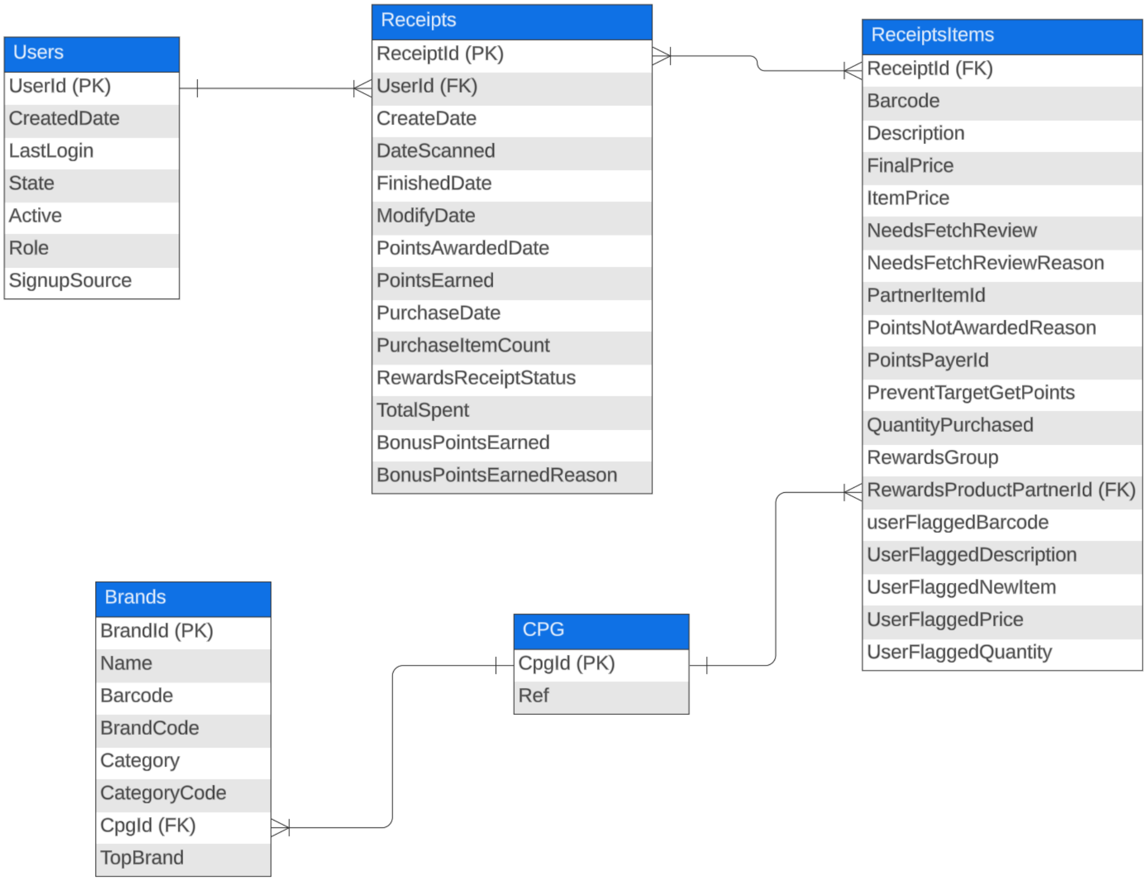
Fetch Rewards Coding Exercise - Analytics Engineer

Author: Prayansh Maheshwari

Part 1: Diagram of new Structured Relational Data Model



In the above diagram, I started with 3 tables, one for each data schema. Then in order to further normalize these 3 tables, I created 2 more tables as ReceiptItems and CPG. The ReceiptItems table consists of all the items present in a receipt and CPG table consists of all the cpg related information for all brands. The relationship between tables are as follows:

1. **Users to Receipts tables:** can be joined on UserId in both the tables
2. **Receipts and ReceiptsItems tables:** can be joined on ReceiptId in both the tables
3. **ReceiptItems and CPG tables:** can be joined on RewardsProductPartnerId and CpgID in respective tables
4. **Brands and CPG tables:** can be joined on CpgId in both the tables

Part 2: SQL Queries

Based on the status of current relational database model, we can answer 2 of the following business problems given by stakeholders directly:

Business Problem 3: When considering average spend from receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?

Assumption: There was no rewardsReceiptStatus as ‘Accepted’ in the table, so I’m considering Finished as “Accepted’

```
WITH receipt_spend AS (
   SELECT r.rewardsReceiptStatus, SUM(r.totalSpent) AS total_spent
   FROM Receipt r
   WHERE r.rewardsReceiptStatus IN (‘FINISHED’, ‘REJECTED’)
   GROUP BY r.rewardsReceiptStatus
)
SELECT AVG(total_spent) AS average_spend
FROM receipt_spend;
```

Business Problem 4: When considering total number of items purchased from receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?

Assumption: There was no rewardsReceiptStatus as ‘Accepted’ in the table, so I’m considering Finished as “Accepted’

```
WITH item_counts AS (
   SELECT r.rewardsReceiptStatus, SUM(ri.quantityPurchased) AS total_items
   FROM Receipt r
   JOIN ReceiptItem ri ON r.id = ri.receipt_id
   WHERE r.rewardsReceiptStatus IN (‘FINISHED’, ‘REJECTED’)
   GROUP BY r.rewardsReceiptStatus
)
SELECT SUM(total_items) AS total_number_of_items
FROM item_counts;
```

Note:

If we can incorporate BrandId or barcode column from Brand table into ReceiptItems table assuming that every entry in the brand tables has a unique BrandId or barcode, then we can solve all the other 4 business problems as well:

Business Problem 1: What are the top 5 brands by receipts scanned for most recent month?

```
SELECT b.name AS brand_name, COUNT(r.id) AS receipts_scanned
FROM Receipts r
JOIN ReceiptItem ri ON r.id = ri.receipt_id
JOIN Brand b ON ri.brandId = b. brandId
AND DATE_FORMAT(r.dateScanned, '%Y-%m') = DATE_FORMAT(CURRENT_DATE() -
INTERVAL 1 MONTH, '%Y-%m')
GROUP BY b.name
ORDER BY COUNT(r.id) DESC
LIMIT 5;
```

Business Problem 2: How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?

```
WITH recent_month AS (
        SELECT a.*,
        ROW_NUMBER() OVER(ORDER BY receipts_scanned DESC) as RN
        FROM(
                SELECT b.name AS brand_name, COUNT(r.id) AS receipts_scanned
                FROM Receipts r
                JOIN ReceiptItem ri ON r.id = ri.receipt_id
                JOIN Brand b ON ri.brandId = b. brandId
                AND DATE_FORMAT(r.dateScanned, '%Y-%m') =
                DATE_FORMAT(CURRENT_DATE() - INTERVAL 1 MONTH, '%Y-%m')
                GROUP BY b.name
                ORDER BY COUNT(r.id) DESC
                LIMIT 5
                )a
),
previous_month AS (
        SELECT b.*,
        ROW_NUMBER() OVER(ORDER BY receipts_scanned DESC) as RN
        FROM(
                SELECT b.name AS brand_name, COUNT(r.id) AS receipts_scanned
                FROM Receipts r
                JOIN ReceiptItem ri ON r.id = ri.receipt_id
                JOIN Brand b ON ri.brandId = b. brandId
```

```
                        AND DATE_FORMAT(r.dateScanned, '%Y-%m') =
                        DATE_FORMAT(CURRENT_DATE() - INTERVAL 2 MONTH, '%Y-%m')
                        GROUP BY b.name
                        ORDER BY COUNT(r.id) DESC
                        LIMIT 5
                        )b
)
SELECT
rm.brand_name AS recent_brand_name,
pm.brand_name AS previous_brand_name,
rm.receipts_scanned AS recent_receipts_scanned,
pm.receipts_scanned AS previous_receipts_scanned,
((rm.receipts_scanned/ pm.receipts_scanned) -1)*100 as percent_change
FROM recent_month rm
JOIN previous_month pm ON rm.rn = pm.rn
```

Business Problem 5: Which brand has the most spend among users who were created within the past 6 months?

```
WITH recent_users AS (
    SELECT DISTINCT Id
    FROM Users
    WHERE createdDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 6 MONTH)
), brand_spending AS (
    SELECT b.name AS brand_name, SUM(r.totalSpent) AS total_spent
    FROM Receipt r
    JOIN ReceiptItem ri ON r.id = ri.receipt_id
    JOIN Brand b ri.brandId = b. brandId
    JOIN recent_users ru ON r.userId = ru.Id
    GROUP BY b.name
)
SELECT brand_name, total_spent
FROM brand_spending
ORDER BY total_spent DESC
LIMIT 1;
```

Business problem 6: Which brand has the most transactions among users who were created within the past 6 months?

```
WITH recent_users AS (
    SELECT DISTINCT Id
    FROM Users
    WHERE createdDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 6 MONTH)
), brand_transactions AS (
    SELECT b.name AS brand_name, COUNT(distinct r.id) AS transaction_count
    FROM Receipt r
    JOIN ReceiptItem ri ON r.id = ri.receipt_id
    JOIN Brand b ri.brandId = b. brandId
    JOIN recent_users ru ON r.userId = ru.Id
    GROUP BY b.name
)
SELECT brand_name, transaction_count
FROM brand_transactions
ORDER BY transaction_count DESC
LIMIT 1;
```

Part 3: Evaluate Data Quality Issues in the Data Provided

Please refer ‘Fetch Assessment – Data Quality Issues’ python file attached along with this file on Github. Please find below the short summary of the data quality issues found in the above-mentioned file:

Null Values:

- There are many null value columns present in all three datasets.
- Some columns have over 50% null values, especially in the "brands" table, which could complicate preprocessing and analysis.
- **Recommendation**: Decide on strategies to handle null values, such as imputation, removal of columns, or other techniques based on the analysis goals.

Duplicates:

- The "users" table contains over 250 duplicate rows.
- **Recommendation**: Remove duplicate rows from the "users" table to ensure data integrity.

Receipt Table Issues:

- Upon manual inspection of the "receipt" table, additional data quality issues were identified.
    - A major issue is the absence of a column related to brands, which complicates joining with the "brands" table.
    - Another issue found is the presence of duplicate receipt items within a single receipt.

- **Recommendation**: If possible, add a column related to brands in the "receipt" table for easier integration with the "brands" table and address the issue of duplicate receipt items, possibly by investigating their source and deciding on appropriate actions.

Further Analysis:

Consider further exploratory data analysis to gain insights into relationships between tables and to understand the impact of data quality issues on analysis results.

Part 4: Communicate with Stakeholders

Subject: Initial Data Analysis Findings and Recommendations

Hi (Product/Business manager's Name),

I wanted to share some initial insights from my current analysis of the three datasets I'm working with. This information is important in terms of ensuring the quality and effectiveness of our data and data-driven decision making moving forward.

Data Quality Findings:

- **Null Values:**
  - There are many null value columns present in all three datasets.
  - Some columns have over 50% null values, especially in the "brands" table, which could complicate preprocessing and analysis.
- **Duplicate Rows:**
  - The "users" table contains over 250 duplicate rows, which could skew our analysis results.
- **Receipt Table Issues:**
  - Along with the nulls found in bonus related columns, upon manual inspection of the "receipt" table, I discovered a couple of critical issue.
  - A major issue is the absence of a column related to brands, which complicates joining with the "brands" table and brand level analysis.
  - Additionally, I found duplicate items present within a single receipt.

Questions and Next Steps:

1. **Questions:**
   - I would like to know the business context of the absent data in the "brands" table. Understanding why this data is missing will help us make informed decisions on how to handle it.
   - What is the best approach to link the brands and receipts tables effectively? Understanding this will help in conducting comprehensive analyses at both the brand and receipt levels.
   - Having more context around the description of rewardsReceiptItemList would improve our understanding of the columns nested under it

2. **Discovery Process:**
   - I identified these issues through a systematic review of the datasets, focusing on null values, duplicates, and manual checks of the of all the table.

3. **Resolution Needs:**
   - To resolve the data quality issues, we would need to decide on strategies such as imputation for null values, removal of duplicate rows in the "users" table, and possibly creating a brand-related column in the "receipt" table.

4. **Optimization Information:**
   - It would be helpful to understand the primary business objectives we aim to achieve with this analysis. This will guide us in optimizing the data assets accordingly.

5. **Performance Concerns:**
   - In production, we anticipate potential issues with dataset size, particularly given the large number of null values in the tables.
   - Our strategy for addressing these concerns includes implementing efficient data cleaning processes, optimizing queries for performance, and investigating ways to simplify data integration.

I believe that fixing these data quality issues will considerably improve the accuracy and reliability of our analysis, resulting in better-informed decision-making. Your insights and guidance on the business context will be invaluable as we continue to optimize these data assets.

Please let me know if you have any questions or if there's any additional information you'd like to discuss. Looking forward to your feedback.

Best Regards,
Prayansh Maheshwari