Code Test - Data Analyst

Aim of this coding test

The aim of this coding task is to assess the **way you approach problems and design solutions**, as well as providing insight into your **coding style, expertise and willingness to experiment**. It will also provide us with a **common ground for the final interview**.

There is no hard time limit set for this task, but we recommend allocating **up to 3 hours** to complete this task. Due to time constraints, we **don't expect a perfect solution** with all the edge cases covered. You're encouraged to **focus on your core strengths** and things that you think are important — feel free to **leave notes and TODOs** if there are parts of implementation you didn't manage to complete.

We'd love to see what kind of solution you come up with to the task below and how you approach problem solving!

Task Description

Scenario

Spond provides a platform for organizing sports teams, events, communication, and payments. In this challenge, you'll simulate a realistic work scenario by producing insights and recommendations from Spond's internal data.

The focus is on:

1. **Data Cleaning** - detect data-quality issues, clean the dataset, and clearly document every assumption or fix you apply.
2. **Aggregation & KPIs** - aggregate the cleaned data into the required KPIs and highlight any meaningful patterns or anomalies you discover.
3. **Visual Storytelling -** build clear, well-labelled charts that translate the KPIs into actionable recommendations.

Data Description

Table	Column	Description
teams	team_id (string)	Unique ID
	team_activity (string)	Activity type of team e.g., football, cricket, rugby, etc.
	country_code (string)	Alpha-3 country code of team e.g., NOR=Norway; GBR=United Kingdom; etc.
	created_at (UTC timestamp)	System generated creation timestamp
events	event_id	Unique ID
	team_id	Foreign Key
	event_start (UTC timestamp)	User-defined event start timestamp
	event_end (UTC timestamp)	User-defined event end timestamp
	created_at (UTC timestamp)	System generated creation timestamp
payments	payment_id	Unique ID
	membership_id	Foreign key
	team_id	Foreign Key
	currency	Currency (e.g., NOK, GBP, USD)
	volume_local_currency	Amount paid by the member (denoted in local currency)
	created_at (UTC timestamp)	System generated creation timestamp
exchange_rates	date	Exchange rate at date
	currency	Currency (e.g., NOK, GBP, USD)
	exchange_rate	NOK as the base currency

Tasks

1. KPI Design

Using the raw tables:

1. **Activation** – Design a metric that show whether groups have *become active* for the first time. The goal is to measure how well we are at activating groups that get created.
2. **Keep-using behavior** – Design a metric that show whether groups continue to use Spond week-over-week. The goal is to measure how well we are at retaining groups that activate.
3. **Engagement depth** – Design a metric that captures the engagement of groups that activate. The goal is to measure how frequently groups are using Spond.
4. **Monetization** – Design a metric that captures Spond's revenues from payment processing. The goal is to measure how well we are monetizing groups that become active (Assume that revenue is 2% of the total amount).

For every metric you define, briefly explain:

- Why you chose it
- How it's computed
- Any assumptions

Tip: There's no single "correct" KPI. We will score on the clarity of your rationale and the correctness of your calculations based on the available data.

2. KPI Implementation

For two of the metrics you designed above, implement the code you'd use to produce it from the raw tables.

3. Visualization & Insights

For **each metric you implemented**, do the following:

1. **Surface meaningful insights**

   Example prompts to explore (choose the ones that make sense for your data):

   - Are there clear differences between countries, sports, or other relevant segments?
   - Which cohorts are growing or declining over time?
   - Do event-heavy groups behave differently from payment-heavy groups?

2. **Spell out product implications**

   - Explain **where the product team should focus** next (e.g., “focus on cricket clubs in the UK” or “simplify the first-payment for football teams”).
   - Tie each recommendation directly back to the evidence you’ve surfaced.

Technology

- Use the programming language that you’re the most comfortable with. Python, PySpark, and SQL are widely used in Spond.
- Use any other standard libraries that you may find helpful (e.g., Pandas, Numpy, Matplotlib, etc. for Python).

Final result should consist of:

- Source code with instructions on how to run it in a Git repository we can access (Github, Bitbucket etc.).
- Extra points for test coverage.
- Charts and/or dashboards
- We encourage you to add a description of improvements to your solution that you think would be natural next steps.