Desafio 9

2025-09-30

```
library(DBI)
library(RSQLite)
library(readr)
```

```
con <- dbConnect(SQLite(), "voos.sqlite3")
```

```
airlines <- read_csv("airlines.csv")
```

```
## Rows: 14 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): IATA_CODE, AIRLINE
##
## 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.
```

```
airports <- read_csv("airports.csv")
```

```
## Rows: 322 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): IATA_CODE, AIRPORT, CITY, STATE, COUNTRY
## dbl (2): LATITUDE, LONGITUDE
##
## 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.
```

```
dbWriteTable(con, "airlines", airlines, overwrite = TRUE)
dbWriteTable(con, "airports", airports, overwrite = TRUE)
```

```
lerDados <- function(input, pos) {
  message("Leitura atingiu a linha ", pos)

  # Filtrar apenas voos dos aeroportos indicados
  aeroportos <- c("BWI", "MIA", "SEA", "SFO", "JFK")
  input_filtrado <- subset(
    input,
    ORIGIN_AIRPORT %in% aeroportos | DESTINATION_AIRPORT %in% aeroportos
  )

  # Gravar no banco (append para acumular chunks)
  dbWriteTable(con, "flights", input_filtrado, append = TRUE)
}

# ------------------------------------------------------------------------
# 4. Leitura do flights.csv em chunks
# ------------------------------------------------------------------------
```

```r
# Callback para processar cada chunk
callback <- SideEffectChunkCallback$new(lerDados)

# Definir apenas as colunas de interesse
colunas_usadas <- cols_only(
  YEAR = col_integer(),
  MONTH = col_integer(),
  DAY = col_integer(),
  AIRLINE = col_character(),
  FLIGHT_NUMBER = col_integer(),
  ORIGIN_AIRPORT = col_character(),
  DESTINATION_AIRPORT = col_character(),
  ARRIVAL_DELAY = col_double()
)

# Ler em chunks de 100 mil linhas
read_csv_chunked(
  "flights.csv",
  callback = callback,
  chunk_size = 100000,
  col_types = colunas_usadas
)
```

```
## Leitura atingiu a linha 1
## Leitura atingiu a linha 100001
## Leitura atingiu a linha 200001
## Leitura atingiu a linha 300001
## Leitura atingiu a linha 400001
## Leitura atingiu a linha 500001
## Leitura atingiu a linha 600001
## Leitura atingiu a linha 700001
## Leitura atingiu a linha 800001
## Leitura atingiu a linha 900001
## Leitura atingiu a linha 1000001
## Leitura atingiu a linha 1100001
## Leitura atingiu a linha 1200001
## Leitura atingiu a linha 1300001
## Leitura atingiu a linha 1400001
## Leitura atingiu a linha 1500001
## Leitura atingiu a linha 1600001
## Leitura atingiu a linha 1700001
## Leitura atingiu a linha 1800001
## Leitura atingiu a linha 1900001
## Leitura atingiu a linha 2000001
```

```
## Leitura atingiu a linha 2100001
## Leitura atingiu a linha 2200001
## Leitura atingiu a linha 2300001
## Leitura atingiu a linha 2400001
## Leitura atingiu a linha 2500001
## Leitura atingiu a linha 2600001
## Leitura atingiu a linha 2700001
## Leitura atingiu a linha 2800001
## Leitura atingiu a linha 2900001
## Leitura atingiu a linha 3000001
## Leitura atingiu a linha 3100001
## Leitura atingiu a linha 3200001
## Leitura atingiu a linha 3300001
## Leitura atingiu a linha 3400001
## Leitura atingiu a linha 3500001
## Leitura atingiu a linha 3600001
## Leitura atingiu a linha 3700001
## Leitura atingiu a linha 3800001
## Leitura atingiu a linha 3900001
## Leitura atingiu a linha 4000001
## Leitura atingiu a linha 4100001
## Leitura atingiu a linha 4200001
## Leitura atingiu a linha 4300001
## Leitura atingiu a linha 4400001
## Leitura atingiu a linha 4500001
## Leitura atingiu a linha 4600001
## Leitura atingiu a linha 4700001
## Leitura atingiu a linha 4800001
## Leitura atingiu a linha 4900001
## Leitura atingiu a linha 5000001
## Leitura atingiu a linha 5100001
## Leitura atingiu a linha 5200001
## Leitura atingiu a linha 5300001
## Leitura atingiu a linha 5400001
## Leitura atingiu a linha 5500001
## Leitura atingiu a linha 5600001
```

```
## Leitura atingiu a linha 5700001
```

```
## Leitura atingiu a linha 5800001
```

```
## NULL
```

```
# ------------------------------------------------------------------------
# 5. Consulta SQL: atraso médio por aeroporto de destino e companhia
# ------------------------------------------------------------------------
query <- "
SELECT
  AIRLINE,
  DESTINATION_AIRPORT,
  AVG(ARRIVAL_DELAY) AS atraso_medio
FROM flights
WHERE ARRIVAL_DELAY IS NOT NULL
GROUP BY AIRLINE, DESTINATION_AIRPORT
"

resultados <- dbGetQuery(con, query)
```

```
# Para cada companhia, pegar o maior atraso médio
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
```

```
resumo <- resultados %>%
  group_by(AIRLINE) %>%
  slice_max(order_by = atraso_medio, n = 1) %>%
  arrange(desc(atraso_medio))

print(resumo)
```

```
## # A tibble: 14 x 3
## # Groups:   AIRLINE [14]
##    AIRLINE DESTINATION_AIRPORT atraso_medio
##    <chr>   <chr>                      <dbl>
##  1 OO      DTW                        143
##  2 DL      JAX                         41
##  3 US      IAH                         36.2
##  4 UA      MSN                         34.5
##  5 NK      ATL                         25.8
##  6 F9      LGA                         24.0
##  7 AA      BNA                         22
##  8 MQ      RIC                         16.7
##  9 HA      HNL                         14.9
## 10 WN      CLT                         14.3
## 11 B6      PDX                         14.3
## 12 VX      DAL                          9.61
```

```
## 13 EV      EWR                         9.58
## 14 AS      AUS                         8.98
```

```
# 6. Data e hora da compilação
# ------------------------------------------------------------------------
cat("Arquivo compilado em:", format(Sys.time(), "%d/%m/%Y %H:%M:%S"), "\n")
```

```
## Arquivo compilado em: 30/09/2025 10:11:35
```

```
# ------------------------------------------------------------------------
# Encerrar conexão
# ------------------------------------------------------------------------
dbDisconnect(con)
```