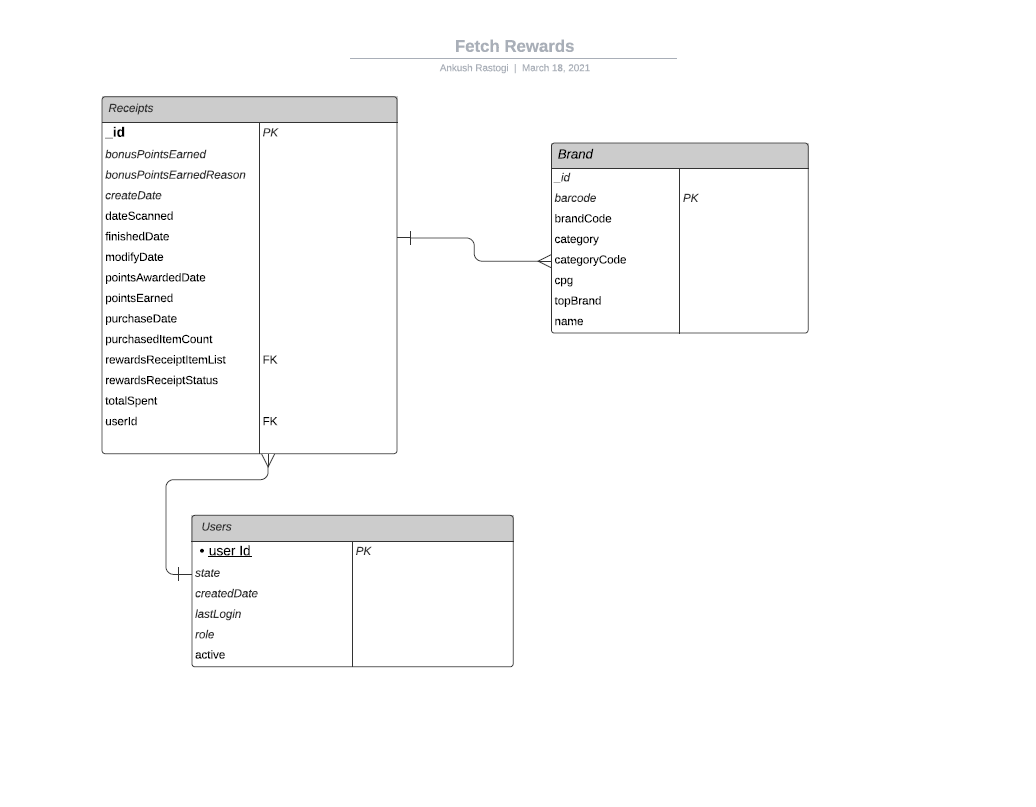
**ER Diagram**

NOTE: if we are using this system for reporting purpose then I will create one more table that will contains all the elements for rewardsReceiptitemList associated to receipt Id. By doing this it will be easy to join with brand dataset and create multiple report out of this.

Relationships between the three datasets:

One to Many:

Receipts <Brand

One to Many:

Users < Receipts

Table Structure:

Receipts:

|  |  |
| --- | --- |
| Column Names  \_id  bonusPointsEarned  bonusPointsEarnedReason  createDate  dateScanned  finishedDate  modifyDate  pointsAwardedDate  pointsEarned  purchaseDate  purchasedItemCount  rewardsReceiptItemList  rewardsReceiptStatus  totalSpent  userId | Datatype  Nvarchar(255)  Int  Nvarchar(2000)  Bigint  Bigint  Bigint  Bigint  Bigint  Float  Bigint  Int  Nvarchar(max)  Nvachar(50)  Float  Nvarchar(225) |

rewardsReceiptItemList is collection; so, I am using Nvarchar(max) datatype since it could have multiple elements in the collection

Users:

|  |  |
| --- | --- |
| \_id  state  createdDate  lastLogin  role  active | Nvarchar(225)  Char(2)  Bigint  Bigint  Nvarchar(100)  Bit |

Brand:

|  |  |
| --- | --- |
| \_id  barcode  brandCode  category  categoryCode  cpg  topBrand  name | Nvarchar(255)  Bigint  Nvarchar(50)  Nvarchcar(200)  Nvarchcar(200)  Nvarchar(max)  Bit  Nvarchar(200) |

Cpg is collection; so I am using Nvarchar(max) datatype since it could have multiple elements in the collection

topBrand: I am using it as a bit and can be store as a 1 and 0

**SQL Queries**

2: Is the average spend from receipts with 'rewardsReceiptStatus = Rejected' greater than the average spends of accepted receipts? What about the total number of items purchased?

with cte1 as

(

select avg(totalSpent) as avg\_spend,\_id as uuid

from Receipts where rewardsReceiptStatus = 'Rejected'

group by \_id

), cte2 as

(select avg(totalSpent) as avg\_spend,\_id as uuid

from Receipts where rewardsReceiptStatus = 'accepted'

group by \_id)

select case when cte1.avg\_spend> cte2.avg\_spend

then 'greater' else 'not greater' end as 'avg\_spend\_results'

from cte1 join cte2

on cte1.uuid=cte2.uuid

**‘Total number of items purchased’**

with cte1 as

(

select sum(purchasedItemCount) as ItemCountSum, \_id as uuid

from Receipts where rewardsReceiptStatus = 'Rejected'

group by \_id

), cte2 as

(select sum(purchasedItemCount) as ItemCountSum , \_id as uuid

from Receipts where rewardsReceiptStatus = 'accepted'

group by \_id)

select sum(cte1. ItemCountSum + cte2. ItemCountSum) as ‘total number of items purchased’

from cte1 join cte2

on cte1.uuid=cte2.uuid

**What are the top 5 brands that Fetch has seen receipts for in the most recent month of data? How do the ranking of those brands compare to the previous month?**

This query will fetch each barcode associated with receipt id

Select \_id, v.value as barcode , totalSpent into #getAllBarcodes from Receipts

cross apply **openjson**(rewardsReceiptItemList, '$.rewardsReceiptitemList') t

cross apply **openjson**(t.value,'$.') v where createdDate>dateadd(month,-6,getdate())

**brands count for current month**

with cte as (

select count(receipts .\_id) over(order by name desc) as rn ,brand from receipts join users on receipts.userId=users.\_id join #getAllBarcodes on #getAllBarcodes.\_id=receipts.\_id

where role =’ fetch-staff’ and Month(createDate)= month(getdate())

group by brand

) select \* into #current\_month from cte where rn<6

**brands count for previous month**

with cte as (

select count(receipts .\_id) over(order by name desc) as rn ,brand from receipts join users on receipts.userId=users.\_id join #getAllBarcodes on #getAllBarcodes.\_id=receipts.\_id

where role =’ fetch-staff’ and Month(createDate)= dateadd(month,-1,month(getdate()))

group by brand

) select \* into #prev\_month from cte where rn<6

We can compare the brands for both the months by using the following query

Select #prev\_month.rn as count1, # current \_month.rn as count1 from #current\_month join #prev\_month on #current\_month.brand=#prev\_month.brand

**3: Write a short email or Slack message to the LOB asking clarifying questions. What other information would you need to help you optimize the data assets you're trying to create?**

What data quality issues have you found in the data?

I don’t see any issue with the dataset as per my understanding. Let’s assume that this data is stored in the NoSQL database and if we want to move this data to RDMS then it can be easily done by writing a script to fetch specific JSON elements that will move data to RDMS.

A couple of things we can do to make this process efficient: We can do indexing; An **index** in **MongoDB** is a special data structure that holds the data of few fields of documents on which the **index** is created. **Indexes** improve the speed of search operations in the database because instead of searching the whole document, the search is performed on the **indexes** that hold only a few fields.

How did you discover those issues and what kinds of questions would you ask to resolve them?

I have used JSON viewer to see the structure of datasets.

The question that I would ask will be:

How big the tables would be?

Do I need to do partitions?

How much data will be stored in tables; will it be days, months, and years’ worth of data