**First: Review Existing Unstructured Data and Diagram a New Structured Relational Data Model**

Created receipts, brands , users and rewards schema.

A screenshot of a computer screen

Description automatically generated

**Second: Write queries that directly answer predetermined questions from a business stakeholder**

**Used MySQL**

|  |
| --- |
| * **What are the top 5 brands by receipts scanned for most recent month?** |
| with receiptsrewards as (  SELECT id  FROM receipts  WHERE STRFTIME('%m-%Y', date\_Scanned) = (  SELECT MAX(STRFTIME('%m-%Y', date\_Scanned))  FROM receipts  )    )  SELECT b.name AS BrandName, COUNT(r.id) AS ReceiptCount  FROM receiptsrewards r  JOIN rewards ri ON r.id = ri.id  JOIN brands b ON ri.barcode = b.barcode  GROUP BY b.name  ORDER BY ReceiptCount DESC  LIMIT 5 |
| * **How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?** |
| Similar to above code we will create 2 CTEs, one for recent month and another for previous month. And 2 more ctes for the ranks of the top 5 brands for the recent and previous month. Then the final query to compare the rank from these for top 5 brands |
| * **When considering average spend from receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?** |
| SELECT  rewardsReceiptStatus,  COALESCE(AVG(totalSpent),0) AS average\_spend  FROM  receipts  where  rewardsReceiptStatus IN ('FINISHED', 'REJECTED')  GROUP BY  rewardsReceiptStatus  Note : Used ‘FINISHED’ as there was no ‘ACCEPTED’ in the data  Output : The average spend is greater for rewardstatus ‘Accepted’  A black background with white text  Description automatically generated |
| * **When considering total number of items purchased from receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?** |
| SELECT  rewardsReceiptStatus,  COALESCE(SUM(purchasedItemCount),0) AS total\_items\_purchased  FROM  receipts  WHERE  rewardsReceiptStatus IN ('FINISHED', 'REJECTED')  GROUP BY  rewardsReceiptStatus  Note : Used ‘FINISHED’ as there was no ‘ACCEPTED’ in the data  Output : The total number of items purchased is greater for rewardstatus ‘Accepted’  A black screen with white text  Description automatically generated |
| * **Which brand has the most spend among users who were created within the past 6 months?** |
| SELECT  b.name AS brand\_name,  SUM(r.totalSpent) AS total\_spent  FROM  users u  JOIN  receipts r ON u.id = r.userId  JOIN  rewards ri ON r.id = ri.id  JOIN  brands b ON ri.barcode = b.barcode  WHERE  u.created\_Date >= DATEADD(month, -6, GETDATE())  GROUP BY  b.name  ORDER BY  total\_spent DESC  LIMIT 1; |
| * **Which brand has the most transactions among users who were created within the past 6 months?** |
| SELECT  b.name AS brand\_name,  COUNT(ri.id) AS transaction\_count  FROM  users u  JOIN  receipts r ON u.id = r.userId  JOIN  rewards ri i ON r.id = ri.id  JOIN  brands b ON i.barcode = b.barcode  WHERE  u.createdDate >= DATEADD(month, -6, GETDATE())  GROUP BY  b.name  ORDER BY  transaction\_count DESC  LIMIT 1;  Note: The max createdDate of users is 2021-02, so I modified the code (in the notebook) to consider users created 6 months from the max date and found the brand with most transactions  A black and white photo of a name  Description automatically generated |

**Third: Evaluate Data Quality Issues in the Data Provided**

* The data is in json format contains complex structures with nested objects or arrays, which can be difficult to parse and flatten for analysis. Firstly, not all key-value pairs are present for each id in the receipts table for ids which have rewards.
* JSON allows for flexibility, meaning certain objects may have fields missing entirely, especially if the data structure is sparse or generated from different sources. And this issue seems to persist in these data. Converting to relational schema gives sparse data
* Considering data level issues:
  + Missing data:

All the data have missing data in some columns (shown in the notebook)

Due to json format and 1:n cardinality of receipts with rewards, rewards data has many missing values.

For some rewards, even the barcode considered to be primary key is missing

Not all the barcode in the rewards is present in the brands data, which is data flaw.

* Duplicate values:

Users data has 283 duplicate values which is more than half of the total data size(495)

Rewards also has 583 duplicate values.

* Inconsistent data in categorical columns:

The column cpg in brands column has 2 values : (Cogs, Cpgs) could be a typo

* Incorrect data type

Price columns to are object datatype.

The barcode had different datatypes in different data

**Fourth: Communicate with Stakeholders**

***Subject: Grocery Reward App Data: Potential and Challenges***

I'm reaching out today with some insights from digging into the data for our automatic grocery reward fetching app. There's some great potential here, but I've also identified some data quality issues that could impact user experience and the overall effectiveness of the program.

1. Data Quality Issues: Our data is a bit like a mixed bag of groceries from different stores—some fresh, some not so much. Here are the key issues we’ve identified:

Inconsistent Data: Each grocery shop provides data in its own format. For example, some stores use product IDs, while others use product names. This inconsistency makes it challenging to create a unified view.

Errors and Missing Data: We’ve noticed discrepancies, such as incorrect purchase amounts or missing customer information. These errors affect our ability to accurately track rewards and provide a seamless experience.

2. Questions and Considerations: To address these issues, we need to ask a few questions:

Data Mapping: What is the source of data, is it external party data?How can we map different data formats to a common structure?For instance, can we create a standardized product catalog?

Data Validation: How do we validate the accuracy of incoming data? Are there automated checks we can put in place?

Data Enrichment: Can we enhance our data with additional information, like customer demographics or purchase history?

3. Optimization Opportunities: Here’s where we can optimize our data assets:

Consolidation: Let’s create a central repository where all data is transformed into a consistent format. This will improve reporting and analytics.

Cleaning: We’ll need to clean up erroneous entries and fill in missing data. Regular audits can help maintain data quality.

Performance and Scaling:

Performance: Using SQL optimization techniques should be used like – optimizing joins, using temp tables and avoiding subqueries, using the correct datatypes , etc

Scaling Concerns: As our user base grows, we must ensure our system can handle the load. We’ll need to monitor performance metrics like response time and resource utilization.

Action Plan: Implement caching, optimize database queries, and consider cloud-based and platform agnostic solutions for scalability.

4. Pros and Cons:

Pros: Once we clean up the data, we’ll have a more accurate picture of customer behavior. This will enable targeted marketing, personalized offers, and better reward management.

Cons: The initial cleanup process may be time-consuming, and we’ll need to allocate resources for ongoing maintenance.

5. Action Steps:

Data Audit: Conduct a thorough audit to identify specific data issues.

Standardization: Work with the IT team to standardize data formats.

Validation Rules: Define validation rules to catch errors early.

Data Governance: Establish guidelines for data entry and maintenance.

Scalability Plan: Collaborate with our tech team to prepare for growth.

I’m confident that by addressing these data quality challenges, we’ll enhance our app’s performance and provide a delightful experience for our users.