Data Engineering Challenge

This is the technical challenge for SCRM's Data Engineering team. Feel free to use whichever platform you want to write and send us back the exercises. You can use any IDE, code in a Notebook or source files. Whatever you feel more comfortable with!

Spark

The first part of the challenge is about coding a Spark Application. You will have a data model and some testing data. Data will be ingested daily and processed in batch mode. You will have to code some transformations on it using **Apache Spark**. Use any Spark compatible language or any Spark API, but we will favor both the Scala and the Python flavors as those are the ones we work with. Generally, there is no such thing as a correct answer, but we value that you know how to justify a decision.

The data model for the challenge encompasses 3 tables: **Products**, **TicketLines**, and **Stores**:

- **Products**: the representation of a product and its features. This entity contains a list of internal category ID codes with its corresponding name (`categories`).
- **Stores**: the representation of a Lidl store. It includes the `country` it belongs as Lidl is a multinational company, also the team versioned the data so a field version is included.
- **TicketLine**: It's the instance of selling some **products** in a **store**.

You will find 3 CSV files describing the data attached to the test; use these samples.
products.json
ticket_line.csv
store.csv

Tables

Products	
product_id	BigInt
product_name	String
categories	Array[Struct(category_id: Int, category_name: String)]

TicketLines	
ticket_id	BigInt
product_id	BigInt
store_id	BigInt
date	Date
quantity	Int

Stores	
store_id	BigInt
country	String
version	String

Relationships

Products.product_id	1..n	TicketLines.product_id
Stores.store_id	1..n	TicketLines.store_id

Questions:

1. Find how many different stores is each product being sold. Please consider only the stores provided in the store.csv file, as not all stores are included in the Lidl Plus program.

2. Calculate the 2nd most selling store for each product as we need a target for advertisement. As the previous one, consider only the stores that are included in the store.csv file.

3. The marketing team wants to group all these second stores by product category, so they can focus on different stores by using the same advertisement approach. As they don't care about internal id's, please provide **one row** per product *category_name* and include all the stores within that row.

4. Now, let's imagine that the integration team is developing a new version for the stores model. They will send this new data for the "version 2" of stores available only for some countries. (That means that you will be receiving both versions simultaneously). They have changed the meaning of *store_id* and country, so the new *store_id* has the country prepended to it like "FR99" and the country field is omitted.

Please, integrate this new source with the existing store source and make your code work seamlessly with both versions.
You will also read now the example file stores_v2.csv

5. Considering this ETL complete, we need to productize it: prepare it on the version control system, automatically deploy the changes, ensure that there is no software regression on new releases, and check that the process works fine. Which steps would you take?

Data Architecture

We want you to design a data system. Let us think of an actual use case: a team is developing an app for Lidl's cycling service that we are about to launch in Deutschland. Consider the following scenarios:

- An external bike provider is responsible for the stock and maintenance of the bikes. They send us their master data (about the bikes) in CSV files through an FTP connection and leave the files in a cloud bucket once every 24h at night.
- The mobile app tracks the user's behavior regarding the bike's service and sends many events (more than 1 million/second) as soon as they are created in compressed format.

The needs of our system:

- We need to handle and be able to query the app's events in real-time.

- Somehow, we need to coordinate the info of our bike provider with the availability, the status, and id of the bikes before 6:30 am (the hour that our bike service opens).
- We need real-time monitoring dashboards.
- The relevant data for every stakeholder should be available as soon as possible (data science team, data analyst, business, developers, etc.).
- You must choose the solution cost-wise (mind efficiency, there is no need to justify the detail of costs).

The challenge deliverable:

- Draw a diagram (flow or persona, there's no need to be a low level one) with your data architecture solution.

Bear in mind that you need to be able to defend your choice vigorously. We do not want you to spend too much time on this challenge since it is just an excuse to have something to discuss in the interview so consider that you can be asked for things like:

- Explain the data architecture you have chosen to solve the problem (kappa, lambda, lake house, etc.).
    - Justify your decision.
    - Explain the components and technologies you used very briefly.
- How would you test and monitor?
- How to govern/audit this architecture?
- How would you make this system resilient and fault-tolerant?
- How would you handle and configure the throughput?