FETCH ASSESSMENT - ANALYTICS ENGINEER

Rishabh Gundavarapu

This document is a collection of thoughts about my decision-making process for the first two steps

STEP 1 - Initial thoughts and ER diagram

Looking at the unstructured files given, there is a lot to unpack.
First, some of the files contain nested data, for example: the rewardsReceiptItemList in the receipts file. Each record for this field contains multiple sub values of varying data values. Furthermore, these sub values are not present across all the records which might pose a challenge when gathering specific insights and information from the data.
Second, there is a lot of missing data from a simple visual inspection, but this can be tackled later.
Creating an Entity-Relationship (ER) digram will help us visualize the database structure making it easier to design the database and improve performance when querying for insights especially in the case of unstructured data right now. I am using LucidChart to make an ER diagram.
The diagram I created consists of 5 entities/tables . I wanted to create a fact table as it can help me establish relationships between the data files especially for a data warehouse like this but decided against it as there were only a couple of tables.
Upon inspection, the receiptsRewardsItemList contains info about every item bought in the receipt furthermore, the data is not consistent across the records, meaning we
Will have to manually find out all the unique key columns present across all the records to get a consistent analysis.
I updated the ER diagram after finding all the unique keys and modified receiptsRewardsItemList as RewardsReceiptItemInfo as a separate entity.
The Primary keys in each of the dimension tables are in bold with a bigger font and denoted with PK. The foreign keys are denoted with FK.
Users
Primary key - userId
Receipts
Primary key - receiptId
Foreign key to users table - userId
Brands
Primary key - brandsId
Foreign key to Cpg table - cpgId
Cpg
Primary key - cpgId
RewardsReceiptItemInfo
Foreign key - receiptId to Receipts table
One user can have multiple receipts (one to many)
One receipt might have information about multiple sets of information about rewards for each of the items (one to zero or many)

STEP 2 - Creating the tables and answering business questions using SQL queries
As the data is unstructured and a bit messy (lot of nested values inside certain columns), I wrote functions that will help me clean up the data frames and
Assist me in writing sql queries to answer business questions
These data processing steps make the data analysis easier
I will be using SQLite dialect using pandasql to write sql queries
Creating a relationship between brands and other tables was tricky, I was only able to join brands with others using brandCode which mostly turned out to be null. This must be flagged and dealt with appropriately.
Steps 3 and Steps 4 are in the ipynb notebook and as a pdf respectively.