Confirmed Data Engineer Case Study

# 1. Data Modeling/ SQL exercise

Important

- For the database design you can use any tool (eg: SQL workbench, Google draw, etc)
- Provide complete documentation about your tables/ models
- Our community counts over 100 million+ travelers in 22 countries, so the model should scale

Carpool is the first and most profitable product at BlaBlaCar. We aim to ensure that passengers find a car that suits their needs and drivers have passengers to share their travel cost.

Publishing a trip

Lisa is driving from Paris to Lyon to celebrate the birthday of her cousin. As she is driving alone, she decides to publish the trip on BlaBlaCar & take advantage of carpooling.

On BlaBlaCar, she can put the origin, destination, the date & time of travel & how many passengers/ seats are available, etc.
There is also the possibility to select the route, give additional stops (to increase possibility for picking more passengers).
The app then proposes a price per seat which she can choose or add a custom one.

Once all the information is provided a ***Trip-Offer*** is published.

A passenger then can choose to travel on her published Trip-Offer. Passengers can also choose to travel on just a part of the Trip that is between stops.

# Exercise

As a Data Engineer, you are asked to build a data model for the above use case. Your main stakeholders/ users are data analysts & business analysts. ***You should build an Analytics data model.***

1. Please provide an Analytics data model to store the above scenario
   a. Hint: Start with an OLTP process model & then convert it into an OLAP process model
   b. Hint: OLAP process models are generally quite big, use proper partition/ cluster on tables

2. Provide documentation for your OLAP model which will be used by Data analysts/ Business analysts.
   a. Hint: Describe table, relationships, columns, partition, clusters, etc

3. The target users: data analysts want to understand the data, query it for exploration and create dashboards from it.
    a. *Provide them with some example queries (SQL) for your model:*
        i. How many trip offers have been published last month?
        ii. What country had the highest number of publications last month?

Hint - Go over https://www.blablacar.fr/ & use the platform to publish a test ride!

# 2. Python exercise

Important:
- Please submit the code via GitHub repository (access to be shared with interviewers)
- Documentation for executing your code
- DO NOT setup the Airflow infrastructure (docker, etc)

In this exercise, the goal is to write a Python script which queries an API & stores the results in a table as raw data (You can choose any DB solution).
We are going to use the public API for "Transport for The Netherlands" which provides information about OVAPI, country-wide public transport (API description: https://github.com/skywave/KV78Turbo-OVAPI/wiki/)
We will use the Per Line endpoint

Base_url : http://v0.ovapi.nl/
Endpoint: /line/
Authorization: Not needed

#Exercise

Step1: Write a Python script which queries the endpoint & extract response. The data from the response is to be stored in a table (next step)
- Test your script so it can handle edge cases/ missing data/ changes in data structure

Step2: Write SQL Script for storing data coming from Step1
- DDL for Table creation
- DML for insertion/ update

Step3: Improve your solution to handle production scenarios
- Hint: ETL should be able to handle large amount of data (ingestion, transformation & loading)
- Hint: Tables should be optimized for read/ write

Step4: Write your Airflow DAG (Bonus - Do not spend time in Airflow setup)
- You can use PythonOperator

About the Table:

- Use proper data types
- Add technical columns
- Think about partitioning, clustering

About your Script:

- Use Python to complete the assignment.
- No action should be needed before running your code. Otherwise, please provide documentation to initialize your project.
- The code should be written like it is executed every day.
- Your code is idempotent.
- Your code is well documented and organized