🔍 BI Engineer Assessment

📝 Objective

Your task is to simulate a basic end-to-end **ETL pipeline** for a gaming data source. The final output should match the format shown in the provided sample dataset screenshot, aggregating casino user performance metrics by demographic and gaming attributes.

🗀 Provided Files (5 CSVs)

You will be working with the following datasets:

1. casinodaily.csv – Raw game performance metrics per user.
2. casinomanufacturers.csv – Contains casino manufacturer names with effective date logic.
3. casinoproviders.csv – Maps casino provider IDs to names.
4. currencyrates.csv – Exchange rate from local currency to EUR.
5. users.csv – Contains user profiles, including birthdate, sex, VIP status, etc.

Each file contains essential fields required for the transformation pipeline.

🎯 Task Requirements

1. **Set Up Your Environment**
    - Load all the CSVs into a SQL-compatible engine (e.g. SQLite, PostgreSQL, DuckDB) **OR** process the files using **Python.**
    - Document the process and reasoning behind your data structure choices.
2. **ETL Goals**
    - Perform **data cleaning**
    - Ensure you always use the **latest manufacturer record**
    - Perform an **age calculation**
    - Apply **currency conversion** for GGR and Returns using currencyrates.csv

- **Aggregate** the metrics at the level of:

Date, Country, Sex, AgeGroup, VIPStatus, CasinoManufacturerName, CasinoProviderName

- **Convert Age into groups:**

Under 18
21-26
27-32
33-40
41-50
50+'

⚙ Additional Guidelines

- Your **ETL process must be controlled and executed from Python**.
  You can combine **Python and SQL** as needed, for example, using SQL queries embedded within Python scripts or notebooks (e.g., with pandas, duckdb, sqlalchemy, or sqlite3).
- The ETL should be built to **run daily**, based on a configurable date range or current date logic.
- You are free to use any Python tools or libraries to support data extraction, transformation, joining, and aggregation.
- Your final code can be submitted as:
- A GitHub repository (preferred)
- A Jupyter Notebook
- A standalone Python script with comments
- A PDF or Markdown report including code and documentation

🧠 Bonus Case Study: Tableau + Big Data

You've been asked to expose a **Gold-level table** (highly curated, production-ready data) containing **over 100 million records** to **Tableau** for dashboarding purposes.

As a BI Engineer, you're responsible for ensuring performance, scalability, and maintainability of the reporting solution.

✅ Your Task

Describe **how you would expose this data to Tableau**.

🏵 What We're Looking For

- Your ability to **evaluate trade-offs** (e.g., performance vs freshness, complexity vs scalability).
- A clear **recommendation of the best approach** for this scenario.
- Any **specific tools, technologies, or strategies** you would use.
- Considerations for **future growth**, **refresh schedules**, and **user experience**.

This is a theoretical question, feel free to write your answer as a short design document, a markdown cell in a notebook, or just a clean explanation in plain text.