Intro

We don’t want to take up too much of your time: don’t spend more than 2h on Part 1 and 30 minutes on Part 2 (unless you enjoy the work!). Include argumentation for your decisions and feel free to demonstrate your areas of interest and expertise.

We value your original thoughts and creativity: please refrain from using external aids (chatbots, language models) when writing the submission. If you do use any, make it clear in your submission: add some reasoning, prompts you used etc.

Part 1

Bolt’s mission is to make cities for people, not cars. To fulfil that mission, we are offering a number of mobility and other services that reduce the need for private cars: ride hailing, micromobility, carsharing and delivery. To further support the cause, we have (hypothetically!) launched another service: Air Boltic.

Air Boltic is a marketplace for sharing aeroplane rides. On one side of the market there are aeroplane operators, ready to fly from anywhere to anywhere. On the other side of the market there are individuals and groups who need to get from A to B.

After a few months of operating in Europe, we see that the service is doing great in some regions, while it hasn’t taken off in others areas. We want to replicate the success in all of our regions and expand outwards, envisioning to facilitate 20% of all aeroplane rides across the globe in 2030. To achieve the goal, we need to understand what are the drivers of growth: the kind of customers we serve well, the kind of use cases we cover (short or long, from where to where, low cost or premium, small or big planes etc). To make sure the business fits into our portfolio, we also monitor daily/weekly/monthly active users, revenue and other metrics that we can compare to our older business lines.

We are providing you with small samples of datasets from our Air Boltic system:

- aeroplane_model.json - a JSON file that holds information about aeroplane models
- Air Boltic data - a Google Sheets file with multiple tabs, representing tables about trips, orders, aeroplanes, customers and customer groups

The aeroplanes table holds information about individual aircrafts that operate on the platform, while the aeroplane models file describes the available models on the market. Trips represent aeroplane rides, while orders represent customers purchasing seats on those trips. You will quickly figure out the other details from the column names in the tables.

Your task is to design a data model that enables monitoring and self-service analysis of the Air Boltic service. Keep in mind the scale we plan to achieve. Apply industry best practices to ensure reliability, scalability, maintainability, high quality and good user experience for your data users.

Please submit:

- a data model in a format that best describes all the data structures that you have envisioned for the use case, for example an ERD (entity relationship diagram) or a set of CREATE TABLE statements in SQL
- a code repository with a relevant part of the data model actually implemented in a tool
    - doesn't have to be the full data model, present what you think shows your skills
    - the repository doesn't have to be a functioning deployment (it's ok to submit code that can't be executed against anything), but the files you choose to submit should be logically complete and demonstrate your skillset
    - if the format of your submission requires, you can provide access to the home task reviewers andre.paal@bolt.eu, kevin.sutanto@bolt.eu and silja.mardla@bolt.eu
- additional context on why you have designed such a data model
- comments on what would you do if you had more time for the task

This is the tooling that we currently have available for analytics:

- S3 for data storage
- Databricks for compute and data exploration
- dbt for data transformations
- Looker for reporting and self-service analytics
- GitHub to keep our scripts organised

We encourage submissions that assume or make use of this stack, but alternatives are also welcome.

Part 2

Let's imagine you have implemented a pipeline that makes your data model from Part 1 available to users. As you can imagine, our business will evolve and become more complex over time, requiring changes in your data model. Let's also imagine you have absolutely no limitations on tooling or resources. **How would you envision the ideal CI/CD process to implement these changes over time?** What kind of environments, processes, tests and tools might be involved?

How would your answer differ in the real world use case where resources are limited and perfect tooling might not be available? What are some of the low effort/short term and high effort/long term things you would suggest we implement?

In your answers, share relevant experience from the past to back up your suggestions. In this part, we do not expect any practical output.