.

Instructions:

- **API Integration:**
  Retrieve data using the endpoints provided in the VATComply API documentation.
  - One endpoint should return a list of currencies.
  - Another should provide conversion rates based on a given base currency (e.g., “NOK”).
- **Database Schema:**
  Create two tables:
  - A **currencies** table to store the currency code, name and symbol.
  - A **currency_conversion_rates** table to store conversion rates between currencies with appropriate foreign key relationships with **currencies**.
- **Process:**
  - Develop a Python script that fetches the data from the API.
  - Implement “upsert” logic to insert new records or update existing ones based on conflicts (using primary key constraints).

Task 2: Transaction Data Ingestion from CSV

Objective:
You will create a small, idempotent, incremental ingestion pipeline.

Create a new PostgreSQL table and ingest item price data from CSV files into it. Ensure that records are updated based on the latest timestamp.

Instructions:

- **Data Source:**
  In the email you will find a zip file containing a folder named `data` with three CSV files:
  - `batch1.csv`
  - `batch2.csv`
  - `batch3.csv`
- **Target Table:**
  As the first step, create a table named `public.item_prices` with the following schema:
  - `id`
  - `item`

    - `price`
    - `currency`
    - `created_at`
    - `updated_at`
    - `system_timestamp`

- **Processing Requirements:**
    - **Iteration:**
      The CSV files must be processed in the correct order. Your script should either automatically process the files sequentially or allow an argument to specify which CSV file to process next.
    - **Checkpointing:**
      Implement a checkpoint mechanism that stores the highest `system_timestamp` processed. For subsequent runs, only process CSV rows with a `system_timestamp` later than the stored checkpoint.

Task 3: Creating a SQL View for Currency Conversion

Objective:
Develop a SQL view that displays all transactions with prices converted into NOK currency.