# Data Engineer (Spark) - Case Study

## Context

The data team at Octopus are responsible for building robust, reliable, well documented and scalable pipelines to move data from source to target and get it in shape to be analysed.
This often requires combining data from multiple data sources.

## Task

Our data analysts want to be able to query the smart meter readings data and join this to the agreement, product and meterpoint data. They also need a set of pre-built final tables they can use for reporting or building dashboards off. To achieve this we would like you to:

- Build a data pipeline to take the readings data and the static db data and move it into a single datastore using a spark related technology, i.e. delta tables, hive tables, etc.
- Build the below final analytics tables in the same datastore for the analysts to use
    - Table of all active agreements on 1st January 2021 (an active agreement has a valid_from less than the 1st January 2021 and a null valid_to or a valid_to greater than 1st January 2021). Should be one row per agreement and include:
        - Agreement id
        - Meterpoint id
        - Product display name
        - Product is_variable

        - Table of the aggregate total consumption and count of meterpoints for all the meterpoints for each half hour. Should be one row per half hour and include:
            - Datetime of half hour
            - Count of distinct meterpoints with readings
            - Sum of consumption in kWh
        - Table of total consumption in kWh per product per day. Should be one row per product-day and include:
            - Product display name
            - Date
            - Count of distinct meterpoints with readings
            - Sum of consumption in kWh
- Using spark build the below gold tables:
    - Aggregate consumption per half hour
    - Average consumption per half hour by product

## Data

We've provided you with two data sources:

- Raw smart meter reading json files data/readings/* containing:

    Json files of half hourly electricity smart meter readings for 3 days in January 2021

Each reading has

- an interval-start (in local time)
- a meterpoint id - this is a unique id for the meterpoint
- a consumption delta which is the energy in kWh consumed by that meter point the half hour interval

A sqlite database containing tables of:

- agreement - an agreement is a contract that links a customer account and a meterpoint. It effectively says that the customer account is responsible for a meterpoint between the valid dates. The agreement also states which product the customer is signed up

to. A meterpoint can only have one agreement at a time but may have multiple agreements through time.
- meterpoint - a meterpoint is a connection to the electricity network. Every household connected to the UK networks has a meterpoint with a unique id. We have provided the region for each meterpoint.
- product - this is the tariff the customer is on. It has a name (e.g. Octopus 12M Fixed) and a flag to say whether it is a variable product (i.e. the rates can change)

## Hints

- You should assume that the readings files come in every day before 8am
- You should assume the database gets updated once a day at midnight

## Guidance

- You should spend around 3 hours on the problem. If you can't finish don't worry, just write down what you would do
- Add a readme file with a description of the chosen design, possible flaws, and future work to improve it
- This isn't a test of electricity industry knowledge so if anything is unclear just ask
- You're pipeline should be written in Python, making use of Pyspark, but feel free to use frameworks, libraries or other tools
- That said, this isn't a test of which libraries you know. Credit will be given for simple, reliable and well engineered solutions.
- Make the code as production ready as possible (within reasonable limits given the time constraints)
- Feel free to choose your own database or storage format for the target
- Your pipeline should be built in a way that someone else could deploy and maintain it easily
- Although the data volumes for this task are small you should structure your pipeline to scale as data volumes grow

You should be able to automate your pipeline or at least be able to describe how you would automate it

Do not share your solution on github/gitlab as it would give an advantage to other candidates