1. Structured Relational Data Model

2. Queries for predetermined business requirements.

a. What are the top 5 brands by receipts scanned for most recent month?  
Solution:  
  
SELECT

b.name AS brand\_name,

COUNT(ri.itemId) AS total\_receipts

FROM (

SELECT

r.\_id AS receipt\_id

FROM Receipts r

INNER JOIN ReceiptStatus rs ON r.\_id = rs.receiptId

WHERE DATE\_FORMAT(rs.dateScanned, '%Y-%m') = DATE\_FORMAT(CURRENT\_DATE, '%Y-%m')

) lm

INNER JOIN ReceiptItems ri ON lm.receipt\_id = ri.receiptId

INNER JOIN Brands b ON ri.brandId = b.\_id

GROUP BY b.name

ORDER BY total\_receipts DESC

LIMIT 5;

b. When considering average spend from receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?  
  
SELECT rs.rewardsReceiptStatus, AVG(r.totalSpent) AS avg\_spend FROM Receipts r INNER JOIN ReceiptStatus rs ON r.\_id = rs.receiptId WHERE rs.rewardsReceiptStatus IN ('Accepted', 'Rejected') GROUP BY rs.rewardsReceiptStatus;

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

SELECT rs.rewardsReceiptStatus, SUM(r.purchasedItemCount) AS total\_items FROM Receipts r INNER JOIN ReceiptStatus rs ON r.\_id = rs.receiptId WHERE rs.rewardsReceiptStatus IN ('Accepted', 'Rejected') GROUP BY rs.rewardsReceiptStatus;

3. Generate a query to capture data quality issues against the new structured relational data model

a. Missing Values: Identify the missing values in any of the tables by using the NULL values identification. This query will identify the NULL values and we can further decide on removing those records or filling the values in those records either based on mean values or forward fill or backward fill.

SELECT 'Receipts' AS table\_name, COUNT(\*) AS missing\_count FROM Receipts WHERE userId IS NULL OR purchaseDate IS NULL OR totalSpent IS NULL

UNION ALL

SELECT 'ReceiptStatus', COUNT(\*) FROM ReceiptStatus WHERE receiptId IS NULL OR rewardsReceiptStatus IS NULL OR dateScanned IS NULL

UNION ALL

SELECT 'ReceiptItems', COUNT(\*) FROM ReceiptItems WHERE receiptId IS NULL OR brandId IS NULL OR itemName IS NULL OR itemPrice IS NULL;

b. Items without corresponding receipts: Identify the records where id is not present or deleted whereas receipt item table has corresponding record present.

SELECT COUNT(\*) FROM ReceiptItems ri LEFT JOIN Receipts r ON ri.receiptId = r.\_id WHERE r.\_id IS NULL;

c. Find duplicated receipt entries: Records where receipt id is duplicated

SELECT receiptId, COUNT(\*) FROM ReceiptStatus GROUP BY receiptId HAVING COUNT(\*) > 1;

d. Standardize rewardsReceiptStatus: Find out if there are any records where any value other than accepted, rejected or pending exists and set that value to pending.

UPDATE Receipts SET rewardsReceiptStatus = 'Pending' WHERE rewardsReceiptStatus NOT IN ('Accepted', 'Rejected', 'Pending');

e. Same brand name with different brandId: Verify and remove if there are any records that might cause ambiguity in brand names as more than one brand is present.

DELETE FROM Brands WHERE \_id NOT IN (SELECT MAX(\_id) FROM Brands GROUP BY name);

4. Short email or Slack message to the business stakeholder

I looked into the data schema and json tables and have done some analysis to identify the general traits, data trends and recommended actions with few insights. I have addressed few business questions as well:  
1. Identified the top 5 brands by receipts scanned in most recent month.

2. Listed the average spend for receipts with 'Accepted' vs. 'Rejected' statuses so that further action could be taken.

3. Listed total number of items purchased with 'Accepted' vs. 'Rejected' statuses

I went ahead to understand the anomalies in the data to get it rectified before consuming it further. This step helped me in enhancing the data quality of records. For example:

1. Missing values in critical fields like userId, purchaseDate, and totalSpent.
2. Orphaned records in the ReceiptItems table without corresponding receipts.
3. Duplicate entries in the ReceiptStatus table.
4. Standardized the RewardsReceiptStatus.

These are the few stated ways to detect the anomalies. Further, I would recommed to take actions beforehand. For example:

1. Implementing constraints and validations during data ingestion.
2. Regularly auditing data to detect and resolve anomalies.
3. Enhancing the ETL pipeline to enforce referential integrity.

Finally, scaling considerations include optimizing joins across the normalized schema and indexing frequently queried fields. For example: Indexing on ReceiptId and maintaining constraint of referential integrity when creating the receipt status and receipt items table could help in avoiding any records which are orphaned.   
If the data size is huge and expected ae the daily new updates, then partitioning could be introduced on daily or monthly records as It will help to decrease the execution time for most recent spend calculated.