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
```

Database connection

Connect to the MySql database using 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:

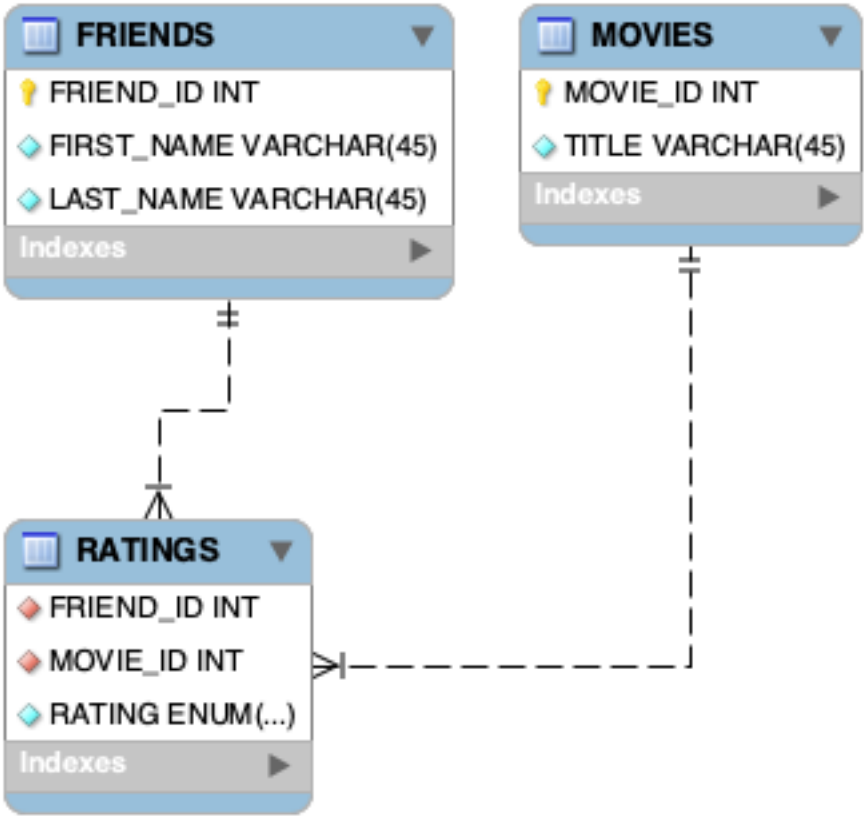
```
dbListTables(db)
```

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

Database schema

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

Furthermore, for referential integrity, **foreign keys** have been added to RATINGS table. The ER diagram for the schema is show below:



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 movie ratings. Since the database schema is normalized, join the FRIENDS, MOVIES, and RATINGS tables to load rating. Ignore 'missing' ratings - the RATINGS.RATING column is an enumeration [0, 1, 2, 3, 4, 5] where 0 (default) denotes 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:0           Length:0           Length:0           Length:0
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
head(ratings)
```

```
## [1] FIRST_NAME LAST_NAME  TITLE      RATING
## <0 rows> (or 0-length row.names)
```

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:0           Length:0           Length:0           Min.   : NA
##  Class :character   Class :character   Class :character   1st Qu.: NA
##  Mode  :character   Mode  :character   Mode  :character   Median : NA
##                                                           Mean   :NaN
##                                                           3rd Qu.: NA
##                                                           Max.   : NA
```

```
head(ratings)
```

```
## [1] FIRST_NAME LAST_NAME  TITLE      RATING
## <0 rows> (or 0-length row.names)
```

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:0           Length:0           Length:0           Min.   : NA
##  Class :character   Class :character   Class :character   1st Qu.: NA
##  Mode  :character   Mode  :character   Mode  :character   Median : NA
##                                                           Mean   :NaN
##                                                           3rd Qu.: NA
##                                                           Max.   : NA
```

```
head(ratings)
```

```
## [1] FIRST_NAME LAST_NAME  TITLE      RATING
## <0 rows> (or 0-length row.names)
```

Close (disconnect) database connection:

```
dbDisconnect(db)
```