Akasa Air

IT'S YOUR SKY

Interview Task – Data Engineer

Akasaair.com

Contents

OBJECTIVE

To process and analyze customer and order data from multiple sources (CSV and XML) using both database table and data-frame (in-memory) approaches. Assume that the data will be share once a day from the source, with new records received on the previous day only. Add missing data if you think there's a need. The goal is to derive key business insights, such as repeat customers, monthly order trends, regional revenue, and top spenders in the last 30 days.

DETAILED REQUIREMENTS

1. Data Sources:

Customer Data (CSV): Fields: customer_id, customer_name, mobile_number, region
Orders Data (XML): Fields: order_id, mobile_number, order_date_time, sku_id, sku_count, total_amount

2. KPIs:

- **Repeat Customers:** Identify customers with more than one order.
- **Monthly Order Trends:** Aggregate orders by month to observe trends.
- **Regional Revenue:** Sum of total_amount grouped by region.
- **Top Customers by Spend** (Last 30 Days): Rank customers by total spend in the last 30 days.

3. Processing Approaches:

A) TABLE-BASED APPROACH

1. Data Cleaning & Loading:
    a. Parse and clean CSV and XML files.
    b. Load into relational database tables (customers, orders).
2. Querying for KPIs:
    a. Use SQL queries to compute each KPI efficiently.
    b. Ensure indexes and joins are optimized for performance.

B) IN-MEMORY APPROACH

1. Data Cleaning:
    a. Use Python/Java/Scala to read and clean CSV/XML files.
    b. Normalize date formats, handle missing values, and ensure type consistency.
2. KPI Functions:
    a. Implement modular functions for each KPI.

4. SECURITY:

a. Use parameterized queries or ORMs (like SQLAlchemy) to prevent injection attacks.
b. Avoid string concatenation in SQL statements.
c. Store credentials securely (e.g., in .env files or secret managers).
d. Avoid exposing credentials in code or logs.

5. IMPORTANT CONSIDERATIONS:

a. Data Freshness: Ensure timely ingestion and transformation to reflect recent trends.
b. Scalability: Design queries and functions to handle growing data volumes.
c. Error Handling: Implement logging and exception handling for file parsing and data transformation.
d. Time Zone Awareness: Normalize timestamps for accurate 30-day calculations.

6. DOCUMENTATION:

a. Provide detailed documentation on how to set up and run the application.

GUIDELINES

- You are free to choose any programming language, framework, or technology stack for the implementation. (Java, Python, scala)
- It is recommended to use a Mysql Database for data storage.
- Focus on code quality, maintainability, and scalability of the application.
- You may use third-party libraries or frameworks, if necessary, but clearly mention any dependencies.

DELIVERABLES

- Source code including all necessary files and resources should be shared via **GitHub**. Share the github repository for this project.
- Documentation explaining the design, implementation, and "how to run" the application.
- Briefly explain any additional features or improvements you would consider if given more time.

Note: We are looking for your understanding of the problem, your approach to solving it, and your ability to design and implement a basic version of the ingestion pipeline, **with all requirements met**.

Please complete the task within the given time frame and submit the deliverables for evaluation. All the Best!

Akasa Air
IT'S YOUR SKY

Thank You!

Note: SNV Aviation Private Limited (“**Akasa Air**”) is the owner and copyright holder of the contents of this document, which are strictly confidential. Unauthorized access of any information contained in this document is prohibited. This document and/or any of its contents must not be copied in whole or in part by any means, without the prior written authorization of Akasa Air.

Disclaimer: Any trademarks, logos, etc. pertaining to third parties have been used for illustration purposes only and remain the exclusive property of their respective owners.

AkasaAir.com