



Technical exercise

Craft Beer Emporium

Overview

You are free to structure the project as you like. During the technical interview, we will be looking at your decision making, technical implementation, and how you go about to solve a given problem.



If something is unclear in the following description, please make reasonable assumptions and explain them during the presentation.

Your Assignment

Craft Beer Emporium, a fast-growing online retailer of craft beers, is in the midst of a technological transformation. Having outgrown their monolithic architecture, they are embracing a microservice-oriented approach and migrating more workloads to the cloud. While this move promises better scalability and reliability, it also means keeping a close eye on rising cloud usage costs.

Recently, the finance team at Craft Beer Emporium noticed that their cloud billing statements had become unwieldy and lacking in actionable detail. To address this, they obtained raw cost and usage data from Microsoft Cost Management. However, they need to convert it into a standardized format called [FOCUS](#) to gain clearer insights and seamlessly integrate it with their internal reporting systems.

Your task is to help Craft Beer Emporium ingest, cleanse, and transform these raw cost files into the FOCUS standard. Once transformed, this data will be loaded into a database and analyzed for trends, potential savings, and actionable recommendations. The ultimate goal is to equip the DevOps and finance teams with robust cost analytics so they can continue brewing success—without worrying that their cloud costs have gone flat.



Your solution should be maintainable when Craft Beer Emporium requests new features and should be ready for production usage. This does not necessarily mean that you need to implement every detail that is relevant for production usage, but you do need to be prepared to answer questions during the presentation that will challenge your thinking in this regard.

Implementation Details

The finance team at Craft Beer Emporium has received raw cost data from Microsoft Cost Management, but it needs to be **converted to the FOCUS standard**.

Available Files (From Microsoft Cost Management)

1. [EA-Cost-Actual](#)
2. [EA-Prices](#)

These files are **not yet** in the FOCUS standard. Your first step is to **transform** the data so that it conforms to the FOCUS format (naming conventions, date formats, field normalization, etc.).

Once transformed, the data should enable Craft Beer Emporium's teams to:

- **Monitor** current cloud spend (actual and amortized).
- **Analyze** consumption trends.
- **Evaluate** reservation effectiveness and recommendations for potential cost savings.
- **Inform** future architecture and capacity decisions in a microservice environment.



If time is an issue, scope down to the most important parts and implement a simple Proof of Concept.

Exercise

Craft Beer Emporium is expanding its online retail operations and wants better visibility over its cloud usage costs. You've been given multiple CSV files obtained from Microsoft Cost Management—things like *EA-Cost-Actual*, *EA-Prices*, etc. These files need to be standardized into the **FOCUS** format. Once the data is in that standard, the finance team will use it to analyze current costs, identify potential savings, and plan future budgets.

The core idea for this exercise is to build a **data pipeline** that ingests, cleans, and transforms those CSVs. You can then store the transformed data somewhere (e.g., a local database) and perform a few basic analyses or queries to demonstrate that the pipeline works. A highly sophisticated solution isn't required—focus on clarity, correctness, and reusability.

No strict framework is mandated. You could use Python, Spark, or any other tool you feel comfortable with. If you do need a database, you may opt for something like **SQLite** or **PostgreSQL**. If you want to push it further, you can use Docker, an orchestration tool (Airflow, Prefect), or maybe generate a couple of neat visualizations.

Things you can cover (but it's up to you how far you go):

- A simple script or service to read and parse the raw CSVs into a consistent schema.
- A transformation step to standardize the data (e.g., converting to FOCUS naming conventions, normalizing dates, handling missing values, etc.).
- A place to store the transformed data (could be flat files, an in-memory table, or a local DB).
- Some example queries or aggregations to demonstrate how Craft Beer Emporium might analyze total costs, monthly trends, or recommended reservations.
- A README describing how your solution is structured, how to run it, and any assumptions you've made.

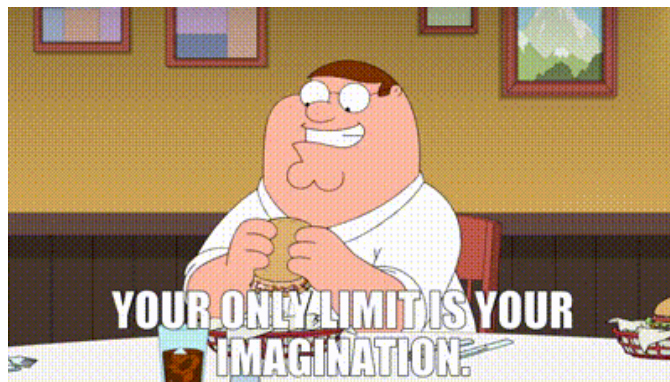
You don't have to build a big UI or a polished dashboard. However, if you feel like visualizing the resulting data, feel free to do so with any tool (Power BI Desktop, Apache Superset, or even a quick Python-based chart).

When you're done, please upload your work to a **GitHub repository** and share it with us. Include all commits you made, and add a **Readme** with details about your approach, instructions for running it, and any interesting observations you gathered from the data. Feel free to use any free tools or platforms that you like—Databricks Community Edition, local Spark, or plain Python scripts with Pandas. We're eager to see how you structure your solution and how you tackle the problem of converting raw data into actionable insights.

Want to go the extra mile? Here are some ideas:

Let your imagination run wild and give the application that touch that makes it stand out from the crowd.

- Show how you'd automate this: maybe a simple Airflow DAG, a script that runs on a schedule.
- Include tests that verify your data transformations (e.g., date formats are valid, numeric fields make sense, etc.).
- Briefly describe how you'd adapt this pipeline for real-time or near-real-time ingestion if Craft Beer Emporium wanted continuously updated cost data.



Potential tools

Data Handling: Python (3.8+ recommended), or similar scripting language for data manipulations (e.g. PySpark if using Spark) using libraries such as pandas, pyarrow or sqlalchemy. You can also get access to [Databricks Community Edition](#)

Database interaction: SQL running local database like SQLite or PostgreSQL

Reporting tools: Power BI Desktop (free for Windows), or open-source equivalents like Apache Superset or Metabase.

Workflow orchestration: Free tools such as Apache Airflow, or Prefect.

VSCode, IntelliJ or whatever tool you feel more comfortable using

Useful link: [FOCUS Converter](#)

Dataset link: [Data](#)

Maximum time allowed

No limit