Fetch Rewards, Analytics Engineering

Tools used: Python - Jupyterhub (Data Transformation & Cleaning)
SQL - Sqlite Enginer using DBeaver (Business Problem queries)

High Level Documentation

First: Review Existing Unstructured Data and Diagram a New Structured Relational Data Model

Dataset Description.

Three main datasets are presented to us
i. **Receipts** - details about the receipts, number of points earned, items list and purchase date and other details that are available in a receipt.
ii. **Brands** - details about the Brands like Barcode, Category Code, and other Brand details
iii. **Users** - Details about when the user was created, when they last logged in and their sign up source.

The source formats of the files are JSON files.

While designing a data pipeline or a framework.
After listing the various datasets based on business needs, it is essential to identify sources, cadence, file types, size, and some metadata that help guide decisions about resource utilization that can have cost implications.

Metadata observations for the datasets above have been highlighted in the Metadata Section in the Jupyter Notebook.

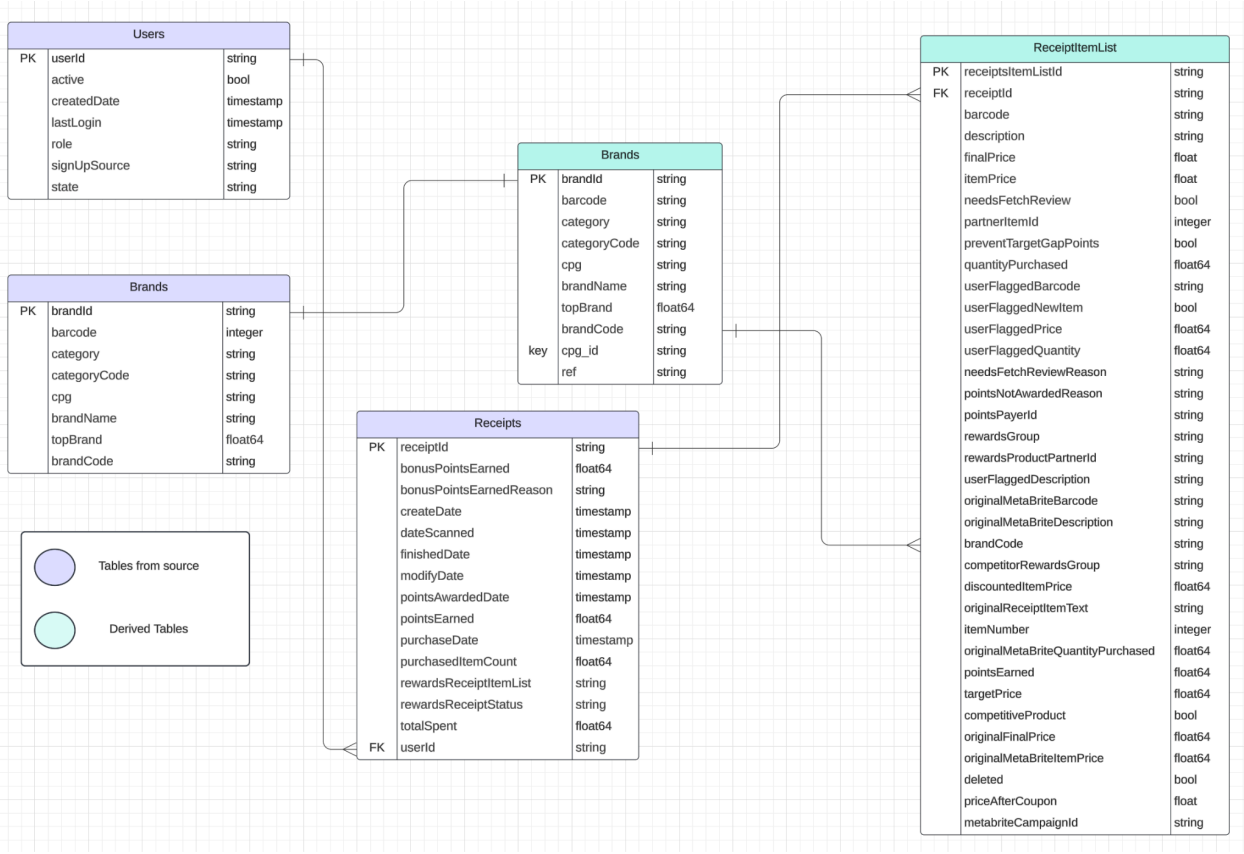
Some brief observations are listed here.
i. **Receipts**, **Brands**, and **Users** have **1119**, **1167** and **495** records respectively.
ii. Each of the datasets has some columns with **NaN** or **null** values that might warrant a null handling process during the transformation phase based on requirements
iii. **_id -** is the primary key in all the three datasets. Need to rename the columns appropriately.

iv. Some of the values within the json objects are within another object or in a different format, these values need to be standardized and cleaned across the three datasets.
v. the column `rewardsReceiptItemList` from the receipts dataset has an unstructured data stored in a dictionary. There is no definite schema that is identifiable across all records, so we will need to unpack the dictionary and define the table and introduce appropriate data handling mechanism for it.

Data Cleaning & Transformation

The values nested with json objects like **_id**, **date fields, cpg** from brands and unpacking **rewardsReceiptItemList** are all handled in this section of the notebook.
The transformations were performed in the following steps.

- The data were read from the json files into the respective dataframes **users_df**, **brands_df** and **receipts_df**
- Data from cpg in brands_df was unpacked and added into 2 new columns **cpg_id** and **ref**
- Each of the dataframe were then treated (extract_data) to remove the data from the objects and store them appropriately into a new data frame, namely **users**, **brands** and **receipts**
  (The reason for storing them in a new data frame is to simulate the process of having the original raw file saved in a transient object in order to not rerun file ingestion process over and over again incase of infrastructural challenges)
- Null value treatment performed. **'Unknown'** for **string** nulls and
  0 for **integer** and **float** based columns.
  Null value treatment **not performed on Boolean** because true and false are sensitive indicators, and can be treated accordingly based on the business use case.
- The data from **rewardsReceiptItemList** is then unpacked into a new dataframe which is similar to designing a fourth table by the name `receiptsItemList`
- Aside from unpacking the values, a new uuid is generated '**receiptItemListId**' which is created using **uuid5,** based on the unique combination of **receiptId** and **partnerItemId**
- Datatype conversion into their suitable types can be helpful for comparison and querying once the SQL engines for Querying and Reporting reference them.



ERD Diagram

Tables from source refer to the tables that we directly get from the files as is with minimal data transformation

Derived Tables are the ones where we do some kind of a transformation and either create new columns or a completely new dataframe (table)

Second: Write queries that directly answer predetermined questions from a business stakeholder.

1. What are the top 5 brands by receipts scanned for most recent month?

Approach: defined the Top 5 Brands by the frequency (count) of the brandCode's appearance in receiptsItemsList for the latest month scanned.

Link to Query:

https://github.com/arclive202/FetchRewards_AnalyticsEngineer/blob/main/BusinessProblem_1.sql

Caveat: Since there are a large amount of BrandCode's missing from the receiptItemsList for the receipts that were scanned, this implies that the internal data capturing mechanism needs to improve its ability to match the Brand appropriately with the items from the scanned receipt and map them accurately before sending over the data via the JSON file. And improvement in the Receipt Scanned to Brand Match Rate will enhance the reporting of this Business Problem.

2. How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?

Approach: Identifying the Top 5 Brands for the previous month and Current Month remains the same (by count desc)

However, displaying the data is where there is a difference.

I selected the top 5 Brands from the Previous Month and Current Months and displayed their performance Side by Side in the following format

BrandCode	Recent_Month_Counts	Recent_Month_Rank	Previous_Month_Counts	Previous_Month_Rank

Link to Query:

https://github.com/arclive202/FetchRewards_AnalyticsEngineer/blob/main/BusinessProblem_2.sql

Caveat: Similar to the issue with the previous question, we need to work with the upstream data aggregators to match the brandCode data better with the receipts that was scanned and define an accurate brandCode key mapping to the brand details.

3. When considering average spend from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?

Approach: Grouping by the rewardsReceiptStatus where status = ('FINISHED','REJECTED')
And displaying the average spend.

A-Z rewardsReceiptStatus
FINISHED
REJECTED
FLAGGED
SUBMITTED
PENDING

Based on the distinct rewardsReceiptStatus observed **Finished** can be considered to be the same as **'Accepted'**

Keeping this in mind the average spend of **'Finished'** is greater than **'Rejected'**

A-Z rewardsReceiptStatus	123 Average_Spend
FINISHED	80.85
REJECTED	23.33

Link to Query:

https://github.com/arclive202/FetchRewards_AnalyticsEngineer/blob/main/BusinessProblem_3.sql

Caveat: Here the status 'Finished' is treated as 'Accepted' we need to align the business expectation with the data that is being captured in order to ensure that these values mean the same and get a soft approval on these conditions.

4. When considering total number of items purchased from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?

Approach: Sum of PurchasedItemCount over the respective rewardsReceiptStatus groups to take a look at the number of purchased items

A-Z rewardsReceiptStatus	123 sum(purchasedItemCount)
FINISHED	8,184
REJECTED	173

Link to Query:

https://github.com/arclive202/FetchRewards_AnalyticsEngineer/blob/main/BusinessProblem_4.sql

Caveat: Same as 3.

5. Which brand has the most *spend* among users who were created within the past 6 months?

Approach: Sum of the total spent grouped by brandCode from receiptItemsList for users created in the last 6 months by getting a list of all the receiptIDs that were scanned and summing up the ***finalPrice * quantityPurchased*** from the receiptItemsList for those particular receipts and grouping by brand

The brand with the most spend observed was actually ***unknown.*** However, since unknown was a null treated value on brandCode on whom we did not have any data. It could possibly mean some kind of an error in scanning or data not being captured properly. However, the next best spend by Brand is **HEMPLER'S.**

Where the total amount spent on it by users created in the last 6 months is $5611.11

A-Z brandCode	123 total_spend
unknown	54,547.87
HEMPLER'S	5,611.11

Link to Query:
https://github.com/arclive202/FetchRewards_AnalyticsEngineer/blob/main/BusinessProblem_5.sql

Caveat: One way to handle **unknown** BrandCode can be by excluding them entirely from all the calculations, but it would be better to improve the data capturing capabilities.

6. Which brand has the most transactions among users who were created within the past 6 months?

Approach: Similar to the previous question, find the list of users created in the last 6 months and then looking at the receipts scanned by them we take a sum of the total quantityPurchased group by brandCode to take a look at the most transacted product.

The most transacted brand is **unknown,** but like before since these values were treated for null these could possibly mean some error in data capture. Excluding this, the Brand that was transacted the most number of times in HY-VEE.

A-Z brandCode	123 total_transactions
unknown	3,332
HY-VEE	308

Link to Query:

https://github.com/arclive202/FetchRewards_AnalyticsEngineer/blob/main/BusinessProblem_6.sql

Caveat: Same as 5.

Third: Evaluate Data Quality Issues in the Data Provided

There are three main data quality checks that we decided to perform on all the data sources at a holistic level.
i. Duplicate Checks
ii. Null Value Checks
iii. Data Recency Checks

The approach to identify these checks were done in the form of defining one function for each of the checks that reports a report for the respective dataframe that was passed.

We can set these up as an alerting mechanism in the data pipeline that raises and alert takes appropriate action before proceeding to the following steps of data processing in order to generate reports.

Duplicate Checks.
i. Duplicates were observed in the Users dataset on the userId column which can lead to issues related to primary key uniqueness checks and have additional issues while joining queries on Users table for reporting purposes.

	Column	Null Count	Null Percentage
0	_id	0	0.00
1	barcode	0	0.00
2	category	155	13.28
3	categoryCode	650	55.70
4	cpg	0	0.00
5	name	0	0.00
6	topBrand	612	52.44
7	brandCode	234	20.05
8	cpg_id	0	0.00
9	ref	0	0.00

Action: Need to have a discussion with the data and business stakeholders on the approach required to prune the duplicate user data

ii. The remaining dataframes did not have any duplicates on their respective primary key.

Null Value Checks.

i. For some of the users that were created, there were null values observed on the lastLogin time, signUp source and State (This could have an impact on analysis on the user grain)

ii. There were a lot of null values observed for different columns across Brands data
Especially on features like category, categoryCode and brandCode which are considered as essential keys/referrences to do analytics based on brand

	Column	Null Count	Null Percentage
0	_id	0	0.00
1	barcode	0	0.00
2	category	155	13.28
3	categoryCode	650	55.70
4	cpg	0	0.00
5	name	0	0.00
6	topBrand	612	52.44
7	brandCode	234	20.05
8	cpg_id	0	0.00
9	ref	0	0.00

iii. Following are the null value observed for receipts we have data missing on columns like finishedDate, rewardsItemList

	Column	Null Count	Null Percentage
0	_id	0	0.00
1	bonusPointsEarned	575	51.39
2	bonusPointsEarnedReason	575	51.39
3	createDate	0	0.00
4	dateScanned	0	0.00
5	finishedDate	551	49.24
6	modifyDate	0	0.00
7	pointsAwardedDate	582	52.01
8	pointsEarned	510	45.58
9	purchaseDate	448	40.04
10	purchasedItemCount	484	43.25
11	rewardsReceiptItemList	440	39.32
12	rewardsReceiptStatus	0	0.00
13	totalSpent	435	38.87
14	userId	0	0.00

Data Recency Checks

If we want to set up a continuous data process which is expected to run on a daily cadence for the business team to generate reports on a daily/weekly/monthly basis.
It is highly important to set data recency checks on tables like Users and Receipts to ensure that we capture the latest up to date data from our data sources.

An example of this can be.

i. Recency Check on Users & Receipts can ensure that our data is up to date and covers all the day's data until the previous day. If the data process runs on a daily cadence

```
1 user_report
```

	Column	Max Date	Threshold Date	Valid
0	createdDate	2021-02-12 14:11:06	2025-01-22	False
1	lastLogin	2021-03-05 16:52:23	2025-01-22	False

An alert can be raised on the threshold not being met which can trigger an action from the data team to look into the upstream data deliveries to ensure data quality and integrity is maintained.
