Data Engineer

Take-Home Data Engineering Exercise

Ferovinum provides capital to businesses in the wine and spirits industry. As we grow, our data needs require robust pipelines to extract and transform operational information. We value core data engineering skills, problem-solving, and clear communication of your approach.

Overview

We've prepared a simplified scenario to test your ability to ingest, transform, and analyze data from multiple sources. We expect you to take 2 hours to solve this problem, and no more than 4 hours.

What You Need to Deliver

Build a small pipeline (with accompanying documentation) that accomplishes the following:

1. **Demonstrates** how you ingest, join, and transform the provided raw data into a clean, well-modeled dataset.
2. **Provides** intermediate steps or outputs (e.g., partial tables, notebooks, or logs) that clarify how you perform key transformations.
3. **Produces** final insights answering two key questions:
    - **Which regions have the highest transaction volume for each SKU (by quarter, by year)?**
    - **Which two brands are the most profitable during the first 21 weeks of 2024?**

Please structure your solution end-to-end: from raw data ingestion to a final cleaned dataset that could be easily used by BI tools or analysts for additional exploration.

Requirements & Deliverables

1. **Data Ingestion**
    - Read and parse the provided application logs.
    - Read and parse the SKU metadata (JSON files).
    - Read and parse the market pricing data.
2. **Data Modeling & Transformation**
    - **Join** the logs with SKU metadata and market prices.
    - Use the most recent market price prior to each transaction to calculate transaction value.

- Create a data model or schema that captures relevant dimensions and facts (e.g., time periods, SKUs, pricing, regions, etc.).
- Show **intermediate outputs** or transformations. Explain your design decisions (e.g., why you chose certain schemas, how you handle missing data, etc.).

3. **Analysis & Output**
   - Produce final outputs (tables, CSVs, JSON, notebooks, etc) that answer the key questions above.
   - Export a **cleaned, well-modeled dataset** (e.g., a star schema table or aggregated file) that can be consumed by BI tools for further analysis.
   - Feel free to include additional "views" or aggregated tables if they help clarify your approach.
4. **Documentation**
   - Provide a brief README that includes:
     - How to run your pipeline (e.g., Python scripts, SQL, or other tools).
     - Any assumptions you have made (e.g., time zones, missing data handling, etc.).
     - Notes explaining your data modeling and pipeline design decisions.
   - (Optional) Include a Dockerfile or environment setup instructions if it simplifies running your solution.

This exercise is intentionally open-ended to allow you to demonstrate your skills and approach to data engineering.

Time & Submission

- **Time Expectation**: Approximately 2–4 hours of focused work.
- **Submission**: Please share your code and instructions (e.g., a Git repository link). We will clone and run your solution in a clean environment.

We appreciate your effort on this exercise and look forward to seeing your approach!

Appendix – Application Context & Data Sources

Application Context

Imagine there is a simple upstream application that manages client orders for wine, whisky, and bourbon products. This application handles two command types:

- `buy [sku] [quantity]` – A client buys a certain quantity of a product (SKU).

- `sell [sku] [quantity]` – A client sells a certain quantity, using FIFO rules.

Whenever a customer issues a `buy [SKU] [quantity]` or `sell [SKU] [quantity]` command, the system tracks the transaction details in real time. By design, the system always follows a First In, First Out (FIFO) principle, meaning when it needs to sell inventory, it uses up the oldest stock first. If sufficient inventory isn't available, the system records the order as failed but continues running without interruption.

Additionally, every transaction is priced at the most recent market price known at the time the order is placed. This allows us to calculate transaction values for each buy or sell command, reflecting current market conditions. The system logs each step—whether an order was placed, how many items were transacted, the outcome (success or failure), and any relevant messages describing what happened.

To provide more insight into each product, the system also uses metadata for every stock-keeping unit (SKU). This metadata includes details like a wine's region or brand, or a whisky's cask type. Combining the SKU's metadata with the transaction history makes it possible to analyze trends (for example, which regions or brands are seeing the most activity) and to break down performance by product attributes.

Because market prices fluctuate, the system relies on an external source of pricing data. The latest price available for a given SKU and timestamp is applied to each transaction. This means any downstream analysis or reconstruction of the order history must correctly align transaction times with the relevant market quotes.

To maintain confidence that stock levels and financials are correct, an inventory snapshot is periodically taken—in this example, we have a partial report from the end of January 2025. By comparing this snapshot with the logs (and referencing the pricing data), we can verify that buy and sell quantities, as well as the associated values, match real-world inventory counts.

Your job is to combine these data sources—logs, metadata, and prices—into a clean, consolidated dataset and to deliver final insights on transaction volume and profitability.

Data Sources

You will be provided with the following data:

- **Application Logs**: A set of application logs that capture `buy` and `sell` transactions for various SKUs.
- **SKU Metadata:** A set of SKU metadata that provides extra attributes for each SKU (e.g., region, brand).
- **Market Prices**: A set of market pricing data that determines transaction value at the time of each order.

- **Ending Inventory Report:** An ending inventory report that provides the ending inventory for a subset of SKUs at the end of January 2025.

Application Logs

Found in the `data/logs` folder.

The application logs lines of output for each order as it is processed. The log is formatted as follows:

- `timestamp` - The timestamp of the log
- `message_type` - The type of message that that the log contains
- `trace_id` - The trace ID of this request
- `details` - The message details of the log

```
timestamp           | message_type  | trace_id         | details
-----------------------------------------------------------------
2024-03-15 10:30:00 | ORDER         | a1b2c3d4e5f6g7h8 | sell WINE-CAB-001 50
2024-03-15 10:30:00 | TRANSACTION   | a1b2c3d4e5f6g7h8 | -50
2024-03-15 10:30:00 | RESULT        | a1b2c3d4e5f6g7h8 | COMPLETED | SUCCESS
2024-03-15 10:30:00 | RESPONSE      | a1b2c3d4e5f6g7h8 | Successfully sold 50 units of WI
NE-CAB-001
2024-03-15 10:30:01 | ORDER         | b2c3d4e5f6g7h8i9 | buy BRBN-OAK-002 100
2024-03-15 10:30:01 | RESULT        | b2c3d4e5f6g7h8i9 | FAILED | NO_AVAILABLE_STOCK
2024-03-15 10:30:01 | RESPONSE      | b2c3d4e5f6g7h8i9 | Insufficient available stock for S
KU BRBN-OAK-002. Requested: 100, Available: 0
2024-03-15 10:35:15 | ORDER         | c3d4e5f6g7h8i9j0 | sell WINE-MER-003 25
2024-03-15 10:35:15 | TRANSACTION   | c3d4e5f6g7h8i9j0 | -25
2024-03-15 10:35:15 | RESULT        | c3d4e5f6g7h8i9j0 | COMPLETED | SUCCESS
2024-03-15 10:35:15 | RESPONSE      | c3d4e5f6g7h8i9j0 | Successfully sold 25 units of WIN
E-MER-003
...
```

The system retains a running, FIFO-based ledger of `sell` and `buy` activity, never returning errors—if a `buy` can't be fulfilled, it will fail the order. At the time of the order, the system will price the order at the most recent market price and use that price for the order.

SKU Metadata

Found in the `data/skus` folder.

Each item bought and sold has additional metadata about the item that can be used to enrich the data. For example, a wine might have the following JSON metadata:

```
{
  "id": 1,
  "code": "WINE-CAB-001",
  "product_category": {
    "id": 1,
    "name": "Wine"
  },
  "region": {
    "id": 1,
    "name": "Napa Valley",
    "country": {
      "id": 1,
      "name": "United States"
    }
  },
  "sub_varietal": {
    "id": 1,
    "name": "Cabernet Sauvignon"
  },
  "brand": {
    "id": 1,
    "name": "Robert Mondavi"
  },
  "attribute_json": {
    "abv": 14.5,
    "bottle_ml": 750.0,
    "year": 2019,
    "age": 5,
    "grape_variety": "Cabernet Sauvignon"
  }
}
```


Market Pricing / Exchange Rates

Found in the `data/market_prices` *folder.*

The market pricing data contains a timestamp, a quote_id, and a price_usd quoted at that time.

```
timestamp            | quote_id     | price_usd
-------------------------------------------------------
2024-12-01T09:00:00Z     | WINE-CAB-001 | 76.55
2024-12-01T09:00:00Z     | BRBN-OAK-002 | 120.55
```

```
2025-01-01T09:00:00Z      | WINE-CAB-001 | 75.54
2025-02-01T09:00:00Z      | WINE-CAB-001 | 74.53
...
```

You can find the data available at the link below:

https://interviewtask.s3.eu-west-2.amazonaws.com/fero_data_engineering_takehome.zip?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIASFZMM5IWHPM3FLAF%2F20250331%2Feu-west-2%2Fs3%2Faws4_request&X-Amz-Date=20250331T133333Z&X-Amz-Expires=604800&X-Amz-SignedHeaders=host&X-Amz-Signature=03e436f0a64a9bfd56df12c92f895dac8717b89b6671cc228c9d6ae73484a31d

Ending Inventory Report

As part of this exercise, imagine that our accounting department has done a partial inventory count. They have provided us with the ending inventory for a subset of SKUs at the end of January 2025 that you can use to verify your results.

Below is their report:

SKU	Ending Inventory
WINE-OPU-001	8
WINE-OPU-003	18, 592
WINE-LAF-004	5, 676
WHKY-GLE-018	3, 546
BRBN-MAK-024	130, 049

Thank you for taking the time to complete this exercise. We look forward to reviewing your submission!