Data Engineering Challenge

Nordeus - Job Fair 2023

About the challenge

It's May 22, 2010. About two weeks had passed since the release of a football manager game, which launched on May 8th, 2010. It's about time we see how the world is reacting to a game we launched, and that is where we need your help. Time to analyze some data!

We want to see how our marketing campaigns are panning out and how much time a user spends playing the game(how exciting our game is for users)!

We are particularly interested in 2 types of users: organic and paid.

Organic users: These are the people who find and use a mobile app or website without the company having to pay for ads to get them. For example, if you tell your friends about a cool new app, and they start playing it because of your recommendation, those new users are considered organic.

Paid users: These are the people who start using an app or website because the company paid for marketing campaigns to get their attention. So, when you see an ad for a game on your favorite social media platform and decide to try it out, you become a paid user because the company spent money on that ad to attract you.

Game developers use user **sessions** to see how long people play, what they do during a session, and if they come back to play more. It helps them make the game better and see how much people enjoy it.Think of a user session in a game like one 'session' of playing. It starts when you login into the game and ends when you logout and it must be at least 1 second long(keep in mind that a user can login on one day and logout on the next).

You are given as input a dataset (JSONL format) where each line is a valid JSON object that represents an event generated by a football manager game. Each event emitted from the game has several fields/parameters that describe the action that is happening inside the game. Some examples of events are: the user has registered, logged in, or made a transaction, the match started, the goal scored, the match ended, the user logged out, etc.

What is your goal?

For our analysis, we are interested in four types of events: registration, login, logout, and transaction.

- Registration event is generated when a user registers. At that time, the user receives a unique identifier that we use for all other events as a way to keep track of them. This event also includes the user's country of origin, the OS of the device they use, and the marketing campaign from which the user originated.
- Login event is generated each time a user logs into the game.

- Transaction event is generated each time our user decides to buy something that is offered in our game (In-App Purchase).
- Logout event is generated each time a user logs out of the game or the app is closed(user gets automatically logged out when the game is closed).

Below you will find the description of the fields each event contains, and also the fields specific to each one. All the fields are required

Dataset (events.jsonl)

Parameter name	Parameter type	Parameter description
event_id	INT	Unique identifier representing an event
event_timestamp	INT	Time of event represented as Unix time
event_type	STRING	One of the following: registration, login, logout
event_data	JSON OBJECT	JSON object containing all event-specific data (check event data below)

Required event data for registration

Parameter name	Parameter type	Parameter description
user_id	STRING	Unique identifier representing a user
country	STRING	Country that the user comes from
device_os	STRING	OS of the device user registered from. Valid values are iOS, Android and Web
marketing_campaign	STRING	Name of marketing campaign user came from(can be NULL or empty string)

Required event data for transaction

Parameter name	Parameter type	Parameter description
user_id	STRING	Unique identifier representing a user
transaction_amount	FLOAT	Amount that the user paid in the transaction. Valid values are 0.99, 1.99, 2.99, 4.99, 9.99
transaction_currency	STRING	Currency of the amount the user paid. Valid values are EUR, USD

Required event data for login and logout

Parameter name	Parameter type	Parameter description
user_id	STRING	Unique identifier representing a user

Your task is to process, clean, and transform the source dataset into a new data model that will be capable of answering certain questions about the state of the game.

Requirements

Data cleaning requirements:

- Discard duplicate events (event_id uniquely identifies an event)
- Discard invalid events. Events are considered invalid if they do not meet the requirements specified above.
- Use common sense to discard other events that don't make any sense.

- Document what and why you discarded

API requirements:

- Get user level stats
  - Input:
    - user_id (required): represents a unique identifier for a user, described above.
    - date in format YYYY-MM-DD (optional): If no date is specified, then calculate all-time stats for the specified user. If it is specified, then calculate user stats for the date specified.
  - Output:
    - Country of the user
    - Name of the user
    - Number of logins
    - How many days have passed since this user last logged in. If no date is specified, calculate how many days have between the last login, and the last date that the given dataset has data for.
    - Number of sessions for that user on that day . If no date is provided give the all time number of sessions
    - Time spent in game in seconds for that day or all time if no date is provided

- Get game level stats
  - Input:
    - date in format YYYY-MM-DD (Optional). If no date is specified, then calculate all-time stats. If it is specified, then calculate stats for that day only.
    - Country (optional). If this argument is specified, group all the results by country.
  - Output:
    - Number of daily active users. If a user has had a login that day, he is considered a daily active user. If no date is specified, then this column represents the number of all users that had at least one login.
    - Number of logins
    - Total revenue (in USD)
    - Number of paid users
    - Average number of sessions for users that have at least one session
    - Average total time spent in game for users that spent at least one second in game

Note: you are given a file (JSONL format) that contains exchange rates necessary for calculations.

Expectations

The input file should be processed **only once** and the target data model should be somehow persisted (in-memory, file, database, etc.) so that queries can be executed efficiently. It is not recommended to process the input file each time a query is executed.

You have the freedom to implement the API in any way you find appropriate and in any language (Python, Java, SQL, etc.) for example:

- REST API implemented in any language
- CLI APP implemented in any language
- Other

One important thing here is that you should add a **documentation** file (possibly README.md). This document should explain your chosen approach for the project and provide clear, user-friendly instructions on how to use the application, including how to start it and install any required dependencies.

Bonus points for

- Implementing a detailed and careful data cleaning process
- Storing the target data model in a SQL database and using SQL to query data
- Implementing the solution as a well-documented REST API

Submission

Please note that each challenge can be completed in less than 2 weeks (e.g. a few days) but we wanted to be mindful of your faculty obligations.

The submission should be sent via email to jobfair@nordeus.com with a link to your GitHub/Gitlab/Bitbucket repository. Email subject should be: **Data Engineering Challenge.**

Challenge is open until November 20, 2023 end of day. Good luck!