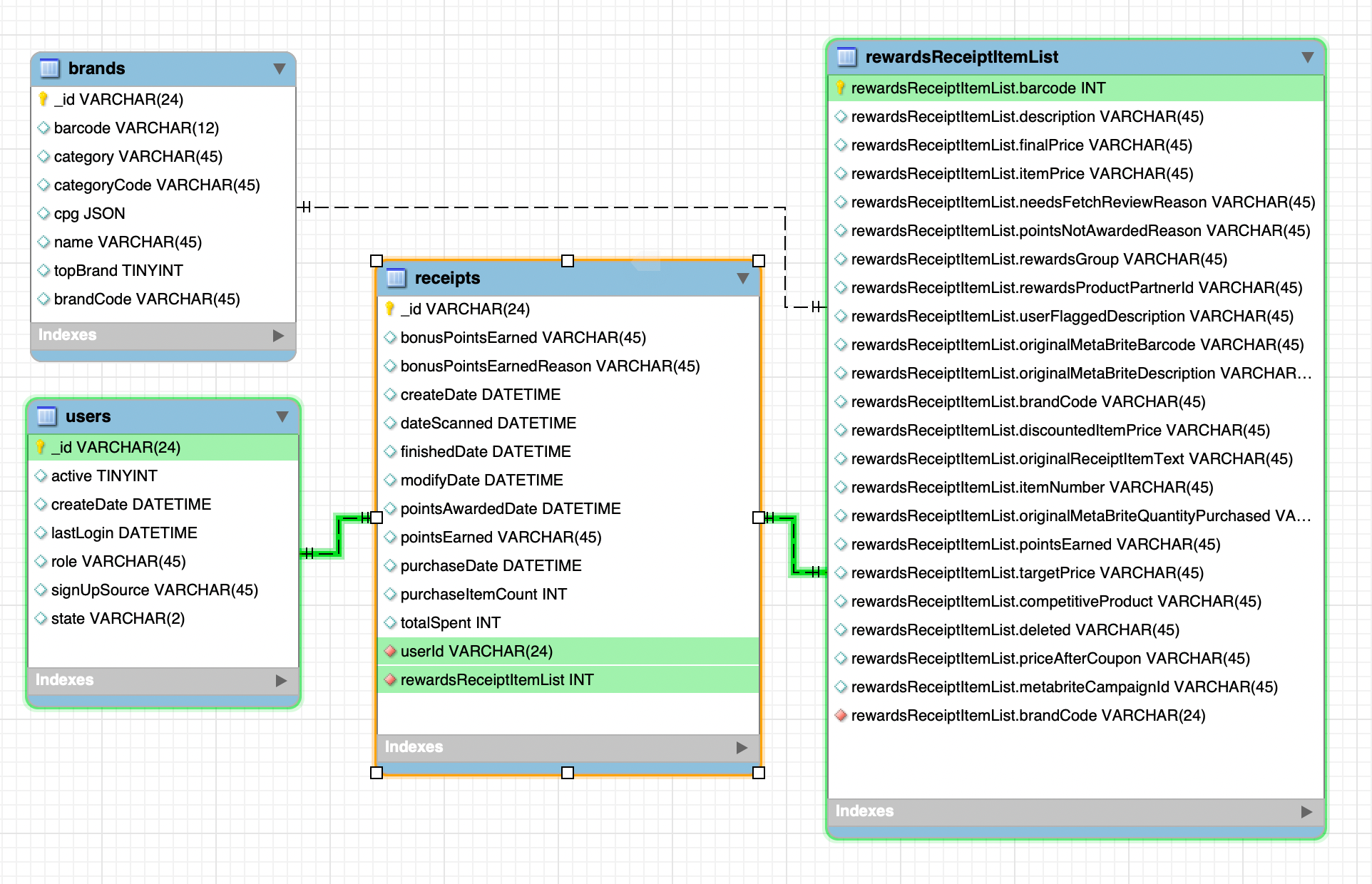
**Jinkai Zhan’s Coding Report**

**First Task - Data Preparation**

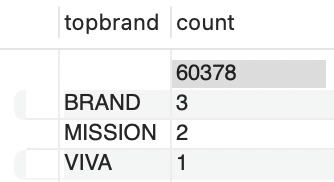
1. Unzip the json format files
2. Import json format files into MongoDB and export csv format spreadsheets
3. Clean the csv format spreadsheets using python: ‘Data Preparation.py’
   1. Substitute the text that is not ascii in ‘brands.csv’
   2. Transfer all ISO 8601 format time in ‘users.csv’ and ‘receipts.csv’ to Datetime format that is readable for MySQL (processing with text type is also adaptable)
4. Plot ERR diagram in MySQL: ‘EER Diagram.mwb’



1. Split the receipts spreadsheet into three subsheets to be corpotated in MySQL

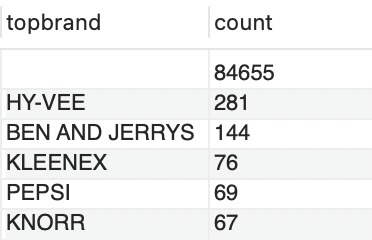
**Second Task - Data Query**

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



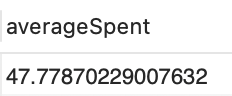
‘Q1.sql’ gives the result while large quantity of data points is missed.

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



‘Q2.sql’ gives the top 5 brands in last month while we cannot conclude the transition trend due to large amount of missing data in the latest month.

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

Rejected: ; Otherwise: 

‘Q3.1.sql’ gives the average spend from receipts with 'rewardsReceiptStatus’ of ‘Rejected’. While I don’t see there is a status called ‘Accepted’ so I consider all other status as ‘Accepted’ and ‘Q3.2.sql’ gives the average spend from receipts with all other status. So the conclusion is:

The receipts from Accepted Status have greater average spend.

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

Rejected: ; Otherwise: 

Similarly ‘Q4.1.sql’ gives the average item count from receipts with 'rewardsReceiptStatus’ of ‘Rejected’. While I don’t see there is a status called ‘Accepted’ so I consider all other status as ‘Accepted’ and ‘Q4.2.sql’ gives the average item count from receipts with all other status. So the conclusion is:

The receipts from Accepted Status have greater average item count.

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

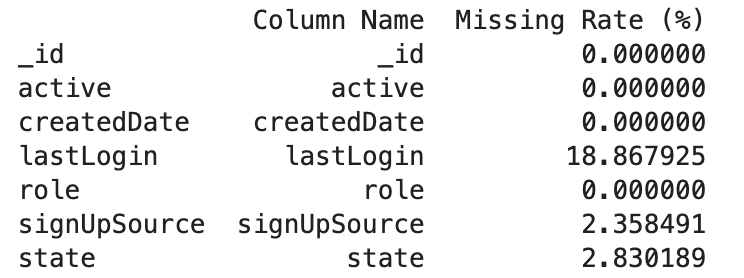
The spreadsheets is too large and it’s a little bit time-consuming to split more subsheets related to ‘rewardsReceiptItemList[0].priceAfterCoupon’ from receipts so I skip this query.

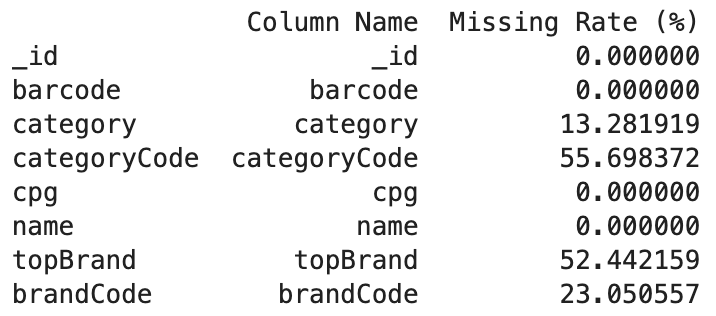
1. Which brand has the most transactions among users who were created within the past 6 months?

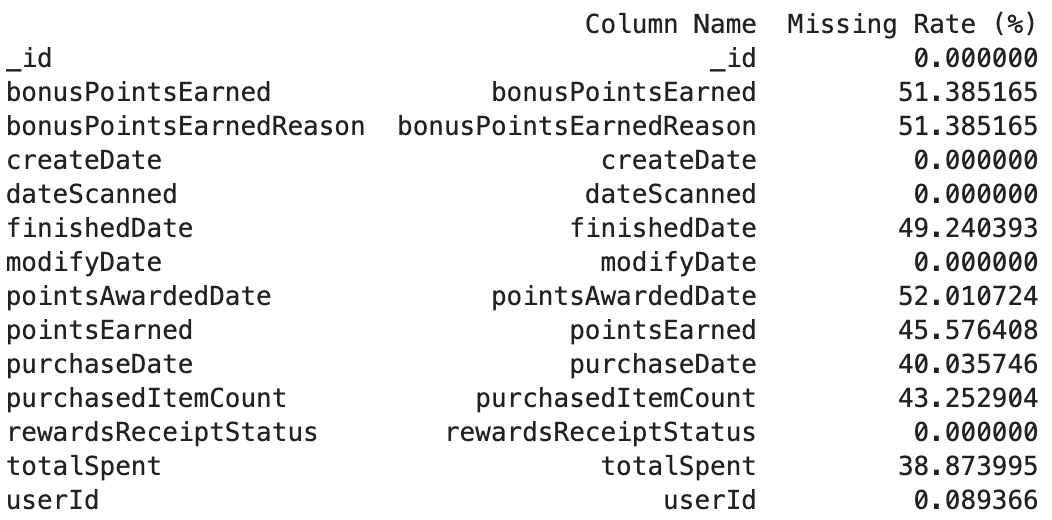
‘Q6.sql’ gives the coding part while I don’t see past 6 months’ data so I don’t run it.

**Third Task - Data Quality**

1. In ‘Data Quality.html’ and/or ‘Data Quality.ipynb’, I give the basic code to identify basic data quality defecient in the spreadsheets. For example, the missing rate of every field in ‘users.csv’ and ‘brands.csv’:

users: 

brands: 

receipts: 

Especially in ‘brands.csv’ and ‘receipts.csv’, many skus’ brandcode is missing, so it’s quite hard to identify their belonged brand during the analysis.

1. I also analyze the missing rate of the brands included in ‘receipts.csv’ while not appeared in the ‘brands.csv’ given. These missing datas confused me a lot during I dealing with the query tasks.



Thus, we cannot actually acquire the brands with the highest sales because many items’ brand informaton is lost.

**Fourth Task - Communicate with Stakeholders**

1. **What questions do you have about the data?**

I have noticed a significant amount of missing data and disorganized cells. For the first query, which requires the top 5 brands over the past month, I found that most cells are missing the brandcode, making it impossible to fulfill this requirement. In the third and fourth queries, I noticed the absence of the "Accepted" status and am unsure if this is expected. Therefore, I treated all non-"Rejected" statuses as "Accepted." Additionally, for the fifth query, due to column expansion, linking item-related data on each receipt is a massive and challenging task.

1. **How did you discover the data quality issues?**

While importing the data into MySQL, I first encountered difficulties querying the JSON format directly. As a result, I re-exported and imported the data using MongoDB into CSV format for analysis. I then discovered that some brands contain non-ASCII characters, which MySQL cannot recognize, requiring further processing. Moreover, the column expansion in the receipts table is a significant issue; I spent a lot of time splitting these tables and writing join queries based on primary keys.

1. **What do you need to know to resolve the data quality issues?**  
    I need to understand how this data was obtained. The prevalence of missing data and disorganized cells may indicate some discrepancies during the data capture process that need to be addressed.
2. **What other information would you need to help you optimize the data assets you're trying to create?**  
    I require all known brand information, including corresponding brandcodes and barcodes, as well as a complete receipts table, to achieve better analysis results.
3. **What performance and scaling concerns do you anticipate in production, and how do you plan to address them?**  
    I used MySQL to address these issues and found that the data layout in "receipts.csv" exhibits column expansion, which can lead to excessive space consumption. I had to create many tables linked by primary keys for join queries. I believe we can optimize this by reshaping the data, using Pivot Tables or other reshaping techniques to convert wide tables into long ones, thereby reducing the number of columns. Alternatively, we could consider nesting structures (e.g., JSON format) for columns with hierarchical data to reduce the column count. Additionally, querying with Python rather than relying solely on SQL could also be beneficial.