Khan_data607_assignment_2

Mehdi Khan

September 8, 2017

The following sql query was used to create a table in SQL Server with movies data:

use data607

CREATE TABLE movies(movie_id int identity(1,1), movie_critic varchar(50), movie_name varchar(50), movie_rate int);

INSERT INTO movies VALUES (‘John’, ‘Death Note’,2), (‘John’, ‘Dunkirk’,5), (‘John’, ‘Mother!’,3), (‘John’, ‘Logan Lucky’,4), (‘John’, ‘Wonder Woman’,3), (‘John’, ‘Unlocked’,2), (‘Angela’, ‘Death Note’,1), (‘Angela’, ‘Dunkirk’,4), (‘Angela’, ‘Mother!’,3), (‘Angela’, ‘Logan Lucky’,4), (‘Angela’, ‘Wonder Woman’,5), (‘Angela’, ‘Unlocked’,1), (‘Jared’, ‘Death Note’,4), (‘Jared’, ‘Dunkirk’,5), (‘Jared’, ‘Mother!’,3), (‘Jared’, ‘Logan Lucky’,3), (‘Jared’, ‘Wonder Woman’,4), (‘Jared’, ‘Unlocked’,2), (‘Steven’, ‘Death Note’,2), (‘Steven’, ‘Dunkirk’,5), (‘Steven’, ‘Mother!’,3), (‘Steven’, ‘Logan Lucky’,3), (‘Steven’, ‘Wonder Woman’,5), (‘Steven’, ‘Unlocked’,1), (‘Becky’, ‘Death Note’,1), (‘Becky’, ‘Dunkirk’,5), (‘Becky’, ‘Mother!’,3), (‘Becky’, ‘Logan Lucky’,4), (‘Becky’, ‘Wonder Woman’,5), (‘Becky’, ‘Unlocked’,2)

Using package RODBC to connect to SQL Server:

```
library(RODBC)
```

Connecting to the data source (SQL Server) through ODBC connection:

```
## the name of the DSN ('data607') and appropriate credentials are
## provided to creare the connection

dataSrc <- odbcConnect("data607", uid = "DataUser", pwd = "Data607@Fall2017")
```

Redaing from the database:

```
## See the accessible tables, tableType = 'TABLE' is used to get
## only table objects otherwise it may list many different data
## objects

sqlTables(dataSrc, tableType = "TABLE")
```

```
##   TABLE_CAT TABLE_SCHEM          TABLE_NAME TABLE_TYPE REMARKS
## 1   data607         dbo              movies      TABLE    <NA>
## 2   data607         sys trace_xe_action_map      TABLE    <NA>
## 3   data607         sys  trace_xe_event_map      TABLE    <NA>
```

```
## get the table (in this case table called 'movies') and assign
## the data to a dataframe
moviesDS <- sqlFetch(dataSrc, "movies", stringsAsFactors = FALSE)
head(moviesDS)
```

```
##   movie_id movie_critic   movie_name movie_rate
## 1        1         John   Death Note          2
## 2        2         John      Dunkirk          5
## 3        3         John      Mother!          3
## 4        4         John  Logan Lucky          4
## 5        5         John Wonder Woman          3
## 6        6         John     Unlocked          2
```

Subsetting the dataframe:

```
## subsetting the data
subset(moviesDS, moviesDS$movie_rate > 3)
```

```
##    movie_id movie_critic   movie_name movie_rate
## 2         2         John      Dunkirk          5
## 4         4         John  Logan Lucky          4
## 8         8       Angela      Dunkirk          4
## 10       10       Angela  Logan Lucky          4
## 11       11       Angela Wonder Woman          5
## 13       13        Jared   Death Note          4
## 14       14        Jared      Dunkirk          5
## 17       17        Jared Wonder Woman          4
## 20       20       Steven      Dunkirk          5
## 23       23       Steven Wonder Woman          5
## 26       26        Becky      Dunkirk          5
## 28       28        Becky  Logan Lucky          4
## 29       29        Becky Wonder Woman          5
```

USing SQL Query to load data (instead of subsetting):

```
## It was also possible to get the required data instead of loading
## all data and subsetting from it. a sql query would do the job

requiredData <- sqlQuery(dataSrc, paste("Select * from \"movies\"",
    "Where \"movie_rate\" > 3"))
head(requiredData)
```

```
##   movie_id movie_critic   movie_name movie_rate
## 1        2         John      Dunkirk          5
## 2        4         John  Logan Lucky          4
## 3        8       Angela      Dunkirk          4
## 4       10       Angela  Logan Lucky          4
## 5       11       Angela Wonder Woman          5
## 6       13        Jared   Death Note          4
```

The connection object needs to be closed once it is no longer needed:

```
odbcClose(dataSrc)
```

ALTERNATIVE APPROACH: Reading and loading data from a CSV file that was created as an output of a sql (select * from movies) to get all the data from the 'movies' table in a sql server databse

```
## The csv file was stored in the working directory, so only the
## name of the file is enough (instead of providing the whole path)
csvMoviesDS <- read.csv("movies.csv", stringsAsFactors = FALSE)
head(csvMoviesDS)
```

```
##   ï..movie_id movie_critic   movie_name movie_rate
## 1           1         John   Death Note          2
## 2           2         John      Dunkirk          5
## 3           3         John      Mother!          3
## 4           4         John  Logan Lucky          4
## 5           5         John Wonder Woman          3
## 6           6         John     Unlocked          2
```

```
## This dataframe (csvMoviesDS) now can be used like any other
## dataframe in R
```

Some data operations:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
filter(moviesDS, movie_rate == 1)
```

```
##   movie_id movie_critic movie_name movie_rate
## 1        7       Angela Death Note          1
## 2       12       Angela   Unlocked          1
## 3       24       Steven   Unlocked          1
## 4       25        Becky Death Note          1
```

dplyr chaining; Selecting the highest rated movies and number of times they received such rating:

```
highestRated <- moviesDS %>% select(movie_name, movie_rate) %>% filter(movie_rate ==
    5) %>% count(movie_name, movie_rate)

names(highestRated)[3] = "count"
highestRated
```

```
## # A tibble: 2 x 3
##     movie_name movie_rate count
##          <chr>      <int> <int>
## 1      Dunkirk          5     4
## 2 Wonder Woman          5     3
```

Raning the movies based on their ratings:

```
## movie list by their total rating
most_loved_hated_movies <- aggregate(moviesDS$movie_rate, by = list(movie_name = moviesDS$movie_name),
    FUN = sum)


## sorting movies by rate
most_loved_hated_movies <- most_loved_hated_movies[with(most_loved_hated_movies,
    order(most_loved_hated_movies$x, decreasing = TRUE)), ]

names(most_loved_hated_movies)[2] = "overall_rating"

most_loved_hated_movies
```

```
##     movie_name overall_rating
## 2      Dunkirk             24
## 6 Wonder Woman             22
## 3  Logan Lucky             18
## 4      Mother!             15
## 1   Death Note             10
## 5     Unlocked              8
```