Fetch Rewards Data Quality Issues

The dataset given at the start of the assessment consisted of three tables:

1. Brands
2. Receipts
3. Users

The final dataset consisted of a total of 9 tables after performing all the transformations. The list of tables are as follows:

1. Brands
2. Categories
3. Items
4. Receipts
5. RewardItems
6. Rewards
7. TransactionItems
8. Transactions
9. Users

After finalizing the tables, there were many data quality issues which I figured out from the data at hand. The details of which are as follows:

1. Missing Data

The missing data was interpreted in the sense that there were a couple of rows in the brands dataset that were the unique identifying features, but there were missing values leading to the data quality issue of mapping the product to the brandCode, as seen in the figure below. I figured out that there was around 23%.

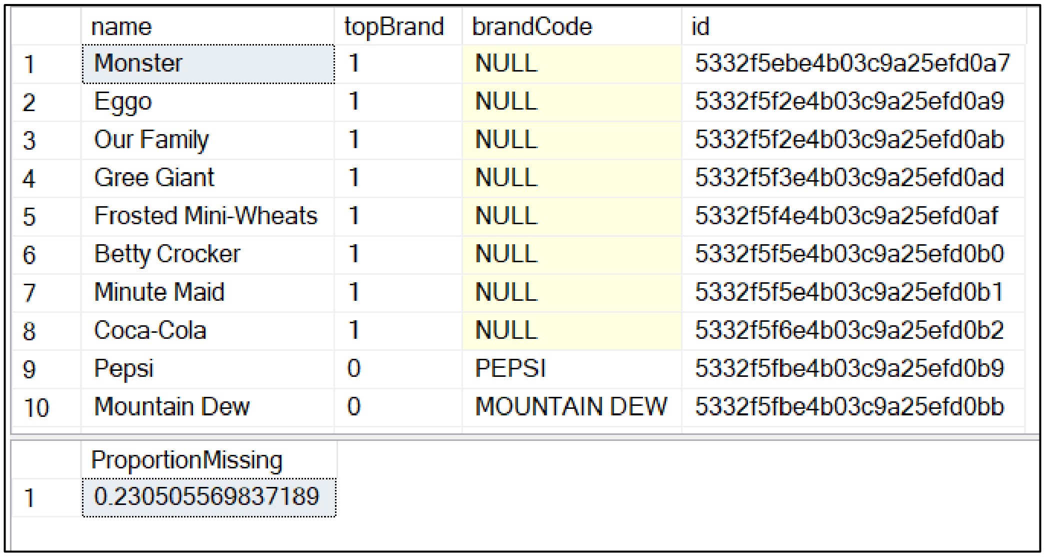
I used the following query to generate the table as shown: (4)

Query:

```
-- BRANDS TABLE
-- listing the entire brands table data
select *
from brands

-- calculating the percentage of missing data in brands table
SELECT CAST(SUM (CASE WHEN brandCode is NULL THEN 1 ELSE 0 END)as float) / COUNT(*) AS
ProportionMissing
FROM brands
```

Output:



	name	topBrand	brandCode	id
1	Monster	1	NULL	5332f5ebe4b03c9a25efd0a7
2	Eggo	1	NULL	5332f5f2e4b03c9a25efd0a9
3	Our Family	1	NULL	5332f5f2e4b03c9a25efd0ab
4	Gree Giant	1	NULL	5332f5f3e4b03c9a25efd0ad
5	Frosted Mini-Wheats	1	NULL	5332f5f4e4b03c9a25efd0af
6	Betty Crocker	1	NULL	5332f5f5e4b03c9a25efd0b0
7	Minute Maid	1	NULL	5332f5f5e4b03c9a25efd0b1
8	Coca-Cola	1	NULL	5332f5f6e4b03c9a25efd0b2
9	Pepsi	0	PEPSI	5332f5fbe4b03c9a25efd0b9
10	Mountain Dew	0	MOUNTAIN DEW	5332f5fbe4b03c9a25efd0bb

	ProportionMissing
1	0.230505569837189

There was a data quality issue of missing data in the categories table also. The categories table consists of only the categoryCode and the category, as we can see in the figure below the categoryCode consists of lots of missing values. Also, the "category id" for the categories table is missing. This is a major data quality issue in the given dataset. The proportion of missing data is around 56%.

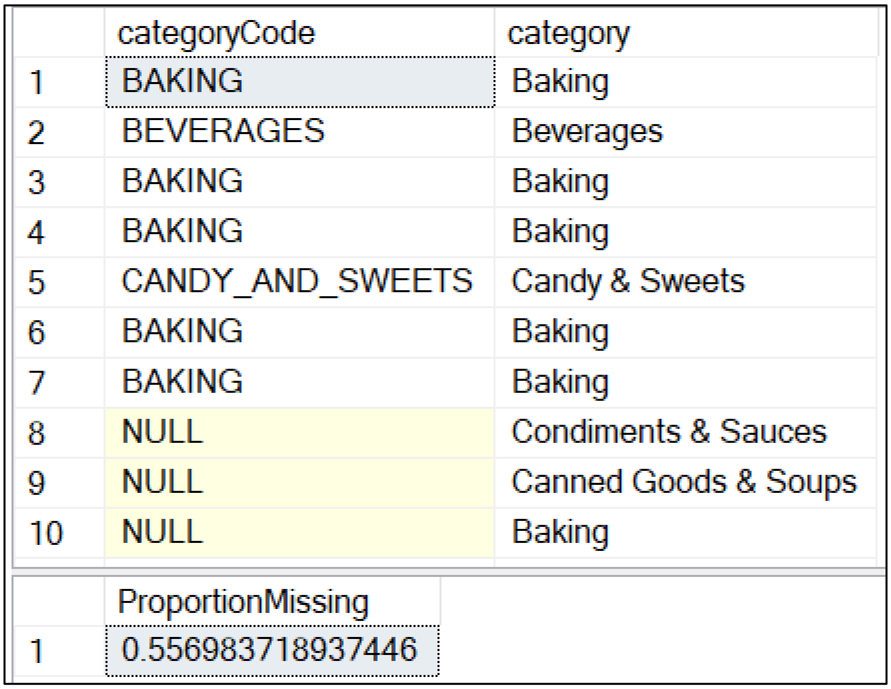
The query used is:

Query:

```
-- CATEGORIES TABLE
-- listing the categories table data
select *
from categories

-- calculating the percentage of missing data in categories table
SELECT CAST(SUM (CASE WHEN categoryCode is NULL THEN 1 ELSE 0 END)as float) / COUNT(*) AS
ProportionMissing
FROM categories
```

Output:



	categoryCode	category
1	BAKING	Baking
2	BEVERAGES	Beverages
3	BAKING	Baking
4	BAKING	Baking
5	CANDY_AND_SWEETS	Candy & Sweets
6	BAKING	Baking
7	BAKING	Baking
8	NULL	Condiments & Sauces
9	NULL	Canned Goods & Soups
10	NULL	Baking

	ProportionMissing
1	0.556983718937446

2. Duplicate Data

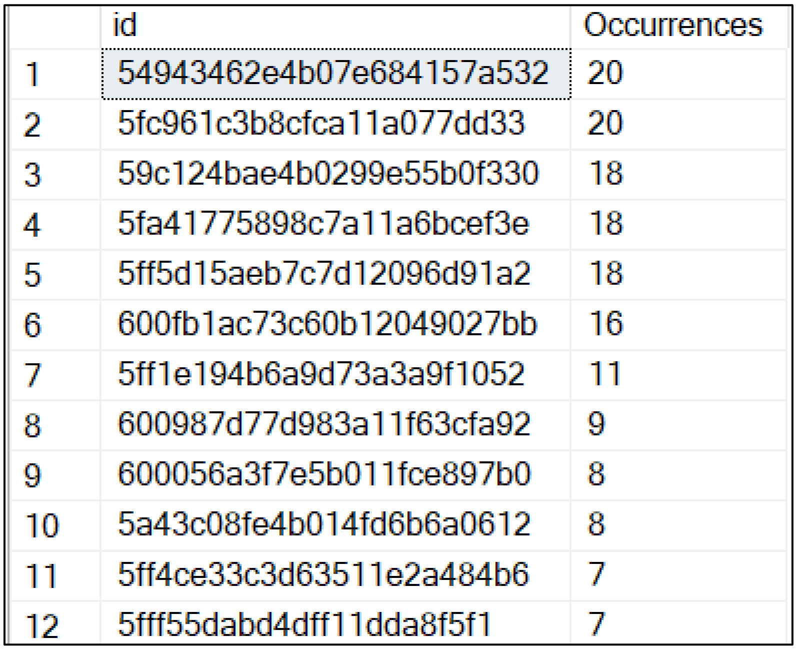
The second data quality issue which I identified is the duplicate data. In the Users table, I found out that many users have repeated Ids, that means the data is redundant for that user. I was able to identify that the user id “54943462e4b07e684157a532” has occurred 20 times, which is highly redundant. I have used the following query to get the desired result.

Query:

```sql
-- USERS TABLE
-- Checking for duplicates in users table
select id, count(*) as Occurrences
from users
-- grouping the data by id of the users
group by id
-- setting the conditions

having count(*) > 1
-- order the results using Occurrences
order by Occurrences desc
```

Output:



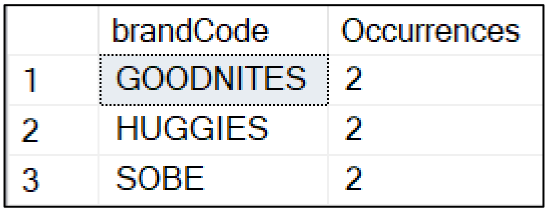
	id	Occurrences
1	54943462e4b07e684157a532	20
2	5fc961c3b8cfca11a077dd33	20
3	59c124bae4b0299e55b0f330	18
4	5fa41775898c7a11a6bcef3e	18
5	5ff5d15aeb7c7d12096d91a2	18
6	600fb1ac73c60b12049027bb	16
7	5ff1e194b6a9d73a3a9f1052	11
8	600987d77d983a11f63cfa92	9
9	600056a3f7e5b011fce897b0	8
10	5a43c08fe4b014fd6b6a0612	8
11	5ff4ce33c3d63511e2a484b6	7
12	5fff55dabd4dff11dda8f5f1	7

Another duplicate data quality issue was in the brands table. The following brandCodes are redundant in the dataset. As can be seen in the output, three brandcodes are repeated twice. The code used for simulating the output is:

Query:

```
-- BRANDS TABLE
-- Checking for duplicate data in brands table
select brandCode,count(id) as Occurrences
from brands
-- grouping the data by brandCode
group by brandCode
-- setting the conditions
having count(id) > 1 and brandCode is not null
-- order the results using Occurrences
order by Occurrences desc
```

Output:



	brandCode	Occurrences
1	GOODNITES	2
2	HUGGIES	2
3	SOBE	2

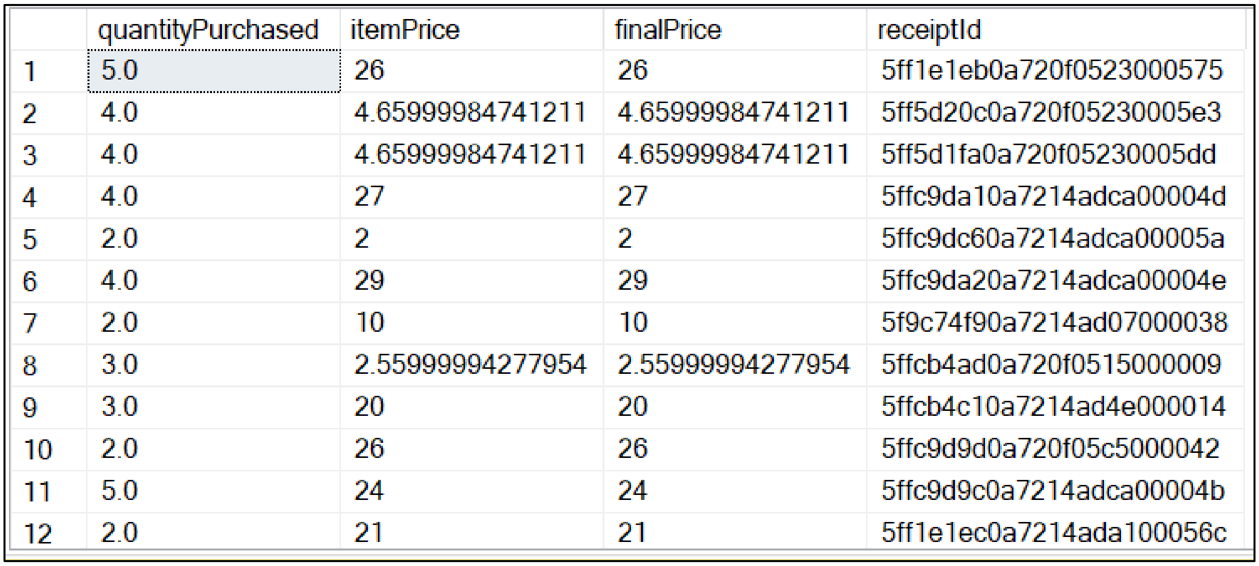
Inconsistent Data

The other major and the final data quality issue is inconsistent data. The issue I figured out was in the transaction and the transactionItems table. The itemPrice for a product purchased was mentioned and match to it the quantity purchased has been mentioned. When running the query, I figured out that even though the quantityPurchased is 5, the itemPrice is 26, ultimately the finalPrice will come out to be 130, but the finalPrice is still 26. This seems to be a very major issue of data inconsistency in our data set. The query used to simulate the data quality issue is mentioned below:

Query:

```
-- selecting the quantitypurchased, itemPrice, finalPrice and receiptID
-- join the transactionItems and transactions table on id
-- condition to test inconsistency is by checking the products with quantities greater
than 1 and where itemPrice and finalPrice are equal
select ti.quantityPurchased, ti.itemPrice, ti.finalPrice, t.receiptId
from transactionItems ti inner join transactions t on ti.transactionId=t.id
where ti.itemPrice=ti.finalPrice and cast(quantityPurchased as float) >1
```

Output:



	quantityPurchased	itemPrice	finalPrice	receiptId
1	5.0	26	26	5ff1e1eb0a720f0523000575
2	4.0	4.65999984741211	4.65999984741211	5ff5d20c0a720f05230005e3
3	4.0	4.65999984741211	4.65999984741211	5ff5d1fa0a720f05230005dd
4	4.0	27	27	5ffc9da10a7214adca00004d
5	2.0	2	2	5ffc9dc60a7214adca00005a
6	4.0	29	29	5ffc9da20a7214adca00004e
7	2.0	10	10	5f9c74f90a7214ad07000038
8	3.0	2.55999994277954	2.55999994277954	5ffcb4ad0a720f0515000009
9	3.0	20	20	5ffcb4c10a7214ad4e000014
10	2.0	26	26	5ffc9d9d0a720f05c5000042
11	5.0	24	24	5ffc9d9c0a7214adca00004b
12	2.0	21	21	5ff1e1ec0a7214ada100056c

As a conclusion, missing data, duplicate data and inconsistent data were the three major data quality issues which I found in the dataset.

All queries are also submitted in the SQL Queries folder.