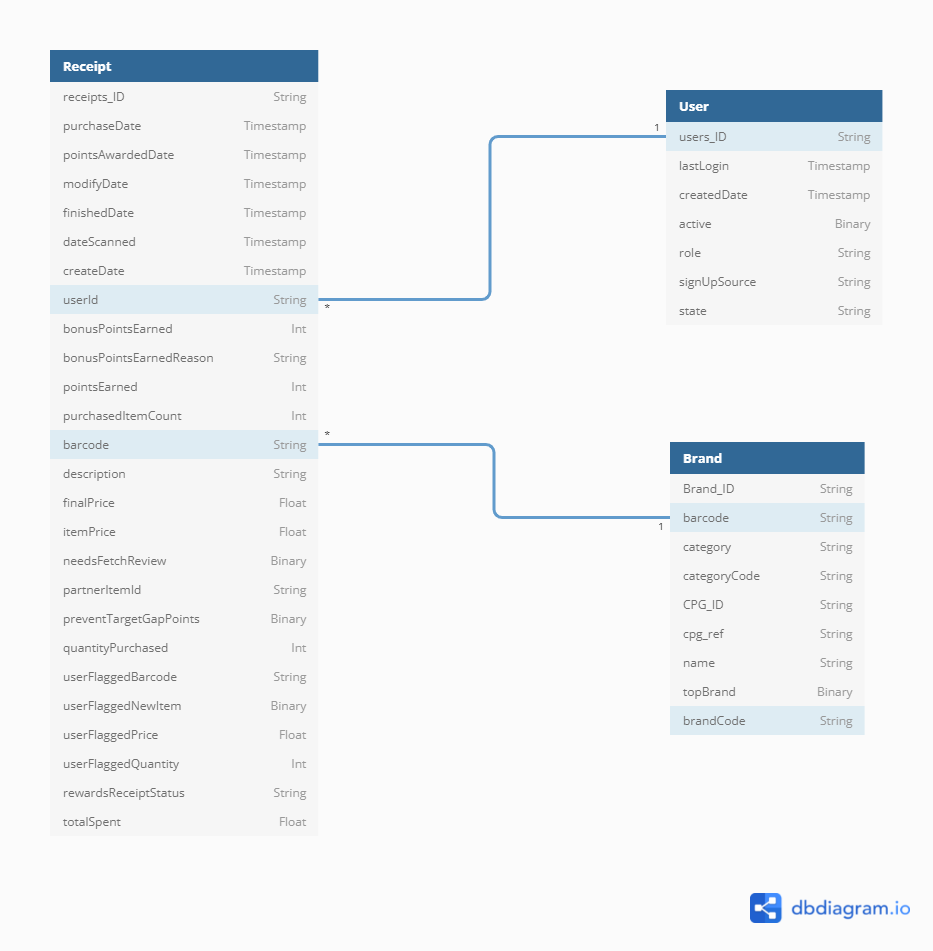
**Fetch Rewards Coding Exercise**

**1: Review unstructured JSON data and diagram a new structured relational data model.**



The ERD diagram can be drawn as shown above with the mappings to each table. After cleaning the Receipts JSON file, a barcode column was found which can be used as a key to the brands table.

|  |  |
| --- | --- |
| Table **Receipt** | Table **User** |
| { | { |
| receipts\_ID String [not null] | users\_ID String [not null] |
| purchaseDate Timestamp | lastLogin Timestamp |
| pointsAwardedDate Timestamp | createdDate Timestamp |
| modifyDate Timestamp | active Binary |
| finishedDate Timestamp | role String |
| dateScanned Timestamp | signUpSource String |
| createDate Timestamp | state String |
| userId String [not null] | } |
| bonusPointsEarned Int |  |
| bonusPointsEarnedReason String | Table **Brand** |
| pointsEarned Int | { |
| purchasedItemCount Int | Brand\_ID String [not null] |
| barcode String | barcode String |
| description String | category String |
| finalPrice Float | categoryCode String |
| itemPrice Float | CPG\_ID String [not null] |
| needsFetchReview Binary | cpg\_ref String |
| partnerItemId String | name String |
| preventTargetGapPoints Binary | topBrand Binary |
| quantityPurchased Int | brandCode String |
| userFlaggedBarcode String | } |
| userFlaggedNewItem Binary |  |
| userFlaggedPrice Float | Ref: Receipt.userId > User.users\_ID // Many to One |
| userFlaggedQuantity Int |  |
| rewardsReceiptStatus String | Ref: Receipt.barcode > Brand.barcode // Many to One |
| totalSpent Float |  |
| } |  |

### 2: Write a query that directly answers a predetermined question

**Q1): What are the top 5 brands by receipts scanned for most recent month?**

SELECT B.Brand\_ID, count(distinct R.receipts\_ID) as purchase\_count

from Receipt R

INNER JOIN Brand B on R.barcode = B.barcode

WHERE YEAR(TO\_DATE(R.purchaseDate)\*100 + MONTH(TO\_DATE(R.purchaseDate))) in

(Select MAX(YEAR(TO\_DATE(purchaseDate)\*100 + MONTH(TO\_DATE(purchaseDate)))) as latest\_month FROM Receipt)

GROUP BY B.Brand\_ID

ORDER BY purchase\_count

LIMIT 5;

***Note: We used an inner query to identify the latest MONTH in the data and find the top 5 brands in terms of no of transactions for that particular MONTH***

**Q2:** How does the ranking of the top 5 brands by receipts scanned for the recent MONTH compare to the ranking for the previous MONTH?

SELECT x.Brand\_Name, x.rank\_1st AS Rank\_MONTH\_1st, y.rank\_2nd AS Rank\_MONTH\_2nd

FROM

(SELECT B1.name AS Brand\_Name, COUNT(DISTINCT R1.receipts\_ID) AS purchASe\_COUNT,

RANK() OVER(ORDER BY COUNT(DISTINCT R1.receipts\_ID) DESC) AS rank\_1st

FROM Receipt R1

INNER JOIN Brand B1 on R1.barcode = B1.barcode

INNER JOIN

(

SELECT Max(TO\_DATE(purchASeDate)) AS latest\_MONTH FROM Receipt

) c on R1.purchASeDate=c.latest\_MONTH

GROUP BY B1.Brand\_ID

ORDER BY purchASe\_COUNT

LIMIT 5) x

LEFT JOIN

(SELECT B2.name AS Brand\_Name, COUNT(DISTINCT R2.receipts\_ID) AS purchASe\_COUNT,

RANK() OVER(ORDER BY COUNT(DISTINCT R2.receipts\_ID) DESC) AS rank\_2nd

FROM receipt R2

INNER JOIN Brand B2 on R2.barcode = B2.barcode

INNER JOIN

(

SELECT YEAR(TO\_DATE(purchASeDate)\*100 + MONTH(TO\_DATE(purchASeDate))) AS YEAR\_MONTH,

rank() over (ORDER BY YEAR(TO\_DATE(purchASeDate)\*100 + MONTH(TO\_DATE(purchASeDate))) desc) AS MONTH\_rank

FROM Receipt

) a on YEAR(TO\_DATE(R2.purchASeDate)\*100 + MONTH(TO\_DATE(R2.purchASeDate))) = a.YEAR\_MONTH

WHERE a.MONTH\_rank = 2) y

on x.Brand\_Name = y.Brand\_Name;

Note: In this query, we first identify the most recent MONTH and identify the top5 brands in terms of transactions and then compare those 5 brands with the lASt MONTH data and get rank of top5 brands for lASt two MONTHs for comparison

**Q3: When considering average spend FROM receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?**

SELECT R.rewardsReceiptStatus, AVG(R.totalSpent) AS Average\_spent

FROM (SELECT DISTINCT receipts\_ID, rewardsReceiptStatus, totalSpent FROM receipt) R

GROUP BY R.rewardsReceiptStatus

ORDER BY Average\_spent desc;

***Note: In this query, we first get unique records for specific columns to get single row for each receipt, and then we group our data with respect to receipt status and calculate average amount spent.***

**Q4: When considering total number of items purchased receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?**

SELECT R.rewardsReceiptStatus, SUM(R.quantityPurchASed) AS items\_purchASed

FROM (SELECT DISTINCT receipts\_ID, rewardsReceiptStatus, quantityPurchASed FROM receipt) R

GROUP BY R.rewardsReceiptStatus

ORDER BY items\_purchASed desc;

**Note: In this query, we first get unique records for specific columns to get single row for each receipt, and then we group our data with respect to receipt status and calculate sum of quantity purchased.**

**Q5: Which brand has the most spend among users who were created within the past 6 months?**

SELECT X.Brand\_Name, SUM(X.finalPrice) AS total\_Spent

FROM

(

SELECT DISTINCT B1.name AS Brand\_Name, U1.user\_ID, R1.finalPrice

FROM Receipt R1

INNER JOIN Brand B1 on R1.barcode = B1.barcode

INNER JOIN User U1 on U1.user\_ID = R1.userID

WHERE DATEDIFF(MONTH, TO\_DATE(R1.createdDate), current\_date) <= 6

) X

GROUP BY X.Brand\_Name

ORDER BY total\_Spent DESC

LIMIT 1;

Note: In this query, we first filter our data for last 6 months using DATEDIFF function from the current date and join all tables to get specific columns. After that, GROUP BY data with respect to. brand name and sum up the final prices paid.

**Q6: Which brand has the most transactions among users who were created within the past 6 months?**

SELECT X.Brand\_Name, COUNT(DISTINCT X.receipts\_ID) AS no\_of\_transactions

FROM

(

SELECT DISTINCT B1.name AS Brand\_Name, U1.user\_ID, R1.receipts\_ID

FROM Receipt R1

INNER JOIN Brand B1 on R1.barcode = B1.barcode

INNER JOIN User U1 on U1.user\_ID = R1.userID

WHERE DATEDIFF(MONTH, TO\_DATE(R1.createdDate), current\_date) <= 6

) X

GROUP BY X.Brand\_Name

ORDER BY no\_of\_transactions DESC

LIMIT 1;

Note: In this query, we first filter our data for last 6 months using DATEDIFF function from the current date and join all tables to get specific columns. After that, GROUP BY data with respect to brand name and count the number of transactions.

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

* To find a common key between receipts table and brands table, the data had to be further cleaned to find barcode column which was nested withing dictionaries.
* Further more, a lot of the data is missing especially data in the barcode column which is important to connect the tables.
* The date columns are stored in Timestamp format. It is better to convert it to Date format and analyze the data.

### **4) Fourth: Communicate with Stakeholders**

Hello Stakeholder!

I was able to work through the datasets, clean them and understand the how the data was structured. The data was quite messy, could you elaborate on the source of these datasets? Moreover, there seems to a good amount of missing data especially in the bar code feature which is the best link between various users, brands and receipts.

It would be useful in future to keep the data cleaner and the timestamps more uniform too. Importantly, I was able to answer the import questions you had put forward with some quick SQL data pulls.

Please let me know if you have any more questions or if you are looking for something specific!

Thanks and Regards!

Sachin Santhosh

Data Analyst