

Take home assessment procedure followed -

Step 1 - JSON data structuring process

I followed the below process to structure the given json data (Tools & Technologies used - Jupyter Notebook, Python, Stack overflow, Lucid Chart)

- Converted each json file (user, brand and receipt) to pandas dataframe by normalizing the columns. Normalization was required since the json data was nested.
- The receipt data contained a column called rewardsReceiptItemList which was a list of json data and normalizing it within the receipt dataset would not result in a good structure.
- Created another table for rewardsReceiptItemList by normalizing each json within the rewardsReceiptItemList where each row in the table would have details for individual items scanned and the receiptId associated with it.
- Total four tables - user, brand, receipt and rewardsitemlist. Created a relational data model for these four tables on Lucid chart
- The four tables were then loaded to the postgresql database by creating a sql engine and establishing connection with postgresql server.

Step 2 - Queries to answer questions from business stakeholders

Accessed the four tables and ran queries (Tools & Technologies used - Postgresql Admin, PostgreSQL)

- When considering *average spend* from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater? [I did not see any rewardsReceiptStatus with a value of 'Accepted'. Hence, I used 'FINISHED' in place of 'Accepted'. FINISHED rewardsReceiptStatus has a higher average totalSpent from the receipts](#)

```
5 -- 3. When considering average spend from receipts with 'rewardsReceiptStatus' of
6 -- 'Accepted' or 'Rejected', which is greater?
7 -- REJECTED or FINISHED
8 select "rewardsReceiptStatus", avg("totalSpent") as average_spend
9 from receipt
10 where "rewardsReceiptStatus" = 'REJECTED' or "rewardsReceiptStatus" = 'FINISHED'
11 group by ("rewardsReceiptStatus");
12
13
14 |
15
16
17
```

	rewardsReceiptStatus text	average_spend double precision
1	REJECTED	23.326056338028184
2	FINISHED	80.854305019305

- When considering *total number of items purchased* from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater? I did not see any rewardsReceiptStatus with a value of 'Accepted'. Hence, I used 'FINISHED' in place of 'Accepted'. FINISHED rewardsReceiptStatus has a higher number of items purchased from the receipts

```

14 -- 4. When considering total number of items purchased from receipts with
15 -- 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?
16 -- REJECTED or FINISHED
17 select "rewardsReceiptStatus", sum("purchasedItemCount") as total_no_of_items
18 from receipt
19 where "rewardsReceiptStatus" = 'REJECTED' or "rewardsReceiptStatus" = 'FINISHED'
20 group by ("rewardsReceiptStatus");
21
22
23

```

Data Output
Messages
Notifications

	rewardsReceiptStatus text	total_no_of_items bigint
1	REJECTED	173
2	FINISHED	8184

Step 3 - Evaluate Data Quality issues

Below are some of the data quality issues I identified (Tools & Technologies used - Jupyter Notebook, Python) -

- **Duplicate records** - The users data has 283 duplicate records.

```
#duplicate records
print("-----Users-----")
print(user_df.shape)
test_user_df = user_df.drop_duplicates()
print(test_user_df.shape)

print("-----Brand-----")
print(brand_df.shape)
test_brand_df = brand_df.drop_duplicates()
print(test_brand_df.shape)

print("-----Receipt-----")
print(receipt_df.shape)
test_receipt_df = receipt_df.drop_duplicates()
print(test_receipt_df.shape)

print("-----Rewards List-----")
print(rewards_item_list_df.shape)
test_rewards_df = rewards_item_list_df.drop_duplicates()
print(test_rewards_df.shape)
```

```
-----Users-----
(495, 7)
(212, 7)
-----Brand-----
(1167, 9)
(1167, 9)
-----Receipt-----
(1119, 14)
(1119, 14)
-----Rewards List-----
(6941, 35)
(6941, 35)
```

- **Null values** - What is the nature of these null values? How do we treat them?

```
#null values
print("-----Users-----")
print(test_user_df.isnull().sum()) #using the user data without duplicates

print("-----Brand-----")
print(brand_df.isnull().sum())

print("-----Receipt-----")
print(receipt_df.isnull().sum())

print("-----Rewards List-----")
print(rewards_item_list_df.isnull().sum())
```

```
-----Users-----
userId      0
active      0
role        0
signupSource 5
state       6
createdDate  0
lastLogin   40
dtype: int64
-----Brand-----
brandId     0
barcode     0
category    155
categoryCode 650
name        0
topBrand    612
cpgId       0
cpg_ref     0
brandCode   234
dtype: int64
```

```
-----Receipt-----
receiptId      0
bonusPointsEarned 575
bonusPointsEarnedReason 575
pointsEarned 510
purchasedItemCount 484
rewardsReceiptStatus 0
totalSpent 435
userId 0
createDate 0
dateScanned 0
finishedDate 551
modifyDate 0
pointsAwardedDate 582
purchaseDate 448
dtype: int64
```

-----Rewards List-----

```

receiptId      0
barcode        3851
description     381
finalPrice     174
itemPrice      174
needsFetchReview 6128
partnerItemId  0
preventTargetGapPoints 6583
quantityPurchased 174
userFlaggedBarcode 6604
userFlaggedNewItem 6618
userFlaggedPrice 6642
userFlaggedQuantity 6642
needsFetchReviewReason 6722
pointsNotAwardedReason 6601
pointsPayerId  5674
rewardsGroup   5210
rewardsProductPartnerId 4672
userFlaggedDescription 6736
originalMetaBriteBarcode 6870
originalMetaBriteDescription 6931
brandCode      4341
competitorRewardsGroup 6666
discountedItemPrice 1172
originalReceiptItemText 1181
itemNumber     6788
originalMetaBriteQuantityPurchased 6926
pointsEarned    6014
targetPrice     6563
competitiveProduct 6296
originalFinalPrice 6932
originalMetaBriteItemPrice 6932
deleted         6932
priceAfterCoupon 5985
metabriteCampaignId 6078
dtype: int64

```

- **Formatting** - The date format by default were in the below format. I converted them to datetime format for better readability and analysis.

Before -

createdDate	lastLogin
1609687444800	1.609688e+12
1609687444800	1.609688e+12

After -

createdDate	lastLogin
2021-01-03 15:24:04.800	2021-01-03 15:25:37.857999872
2021-01-03 15:24:04.800	2021-01-03 15:25:37.857999872

- **Multiple rows** - brandCode has multiple rows in the brand dataset. There are also null brandCodes.

```

3  select * from brand
4  where "brandCode" = 'HUGGIES'
5  ;
6
7
8

```

Data Output Messages Notifications

brandid [PK] text	barcode text	category text	categoryCode text	name text	topBrand boolean	cpgid text	cpgr_ref text	brandCode text
5bd2011f90fa074576779a17	511111704652	Baby	[null]	Huggies	false	550b2565e4b001d5e9e4146f	Cogs	HUGGIES
5c7d9cb395144c337a3cbfb	511111707202	Baby	BABY	Huggies	true	5459429be4b0bfc1e864082	Cogs	HUGGIES

```

24 select * from brand
25 where "brandCode" is null;
26
27

```

Data Output Messages Notifications

barcode text	category text	categoryCode text	name text	topBrand boolean	cpgid text	cpgr_ref text	brandCode text
11111019862	Baking	BAKING	test brand @1612366101024	false	601ac114be37ce2ead4375...	Cogs	[null]
11111102540	[null]	[null]	MorningStar	[null]	5332f5f2e4b03c9a25efd0aa	Cogs	[null]
11111317364	Baking	BAKING	test brand @1605535049181	false	5fb28549be37ce522e165cb4	Cogs	[null]