SPS-DATA607-ASSIGNMENT-2-RESULTS

Tage N Singh

2021-02-14

###**This page provides some sample code and outputs for the Multi-Table Project**

```
## Warning: package ’ggplot2’ was built under R version 4.0.3
```

```
## Warning: package ’tidyverse’ was built under R version 4.0.3
```

```
## Warning: package ’tibble’ was built under R version 4.0.3
```

```
## Warning: package ’readr’ was built under R version 4.0.3
```

```
## Warning: package ’forcats’ was built under R version 4.0.3
```

```
## Warning: package ’RMySQL’ was built under R version 4.0.3
```

```
## Warning: package ’DBI’ was built under R version 4.0.3
```

```
mydb = dbConnect(MySQL(),user=’localtest’,password=’Localtest!’,dbname=’shows’,host=’localhost’)

#**Note** that the credentials are local to this implementation of MySQL

dbListTables(mydb)
```

```
## [1] "netflixratings" "survey"
```

```
rs = dbSendQuery(mydb, "SELECT * FROM survey")

data = fetch(rs)

df1 <- data.frame(data)

tibble (df1)
```

```
## # A tibble: 10 x 12
##    Question title did.not.see.thi~ Very.dissatisfi~ Dissatisfied
##       <int> <chr>            <int>            <int>        <int>
##  1        1 The ~               12                0            0
##  2        2 Brea~               16                0            0
##  3        3 Brid~               23                0            4
##  4        4 Sher~               17               0            0
##  5        5 Ozark               19               2            0
##  6        6 Hann~               25               0            3
##  7        7 The ~               29               1            0
##  8        8 Coll~               25               0            0
##  9        9 Comm~               22               1            0
## 10       10 Schi~               18               1            1
## # ... with 7 more variables: Neither.satisfied.nor.dissatisfied <int>,
## #   Satisfied <int>, Very.satisfied <int>, Total <int>, Total.Responses <int>,
## #   Missing.Responses <int>, Last.Update <chr>
```

```r
rs = dbSendQuery(mydb, "SELECT * FROM netflixratings")

data = fetch(rs)

df2 <- data.frame(data)

tibble(df2)
```

```
## # A tibble: 487 x 7
##    title rating ratingLevel ratingDescripti~ release.year user.rating.sco~
##    <chr> <chr>  <chr>                  <int>        <int>            <int>
##  1 10 T~ PG-13  adult cont~               80         1999               68
##  2 100 ~ TV-MA  For mature~              110         2016                0
##  3 30 R~ TV-14  Parents st~               90         2012               66
##  4 5 to~ R      some sexua~              100         2014                0
##  5 A Mo~ PG     some actio~               60         2011                0
##  6 A We~ TV-MA  For mature~              110         2016                0
##  7 Abso~ TV-14  Parents st~               90         2012               59
##  8 Abst~ TV-14  0                         90         2017                0
##  9 Agen~ PG     Parental g~               60         2014                0
## 10 Air ~ G      General Au~               35         2003                0
## # ... with 477 more rows, and 1 more variable: user.rating.size <int>
```

The Following Query uses a **LEFT JOIN** to demonstrate the fields from the survey table that do not have a value in the netflixratings table will be part of the resultant dataset.

```r
rs = dbSendQuery(mydb, "SELECT survey.title,
survey.‘Very satisfied‘,
survey.‘Satisfied‘,
survey.‘Neither satisfied nor dissatisfied‘,
survey.‘Dissatisfied‘,
survey.‘Very dissatisfied‘,
survey.‘did not see this one‘,
netflixratings.‘rating‘,
netflixratings.‘release year‘,
netflixratings.‘ratingDescription‘
FROM survey LEFT JOIN netflixratings ON (survey.title = netflixratings.title);")


data = fetch(rs,n=50)

df3 <- data.frame(data)
```

```
tibble(df3)
```

```
## # A tibble: 10 x 10
##    title Very.satisfied Satisfied Neither.satisfi~ Dissatisfied Very.dissatisfi~
##    <chr>          <int>     <int>            <int>        <int>            <int>
##  1 The ~             14         6                2            0                0
##  2 Brea~             10         7                1            0                0
##  3 Brid~              2         3                1            4                0
##  4 Sher~              5         7                4            0                0
##  5 Ozark              4         7                2            0                2
##  6 Hann~              4         1                1            3                0
##  7 The ~              2         1                1            0                1
##  8 Coll~              0         7                2            0                0
##  9 Comm~              3         6                1            0                1
## 10 Schi~              5         4                3            1                1
## # ... with 4 more variables: did.not.see.this.one <int>, rating <chr>,
## #   release.year <int>, ratingDescription <int>
```

The Following Query uses a **INNER JOIN** to demonstrate the fields from the survey table that do not have a value in the netflixratings table will *NOT* be part of the resultant dataset

```
rs = dbSendQuery(mydb, "SELECT survey.title,
survey.‘Very satisfied‘,
survey.‘Satisfied‘,
survey.‘Neither satisfied nor dissatisfied‘,
survey.‘Dissatisfied‘,
survey.‘Very dissatisfied‘,
survey.‘did not see this one‘,
netflixratings.‘rating‘,
netflixratings.‘release year‘,
netflixratings.‘ratingDescription‘
FROM survey INNER JOIN netflixratings ON (survey.title = netflixratings.title);")

data = fetch(rs,n=50)

df4 <- data.frame(data)

tibble(df4)
```

```
## # A tibble: 8 x 10
##   title Very.satisfied Satisfied Neither.satisfi~ Dissatisfied Very.dissatisfi~
##   <chr>          <int>     <int>            <int>        <int>            <int>
## 1 Brea~             10         7                1            0                0
## 2 Brid~              2         3                1            4                0
## 3 Hann~              4         1                1            3                0
## 4 Ozark              4         7                2            0                2
## 5 Schi~              5         4                3            1                1
## 6 Sher~              5         7                4            0                0
## 7 The ~              2         1                1            0                1
## 8 The ~             14         6                2            0                0
## # ... with 4 more variables: did.not.see.this.one <int>, rating <chr>,
## #   release.year <int>, ratingDescription <int>
```

```
rs = dbSendQuery(mydb, "SELECT * from survey")

data = fetch(rs,n=50)

df5 <- data.frame(data)

tibble(df5)
```

```
## # A tibble: 10 x 12
##    Question title did.not.see.thi~ Very.dissatisfi~ Dissatisfied
##       <int> <chr>            <int>            <int>        <int>
##  1        1 The ~               12                0            0
##  2        2 Brea~               16                0            0
##  3        3 Brid~               23                0            4
##  4        4 Sher~               17                0            0
##  5        5 Ozark               19                2            0
##  6        6 Hann~               25                0            3
##  7        7 The ~               29                1            0
##  8        8 Coll~               25                0            0
##  9        9 Comm~               22                1            0
## 10       10 Schi~               18                1            1
## # ... with 7 more variables: Neither.satisfied.nor.dissatisfied <int>,
## #   Satisfied <int>, Very.satisfied <int>, Total <int>, Total.Responses <int>,
## #   Missing.Responses <int>, Last.Update <chr>
```

Another Query converted to a dataframe showing the INNER JOIN feature of the table join

```
rs = dbSendQuery(mydb, "SELECT survey.title,
survey.‘Very satisfied‘,
survey.‘Total Responses‘,
netflixratings.‘user rating score‘
FROM survey INNER JOIN netflixratings ON (survey.title = netflixratings.title);")

data = fetch(rs,n=50)

df6 <- data.frame(data)

tibble(df6)
```

```
## # A tibble: 8 x 4
##   title                     Very.satisfied Total.Responses user.rating.score
##   <chr>                              <int>           <int>             <int>
## 1 Breaking Bad                          10              34                97
## 2 Bridgerton                             2              34                96
## 3 Hannibal                               4              34                90
## 4 Ozark                                  4              34                93
## 5 Schitt’s Creek                         5              34                74
## 6 Sherlock                               5              34                95
## 7 The Haunting of Bly Manor              2              34                88
## 8 The Queen’s Gambit                    14              34                97
```

This marks the End of this Test Project