Assignment2 - Sql and R

DATA607 - Acquisition of Data and Management - Instructor: Andrew Catlin

Author: Peter Gatica

Date: 2/12/2021

```
# Load needed libraries
library(devtools)
library(tidyverse)
library(RCurl)
library(plyr)
library(knitr)
library(RMySQL)
```

Function call to MySQL db to connect and use the return command in a function.

```
conn.MyQL <- function(db_parms)
{
  db_conn <- dbConnect(MySQL(), user=db_user, password=db_password, dbname=db_name, host=db_host)
  return(db_conn)
}
```

Source the credentials parameter file from your local directory. Do not store in github repository since R has no encryption capability.

```
filename <- "/Users/Audiorunner13/CUNY MSDS Course Work/DATA607 Spring 2021/Week2/Assignment/Data/MySQL
db_parms <- read.csv(filename)
```

Set the login application credential to pass to the db log in function

```
db_user = db_parms$db_user
db_password = db_parms$db_password
db_name = db_parms$db_name
db_host = db_parms$db_host
db_result_set = ""

db_parms <- c(db_user, db_password, db_name, db_host, db_result_set)
```

Call the conn.MySql connect function to access the movies database

```
db_conn <- conn.MyQL(db_parms)
```

Use the dbListTables() function to list the tables in the database

```r
dbListTables(db_conn)
```

```
## [1] "customer_dim" "movie_dim"    "movie_rank"
```

Use the dbListFields() function to list the fields in a database table

```r
dbListFields(db_conn, "movie_dim")
```

```
## [1] "movie_id"     "movie_name"   "release_year" "afi_100_rank" "actor_1"
## [6] "actor_2"      "actor_3"      "etl_nr"       "etl_dt"
```

Source the movie dimension file from the movies github repository to load to the movie dimension

```r
filename <- getURL("https://raw.githubusercontent.com/audiorunner13/Masters-Coursework/main/DATA607%20S
movies_dim_df <- read.csv(text=filename)
movies_dim_df
```

```
##    movie_id                                          movie_name release_year
## 1         1                                       The Godfather         1972
## 2         2                                          Unforgiven         1992
## 3         3 The Lord of the Rings: The Fellowship of the Riings         2001
## 4         4                             Raiders of The Lost Ark         1981
## 5         5                                Shawshank Redemption         1994
## 6         6                                         Bull Durham         1988
## 7         7                                             Ben-Hur         1959
## 8         8                                           Greyhound         2020
## 9         9                                   News of the World         2020
## 10       10                                        Midnight Sky         2020
## 11       11                                            The King         2019
##    afi_100_rank           actor_1        actor_2         actor_3 etl_nr  etl_dt
## 1             2     Marlon Brando      Al Pacino    Diane Keaton     10 1/25/21
## 2            68    Clint Eastwood Morgan Freeman  Richard Harris     10 1/25/21
## 3            50   Viggo Mortenson   Elijah Woods   Ian McClellan     10 1/25/21
## 4            66     Harrison Ford    Karen Allen                     12  2/1/21
## 5            72       Tim Robbins Morgan Freeman    Clancy Brown     12  2/1/21
## 6            NA     Kevin Costner Susan Sarandon     Tim Robbins     12  2/1/21
## 7           100  Charelton Heston   Stephen Boyd   Haya Harareet     15  2/5/21
## 8            NA         Tom Hanks Stephen Graham  Elisabeth Shue     15  2/5/21
## 9            NA         Tom Hanks  Helena Zengel Mare Winningham     15  2/5/21
## 10           NA    George Clooney Felicity Jones   Kyle Chandler     15  2/5/21
## 11           NA Timothee Chalamet  Joel Edgerton Robert Pattison     17  2/6/21
```

Drop dimension and fact tables if they exist. Dropping and reloading is only recommended when table size and contents is small. Write the database tables from their respective data frames.

```r
if (dbExistsTable(db_conn, "movie_dim"))
    dbRemoveTable(db_conn, "movie_dim")
```

```
## [1] TRUE
```

```
dbWriteTable(db_conn, name = "movie_dim", value = movies_dim_df, row.names = FALSE)
```

```
## [1] TRUE
```

Source the customer dimension file from the movies github repository to load to the customer dimension

```
filename <- getURL("https://raw.githubusercontent.com/audiorunner13/Masters-Coursework/main/DATA607%20S
cust_dim_df <- read.csv(text=filename)
cust_dim_df
```

```
##   cust_id last_name first_name            address_1 address_2        city state
## 1       1    Gatica      Peter 12217 White Birch St        NA San Antonio    TX
## 2       2    Gatica     Leslie 12217 White Birch St        NA San Antonio    TX
## 3       3   Trevino    Gabriel          783 Menefee        21 San Antonio    TX
## 4       4 Rodriguez    Rebecca        300 Queretaro        NA San Antonio    TX
## 5       5     Salas    Liliana  222 Rolling View Dr        NA      Boerne    TX
## 6       6 Rodriguez     Camila   214 W French Place      2201      Austin    TX
##   zip_code etl_nr etl_dt
## 1    78245    100 2/1/21
## 2    78245    100 2/1/21
## 3    78237    110 2/3/21
## 4    78237    110 2/3/21
## 5    78006    112 2/5/21
## 6    75019    112 2/5/21
```

```
if (dbExistsTable(db_conn, "customer_dim"))
    dbRemoveTable(db_conn, "customer_dim")
```

```
## [1] TRUE
```

```
dbWriteTable(db_conn, name = "customer_dim", value = cust_dim_df, row.names = FALSE)
```

```
## [1] TRUE
```

Source the movie rank survey results file from the movies github repository.

```
filename <- getURL("https://raw.githubusercontent.com/audiorunner13/Masters-Coursework/main/DATA607%20S
movie_rank_df <- read.csv(text=filename)
movie_rank_df
```

```
##    cust_id movie_id movie_rank_nr rent_own etl_dt
## 1        1        8             4        r 2/7/21
## 2        1        9             5        r 2/7/21
## 3        1       10             4        r 2/7/21
## 4        1        5             5        o 2/7/21
## 5        1        1             5        o 2/7/21
## 6        1       11             5        o 2/7/21
## 7        2        8             3        r 2/7/21
## 8        2        9             5        r 2/7/21
## 9        2       10             4        r 2/7/21
## 10       2        5             3         o 2/7/21
## 11       2        1             2         o 2/7/21
## 12       2       11             5         o 2/7/21
## 13       3        8             3         r 2/8/21
## 14       3        9             4         r 2/8/21
## 15       3       10             5         r 2/8/21
## 16       3        5             5         r 2/8/21
## 17       3        1             3         r 2/8/21
## 18       3       11             4         r 2/8/21
## 19       4        8             3         o 2/8/21
## 20       4        9             5         r 2/8/21
## 21       4       10             4         r 2/8/21
## 22       4        5             3         o 2/8/21
## 23       4        1             2         r 2/8/21
## 24       4       11             5         o 2/8/21
## 25       5        8             3         r 2/9/21
## 26       5        9             5         r 2/9/21
## 27       5       10             4         r 2/9/21
## 28       5        5             3         r 2/9/21
## 29       5        1             2         r 2/9/21
## 30       5       11             5         r 2/9/21
## 31       6        8             4         o 2/9/21
## 32       6        9             5         o 2/9/21
## 33       6       10             4         o 2/9/21
## 34       6        5             5         o 2/9/21
## 35       6        1             5         o 2/9/21
## 36       6       11             5         o 2/9/21
```

```
if (dbExistsTable(db_conn, "movie_rank"))
    dbRemoveTable(db_conn, "movie_rank")
```

```
## [1] TRUE
```

```
dbWriteTable(db_conn, name = "movie_rank", value = movie_rank_df, row.names = FALSE)
```

```
## [1] TRUE
```

Source the sql file in the movies github repositpry. The sql will extract all survey answers and order by them first name, last name, and movie title and will replace nulls in the AFI 100 Rank field if a movie is not ranked.

```
filename <- "/Users/Audiorunner13/CUNY MSDS Course Work/DATA607 Spring 2021/Week2/Assignment/Sql/movie_
db_sql <- readChar(filename, file.info(filename)$size)
db_sql <- gsub("\n", " ",db_sql)
db_sql
```

```
## [1] "select  cd.first_name as ’First Name’,  cd.last_name as ’Last Name’,  md.movie_name as ’Movie T
```

Execute the sql query joining the fact table to the dimension tables and return all records in the result set. Specify the number of records to return by adjusting the “n =” argument.

```
db_data = dbSendQuery(db_conn, db_sql)
result_set = fetch(db_data, n = -1)
result_set
```

```
##    First Name Last Name          Movie Title Movie Rank        Lead Actor
## 1      Camila Rodriguez            Greyhound          4         Tom Hanks
## 2      Camila Rodriguez         Midnight Sky          4    George Clooney
## 3      Camila Rodriguez    News of the World          5         Tom Hanks
## 4      Camila Rodriguez Shawshank Redemption          5       Tim Robbins
## 5      Camila Rodriguez        The Godfather          5      Marlon Brando
## 6      Camila Rodriguez             The King          5 Timothee Chalamet
## 7     Gabriel   Trevino            Greyhound          3         Tom Hanks
## 8     Gabriel   Trevino         Midnight Sky          5    George Clooney
## 9     Gabriel   Trevino    News of the World          4         Tom Hanks
## 10    Gabriel   Trevino Shawshank Redemption          5       Tim Robbins
## 11    Gabriel   Trevino        The Godfather          3      Marlon Brando
## 12    Gabriel   Trevino             The King          4 Timothee Chalamet
## 13     Leslie    Gatica            Greyhound          3         Tom Hanks
## 14     Leslie    Gatica         Midnight Sky          4    George Clooney
## 15     Leslie    Gatica    News of the World          5         Tom Hanks
## 16     Leslie    Gatica Shawshank Redemption          3       Tim Robbins
## 17     Leslie    Gatica        The Godfather          2      Marlon Brando
## 18     Leslie    Gatica             The King          5 Timothee Chalamet
## 19    Liliana     Salas            Greyhound          3         Tom Hanks
## 20    Liliana     Salas         Midnight Sky          4    George Clooney
## 21    Liliana     Salas    News of the World          5         Tom Hanks
## 22    Liliana     Salas Shawshank Redemption          3       Tim Robbins
## 23    Liliana     Salas        The Godfather          2      Marlon Brando
## 24    Liliana     Salas             The King          5 Timothee Chalamet
## 25      Peter    Gatica            Greyhound          4         Tom Hanks
## 26      Peter    Gatica         Midnight Sky          4    George Clooney
## 27      Peter    Gatica    News of the World          5         Tom Hanks
## 28      Peter    Gatica Shawshank Redemption          5       Tim Robbins
## 29      Peter    Gatica        The Godfather          5      Marlon Brando
## 30      Peter    Gatica             The King          5 Timothee Chalamet
## 31    Rebecca Rodriguez            Greyhound          3         Tom Hanks
## 32    Rebecca Rodriguez         Midnight Sky          4    George Clooney
## 33    Rebecca Rodriguez    News of the World          5         Tom Hanks
## 34    Rebecca Rodriguez Shawshank Redemption          3       Tim Robbins
## 35    Rebecca Rodriguez        The Godfather          2      Marlon Brando
## 36    Rebecca Rodriguez             The King          5 Timothee Chalamet
##    AFI 100 Rank Year Released
## 1    Not Ranked          2020
## 2    Not Ranked          2020
## 3    Not Ranked          2020
## 4            72          1994
## 5             2          1972
## 6    Not Ranked          2019
## 7    Not Ranked          2020
## 8    Not Ranked          2020
## 9    Not Ranked          2020
## 10           72          1994
## 11            2          1972
## 12   Not Ranked          2019
## 13   Not Ranked          2020
## 14   Not Ranked          2020
## 15   Not Ranked          2020
## 16           72          1994
## 17            2          1972
## 18   Not Ranked          2019
## 19   Not Ranked          2020
## 20   Not Ranked          2020
## 21   Not Ranked          2020
## 22           72          1994
## 23            2          1972
## 24   Not Ranked          2019
## 25   Not Ranked          2020
## 26   Not Ranked          2020
## 27   Not Ranked          2020
## 28           72          1994
## 29            2          1972
## 30   Not Ranked          2019
## 31   Not Ranked          2020
## 32   Not Ranked          2020
## 33   Not Ranked          2020
## 34           72          1994
## 35            2          1972
## 36   Not Ranked          2019
```

The result_set containing the extracted data is a data.frame.

```
class(result_set)
```

```
## [1] "data.frame"
```