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

Created receipts, brands , users and items schema.

A diagram of a user flow

Description automatically generated with medium confidence

**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 recieptitems 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?** |
| WITH CurrentMonth AS (  SELECT b.name AS brand\_name, COUNT(ri.receipt\_id) AS receipt\_count  FROM Receipts r  JOIN ReceiptItems ri ON r.receipt\_id = ri.receipt\_id  JOIN Brands b ON ri.barcode = b.barcode  WHERE r.date\_scanned >= DATE\_TRUNC('month', CURRENT\_DATE - INTERVAL '1 month')  GROUP BY brand\_name  ),  PreviousMonth AS (  SELECT b.name AS brand\_name, COUNT(ri.receipt\_id) AS receipt\_count  FROM Receipts r  JOIN ReceiptItems ri ON r.receipt\_id = ri.receipt\_id  JOIN Brands b ON ri.barcode = b.barcode  WHERE r.date\_scanned >= DATE\_TRUNC('month', CURRENT\_DATE - INTERVAL '2 month')  AND r.date\_scanned < DATE\_TRUNC('month', CURRENT\_DATE - INTERVAL '1 month')  GROUP BY brand\_name  )  SELECT  cm.brand\_name,  cm.receipt\_count AS current\_month\_count,  pm.receipt\_count AS previous\_month\_count,  RANK() OVER (ORDER BY cm.receipt\_count DESC) AS current\_month\_rank,  RANK() OVER (ORDER BY pm.receipt\_count DESC) AS previous\_month\_rank  FROM CurrentMonth cm  LEFT JOIN PreviousMonth pm ON cm.brand\_name = pm.brand\_name  ORDER BY current\_month\_rank  LIMIT 5; |
| * **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  receiptitems 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  receiptitems 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 items.
* 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
* Summary of Issues Found
  + Missing Data:
    - Null values in key fields like purchaseDate, userId, brandId,brandcode.
  + Inconsistent References:
    - userId in receipts not found in users ***(2604)***
    - Items in rewardsReceiptItemList missing associated brandId ***(6859)***
    - To join to Brands collection, this receipts collection does not seem to have an obvious mapping key to the Brands collection. The receipts barcode key does not match with the brands barcode key.
  + Invalid Values:
    - Zero totalSpent and purchasedItemCount
    - Invalid or missing dates
    - Considering pointsEarned, there is a sparse distribution beyond 1000 points and then a few extreme values at 8000 to 10000 points. The discontinuity suggests that these are not trivial scenarios and may inform the company to set thresholds beyond certain limits or revise any bulk point promotions. More analysis can be done on correlation between points and bonus points to ensure data validity.
    - Finally, considering bonusPointsEarned, there are distinct peaks at different points such as 50, 200, 300, 500, and 700+ indicating a potential tiered reward system. Alternatively, this could also indicate errorneous duplicates or artificial clustering in the data. The high spike at 700 is unusual indicating a special promotion or an incorrect data capture. It may be worth to revisit the data source and understand whether the values are computed, manuall entered, or pulled from an external system.
  + Duplication:
    - Duplicate rows in receipts and users
      * After deduplication, there are only 212 unique users
* Recommendations for Resolution
* Handle Missing Data:
  + Impute or drop rows with null critical fields based on business rules
* Fix Referential Integrity:
  + Investigate mismatched userId and brandId and align datasets
* Validate and Clean Data:
  + Correct invalid purchaseDate and totalSpent values
* Remove Duplicates:
  + Drop duplicate rows where applicable

**Fourth: Communicate with Stakeholders**

***Subject: Data Quality and Optimization for Rewards Program***

I'm reaching out to discuss some data quality issues and optimization opportunities I've encountered while working on the Rewards fetching project. Our goal is to build a robust data warehouse that provides valuable insights into customer behavior and purchasing patterns. To achieve this, we need to ensure our data is accurate, reliable, and scalable.

**Questions About the Data:**

1. **What constitutes a valid receipt?**  
   Some receipts have incomplete fields (e.g., purchaseDate or rewardsReceiptItemList). Do we consider these valid, or should they be excluded?
2. **How do we handle edge cases?** 
   * Users without a state value in the Users table.
   * Missing or inconsistent brand\_id values in the ReceiptItems table.
3. **What is the threshold for active/inactive users?**  
   Understanding this definition is critical for accurate segmentation.

**How I Discovered Data Quality Issues:**

1. **Null Values:** Found during preliminary aggregation and validation (e.g., calculating spend per brand).
2. **Inconsistent IDs:** Mismatched brand\_id across ReceiptItems and Brands tables caused aggregation mismatches.
3. **Duplicate Entries:** Identified duplicate rows in the Receipts table when counting unique transactions.

**What I Need to Resolve Data Quality Issues:**

1. Confirmation on business logic for missing or null fields.
2. Guidelines for deduplication—should we keep the most recent row, or merge duplicates?
3. Insights into how data is ingested—are there upstream processes or known issues contributing to these inconsistencies?

**Other Information Needed to Optimize Data Assets:**

1. **Data Latency:** How frequently are updates made to these tables?
2. **Retention Policies:** How long should we store user and receipt data?
3. **Integration Goals:** Should we align with other teams' datasets or tools for cross-functional reporting?

**Performance and Scaling Concerns:**

1. **Query Performance:** Some tables (e.g., Receipts) have high volumes of records, which can slow down queries.  
   *Plan:* Introduce indexing on frequently queried fields like purchaseDate and user\_id.
2. **Data Growth:** With an increasing number of receipts and users, storage and processing needs may become a bottleneck.  
   *Plan:* Explore partitioning strategies (e.g., by month or user region) and monitor cloud storage limits.
3. **Real-Time Needs:** If there’s a need for real-time analytics, we may need to rethink our architecture.

I'm confident that by addressing these data quality issues and implementing optimization strategies, we can build a valuable data asset that drives informed decision-making. I'd appreciate it if you could provide the requested information and collaborate with me to ensure the success of this project.

Best regards,

Karthika