Monday, April 18, 2022 11:24 AM

Rece.Pts

- **_id:** uuid for this receipt
- **bonusPointsEarned:** Number of bonus points that were awarded upon receipt completion
- **bonusPointsEarnedReason:** event that triggered bonus points
- **createDate:** The date that the event was created
- **dateScanned:** Date that the user scanned their receipt
- **finishedDate:** Date that the receipt finished processing
- **modifyDate:** The date the event was modified
- **pointsAwardedDate:** The date we awarded points for the transaction
- **pointsEarned:** The number of points earned for the receipt
- **purchaseDate:** the date of the purchase
- **purchasedItemCount:** Count of number of items on the receipt
- **rewardsReceiptItemList:** The items that were purchased on the receipt
- **rewardsReceiptStatus:** status of the receipt through receipt validation and processing
- **totalSpent:** The total amount on the receipt
- **userId:** string id back to the User collection for the user who scanned the receipt
- brandId

Receipts

From <https://fetch-hiring.s3.amazonaws.com/data-analyst/ineeddata-data-modeling/data-modeling.html>

This is the simplest structure based upon what was given already, just linking main tables by adding the obvious key pairs that would connect the user table and brand table. I could also see a more complete structure change, by breaking receipts into basically just a keyed table, and connecting/making a **date table** and a **point table**, that would simplify receipts into a more usable form depending on how much processing needs to be done.

Users

- **_id:** user Id
- **state:** state abbreviation
- **createdDate:** when the user created their account
- **lastLogin:** last time the user was recorded logging in to the app
- **role:** constant value set to 'CONSUMER'
- **active:** indicates if the user is active; only Fetch will de-activate an account with this flag

Users Data

From <https://fetch-hiring.s3.amazonaws.com/data-analyst/ineeddata-data-modeling/data-modeling.html>

Seems redundant as a constant here

Could just automate/link to lastLogin, say if lastLogin > 180 days, something like that

Brands

Need way to link these tables
Uncertain if barcode is a single unique identifier to a brand, or to one brand product?

- **_id:** brand uuid
- **barcode:** the barcode on the item
- **brandCode:** String that corresponds with the brand column in a partner product file
- **category:** The category name for which the brand sells products in
- **categoryCode:** The category code that references a BrandCategory
- **cpg:** reference to CPG collection
- **topBrand:** Boolean indicator for whether the brand should be featured as a 'top brand'
- **name:** Brand name

Brand Data

From <https://fetch-hiring.s3.amazonaws.com/data-analyst/ineeddata-data-modeling/data-modeling.html>

2

Monday, April 18, 2022 11:43 AM

- What are the top 5 brands by receipts scanned for most recent month?
- How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?
- When considering *average spend* from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?
- When considering *total number of items purchased* from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?
- Which brand has the most *spend* among users who were created within the past 6 months?
- Which brand has the most *transactions* among users who were created within the past 6 months?

From <https://fetch-hiring.s3.amazonaws.com/data-analyst/ineeddata-data-modeling/data-modeling.html>

1.
```
SELECT COUNT(brand_id) as brandTotal, dateScanned
FROM receipts
WHERE MONTH(dateScanned) == MONTH(getdate())
ORDER BY brandTotal
```

2.
```
SELECT COUNT(brandId) as brandTotal, dateScanned
FROM receipts
WHERE MONTH(dateScanned) == MONTH(DATEADD(mm,-1, getdate())
ORDER BY brandTotal
DATEADD (datepart , number , date )
```

Then compare this to the previous results. Could also store each of the last in a new table, where we could join tables and compare, or ideally information like this would be automatically queried each month and add to this hypothetical table. Then you could take aggregate statistics over the months, like finding an average ranking of each brand yearly.

3.
```
SELECT avg(totalSpent), rewardsReceiptStatus
FROM receipts
GROUP BY rewardsReceiptStatus
```

4.
```
SELECT avg(rewardsItemCount), rewardsReceiptStatus
FROM receipts
GROUP BY rewardsReceiptStatus
```

Assuming for these last two, that there is no other options other than 'Accepted' or 'Rejected', otherwise could add a WHERE clause to select just those two conditions, let's say if there was a pending. Not a huge deal though, would just give a 3rd metric.

5.

```
SELECT avg(totalSpent), userId
FROM receipts
INNER JOIN users ON userId
WHERE MONTH(createdDate) == MONTH(DATEADD(mm, -6, getdate())
GROUP BY brandId
```

6.

```
SELECT COUNT(receiptId), userId
FROM receipts
INNER JOIN users ON userId
WHERE MONTH(createdDate) == MONTH(DATEADD(mm, -6, getdate())
GROUP BY brandId
```

3

Tuesday, April 19, 2022 10:52 AM

I downloaded the jsons, and converted them to a csvs using a tool, after having many issues with nested jsons working with the R libraries I was using, then imported them into R studio, where I looked at some statistics. Checking table summaries with pysch, and then checking for null values with tidyverse (could also be done with normal R). I'm sure there is a lot that I could dig into, just after visual inspection there is some table inconsistencies with dimensions, and a lot to dig through in receipts, but I would need to optimize things more, as it is a rather large amount of data to load in one chunk into R.

So overall I would say there is a data quality issue just with the NULL values alone, but there are some other easily diagnosed issues as well, such as making sure all values are set to the correct data type.

```{r}
#load data
users <- read.csv('users.csv')
brands <- read.csv('brands.csv')
receipts <- read.csv('receipts.csv')
```

```{r}
library('psych')

#Numerical analysis
describe(users)
describe(brands)
describe(receipts)
```
```{r}
library('tidyverse')

#checking for NULL values
a <- tibble(users)
sum(map_dbl(a, ~length(~.x) > 0))

b <- tibble(brands)
sum(map_dbl(b, ~length(~.x) > 0))

c <- tibble(receipts)
sum(map_dbl(c, ~length(~.x) > 0))

```

4

Wednesday, April 20, 2022 12:29 PM

To whomever it may concern,

I am writing to you to discuss some inquiries about a portion of the Fetch data, related to Users, Brands, and Receipts, so that I may do a better analysis of it.

Firstly, to clarify about some of the data itself, I am uncertain why the Users data contains a "role" value, when it appears that this should always be set to "CONSUMER". Do you know someone who I should contact who originated this structure, so that I can remove or keep this field? I also need to know what dictates a "Top Brand", so that when I analyze the data, if I need to filter these into more distinct metrics.

On to the data, under some of the first data quality analysis I did, I discovered a number of NULL values, when checking all of these data. This was just done quickly in R Studio, and I will continue looking for other quality issues, but this was the earliest and most obvious issue.

I believe that this needs to be addressed quickly, particularly in the Receipts data. If you could help me get into contact with someone who knows how the Receipts data is generated, I may be able to resolve these issues, or even collect this data depending on how the NULL values were generated.

Also to make my analysis more complete, the point generation will be extremely important, so that I weight my analysis later on, especially if I introduce any ML after this, do you know who would be knowledgeable on the point information?

That's the end of my questions, I think that once I know these, I will be able to start optimizing my analysis. My only other concerns are on my end, I think that I will have to make a system to import and analyze the data as it being produced, as the receipts data is already getting too large for what would be usable in my environment. I would just have to setup some connections at a later date, when I fully map out this plan.

If you have any questions for me, just let me know, I know that this is a dense message. Thank you!

-Patrick Lutz