Hello Sir/ Madam,

Hope you are doing fine!

After analyzing the datasets of Users, Receipts, and Brand, I had some queries,
It would be great if you could provide your insights on them.

Questions regarding the data:

I have one question regarding the brands.json and items list in receipts.json.
columns of brands.json are:

_id, barcode, brandCode, category, categoryCode, cpg, topBrand, name

while columns in the items list of receipts.json are:

barcode, description, finalPrice, itemPrice, needsFetchReview, partnerItemId, pointsNotAwardedReason, pointsPayerId, preventTargetGapPoints, quantityPurchased, rewardsGroup, rewardsProductPartnerId, userFlaggedBarcode, userFlaggedDescription userFlaggedNewItem, top brand, userFlaggedPrice, userFlaggedQuantity

So, I have a query about whether some of these columns can be merged into one to remove some redundancy.

Ways to discover data quality issues:

For recognizing the data quality issue, I applied EDA (Exploratory Data Analysis) to the dataset. I have converted **JSON** to python library **pandas** library **data frame**. I calculated the number of rows and unique row count of most likely primary IDs. There were many date columns hence I checked for any invalid dates for instance any transaction happening in the future. For categorical variables like the **role** column in *users.json,* I calculated frequency distribution to find any discrepancy present.

What information needs to know to resolve the issue?

For resolving the data quality issue, I need to understand the more detailed understanding of columns in *rewardsReceiptItemList* (The items that were purchased on the receipt) in *receipts.json*, so that I can remove redundancy as much as possible to save storage.

What other information do I need to optimize the data assets?

One change that I recommended while performing EDA (Exploratory Data Analysis) on the dataset, is to have a parent class named **Person**, and subset classes of **Customer**, and **Fetch-staff**. I required information about fetch-staff members that the database would require to save like employee ID.

What performance and scaling concerns do I need to anticipate?

In the future when it needs to be handled on large scale, there could be latency issues in processing multiple requests simultaneously so, for that cause duplicate database idea would be very helpful. During one request, the database gets locked and other requests get in the queue.

Looking forward to hearing back from you soon.

Thank you
Deepak Singhal
(Masters student in Business Analytics at the University of Illinois, Chicago)