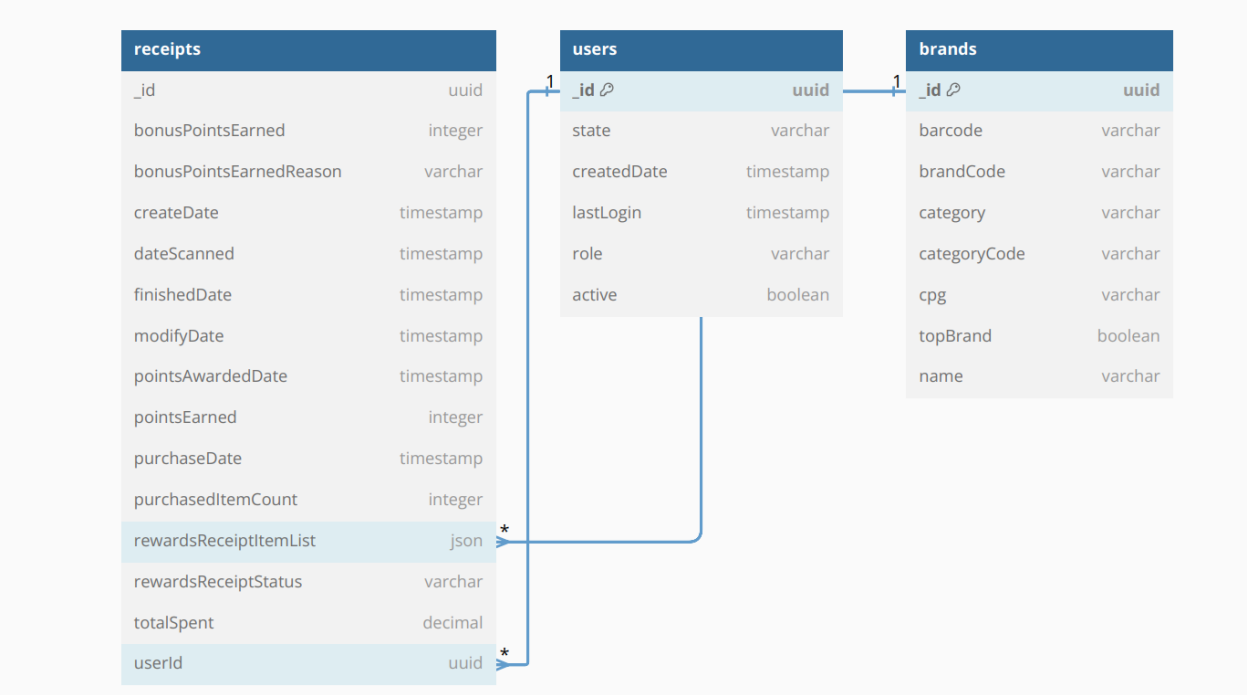
First: Review Existing Unstructured Data and Diagram a New Structured Relational Data Model



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

1. **What are the top 5 brands by receipts scanned for the most recent month?**

```
WITH RecentMonth AS (
   SELECT DATE_TRUNC('month', MAX(date_scanned)) AS last_month
   FROM Receipts
)
SELECT b.name, COUNT(r.receipt_id) AS receipts_scanned
FROM Receipts r
JOIN ReceiptItems ri ON r.receipt_id = ri.receipt_id
JOIN Brands b ON ri.brand_id = b.brand_id
WHERE DATE_TRUNC('month', r.date_scanned) = (SELECT last_month FROM RecentMonth)
GROUP BY b.name
ORDER BY receipts_scanned DESC
LIMIT 5;
```

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 RecentMonths AS (
   SELECT
```

```
    DATE_TRUNC('month', MAX(date_scanned)) AS last_month,
    DATE_TRUNC('month', MAX(date_scanned) - INTERVAL '1 month') AS previous_month
  FROM Receipts
),
BrandRankings AS (
  SELECT
    b.name,
    DATE_TRUNC('month', r.date_scanned) AS month,
    COUNT(r.receipt_id) AS receipts_scanned,
    RANK() OVER (PARTITION BY DATE_TRUNC('month', r.date_scanned) ORDER BY
COUNT(r.receipt_id) DESC) AS rank
  FROM Receipts r
  JOIN ReceiptItems ri ON r.receipt_id = ri.receipt_id
  JOIN Brands b ON ri.brand_id = b.brand_id
  WHERE DATE_TRUNC('month', r.date_scanned) IN (SELECT last_month FROM
RecentMonths)
    OR DATE_TRUNC('month', r.date_scanned) IN (SELECT previous_month FROM
RecentMonths)
  GROUP BY b.name, DATE_TRUNC('month', r.date_scanned)
)
SELECT
  br1.name,
  br1.month AS last_month,
  br1.rank AS last_month_rank,
  br2.month AS previous_month,
  br2.rank AS previous_month_rank
FROM BrandRankings br1
LEFT JOIN BrandRankings br2 ON br1.name = br2.name AND br1.month = br2.month +
INTERVAL '1 month'
WHERE br1.month = (SELECT last_month FROM RecentMonths)
ORDER BY br1.rank
LIMIT 5;
```

3. **When considering average spend from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?**

```
SELECT
  rewards_receipt_status,
  AVG(total_spent) AS average_spend
FROM Receipts
WHERE rewards_receipt_status IN ('Accepted', 'Rejected')
GROUP BY rewards_receipt_status
ORDER BY average_spend DESC;
```

4. **When considering total number of items purchased from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?**

```
SELECT
```

```
    rewards_receipt_status,
    SUM(purchased_item_count) AS total_items_purchased
FROM Receipts
WHERE rewards_receipt_status IN ('Accepted', 'Rejected')
GROUP BY rewards_receipt_status
ORDER BY total_items_purchased DESC;
```

5. **Which brand has the most spend among users who were created within the past 6 months?**

```
WITH RecentUsers AS (
    SELECT user_id
    FROM Users
    WHERE created_date >= NOW() - INTERVAL '6 months'
)
SELECT b.name, SUM(r.total_spent) AS total_spend
FROM Receipts r
JOIN ReceiptItems ri ON r.receipt_id = ri.receipt_id
JOIN Brands b ON ri.brand_id = b.brand_id
WHERE r.user_id IN (SELECT user_id FROM RecentUsers)
GROUP BY b.name
ORDER BY total_spend DESC
LIMIT 1;
```

6. **Which brand has the most transactions among users who were created within the past 6 months?**

```
WITH RecentUsers AS (
    SELECT user_id
    FROM Users
    WHERE created_date >= NOW() - INTERVAL '6 months'
)
SELECT b.name, COUNT(r.receipt_id) AS transaction_count
FROM Receipts r
JOIN ReceiptItems ri ON r.receipt_id = ri.receipt_id
JOIN Brands b ON ri.brand_id = b.brand_id
WHERE r.user_id IN (SELECT user_id FROM RecentUsers)
GROUP BY b.name
ORDER BY transaction_count DESC
LIMIT 1;
```

Third : Evaluate Data Quality Issues Data Quality Issues Identified.

1. **Missing Data**:
   - Some receipts may have missing rewardsReceiptStatus.
   - Some users may not have a state or lastLogin date.

2. **Inconsistent Data**:
   - rewardsReceiptStatus may have inconsistent values (e.g., typos like 'Accept' instead of 'Accepted').
   - brandCode in Brands may not be unique or may be missing.

3. **Data Type Issues**:
   - totalSpent in Receipts should be a numeric type, but if it's stored as a string, it could cause issues in calculations.
4. **Duplicate Data**:
   - There may be duplicate entries in the Brands table with different brand_id but the same brandCode.
5. **Referential Integrity**:
   - userId in Receipts should always reference a valid user_id in Users. If not, it could lead to orphaned records.

SQL Queries to Identify Data Quality Issues

1. **Check for Missing rewardsReceiptStatus**:
   ```
   SELECT COUNT(*)
   FROM Receipts
   WHERE rewards_receipt_status IS NULL;
   ```

2. **Check for Inconsistent rewardsReceiptStatus**:
   ```
   SELECT DISTINCT rewards_receipt_status
   FROM Receipts;
   ```

3. **Check for Missing state in Users**:
   ```
   SELECT COUNT(*)
   FROM Users
   WHERE state IS NULL;
   ```

4. **Check for Duplicate brandCode in Brands**:
   ```
   SELECT brand_code, COUNT(*)
   FROM Brands
   GROUP BY brand_code
   HAVING COUNT(*) > 1;
   ```

5. **Check for Orphaned userId in Receipts**:
   ```
   SELECT COUNT(*)
   FROM Receipts r
   LEFT JOIN Users u ON r.user_id = u.user_id
   WHERE u.user_id IS NULL
   ```

Fourth: Communicate with Stakeholders

Hi Team,
I'm writing to inform you about some critical data quality issues we've discovered that impact on the reliability of our reporting and analysis. These issues could potentially lead to inaccurate insights and flawed decision-making.

First, I would like to discuss some of the questions regarding the data.

- Why is data missing or inconsistent? Are there issues with data entry, system integrations, or other processes?
- Where does the receipt data originate from? Are there any known issues with the data collection process?
- Who is responsible for maintaining the brand and user data, and can we collaborate with them to improve data quality?
- Are there existing data standards documentation that we can use?
- What is the expected volume of new data ingested daily, and are there peak usage times that could impact data processing?

Next, I want to discuss the data quality issues I found in the data.

- **Missing Data:**
  - We're seeing gaps in data related to receipt statuses and user information (like state and login dates). This makes it hard to get a complete picture of user behavior and receipt processing.
- **Inconsistent Data:**
  - We've found variations and errors in how receipt statuses and brand codes are recorded. For example, slight spelling mistakes create separate categories. This makes it difficult to aggregate and analyze the data accurately.
- **Data Type Problems:**
  - Some numerical data, like total spending on receipts, might be stored as text, which will cause errors in calculations.
- **Duplicate Data:**
  - We have found duplicate brand codes. This will cause inaccurate reporting about the number of brands we work with.
- **Broken Links:**
  - Some receipt records are linked to users that don't exist, which creates "orphaned" data.

Now, here are the steps that I use to identify these data quality issues.

- Ran SQL queries to check for missing values, inconsistencies, duplicates, and referential integrity issues.
- Compared rewardsReceiptStatus values against expected categories and found inconsistencies (e.g., "Accept" vs. "Accepted").
- Detected totalSpent stored as a string instead of a numeric type, which would cause calculation errors.

- Identified duplicate brandCode values in the Brands table, indicating possible data redundancy or entry errors.
- Found orphaned userId values in Receipts that do not match any valid user_id in Users, suggesting referential integrity issues.

Here are some of the important points that we, as a team, need to know about the data.

- What are the business rules for handling missing or incorrect rewardsReceiptStatus values? Should they be inferred, corrected, or removed?
- Should brandCode be enforced as a unique constraint to prevent duplicates, or are there valid cases where duplication exists?
- What is the acceptable threshold for missing data in fields like state and lastLogin?
- Should orphaned userId values in Receipts be deleted, or is there a reconciliation process to match them with users?
- Are there logging mechanisms in place to track data quality issues over time?

Below are some of the important points that we need to know that will help us to optimize the data assets we're trying to create.

- Understanding the business impact of these data quality issues—are they affecting revenue, reporting accuracy, or customer insights?
- Documentation on data lineage to understand how this data is ingested, transformed, and consumed by other teams or applications.
- Performance benchmarks for existing ETL pipelines to assess whether optimizations are needed.
- Expected retention policy for historical data to determine how much data should be cleaned and archived.

These are the performance and scaling concerns that I anticipate in production, and a plan to address them.

- **Increasing Data Volume:** As more receipts and users enter the system, the queries must be optimized using indexing, partitioning, and caching mechanisms.
- **ETL Bottlenecks:** Ensuring that ETL jobs run efficiently by parallelizing queries and batch processing instead of row-by-row operations.
- **Referential Integrity Enforcement:** Using foreign key constraints or validation scripts to prevent orphaned records.
- **Automated Data Quality Checks:** Implementing real-time alerts or scheduled audits to proactively identify and address data inconsistencies.
- **Scalability:** Migrating to a distributed processing architecture (e.g., Snowflake, BigQuery, or Redshift) to handle larger datasets and more complex queries efficiently.

I'd like to schedule a brief meeting to discuss these issues in more detail and plan our next steps.

Thanks,
Vamshi