# desafio 13

Sophia ra277230

2025-10-16

Desafio 13

```
Sys.time()
```

```
## [1] "2025-10-15 22:16:43 -03"
```

```
getwd()
```

```
## [1] "C:/Users/Shophia/Documents/me315"
```

```
#install.packages(c("RSQLite", "DBI", "readr", "dplyr", "stringr"))
```

```
library(readr)
library(dplyr)
```

```
##
## Anexando pacote: 'dplyr'

## Os seguintes objetos são mascarados por 'package:stats':
##
##     filter, lag

## Os seguintes objetos são mascarados por 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(DBI)
library(RSQLite)

# Lê os arquivos TSV compactados
basics <- read_tsv("title.basics0.tsv.gz", na = "\\N", quote = "")
```

```
## Rows: 11144942 Columns: 9

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): tconst, titleType, primaryTitle, originalTitle, genres
## dbl (4): isAdult, startYear, endYear, runtimeMinutes
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
ratings <- read_tsv("title.ratings.tsv.gz", na = "\\N", quote = "")
```

```
## Rows: 1484615 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): tconst
## dbl (2): averageRating, numVotes
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
principals <- read_tsv("title.principals0.tsv.gz", na = "\\N", quote = "")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 76695844 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): tconst, nconst, category, job, characters
## dbl (1): ordering
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

1 Crie um banco de dados SQLite utilizando os 3 arquivos acima. O banco de dados deve conter as seguintes tabelas: basics, ratings e principals

```r
# Criar conexão com banco SQLite (será criado no mesmo diretório)
con <- dbConnect(SQLite(), "movies.sqlite3")

# Salvar os dataframes como tabelas
dbWriteTable(con, "basics", basics, overwrite = TRUE)
dbWriteTable(con, "ratings", ratings, overwrite = TRUE)
dbWriteTable(con, "principals", principals, overwrite = TRUE)
```

2 (Utilizando SQL, responda): Quais são os 5 filmes com as maiores notas (averageRating)? Apresente uma solução capaz de desempatar os filmes baseando-se no número de votos recebidos.

```r
query1 <- "
SELECT b.primaryTitle, r.averageRating, r.numVotes
FROM ratings r
JOIN basics b ON r.tconst = b.tconst
WHERE b.titleType = 'movie'
ORDER BY r.averageRating DESC, r.numVotes DESC
LIMIT 5
"

top5_movies <- dbGetQuery(con, query1)
print(top5_movies)
```

```
##               primaryTitle averageRating numVotes
## 1                   Kaveri            10     1023
## 2                  Kurukku            10      451
## 3 Jedal Dar Omghe 30 Metri            10      142
## 4                Sargashte            10      134
## 5          Gorgeous Rascal            10      115
```

3 (Utilizando SQL, responda): Qual é o gênero mais frequente entre os filmes com nota maior que 8?

```
query2 <- "
SELECT b.genres
FROM ratings r
JOIN basics b ON r.tconst = b.tconst
WHERE r.averageRating > 8 AND b.titleType = 'movie'
"

genres_df <- dbGetQuery(con, query2)

# Separar e contar os gêneros
library(tidyr)
genre_counts <- genres_df %>%
  filter(!is.na(genres)) %>%
  separate_rows(genres, sep = ",") %>%
  count(genres, sort = TRUE)

# Exibir o gênero mais frequente
head(genre_counts, 1)
```

```
## # A tibble: 1 x 2
##   genres          n
##   <chr>       <int>
## 1 Documentary 10564
```

4 (Utilizando SQL, responda): Quais são os 3 atores/atrizes que mais participaram de filmes com nota maior que 7.5?

```
query3 <- "
SELECT p.nconst, COUNT(*) AS num_filmes
FROM principals p
JOIN ratings r ON p.tconst = r.tconst
JOIN basics b ON p.tconst = b.tconst
WHERE r.averageRating > 7.5
  AND b.titleType = 'movie'
  AND p.category IN ('actor', 'actress')
GROUP BY p.nconst
ORDER BY num_filmes DESC
LIMIT 3
"

top3_atores <- dbGetQuery(con, query3)
print(top3_atores)
```

```
##      nconst num_filmes
```

```
## 1 nm0004660        231
## 2 nm0595934        154
## 3 nm3183374        124
```

5

```
dbDisconnect(con)
```