

# Curso de Especialização em Data Science e Estatística Aplicada

## Banco de Dados - Atividade Avaliativa

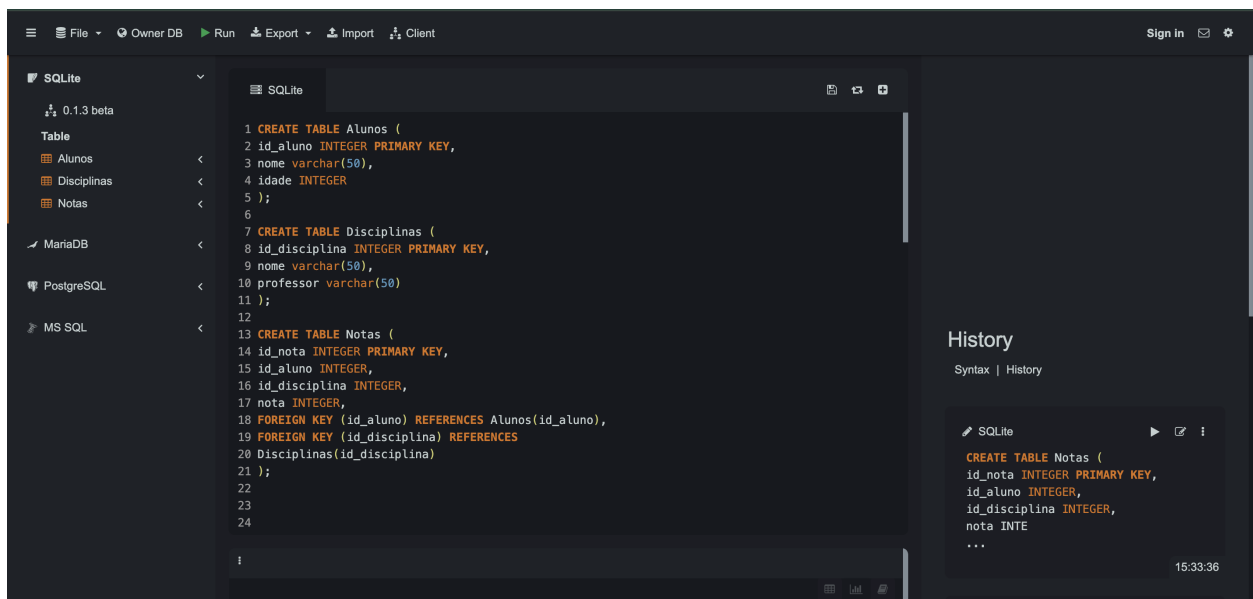
Ana M. Alves da Silva

2024-06-08

**Questão 1.** Na ferramenta SQLite IDE, crie e execute os scripts do BD disponíveis em google drive.

**Solução 1:**

- Passo 1: Criar as tabelas.

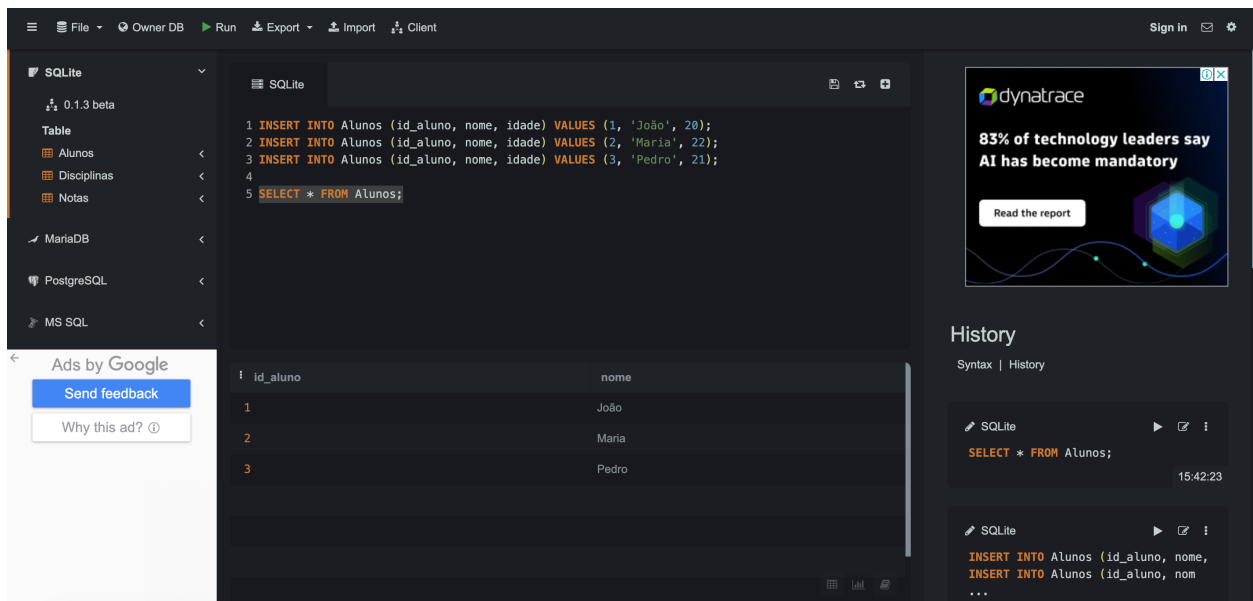


The screenshot shows the SQLite IDE interface. On the left, a sidebar lists database types: SQLite (selected), MariaDB, PostgreSQL, and MS SQL. Under SQLite, a list of tables is shown: Alunos, Disciplinas, and Notas. The main editor area displays SQL code for creating these tables. The code is as follows:

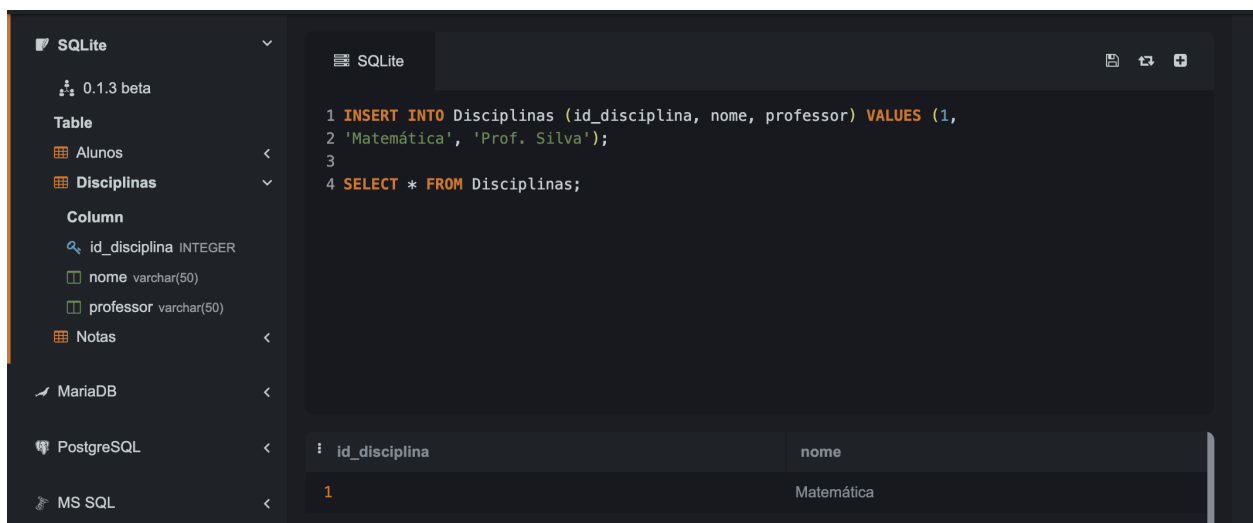
```
1 CREATE TABLE Alunos (  
2 id_aluno INTEGER PRIMARY KEY,  
3 nome varchar(50),  
4 idade INTEGER  
5 );  
6  
7 CREATE TABLE Disciplinas (  
8 id_disciplina INTEGER PRIMARY KEY,  
9 nome varchar(50),  
10 professor varchar(50)  
11 );  
12  
13 CREATE TABLE Notas (  
14 id_nota INTEGER PRIMARY KEY,  
15 id_aluno INTEGER,  
16 id_disciplina INTEGER,  
17 nota INTEGER,  
18 FOREIGN KEY (id_aluno) REFERENCES Alunos(id_aluno),  
19 FOREIGN KEY (id_disciplina) REFERENCES  
20 Disciplinas(id_disciplina)  
21 );  
22  
23  
24
```

On the right side, there is a 'History' panel showing a list of executed queries. The first query is the 'CREATE TABLE Notas' statement. The timestamp '15:33:36' is visible at the bottom right of the history panel.

- Passo 2: Inserir dados na tabela Alunos.



- Passo 3: Inserir dados na tabela Disciplina.



- Passo 4: Inserir dados na tabela Notas.

SQLite 0.1.3 beta

Table

- Alunos
- Disciplinas
- Notas

Column

- id\_nota INTEGER
- id\_aluno INTEGER
- id\_disciplina INTEGER
- nota INTEGER

```

1 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (1,1, 1, 8);
2 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (2,1, 1, 7);
3 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (3,1, 1, 9);
4
5 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (4,2, 1, 6);
6 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (5,2, 1, 8);
7 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (6,2, 1, 7);
8
9 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (7,3, 1, 9);
10 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (8,3, 1, 7);
11 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (9, 3, 1, 8);
12
13 SELECT * FROM Notas;

```

id_nota	id_aluno	id_disciplina
1	1	1
2	1	1
3	1	1
4	2	1
5	2	1
6	2	1
7	3	1

Compre o

- Passo 5: Update de dados na tabela Alunos.

SQLite

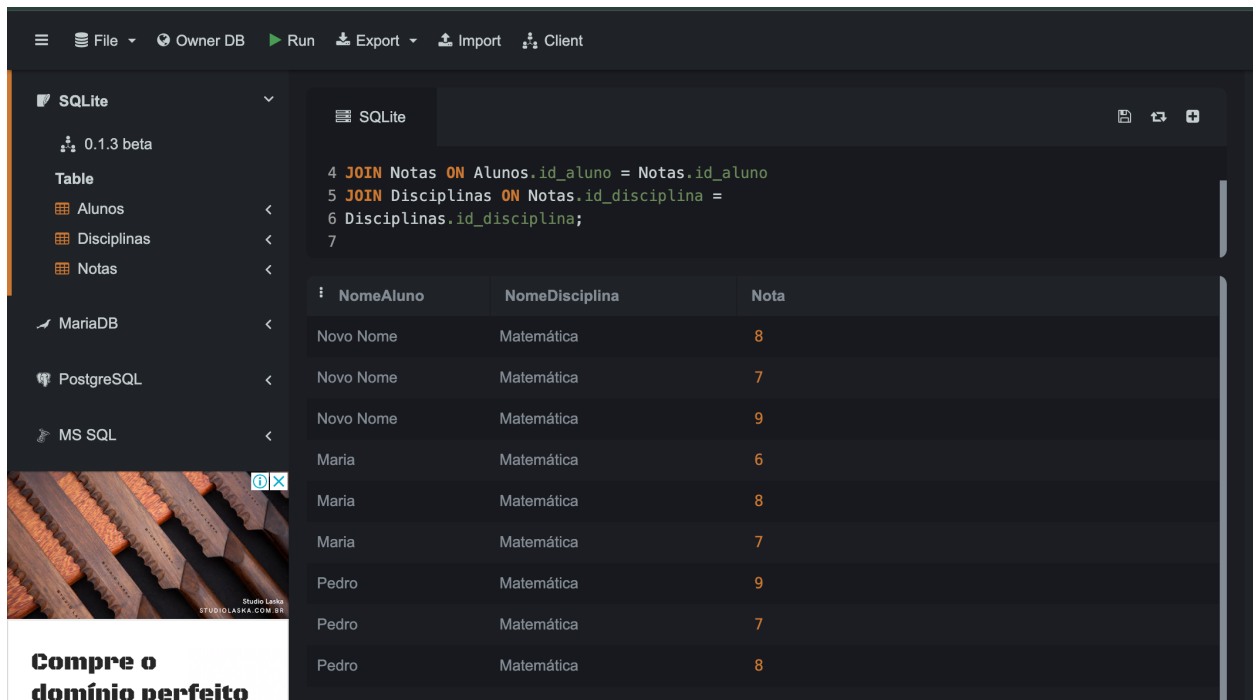
```

1 UPDATE Alunos SET nome = 'Novo Nome' WHERE id_aluno = 1;
2
3 SELECT * FROM alunos;

```

id_aluno	nome
1	Novo Nome
2	Maria
3	Pedro

- Passo 6: Visualizando as tabelas interligadas.



Questão 2. Na ferramenta SQLite IDE, insira linhas em todas as tabelas do BD criado.

Solução 2:

- Passo 1: Insert na tabela Alunos.

SQLite

```
1 INSERT INTO Alunos (id_aluno, nome, idade) VALUES (4, 'José', 25);
2 INSERT INTO Alunos (id_aluno, nome, idade) VALUES (5, 'Paula', 19);
3 INSERT INTO Alunos (id_aluno, nome, idade) VALUES (6, 'Lira', 26);
4
5 SELECT * FROM Alunos;
```

id_aluno	nome
1	Novo Nome
2	Maria
3	Pedro
4	José
5	Paula
6	Lira

- Passo 2: Insert na tabela Disciplinas.

SQLite

```
1 INSERT INTO Disciplinas (id_disciplina, nome, professor) VALUES (2,
2 'Português', 'Prof. Sousa');
3 INSERT INTO Disciplinas (id_disciplina, nome, professor) VALUES (3,
4 'Português', 'Prof. McGonagall');
5
6 SELECT * FROM Disciplinas;
```

id_disciplina	nome	professor
1	Matemática	Prof. Silva
2	Português	Prof. Sousa
3	Português	Prof. McGonagall

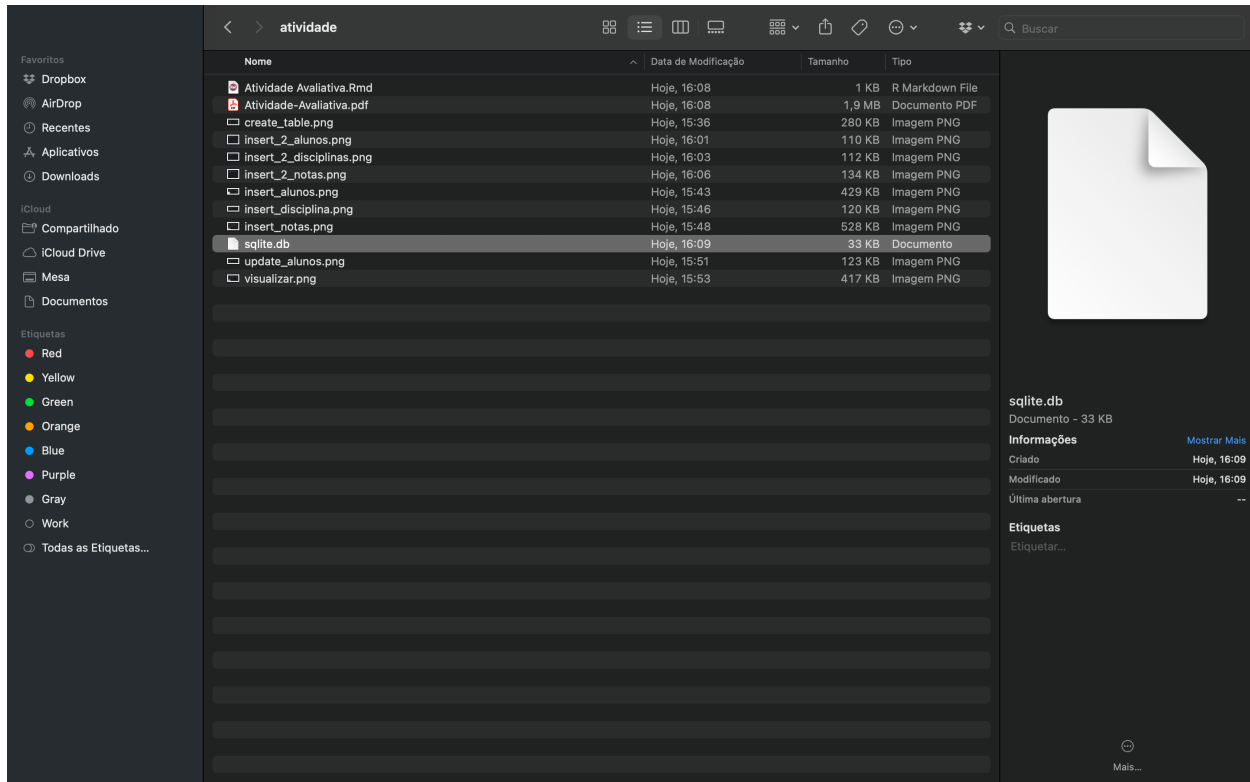
- Passo 3: Insert na tabela Notas.

```
1 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (10,3, 2, 6);
2 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (11,3, 2, 5);
3 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (12, 3, 3, 4);
4 INSERT INTO Notas (id_nota, id_aluno, id_disciplina, nota) VALUES (13, 3, 3, 8);
5
6 SELECT * FROM notas;
```

id_nota	id_aluno	id_disciplina
3	1	1
4	2	1
5	2	1
6	2	1
7	3	1
8	3	1
9	3	1
10	3	2
11	3	2
12	3	3
13	3	3

Questão 3. Na ferramenta SQLite, gere o BD na forma de um arquivo e armazene-o em uma pasta/diretório no seu computador.

Solução 3:



Questão 4. No R, faça a conexão com o BD usando o arquivo gerado no enunciado 3.

Solução 4:

```
library("RSQLite")
setwd("/Users/anamaria/especializacao/modulo_2/atividade/")
conexao <- RSQLite::dbConnect(RSQLite::SQLite(), dbname = "sqlite.db")
```

```
DBI::dbListTables(conexao)
```

```
## [1] "Alunos"      "Disciplinas" "Notas"
```

Questão 5. No R, construa três consultas SQL selecionando diretamente do BD linhas das tabelas utilizando a cláusula WHERE.

Solução 5:



```
-- Consulta na tabela Alunos
SELECT * FROM Alunos where nome ='Paula';
```

Table 1: 1 records

id_aluno	nome	idade
5	Paula	19

```
-- Consulta na tabela Disciplina
SELECT * FROM Disciplinas where nome ='Português';
```

Table 2: 2 records

id_disciplina	nome	professor
2	Português	Prof. Sousa
3	Português	Prof. McGonagall

```
-- Consulta na tabela Notas
SELECT * FROM Notas where nota > 8;
```

Table 3: 2 records

id_nota	id_aluno	id_disciplina	nota
3	1	1	9
7	3	1	9

**Questão 6.** No R, faça a importação das tabelas para data frames.

**Solução 6:**

- Dataframe Alunos:

```
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("tibble")
alunos_tbl <- dplyr::tbl(conexao,"Alunos")
alunos_df <- dplyr::collect(alunos_tbl)
```

```
alunos_df
```

```
## # A tibble: 6 x 3
##   id_aluno nome      idade
##   <int> <chr>    <int>
## 1      1 Novo Nome    20
## 2      2 Maria      22
## 3      3 Pedro      21
## 4      4 José       25
## 5      5 Paula      19
## 6      6 Lira       26
```

- Dataframe Disciplinas:

```
disciplinas_tbl <- dplyr::tbl(conexao,"Disciplinas")
disciplinas_df <- dplyr::collect(disciplinas_tbl)
```

```
disciplinas_df
```

```
## # A tibble: 3 x 3
##   id_disciplina nome      professor
##   <int> <chr>    <chr>
## 1      1 Matemática Prof. Silva
## 2      2 Português Prof. Sousa
## 3      3 Português Prof. McGonagall
```

- Dataframe Notas:

```
notas_tbl <- dplyr::tbl(conexao,"Notas")
notas_df <- dplyr::collect(notas_tbl)
notas_df
```

```
## # A tibble: 13 x 4
##   id_nota id_aluno id_disciplina nota
##   <int>    <int>      <int> <int>
## 1      1      1          1      8
## 2      2      2          1      7
## 3      3      3          1      9
## 4      4      4          2      6
## 5      5      5          2      8
## 6      6      6          2      7
## 7      7      7          3      9
## 8      8      8          3      7
## 9      9      9          3      8
## 10     10     10          3      6
## 11     11     11          3      5
## 12     12     12          3      4
## 13     13     13          3      8
```

Questão 7. No R, faça consultas utilizando `select()` do pacote `dplyr` nos objetos `tibble` correspondentes aos data frames gerados no enunciado 6.

Solução 7:

```
alunos1 <- dplyr::sql("SELECT * FROM Alunos WHERE idade > 22")
alunos_select <- dplyr::tbl(conexao, alunos1)
alunos_db_select <- dplyr::collect(alunos_select)
alunos_db_select
```

```
## # A tibble: 2 x 3
##   id_aluno nome   idade
##   <int> <chr> <int>
## 1     4 José    25
## 2     6 Lira    26
```

```
disciplinas1 <- dplyr::sql("SELECT * FROM Disciplinas WHERE nome != 'Português'")
disciplinas_select <- dplyr::tbl(conexao, disciplinas1)
disciplinas_db_select <- dplyr::collect(disciplinas_select)
disciplinas_db_select
```

```
## # A tibble: 1 x 3
##   id_disciplina nome      professor
##   <int> <chr>      <chr>
## 1     1 Matemática Prof. Silva
```

```
notas1 <- dplyr::sql("SELECT * FROM Notas WHERE nota <= 6")
notas_select <- dplyr::tbl(conexao, notas1)
notas_db_select <- dplyr::collect(notas_select)
notas_db_select
```

```
## # A tibble: 4 x 4
##   id_nota id_aluno id_disciplina  nota
##   <int>   <int>      <int> <int>
## 1     4     2         1     6
## 2    10     3         2     6
## 3    11     3         2     5
## 4    12     3         3     4
```

Questão 8. Considere o data set `Melanoma.xlsx` disponível google drive a) Faça a importação do data set para o R; b) realize operações de manipulação do data frame e do `tibble`; e c) faça consultas usando SQL.

Solução 8:

- Parte a: importação dos dados

```
library(readxl)
melanoma_data <- read_excel("/Users/anamaria/especializacao/modulo_2/atividade/Melanoma.xlsx")
head(melanoma_data)
```

```
## # A tibble: 6 x 7
##   time status sex age year thickness ulcer
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    10      3    1  76 1972     6.76    1
## 2    30      3    1  56 1968     0.65    0
## 3    35      2    1  41 1977     1.34    0
## 4    99      3    0  71 1968     2.9     0
## 5   185      1    1  52 1965    12.1     1
## 6   204      1    1  28 1971     4.84    1
```

- Parte b: operações de manipulação

```
library("dplyr")
melanoma_tibble <- as_tibble(melanoma_data)
melanoma_selected <- select(melanoma_tibble, sex, age)
melanoma_selected
```

```
## # A tibble: 205 x 2
##   sex age
##   <dbl> <dbl>
## 1     1  76
## 2     1  56
## 3     1  41
## 4     0  71
## 5     1  52
## 6     1  28
## 7     1  77
## 8     0  60
## 9     1  49
## 10    0  68
## # i 195 more rows
```

```
melanoma_filtered <- filter(melanoma_tibble, age >= 68)
melanoma_filtered
```

```
## # A tibble: 41 x 7
##   time status sex age year thickness ulcer
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    10      3    1  76 1972     6.76    1
## 2    99      3    0  71 1968     2.9     0
## 3   210      1    1  77 1972     5.16    1
## 4   279      1    0  68 1971     7.41    1
## 5   386      1    0  68 1965     3.87    1
## 6   493      3    1  72 1971    12.6     1
## 7   621      1    1  72 1972     7.06    1
## 8   629      1    1  95 1968     5.48    1
## 9   667      1    0  89 1968    13.8     1
## 10  793      1    1  68 1970     4.84    1
## # i 31 more rows
```

```
melanoma_arranged <- arrange(melanoma_tibble, age)
melanoma_arranged
```

```
## # A tibble: 205 x 7
##   time status  sex  age  year thickness ulcer
##   <dbl>  <dbl> <dbl> <dbl> <dbl>      <dbl> <dbl>
## 1  3385      2    0    4  1968      2.74    0
## 2  3776      2    1   12  1967      7.09    1
## 3   469      1    0   14  1969      2.42    1
## 4  1710      2    1   15  1973      0.58    0
## 5   858      1    0   16  1967      3.56    0
## 6  1062      1    1   19  1966      3.87    1
## 7  4479      2    0   19  1965      1.13    1
## 8  1654      2    1   20  1973      0.97    0
## 9  3441      2    0   20  1968      0.65    0
## 10 3776      2    0   21  1967      1.29    1
## # i 195 more rows
```

- Parte c: Consultas usando SQL

```
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): unable to load shared object '/Library/
##   dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11.so, 0x0006): Library not loaded: /
##   Referenced from: <9A3F5E83-2A35-33C3-9C5A-5255B116A1BE> /Library/Frameworks/R.framework/Versions/4
##   Reason: tried: '/opt/X11/lib/libSM.6.dylib' (no such file), '/System/Volumes/Preboot/Cryptexes/OS/
```

```
## tcltk DLL is linked to '/opt/X11/lib/libX11.6.dylib'
```

```
## Could not load tcltk. Will use slower R code instead.
```

```
result_sql <- sqldf("SELECT * FROM melanoma_tibble WHERE age > 85 AND status = 3")
head(result_sql)
```

```
##   time status sex age year thickness ulcer
## 1  826      3  0  86 1965      8.54    1
```