

Evaluating Data Quality Issue

Before looking at the issues , I used an online tool for converting json files to csv format and have analyzed in Google Colab after uploading it to drive

Issue 1 : Duplicate Entries in Users table

- This is a critical data quality Issue .
- There are more than 50 % duplicated data in the USERS table .
- This issue can serious wreck any actions taken based on User Count .
- I have dropped duplicate entries using drop_duplicates() function in pandas

```
[57] users.shape  
  
(495, 7)  
  
[66] users_no_duplicates = users.copy()  
      users_no_duplicates = users_no_duplicates.drop_duplicates()  
      users_no_duplicates.shape  
  
(212, 7)
```

Real User Count : 212

Duplicated Entries : 283

Issue 2 : Useless Column “ROLE” in Users table

- The column “ROLE” in the USERS table does not provide any information for analysis .
- It established data redundancy and takes up storage space
- The column has been dropped

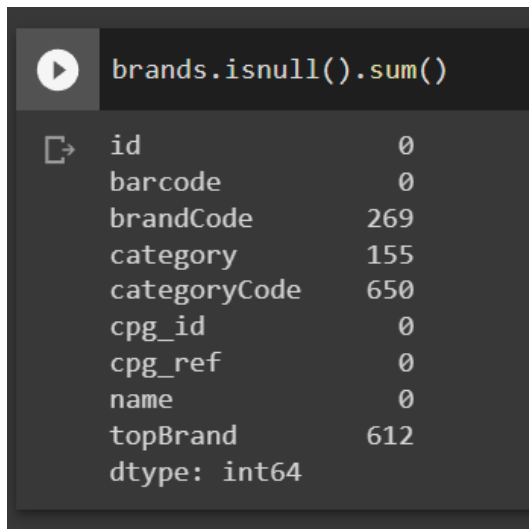
```
[70] users_no_duplicates = users_no_duplicates.drop("role",axis=1)  
  
users_no_duplicates.head()
```

	id	active	createdDate	lastLogin	signUpSource	state
0	5ff1e194b6a9d73a3a9f1052	True	1609687444800	1.609688e+12	Email	WI
3	5ff1e1eacfc6c399c274ae6	True	1609687530554	1.609688e+12	Email	WI
6	5ff1e1e8cfc6c399c274ad9	True	1609687528354	1.609688e+12	Email	WI
7	5ff1e1b7cfc6c399c274a5a	True	1609687479626	1.609687e+12	Email	WI
9	5ff1e1f1cfc6c399c274b0b	True	1609687537564	1.609688e+12	Email	WI

Issue 3 : Numerous missing values in Brands table

- Columns “categoryCode” and “topBrand” has approximately 55% of the data as missing values
- These column provides very less information and can be very misleading when filtering tables

- These column can either be imputed by doing more research or dropped from the table
- The following table shows number of missing values in each column



```
brands.isnull().sum()
```

id	0
barcode	0
brandCode	269
category	155
categoryCode	650
cpg_id	0
cpg_ref	0
name	0
topBrand	612
dtype: int64	

Issue 4 : Numerous TEST values in Brands table

- The columns “brandCode” and “name” in BRANDS table contain a lot of test entries
- It shows that the table was not properly maintained
- It can lead to some misleading information like BrandCount

Observation :

1. Brand_test is a table containing just test brand entries .
2. It has 428 rows of test entries .
3. Doing a deeper analysis reveal that the category for the test entries are “Baking” and “Candy & Sweets” .
4. Removing test entries from the brand table reduces the number of entries in BAKING category to **12 (from 369)** and interestingly **0 (from 71)** entries in the category “Candies and Sweets” (all the entries in this category is removed)
5. This saves a lot of space and provide much better overview of data

SubTable with test entries only

```
[86] brands_test = brands[brands['name'].str.contains('test brand')]
```

```
[88] brands.shape
```

```
(1167, 9)
```

```
[89] brands_test.shape
```

```
(428, 9)
```

Category Count for subtable

```
brands_test.category.value_counts()
```

```
Baking      357
```

```
Candy & Sweets  71
```

```
Name: category, dtype: int64
```

Category Count for brands table

```
[93] brands.category.value_counts()
```

```
Baking      369
```

```
Beer Wine Spirits  90
```

```
Snacks      75
```

```
Candy & Sweets  71
```

```
Beverages   63
```

```
Magazines    44
```

```
Health & Wellness  44
```

```
Breakfast & Cereal  40
```

```
Grocery      39
```

```
Dairy        33
```

```
Condiments & Sauces  27
```

```
Frozen       24
```

```
Personal Care  20
```

```
Baby         18
```

```
Canned Goods & Soups  12
```

```
Beauty        9
```

```
Cleaning & Home Improvement  6
```

```
Deli          6
```

```
Beauty & Personal Care  6
```

```
Household     5
```

```
Bread & Bakery  5
```

```
Dairy & Refrigerated  5
```

```
Outdoor       1
```

```
Name: category, dtype: int64
```

After removing test entries

```
[94] brands_new=pd.concat([brands,brands_test]).drop_duplicates(keep=False)
```

Category Count after removing test entries

```
[97] brands_new.category.value_counts()
```

```
Beer Wine Spirits  90
```

```
Snacks           75
```

```
Beverages        63
```

```
Health & Wellness  44
```

```
Magazines         44
```

```
Breakfast & Cereal  40
```

```
Grocery          39
```

```
Dairy            33
```

```
Condiments & Sauces  27
```

```
Frozen           24
```

```
Personal Care     20
```

```
Baby             18
```

```
Baking           12
```

```
Canned Goods & Soups  12
```

```
Beauty            9
```

```
Cleaning & Home Improvement  6
```

```
Deli              6
```

```
Beauty & Personal Care  6
```

```
Household         5
```

```
Bread & Bakery     5
```

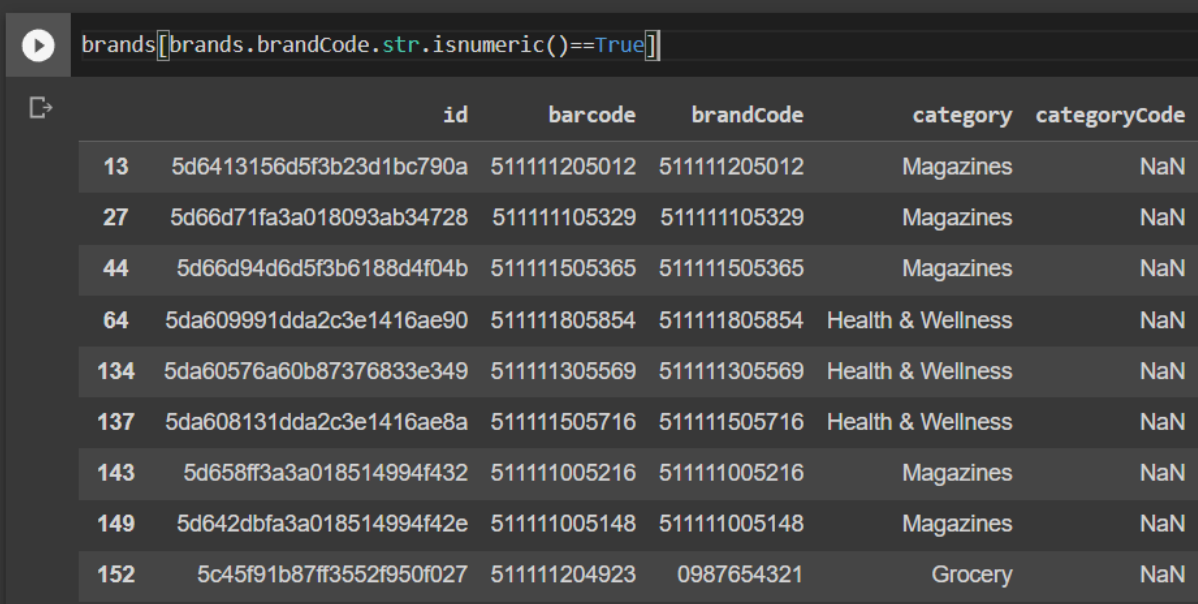
```
Dairy & Refrigerated  5
```

```
Outdoor           1
```

```
Name: category, dtype: int64
```

Issue 5 : Columns have mixed datatypes in Brands table

- Some values in the Column “barcode” are equal to barcode
- Most values in brandCode column is of “string” Datatype
- Column names can be meaningfully kept



The screenshot shows a database query interface. At the top, a query is entered: `brands[brands.brandCode.str.isnumeric()==True]`. Below the query, a table of results is displayed with 7 columns: `id`, `barcode`, `brandCode`, `category`, and `categoryCode`. The results show 10 rows of data. The first 9 rows have a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 10th row has a `category` of 'Health & Wellness' and a `categoryCode` of 'NaN'. The 11th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 12th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 13th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 14th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 15th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 16th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 17th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 18th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 19th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'. The 20th row has a `category` of 'Magazines' and a `categoryCode` of 'NaN'.

	id	barcode	brandCode	category	categoryCode
13	5d6413156d5f3b23d1bc790a	511111205012	511111205012	Magazines	NaN
27	5d66d71fa3a018093ab34728	511111105329	511111105329	Magazines	NaN
44	5d66d94d6d5f3b6188d4f04b	511111505365	511111505365	Magazines	NaN
64	5da609991dda2c3e1416ae90	511111805854	511111805854	Health & Wellness	NaN
134	5da60576a60b87376833e349	511111305569	511111305569	Health & Wellness	NaN
137	5da608131dda2c3e1416ae8a	511111505716	511111505716	Health & Wellness	NaN
143	5d658ff3a3a018514994f432	511111005216	511111005216	Magazines	NaN
149	5d642dbfa3a018514994f42e	511111005148	511111005148	Magazines	NaN
152	5c45f91b87ff3552f950f027	511111204923	0987654321	Grocery	NaN

Issue 6 :Breach of Atomic property /1NF in Receipts table

- Receipts table contain a column “ReceiptRewardsItemList” which breaches the atomic property of relational databases
- These nested columns should be split across multiple tables to prevent duplicate primary keys (if listed)

Eg:{"_id":{"\$oid":"5ff1e1eb0a720f0523000575"},"bonusPointsEarned":500,"bonusPointsEarnedReason":"Receipt number 2 completed, bonus point schedule DEFAULT (5cefdcacf3693e0b50e83a36)","createDate":{"\$date":1609687531000},"dateScanned":{"\$date":1609687531000},"finishedDate":{"\$date":1609687531000},"modifyDate":{"\$date":1609687536000},"pointsAwardedDate":{"\$date":1609687531000},"pointsEarned":"500.0","purchaseDate":{"\$date":1609632000000},"purchasedItemCount":5,"rewardsReceiptItemList":[{"barcode":"4011","description":"ITEM NOT FOUND","finalPrice":"26.00","itemPrice":"26.00","needsFetchReview":false,"partnerItemId":"1","preventTargetGapPoints":true,"quantityPurchased":5,"userFlaggedBarcode":"4011","userFlaggedNewItem":true,"userFlaggedPrice":"26.00","userFlaggedQuantity":5}],"rewardsReceiptStatus":"FINISHED","totalSpent":"26.00","userId":"5ff1e1eacfcf6c399c274ae6"}

Issue 7 : NULL value in Receipts table

- Receipts that have the status “Submitted” has most of the columns NULL
- This consumes a lot of space
- This issue arises due to storing a lot of information on a single table

```
receipts_sub=receipt_fin[receipt_fin.rewardsReceiptStatus=='SUBMITTED']

[74] receipts_sub.shape

(434, 14)

[75] receipts_sub.isnull().sum()

   _id/$oid      0
bonusPointsEarned    434
bonusPointsEarnedReason  434
createDate/$date      0
dateScanned/$date     0
finishedDate/$date    434
modifyDate/$date      0
pointsAwardedDate/$date  434
pointsEarned          434
purchaseDate/$date    434
purchasedItemCount    434
rewardsReceiptStatus  0
totalSpent            434
userId               0
dtype: int64
```