Socialpoint

Data Engineer Technical Test

Source data

The main data sources that feed our Data Warehouse are **log data generated by our mobile applications and third parties API's data**. Log data is in json format and tracks all the actions done by players on mobile applications. The data coming from APIs provides us with information about the impact of our games on social media, as well as universal open data.

Among the log data generated by our games, we are using for this exercise the following sources: **log_monetization_transaction.json** and **log_user_register.json**.

Log_user_register

We receive one row for each user who has registered in a mobile application.

Fields	Type	Description
client_mobile_device	varchar(64)	Name of the client device
client_mobile_os	varchar(64)	Operating System is using the client device
datetime	timestamp	Datetime when the player registered on the app.
ip_country	char(2)	Country provided by the IP
platform	varchar(8)	Mobile operating system (iOS or Android).
user_id	varchar(64)	Unique identifier of the player.
version	varchar(32)	Game application version.
app_code	varchar(10)	Internal code of the game
app_name	varchar(64)	Name of the game seen on the store

Log_monetization_transaction

We receive a row per each purchase done in our mobile applications.

Fields	Type	Description
datetime	timestamp	Datetime when the transaction was performed
game_basic_level	smallint	Game level when the transaction was performed
ip_country	char(2)	Country provided by the IP
order_amount_gross	double precision	Money spent on the transaction
order_payment_provider	varchar(16)	Provider for the transaction
order_transaction_id	varchar(255)	ID of the transaction
platform	varchar(8)	Mobile operating system (iOS or Android).
user_id	varchar(64)	Unique identifier of the player.
version	varchar(32)	Game application version.
app_code	varchar(10)	Internal code of the game
app_name	varchar(64)	Name of the game seen on the store
currency	varchar(10)	Currency used to purchase.

Technical Assessment

We would like to move from user and purchase json files to a more compact summary information. Sadly, we have been very busy lately and haven't had enough time to accomplish it. Could you please help us? :-)

The main objective of this task is to load to the database the log data provided together with the currency exchange data and develop an ETL process to obtain a Data Mart that helps the company to answer questions like:

- Number of players
- Revenue generated
- Number of customers
- Average of revenues per user
- Number of transactions
- Average of transactions per user

Socialpoint

To accomplish this project you will find a zip file called sp-technical-test.zip that contains a small framework that will help you to complete the tasks with Python and SQL. Please read the readme file inside the zip file for installation instructions.

The tasks needed to complete the project are the following:

1. Complete the **load_raw_data.py** (src/scripts/load_raw_data.py) file. Your code has to read the raw data provided and insert it to the database. Keep in mind that data can be reloaded at any time. You will find the files to load in the src/log_files folder.
2. Complete the **load_currency_exchange.py** (src/scripts/load_currency_exchange.py) file. This will pull and store the needed data from this API to perform the currency exchange. Keep in mind that data can be reloaded at any time.
3. Once data is loaded, Include the needed SQL to obtain the requested data mart. Add your queries to the **etl.sql** (src/sql_projects/etl/queries/etl.sql) file.
4. Add a brief description of the data modelling strategy used to **data-modeling-strategy.txt** (src/data-modeling-strategy.txt)
5. Add the SQL to answer the following questions to the **queries.sql** (src/sql/queries.sql) file.
    a. We would like to know the **average revenue per user** by application and platform.
    b. We would like to know the **average revenue per customer** (player who has at least one transaction) by registration country.
    c. We would like to know the generated revenue by **register date** for users registered in May 2020.

Register Date	Revenue (USD)
2020-05-01	150
2020-05-02	30
...	...

    d. We would like to know the accumulated revenue by **register date** and **days since register** (days 0, 3, 7, 15 and 30) for users registered in June 2020 ordered by register date and day since register ascending.

    For example, in the table below the accumulated revenue column tells us the revenue generated by players who registered on the day X since register (Days since register column). This number represents how many days since registration have passed. Day 0 is the same day of registration, day 1 is day of registration + 1 day, etc. E.g. if a player registers on 2020-06-01 23:00:00 and makes a purchase of 2$ on 2020-06-02 09:00:00, revenue on day 0 is 0$ and revenue for day 1 is 2$.

Register Date	Days since Register	Accumulated Revenue (USD)
2020-06-01	0	5.00
2020-06-01	3	12.50

2020-06-01	7	15.50
2020-06-01	15	17.00
2020-06-01	30	18.00
2020-06-02	0	3.00
2020-06-02	3	9.50
2020-06-02	7	14.00
2020-06-02	15	17.75
2020-06-02	30	22.00
...	...	...

Considerations:

- Data regarding purchases is required to be in dollars.
- 'Unknown' values in the currency field could be considered as dollars.
- In case platform value is null in log_monetization_transaction.json, use the platform on which the user registered the first time.
- Please try to keep your answers as simple as possible - don't over-engineer - and add your code only to the suggested files - don't create any new files -.