Data Developer Exercise – Dice Game Data Processing

Simulated Dice Game Data Pipelines & Analysis (80 Marks)

Description:

It's the start of the year 2025. You are being hired into a fictional game app development company who has had a publicly available "Dice Game" app on the market for the last year (2024). It has been available for both limited free-to-play and unlimited subscription-based play online through both browser and mobile app. The company has hired you to process and analyze the data they've been collecting from their user base, including user play sessions, user registrations and subscriptions, user payment history, and more. Based on the first year of the game app's launch, the company is looking to understand a forecast of what they might expect for 2025, where they are performing or underperforming, and any behavioural characteristics they might glean about their users.

The tasks expected of you in your role are:

- Analyze the source data model/data to understand what has been collected.
- Using pyspark, pandas, or the python-based data processing framework of your choice, design an application that processes and transforms the data into a local "data warehouse" or "data lake" structured using either Data Vault 2.0 or Star Schema (Dimensional) patterns. NOTE: Producing these structures into local files (either CSV or parquet) when the application runs is perfectly fine.
- Implement a series of unit tests and/or data quality validations in your application that ensure your code and end results are structured correctly.
- Leverage the application itself or the end data produced to derive 2-3 key insights that might identify to the company what they might need to know for 2025. Examples:
    - How many play sessions took place Online vs on the Mobile App?
    - How many registered users opted for a onetime payment vs a subscription?
    - How much gross revenue was generated from the app?

Scoring:

- Does the application run to completion without errors and produce the intended results? 50%
- Does the application code look well organized? Classes, functions, scripts, and/or unit tests created that match the problem description and tasks. 30%
- Would it be easy to enhance the application for future sources and needs? 10%

- Did the resulting datasets and insights satisfy the problem description? 10%