Yuhan (Caroline) Gao

11/18/2022

Fetch Reward DA Challenge

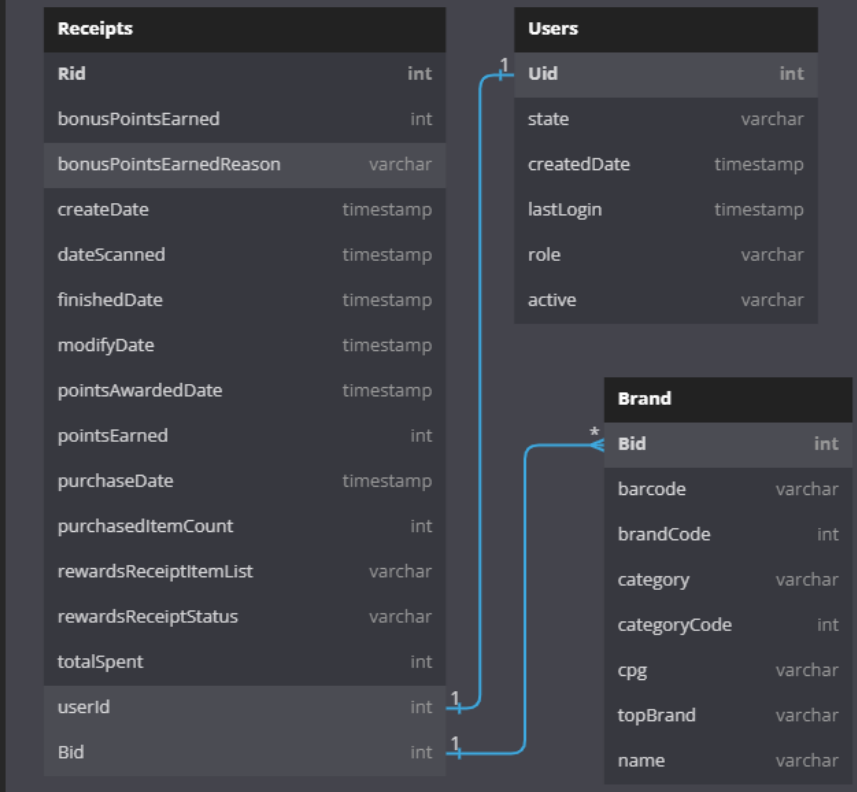
1. Relational Schema between 3 tables: ‘Receipts’, ‘Users’, and ‘Brand’, by using ‘dbdiagram.io’.

```
Table Receipts as R {

  Rid int
  bonusPointsEarned int
  bonusPointsEarnedReason varchar
  createDate timestamp
  dateScanned timestamp
  finishedDate timestamp
  modifyDate timestamp
  pointsAwardedDate timestamp
  pointsEarned int
  purchaseDate timestamp
  purchasedItemCount int
  rewardsReceiptItemList varchar
  rewardsReceiptStatus varchar
  totalSpent int
  userId int [ref: - U.Uid]
  Bid int [ref: < B.Bid]
  Indexes {
    (Rid) [pk]
  }
}

Table Users as U {
  Uid int [pk]
  state varchar
  createdDate timestamp
  lastLogin timestamp
  role varchar
  active varchar
}

Table Brand as B {
  Bid int [pk]
  barcode varchar
  brandCode int
  category varchar
  categoryCode int
  cpg varchar
  topBrand varchar
  name varchar
 }
```



2. SQL Queries for question 1, 5 and 6.

```sql
--What are the top 5 brands by receipts scanned for most recent month?

SELECT TOP 5
R.BID
,B.[NAME]
,SUM(R. TOTALSPENT)  AS [TOP SPEND AMOUNT]
FROM RECEIPTS R
INNER JOIN BRAND B
ON B.BID =R.BID
WHERE MONTH(R.DATESCANNED) = MONTH(GETDATE())
GROUP BY
R.BID
,B.[NAME]
ORDER BY [TOP SPEND AMOUNT] DESC

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

SELECT
MAX(R.TOTALSPENT) AS [SPEND AMOUNT]
,R.BID
,B.[NAME]
FROM RECEIPTS R
INNER JOIN USERS U
ON U.[UID] = R.USERID
INNER JOIN BRAND B
ON B.BID=R.BID
WHERE DATEDIFF(MONTH,U.CREATEDDATE, GETDATE())<=6
GROUP BY
R.BID

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

SELECT
MAX(C.[TRANSACTION AMOUNT])
,C.BRANDNAME
FROM (
    SELECT
    COUNT(DISTINCT R.RID) AS [TRANSACTION AOMUNT]
    ,R.BID AS [BRNADID]
    ,B.[NAME] AS [BRANDNAME]
    FROM RECEIPTS R
    INNER JOIN USERS U
    ON U.[UID] = R.USERID
    INNER JOIN BRAND B
    ON B.BID=R.BID
    WHERE DATEDIFF(MONTH,U.CREATEDDATE, GETDATE())<=6
     )C
GROUP BY
C.BRANDNAME
```

3. Evaluate Data Quality Issues

By using R and Python.

import pandas as pd

from pathlib import Path

import json

```
p = Path(r'C:\Users\Carol\OneDrive\Desktop\receipts.json')
# read json file
with p.open('r', encoding='utf-8') as f:
    data = json.loads(f.read())

#dataframe
df = pd.json_normalize(data)
```

```
import pandas as pd

from pathlib import Path
import json

p = Path(r'C:\Users\Carol\OneDrive\Desktop\receipts.json')
# read json file
with p.open('r', encoding='utf-8') as f:
    data = json.loads(f.read())

#dataframe
df = pd.json_normalize(data)
#data quality issue: exist 'extra data' error: means the input JSON file has more than one object
#per line. In general, there would be only one object per line.



import pandas as pd
from pathlib import Path
import json

# set path to file
p = Path(r'C:\Users\Carol\OneDrive\Desktop\brands.json')

# read json
with p.open('r', encoding='utf-8') as f:
    data = json.loads(f.read())

# create dataframe
df = pd.json_normalize(data)


#############################################################################
import json

with open('brands.json','r') as openfile:
      json_object = json.load(openfile)

print(json_object)
print(type(json_object))
```

Data quality issue (1): exist 'extra data' error, means the input JSON file has more than one object per line. In general, there would be only one object per line.

Issue (2): huge amount of unstructured & malformed data and exist duplicate data.

Issue (3): need to replace newlines and wrap the whole file in a pair of {}, since the json file has a large amount of statements separated by newlines instead of a single unified statement.

Issue (4): For the ‘receipts’ data and the ‘brand’ data, there is no connection between them, which are unable to join together. To solve this problem, we can add the brand uuid into receipts data, and therefore, we can use inner join and other join functions.

```
install.packages("rjson")
install.packages("jsonlite")
install.packages("RJSONIO")
library(jsonlite)
library("rjson")
json_file <- "C:/Users/Carol/OneDrive/Desktop/receipts.json"

result <- fromJSON(file=json_file)


print(result)


############################
library(RJSONIO)

D2 <- RJSONIO::fromJSON("C:/Users/Carol/OneDrive/Desktop/receipts.json")

# convert the numeric vector helpful to one string
D2$helpful <- paste(D2$helpful, collapse = " ")

D2
reviewerID        asin          reviewerName    helpful
[1,] "A3TS466QBAWB9D" "0014072149" "Silver Pencil" "0 0"

D3 <- do.call(cbind, D2)

write.csv(D3, "D3.csv")
##################################2222
json_file2 <- "C:/Users/Carol/OneDrive/Desktop/brands.json"


out <- lapply(readLines("C:/Users/Carol/OneDrive/Desktop/brands.json"), fromJSON)
l <- replicate(
  132,
  as.list(sample(letters, 20)),
  simplify = FALSE
)
df <- data.frame(matrix(unlist(l), nrow=length(l), byrow=TRUE))
```

4. Communication-Email Format
   Dear XXX manager,

   Hope you are doing well!
   There were some data quality issues existed in all the three json files after I put them in detail analysis. All of them needed to be converted to csv format by using R, Python, or other programming languages, in order to use SQL or other data analytical tools to do better & further analysis such as find which customers purchased the most in a specific period, etc. The three datasets needed to be modified by many ways, such as replace newlines and wrap the whole file in a pair of {}, since the json file has a large amount of statements separated by newlines instead of a single unified statement; or add the brand uuid into receipts data, and therefore, we can use inner join and other join functions to better determine which specific product need to be promoted more and attract more sales, etc.

   Best,
   Caroline