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

**A screenshot of a computer

Description automatically generated**

This ER (Entity-Relationship) diagram represents a system involving users, receipts, reward items, and brands. Here’s a breakdown of the relationships and key entities:

1. Users: Each user has a unique ID (\_id), role, created date, last login timestamp, and a flag indicating if they are active or not.
2. Receipts: Each receipt is tied to a user (userId), capturing details like bonus points earned, total amount spent, and the purchase date. The userId is a foreign key from the Users table.
3. RewardsReceiptItems: Each reward receipt item is associated with a receipt (receiptId). It contains information about the item, such as barcode, partner item ID, final price, and the quantity purchased. The receiptId is a foreign key from the Receipts table. This table also links to the Brands table through the brandId field.
4. Brands: Each brand has a unique ID (\_id), a name, and a category.

Relationships:

* Users to Receipts: One-to-many relationship. A user can have multiple receipts.
* Receipts to RewardsReceiptItems: One-to-many relationship. Each receipt can have multiple associated reward items.
* RewardsReceiptItems to Brands: Many-to-one relationship. Each reward item links to a single brand.

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

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

SELECT b.name, COUNT(r.\_id) AS recp\_count

FROM brands b

LEFT OUTER JOIN rewardsreceiptitems ri ON ri.brandId = b.\_id

LEFT OUTER JOIN receipts r ON r.\_id = ri.receiptId

WHERE TO\_TIMESTAMP(r.purchaseDate / 1000) >= DATE\_TRUNC('MONTH', TO\_TIMESTAMP((SELECT MAX(r.purchaseDate / 1000) FROM receipts)))

GROUP BY b.name

ORDER BY recp\_count DESC

LIMIT 5;

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

select b.name, sum(r.totalSpent) totalspent from brands b

LEFT OUTER JOIN rewardsreceiptitems ri ON ri.brandId = b.\_id

left outer join receipts r on r.\_id = ri.receiptId

left outer join users u on u.\_id = r.userId

where TO\_TIMESTAMP(u.createdDate/1000) >= DATEADD(MONTH, -6, current\_date) group by b.name order by r.totalspent Limit1;

3. 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 cte AS (

SELECT \_id, dateScanned

FROM receipts

WHERE dateScanned >= DATEADD(MONTH, -1, CURRENT\_DATE())

),

top\_brand\_rank AS (

SELECT b.name,

COUNT(r.\_id) AS recpt\_count,

ROW\_NUMBER() OVER (ORDER BY COUNT(r.\_id) DESC) AS rnk,

EXTRACT(MONTH FROM TO\_TIMESTAMP(r.purchaseDate / 1000)) AS purchased\_month,

EXTRACT(YEAR FROM TO\_TIMESTAMP(r.purchaseDate / 1000)) AS purchased\_year

FROM brands b

LEFT OUTER JOIN rewardsreceiptitems ri ON ri.brandId = b.\_id

LEFT OUTER JOIN receipts r ON r.\_id = ri.receiptId

GROUP BY b.name

)

SELECT tb.name,

tb.recpt\_count,

tb.rnk,

CASE

WHEN tb.purchased\_month = EXTRACT(MONTH FROM CURRENT\_DATE())

AND tb.purchased\_year = EXTRACT(YEAR FROM CURRENT\_DATE())

THEN 'Current\_Month'

WHEN tb.purchased\_month = EXTRACT(MONTH FROM DATEADD(MONTH, -1, CURRENT\_DATE()))

AND tb.purchased\_year = EXTRACT(YEAR FROM DATEADD(MONTH, -1, CURRENT\_DATE()))

THEN 'Previous\_Month'

ELSE 'other\_month'

END AS comparisons

FROM top\_brand\_rank tb

WHERE tb.rnk <= 5

ORDER BY tb.rnk;

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

1. Observed Nulls in totalSpent column in receipts.Json where receipt Id = ‘5ff5ecb90a7214ada10005f9’, and brands.json where barnd Id = ‘5332f9fce4b03c9a25efd1cd’ also in users.json where userid =’ 5ff36adb0a720f0523000590’

Select count(\*) from receipts where totalSpent is null;

1. Duplicates on PK in users.json ex: id 5ff36a3862fde912123a4460

Select count(\*), \_id from receipt

Group by \_id having count(\*)>1;

1. Inconsitent Data in brands.json and in receipts.json, it should be a proper data.
2. Make sure all Pk in users present in receipts but not in users in receipts ex: 6006f1c5e257127d5d9aedfb

Select userId from receipt r

Left outer join on users u on u.\_id = r.userId

Where u.\_id is not null;

**Fourth: Communicate with Stakeholders:**

Email:

Subject: Current Project update and Analysis

Hello Team,

Hope this email finds you well.

I have analyzed the given data and created a structed relationship diagram to get clear picture about it. Also, I’ve observed few potential quality issues in the data which are mentioned below

1. There are few missing data in the files, especially for ‘totalSpent’ column in the receipts.json file - which I would like to fix it but I need a clarification on how we can handle the missing or negative values.
2. Inconsistent of data – Need some additional information about expected fields behavioral like nullable, field range values.
3. Some userIds which are in receipts.json file but not in users.json – Which raise a missing foreign key issues, need some validation rules to check the relationship between ‘Receipts’ and ‘users’.
4. Also, there are few duplicates on PKs in 3 files- These should be unique and cause inconsistencies in downstream data. Can you please confirm if this is an expected behavior?

Once all these addressed, I’ll begin with optimizing the data sets, to ensure smooth implementation and maintain scalability in production.

Please let me know if you have any questions or need any additional information

Thank you!!

Bhavani M