Finding Aggregate Savings of Customers by Joining Different Tables

Monika Baloda

2023-04-07

The Problem

Work in SQL databases for data files, work for corporate, data competitions like datafest / kaggle, then data is often stored in multiple tables. Analyses are unable to be completed unless you know how to join in the needed data.

This Challenge will require all 3 data sources:

Data-1 contains ID and Savings.

Data-2 contains ID, Tag and several other variables.

Data-3 contains Tag and Customer tiers.

The Goal is to calculate the mean Savings for all 5 customer tiers, answer example:

Customer Tier 1: XXXXXX

Customer Tier 2: YYYYYY

...

Customer Tier 5: ZZZZZZ

Solution

Step 0: loading the library

```
#library(tidyverse)
```

Step 1: Read two files

```
setwd("C:/Users/EndUser/Desktop/UCR/Spring2023/STAT208/Challanges/Challange2")

tag=read.csv("Challenge2_CustomerTag.csv")
saving=read.csv("Challenge2_Savings.csv")
tier=read.csv("Challenge2_TagTier.csv")
```

Step 2: 1:1 Merging dataframes : tag and savings

```
df=merge(tag, saving, by = "ID")
names(df)  #prints columns of merged dataframe. If saving column is there, we are fine
```

```
##  [1] "ID"                 "Encrypted.Password"  "Password"
##  [4] "Gender"             "Town"                "Region"
##  [7] "Account.Age"        "Tag"                 "Customer.Age.Years"
## [10] "Internal.IP.Address" "Savings"
```

Step-3 : 1:n Merging : merging df and tier with tag

```
df_final=merge(df, tier, by = "Tag")
dim(df_final)   #number of rows are 100000, so we are fine
```

```
## [1] 100000     12
```

```
names(df_final)  #if customer.Tier appears in columns, we are fine
```

```
##  [1] "Tag"                 "ID"                  "Encrypted.Password"
##  [4] "Password"            "Gender"              "Town"
##  [7] "Region"              "Account.Age"         "Customer.Age.Years"
## [10] "Internal.IP.Address" "Savings"             "Customer.Tier"
```

```
#head(df_final)  #one can have a look at final dataframe
```

Step-4 : Aggregating Savings based on Customer Tiers

```
agg_df = aggregate(Savings ~ Customer.Tier, data = df_final, FUN = sum)
```

Printing the Final Answer

```
agg_df
```

```
##     Customer.Tier    Savings
## 1 Customer Tier 1 1146992314
## 2 Customer Tier 2 2961512691
## 3 Customer Tier 3 3367355233
## 4 Customer Tier 4 1993978638
## 5 Customer Tier 5 1193150400
```