Hello,

Thank you for giving me the opportunity to work on this exercise. I had fun and my results are explained below.

**These are the questions I have about the data:**

1. What is the difference between barcode, userFlaggedBarcode and originalMetaBriteBarcode?
2. Why do not all items in the receipt table have a valid barcode column?
3. Can I get a data dictionary for rewardsReceiptItemList object?
4. Are some columns in the rewardsReceiptItemList object optional? For example, some items are missing barcodes, others are missing itemDescription.
5. What does the cpgId in the cpg collection refer to?
6. Why is there no ‘Accepted’ status for the column rewardsReceiptStatus in the receipts table? The closest to ‘Accepted’ status was ‘Finished’, are they the same?

**Following are the data quality issues and some steps I took to resolve them:**

I started off by opening the json documents in Notepad++. After looking at all the 3 documents, it was clear that all of them were missing a comma between individual json objects. I wrote a code to add commas to make the raw data a proper json document.

Next up was to figure out the meaning of $ sign before column names in the json document. I then wrote a code to remove them and replace it with blanks as it did not contribute to the data.

After that, I found out that in the json documents the dates were in Unix time in milliseconds rather than seconds. I had to divide the values by 1000 and then use the sql datetime function in SQL to get the date in the desired format.

Another concern that I found out is missing barcodes for some receipts in the receipts\_item table. This causes loss of data when we try to join receipts\_item and brands\_table tables. If we can somehow get barcodes for all items in a receipt or try to back fill it using a different column, we can query all the receipts for analyses.

Finally, the receipts table had a nested object rewardsReceiptItemList which had a lot of columns inside it. In order to avoid redundancy and store data logically, I decided to normalize the table. I took the values from the rewardsReceiptItemList object into a new table and also inserted the receipt id into that table which would be the key. After that as a receipt can have multiple items, I created a new column ‘itemId’ which was auto incremented for each item in a receipt starting from 1.

**This is the information I need to optimize the data assets.**

Having a data dictionary for the object rewardsReceiptItemList inside the receipt table would have been useful. This object was very important as it had the barcode column which would help us join with the brands table. The column ‘barcode’ had a lot of missing values which made it difficult to join with the brands table. If I had some more context about the data, it would have been easier to join the receipts table with the brands table.The column partnerItemId in the object rewardsReceiptItemList seemed to have unique ids for each item in a receipt. However it was not consistent across all receipts. For some receipts, the id started from 1, while for others it started from 1000. To handle that, I created a new column itemId which always started with 1 for all items in a receipt. Some information about the difference between columns barcode, userFlaggedBarcode and originalMetaBriteBarcode would have been useful in understanding the different codes and their importance in the data.

**Performance and scaling concerns:**

Currently with my relational data model, I do not anticipate any concerns in terms of performance. With the normalized tables, there are no duplicate records which helps in reducing query execution time. With regards to scaling, adding new records to the tables will not impact the execution time.

If we plan to add additional columns in the rewardsReceiptItemList object(receipts\_item table), I would recommend further normalizing this table to avoid redundancy. Also, if we plan to collect more information about items from the receipts, then we will have to create new columns in the receipts\_item table.

Please let me know if you have any questions or need any clarifications.

Best Regards,

Abhijay