Subject: Data Quality Insights and Next Steps

Hi [Leader's Name],

I hope this message finds you well! I wanted to share some insights regarding the data quality in our current datasets and outline the next steps for addressing these issues.

Key Findings:

During my recent analysis, I discovered several data quality issues that could impact our decision-making processes and the overall effectiveness of our product:

1. Missing Fields: Some receipts are missing critical information, such as purchaseDate. This could affect our ability to analyze user behavior and spending patterns accurately.
2. Orphaned Receipts: There are receipts linked to non-existent users, which raises concerns about data integrity/fraud.
3. Fetch Staff: Several of the users were fetch staff. I believe that they should be excluded in all computations/queries.
4. Category Codes: More than 50% of brand IDs did not have an associated category code. If the future study involves a deep-dive into brand-related data, we should seek to improve the dataset.

Discovery Process:

These issues were identified through SQL queries that analyze the datasets for missing or inconsistent values. These can also be programmatically run as new data is observed.

Next Steps and Information Needed:

- Clarify Data Sources: Confirm the data sources for user and receipt information to ensure we understand where and how the data is generated.
  For example, is this a persisting users.json data structure that gets updated daily? Weekly? Monthly? Or is it a specific subsection of our users?

- Understand Data Entry Protocols: Review how users and receipts are entered into the system, which will help identify potential points of failure or inconsistency.
  Ilooks like receipts.json consists of several fields for each receipt item, further clarification on the meaning of those fields would help us understand the purpose

of those. Additionally, are we supposed to be excluding/including certain records? Example: FINISHED only, REJECTED only, needsFetchReview = false, etc.

Performance and Scaling Concerns:

1. SQL Query Optimization:
    - In data_quality_queries.py, several queries, such as get_receipts_with_missing_fields_query, need optimization to handle increasing data volume efficiently:
        - Join Operations: Queries like get_inactive_users_with_recent_receipts_query involve joins. Analyzing execution plans will help identify any potential slowdowns as the number of records increases.
        - Aggregate Functions: The use of aggregate functions (e.g., COUNT, SUM) in get_receipts_with_unusual_total_amounts_query can become resource-intensive. We'll need to monitor and potentially refactor these queries to use indexed columns or pre-aggregated tables.
2. Automate Data Quality Checks:
    - Enhance the data_quality_check.py script to automate monitoring for anomalies identified in our SQL queries. For instance:
        - Implement alerts when the count of missing_fields_count in get_receipts_with_missing_fields_query exceeds a threshold, allowing us to proactively address issues.
3. Database Scalability:
    - As outlined in the queries, we need to ensure that key columns (like userId, purchaseDate, and totalSpent) are indexed to facilitate faster lookups and aggregations. This is crucial as our user base and receipt volume continue to grow (likely).
    - We might consider implementing data partitioning strategies for the receipts table to improve performance for frequently accessed queries.
4. Supporting Dashboarding:
    - As we aim to feed data into Tableau/Grafana for visualization, ensuring that our SQL queries return results in a timely manner is critical. This may involve refining our queries to reduce the amount of data processed or returned, focusing on relevant metrics that directly support business decisions.