MIGx Technical Assignment for Data Engineers

Congratulations, you’ve made it to the technical test! Completing this will help us assess your technical and problem-solving skills.

This exercise consists of two distinct parts. The first one is a programming problem, where you are expected to set up a small project from scratch in order to solve the use case problem explained below, using either **Scala** or **Python** as your programming language of choice. How you set up the project and its structure is up to you, and you can use any data processing library you see fit (or no dependencies at all!). In any case, you’ll have to **provide instructions on how to run this project locally**.

The second part are open-ended questions on different aspects of architecture and high-level decisions you may have to take on hypothetical scenarios. You can write down the answers to them in any popular file format (like Word, Markdown, Asciidoc, etc.) and optionally export it to a PDF. Just make sure you give us the raw file.

Keep in mind that you’re **not expected to complete the totality of this assignment**. If you don’t complete some bullet points that’s OK, it will help us guide the follow-up interview where we’ll ask you about what you’ve given to us and why you did (or didn’t) take those decisions.

Programming Challenge

Company Background

A global healthcare company is conducting a study to evaluate the effectiveness of various medications used in clinical trials. The company has gathered data on patients and their participation in these trials. However, the data has been collected from multiple sources and is not yet ready for analysis.

Business Requirement

The company’s data analytics team needs a well-structured, clean dataset to support decision-making. Additionally, the business team is interested in key performance indicators (KPIs) to assess:

- Trial effectiveness.
- Patient demographics.
- Medication distribution across trials.

Tasks

Ingest and Transform

Using your programming language of choice, build an ETL that reads from the provided files and outputs a single file output that's easy to query with SQL. You might notice that these inputs are not as *ideal, clean nor pristine* as you would prefer. How you handle these are part of the exercise and there's not a unique solution to that.

Querying the output data

From the output of the previous task, we would want you to calculate the following Key Performance Indicators (KPIs):

- Number of Patients per Country.
- Average Age of Patients per Diagnosis.
- Most Frequently Used Medications.
- Average Duration of Clinical Trials.
- Time Between Consecutive Trials for the Same Patient.
- Percentage of Trials with Missing or Inconsistent Data.
- Percentage of Patients in Multiple Trials.

Architecture Decisions

- You now have a working solution for the use case and have to deploy it so that it can be executed within any given environment. You have access to a Cloud platform, such as Azure, and also to Databricks, an Apache Spark provider. Explain what resources would you need and how they are connected to each other.
- Schema changes can cause unnoticed, unwanted results or even break the whole pipeline. How would you manage these changes for any given pipeline?
- You have detected that the output of your pipeline is not as intended. The output data happens to not be correct, but it's already in production. What would you do to prevent this kind of situation?