Assignment - SQL and R

Jawaid Hakim

2022-09-07

```
library(RMariaDB)
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
```

```
library(knitr)
```

Database connection

Connect to the MySql database using the **MariaDB** driver. For security reasons, connection details are stored in a **CNF** file in a directory that is accessible by R. e.g., the current working directory.

```
rmariadb.settingsfile <- 'assignment_sql_and_r.cnf'
rmarisdb.db <- 'movie_ratings'
my.dbConnect <- function() {
    db <- dbConnect(RMariaDB::MariaDB(), default.file=rmariadb.settingsfile,group=rmarisdb.db)
    db
}
```

Connect to the database:

```
db <- my.dbConnect()
```

For sanity check, list the tables:

```
dbListTables(db)
```

```
## [1] "FRIENDS" "MOVIES"  "RATINGS"
```

Database schema

The schema is is normalized into 3 tables: FRIENDS, MOVIES, and RATINGS. This design allows all 3 entities to evolve independently. For example, adding additional details to FRIENDS, date of birth and address, can be done without impacting other entities Similarly, more metadata can be added to MOVIES, e.g., name of the director and release date.

To view foreign key relationships, click ER diagram

Load the FRIENDS table:

```
qry <- 'SELECT * FROM FRIENDS ORDER BY FIRST_NAME, LAST_NAME'
rs <- dbSendQuery(db, qry)
friends <- dbFetch(rs, n=-1)
dbClearResult(rs)
head(friends)
```

```
##   FRIEND_ID FIRST_NAME LAST_NAME
## 1         2       Alex  Zakharov
## 2         1    Claudia    Schaab
## 3         4     Howard      Pein
## 4         3       Igor Vaysiberg
## 5         5     Monish     Darda
```

Load the MOVIES table:

```
qry <- 'SELECT * FROM MOVIES ORDER BY TITLE'
rs <- dbSendQuery(db, qry)
movies <- dbFetch(rs, n=-1)
dbClearResult(rs)
head(movies)
```

```
##   MOVIE_ID             TITLE
## 1       14     A QUIET PLACE
## 2       13    A STAR IS BORN
## 3       10 BOHEMIAN RHAPSODY
## 4        9              CODA
## 5       11 CRAZY RICH ASIANS
## 6        5     DON'T LOOK UP
```

Finally, load the RATINGS. Since the database schema is normalized, join FRIENDS, MOVIES, and RATINGS tables to load aggregate ratings data.

Missing Ratings: it is possible, even highly likely, that not all friends have viewed all movies. This database schema accommodates this by design - only rated movies need be loaded into RATINGS table.

However, to allow for 'NA' ratings to be loaded, the RATINGS.RATING columns is defined as an ENUM [0, 1, 2, 3, 4, 5], where 0 (default) is for unrated movies.

```
qry <- 'SELECT f.FIRST_NAME, f.LAST_NAME, m.TITLE, r.RATING
            FROM FRIENDS f, MOVIES m, RATINGS r
            WHERE r.FRIEND_ID = f.FRIEND_ID AND r.MOVIE_ID = m.MOVIE_ID
            ORDER BY f.FIRST_NAME, f.LAST_NAME, m.TITLE'
rs <- dbSendQuery(db, qry)
```

```
ratings <- dbFetch(rs, n=-1)
dbClearResult(rs)
summary(ratings)
```

```
##   FIRST_NAME         LAST_NAME            TITLE             RATING
##  Length:70          Length:70          Length:70          Length:70
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
head(ratings)
```

```
##   FIRST_NAME LAST_NAME             TITLE RATING
## 1       Alex  Zakharov     A QUIET PLACE      3
## 2       Alex  Zakharov    A STAR IS BORN      4
## 3       Alex  Zakharov BOHEMIAN RHAPSODY      4
## 4       Alex  Zakharov              CODA      3
## 5       Alex  Zakharov CRAZY RICH ASIANS      4
## 6       Alex  Zakharov     DON'T LOOK UP      3
```

Enumerations in MySQL as stored as characters. This is not the most convenient representation in R as we may want to do numeric analysis on ratings, e.g. average movie rating. Convert rating from character to integer:

```
ratings <- ratings %>% mutate(RATING = as.integer(RATING))
summary(ratings)
```

```
##   FIRST_NAME         LAST_NAME            TITLE               RATING
##  Length:70          Length:70          Length:70          Min.   :0.000
##  Class :character   Class :character   Class :character   1st Qu.:3.000
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
##                                                           Mean   :3.643
##                                                           3rd Qu.:5.000
##                                                           Max.   :5.000
```

```
head(ratings)
```

```
##   FIRST_NAME LAST_NAME             TITLE RATING
## 1       Alex  Zakharov     A QUIET PLACE      3
## 2       Alex  Zakharov    A STAR IS BORN      4
## 3       Alex  Zakharov BOHEMIAN RHAPSODY      4
## 4       Alex  Zakharov              CODA      3
## 5       Alex  Zakharov CRAZY RICH ASIANS      4
## 6       Alex  Zakharov     DON'T LOOK UP      3
```

A rating of 0 is assigned to unrated movies. Filter out data for unrated movies:

```
ratings <- ratings %>% filter(RATING != 0)
summary(ratings)
```

```
##   FIRST_NAME         LAST_NAME            TITLE               RATING
##  Length:66          Length:66          Length:66          Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:3.000
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
##                                                           Mean   :3.864
##                                                           3rd Qu.:5.000
##                                                           Max.   :5.000
```

```
head(ratings)
```

```
##   FIRST_NAME LAST_NAME             TITLE RATING
## 1       Alex  Zakharov     A QUIET PLACE      3
## 2       Alex  Zakharov    A STAR IS BORN      4
## 3       Alex  Zakharov BOHEMIAN RHAPSODY      4
## 4       Alex  Zakharov              CODA      3
## 5       Alex  Zakharov CRAZY RICH ASIANS      4
## 6       Alex  Zakharov     DON'T LOOK UP      3
```

Close (disconnect) database connection:

```
dbDisconnect(db)
```