Engineering Manager, Data & Analytics
F1 Technical Test

Overview

You'll build a lightweight ELT pipeline against the OpenF1 API (https://openf1.org/). Demonstrate your Data Engineering approach from raw ingestion through to a dbt modeled layer, as well as CI/CD considerations and an end-to-end architecture description.

Task:

1. Ingest these API endpoints into a raw data layer with idempotent writes using Python.

- Drivers
- Laps
- Locations

2. Build a dbt project that:

- Defines the raw data as sources
- Produces at least one factual / dimensional model using these endpoints

3. Provide a CI/CD workflow skeleton showing how extraction, dbt runs, and tests would be automated. Provide this in an example github workflow. (workflows/dbt-ci.yml)

4. Additionally, a brief end-to-end architecture doc describing how you'd productionize F1 API to a cloud data warehouse such as Databricks, Snowflake or Big Query.

The core focus will be on the idiosyncrasies for this work and navigating best practices. You may choose the document type of your preference including (but not limited to):

- Google Docs (limit of 2 pages), or
- Google Slides (limit 4 slides)

As a guide, the document should outline the rationale of your approach for the following areas:

- Orchestration
- Ingestion
- Transformation & storage
- Testing & monitoring
- Scaling & governance

Repo Structure (Example)

```
None
f1_test/
├─ data/raw/…
├─ f1_extract.py (or module)
├─ requirements.txt
├─ README.md
├─ dbt_project/
│   ├─ dbt_project.yml
│   ├─ profiles.yml
│   └─ models/
│       ├─ raw/
│       │   ├─ schema.yml
│       │   ├─ stg_drivers.sql
│       │   └─ stg_laps.sql
│       └─ mart/
│           ├─ schema.yml
│           ├─ drivers_profile.sql
│           └─ lap_data.sql
├─ .github/
│   └─ workflows/dbt-ci.yml
└─ docs/
    └─ architecture.md
```

Deliverables

- A Git repository (or zip) containing all workings: code, configs, and documentation
- README that explains what's included and requirements for set up
- A "Raw Data" folder with example CSV/JSON files for each endpoint
- Python extraction script(s) / module(s) for the endpoints.
- dbt project directory with:
    - Staging Models
    - Source definitions for the tables (Incl. testing would be advantageous)
    - Modeled tables
    - Example profiles.yml
- CI/CD workflow file showing pipeline steps.
- An architecture document (docs/architecture.md)

Please submit your completed repository or zip file containing all of the above. Good luck!