SQLite

Dados Relacionais e SQL

Benilton Carvalho, Guilherme Ludwig, Tatiana Benaglia

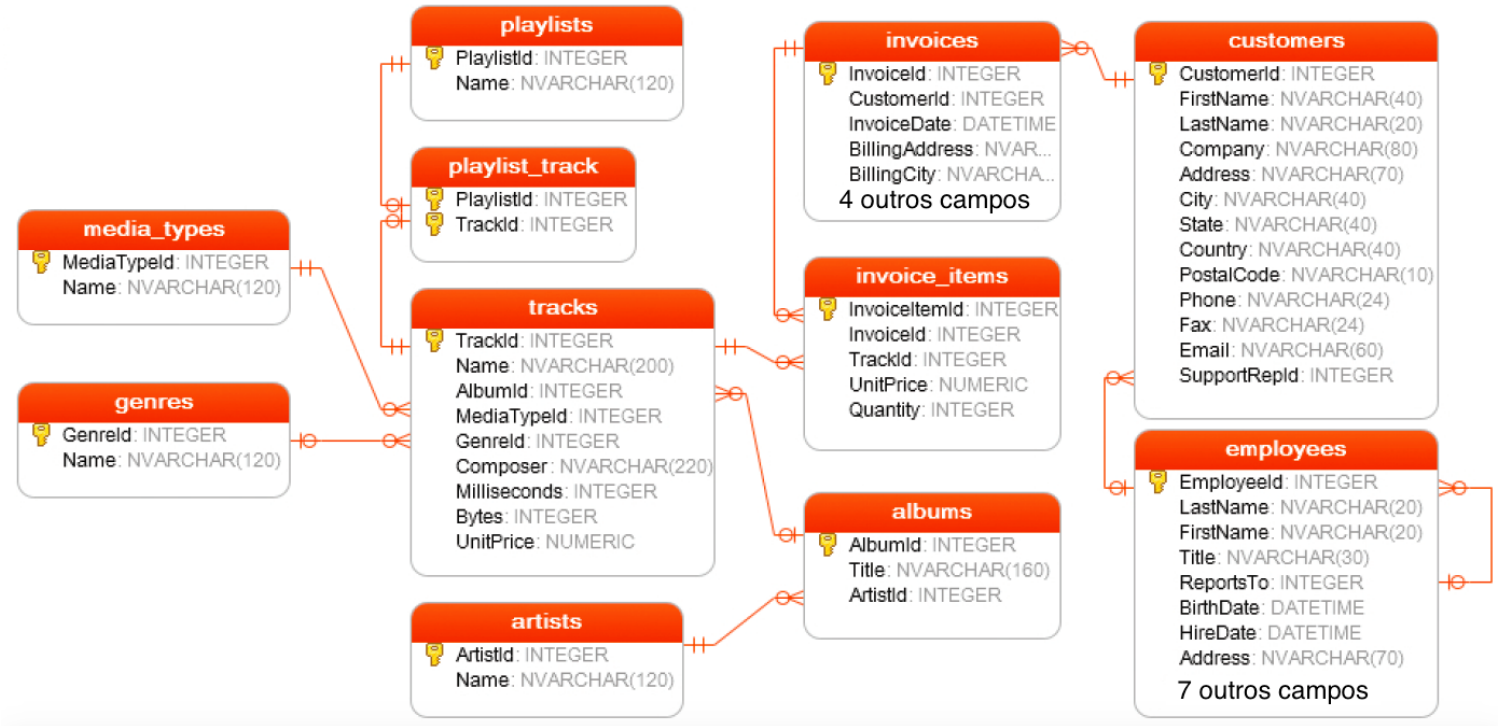
O que é SQLite?

- Biblioteca embutida em um processo que implementa um mecanismo de banco de dados SQL:
  - autônomo;
  - sem servidor;
  - com zero de configuração;
- Código de domínio-público com uso gratuito para qualquer uso:
  - comercial;
  - privado;
- Lê e escreve diretamente de/para arquivos comuns em disco;
- Sistema SQL completo com múltiplas tabelas, índices, gatilhos e visões em um único arquivo em disco;
- O arquivo produzido pelo SQLite funciona em qualquer plataforma;
- Aproximadamente 600KB;
- LTS.

SQLite é o banco de dados mais implantado no mundo

- Photoshop Lightroom;
- Software de vôo para a família Airbus A350 XWB;
- Apple iTunes;
- Sistemas multimídias da GM, Nissan e Suzuki geridos pela Bosch;
- Dropbox;
- Android;
- Preferível pela Biblioteca do Congresso;
- Sistema de armazenamento:
  - Windows 10;
  - Firefox e Thunderbird;
  - Skype;
- Sistema de rastreamento de versões do RPM/RedHat;

Banco de Dados de Exemplo: `disco.db`



Instalando SQLite

- O banco de dados apresentado pode ser acessado diretamente via SQLite;
- Visite o site www.sqlite.org;
- Existem múltiplas versões para download:
  - Código-fonte;
  - Linux;
  - Mac OS X;
  - Windows (32/64 bits);
  - Windows Phone;
  - Android;
- Essencialmente, linha de comando (meu predileto...);
- Existem interfaces gráficas para "facilitar"...

Usando o R como Interface para SQLite

- O R possui um pacote chamado `RSQLite`;
- O `RSQLite` oferece *todos* os recursos do `SQLite` diretamente dentro do R;
- Apesar de, neste caso, estarmos dentro do R, os comandos são passados diretamente para o motor do SQLite, que é instalado automagicamente no momento da instalação do pacote.

```
install.packages("RSQLite")
```

```
library(RSQLite)
db = dbConnect(SQLite(), '../dados/disco.db')
db
```

```
## <SQLiteConnection>
##   Path: /Volumes/Disk2/github/ME315/me315-unicamp.github.io/dados/disco.db
##   Extensions: TRUE
```

Manipulação Básica

- Quais são as tabelas existentes no banco de dados?
- Quais são as colunas na tabela `albums`?

```
dbListTables(db)
```

```
##  [1] "albums"          "artists"         "customers"       "employees"
##  [5] "genres"          "invoice_items"   "invoices"        "media_types"
##  [9] "playlist_track"  "playlists"       "sqlite_sequence" "sqlite_stat1"
## [13] "tracks"
```

```
dbListFields(db, 'albums')
```

```
## [1] "AlbumId"  "Title"    "ArtistId"
```

Manipulação Básica: Selecionando registros

- O comando `dbGetQuery` consulta o banco de dados, extrai os resultados solicitado e **os retorna** ao R;
- A sintaxe é `dbGetQuery(<conexao>, <chamada SQL>)`;
- Extraia da tabela `albums` todas as colunas e todas as linhas e armazene-os em um objeto chamado `album_db`:

```
album_db = dbGetQuery(db, 'SELECT * FROM albums')
head(album_db)
```

```
##   AlbumId                                 Title ArtistId
## 1       1 For Those About To Rock We Salute You        1
## 2       2                      Balls to the Wall        2
## 3       3                      Restless and Wild        2
## 4       4                      Let There Be Rock        1
## 5       5                                Big Ones        3
## 6       6                    Jagged Little Pill        4
```

```
dim(album_db)
```

```
## [1] 347   3
```

O Comando `SELECT`

- O comando `SELECT` é o comando mais utilizado em SQL;
- É, também, um dos comandos mais complexos;
- Pode ser combinado com uma série de argumentos:
  - `ORDER BY`: ordernar o resultado;
  - `DISTINCT`: pesquisar por linhas únicas;
  - `WHERE`: filtrar linhas;
  - `LIMIT`: restringir o número de linhas do resultado;
  - `INNER JOIN`/`LEFT JOIN`: consultar múltiplas tabelas;
  - `GROUP BY`: agrupar e aplicar funções para agregação nos grupos;
  - `HAVING`: filtrar em grupos;

Seleção de colunas específicas e ordenação de resultados

Selecione as colunas `trackid`, `name`, `composer` e `unitprice`. Ordene o objeto resultante por `unitprice`.

```
sql = 'SELECT trackid, name FROM tracks ORDER BY name'
res = dbGetQuery(db, sql)
head(res)
```

```
##   TrackId                                                          Name
## 1    3027                                                         "40"
## 2    2918                                                          "?"
## 3    3412 "Eine Kleine Nachtmusik" Serenade In G, K. 525: I. Allegro
## 4     109                                                       #1 Zero
## 5    3254                                                     #9 Dream
## 6     602                                                 'Round Midnight
```

Seleção de Registros Diferentes

Quais são as cidades de todos os clientes, ordenadas por nome de cidade?

```
sql = 'SELECT city FROM customers ORDER BY city'
ex3a = dbGetQuery(db, sql)
head(ex3a)
```

```
##        City
## 1 Amsterdam
## 2 Bangalore
## 3    Berlin
## 4    Berlin
## 5  Bordeaux
## 6    Boston
```

```
dim(ex3a)
```

```
## [1] 59  1
```

Seleção de Registros Diferentes

Quais são as cidades de todos os clientes, ordenadas por nome de cidade?

```
sql = 'SELECT DISTINCT city FROM customers ORDER BY city'
ex3b = dbGetQuery(db, sql)
head(ex3b)
```

```
##        City
## 1 Amsterdam
## 2 Bangalore
## 3    Berlin
## 4  Bordeaux
## 5    Boston
## 6  Brasília
```

```
dim(ex3b)
```

```
## [1] 53  1
```

Seleção com Condições

Quais são todas as músicas do álbum 1?

```
dbGetQuery(db, 'SELECT name, albumid FROM tracks WHERE albumid=1')
```

```
##                                          Name AlbumId
## 1  For Those About To Rock (We Salute You)       1
## 2                      Put The Finger On You       1
## 3                            Let's Get It Up       1
## 4                           Inject The Venom       1
## 5                                 Snowballed       1
## 6                                 Evil Walks       1
## 7                                     C.O.D.       1
## 8                          Breaking The Rules       1
## 9                  Night Of The Long Knives       1
## 10                                Spellbound       1
```

Seleção com Condições Complexas

Limitando-se a 5 registros, após ordenação por nome, quais são os nomes, ID de álbum e ID de mídia de músicas com mídias de tipo 1 ou 2?

```
sql = paste('SELECT name, albumid, mediatypeid FROM tracks',
            'WHERE mediatypeid IN (1, 2)',
            'ORDER BY name LIMIT 5')
dbGetQuery(db, sql)
```

```
##                                                                 Name AlbumId
## 1                                                               "40"     239
## 2 "Eine Kleine Nachtmusik" Serenade In G, K. 525: I. Allegro     281
## 3                                                            #1 Zero      11
## 4                                                          #9 Dream     255
## 5                                                 'Round Midnight      48
##   MediaTypeId
## 1           1
## 2           2
## 3           1
## 4           2
## 5           1
```

Seleção de Seleção

Quais são as músicas e identificadores de faixa e álbum produzidas pelo artista que identificador 12?

```
sql = paste('SELECT trackid, name, albumid FROM tracks',
            'WHERE albumid IN',
            '(SELECT albumid FROM albums WHERE artistid==12)',
            'LIMIT 5')
dbGetQuery(db, sql)
```

```
##   TrackId                      Name AlbumId
## 1     149             Black Sabbath      16
## 2     150                The Wizard      16
## 3     151 Behind The Wall Of Sleep      16
## 4     152                    N.I.B.      16
## 5     153                Evil Woman      16
```

Seleções Complexas

Quais são as faixas cujos nomes começam com qualquer caracter seguido de 'ere' e terminam com qualquer expressão?

```
sql = "SELECT trackid, name FROM tracks WHERE name GLOB '?ere*'"
dbGetQuery(db, sql)[1:5,]
```

```
##   TrackId                    Name
## 1     324                  Pererê
## 2    1132                Serenity
## 3    1452 Were Do We Go From Here
## 4    1740                  Sereia
## 5    2198                  Jeremy
```

Seleções Complexas

Quais são as faixas cujos nomes possuem algum dígito?

```
sql = "SELECT trackid, name FROM tracks WHERE name GLOB '*[0-9]*'"
dbGetQuery(db, sql)[1:5,]
```

```
##   TrackId                        Name
## 1     109                    #1 Zero
## 2     122              20 Flight Rock
## 3     132          13 Years Of Grief
## 4     343 Communication Breakdown(2)
## 5     347 Communication Breakdown(3)
```

Agregação por Grupos de Variáveis

Quantas faixas por disco?

```
sql = 'SELECT albumid, COUNT(trackid) FROM tracks GROUP BY albumid'
dbGetQuery(db, sql)[1:5,]
```

```
##   AlbumId COUNT(trackid)
## 1       1              10
## 2       2               1
## 3       3               3
## 4       4               8
## 5       5              15
```

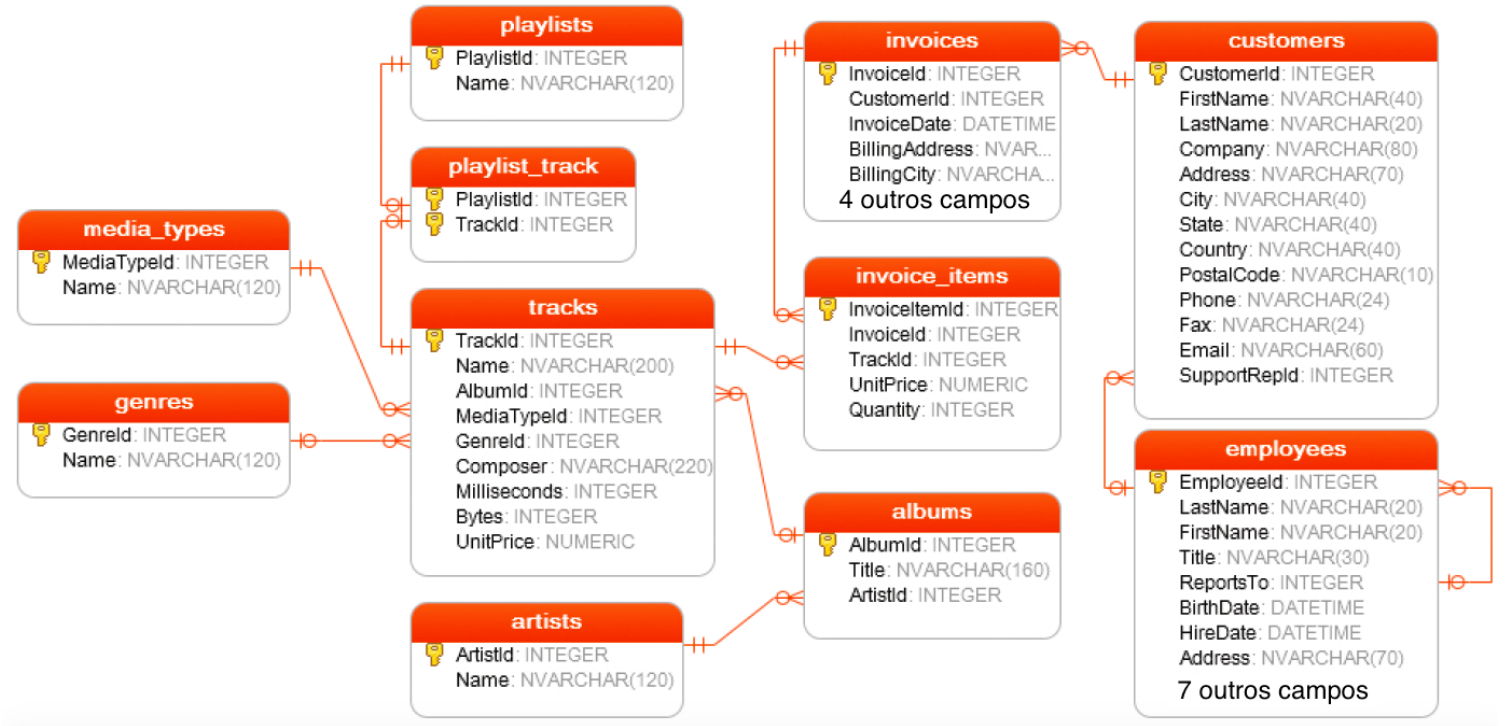
Agregação por Grupos de Variáveis com Filtro

Quantas faixas por disco para o album 1?

```
sql = paste('SELECT albumid, COUNT(trackid)',
            'FROM tracks GROUP BY albumid',
            'HAVING albumid=1')
dbGetQuery(db, sql)
```

```
##   AlbumId COUNT(trackid)
## 1       1             10
```

Banco de Dados de Exemplo: `disco.db`



INNER JOIN

Quais são os nomes de cada faixa com os respectivos títulos dos álbums?

```
sql = paste('SELECT trackid, name, title FROM tracks',
            'INNER JOIN albums ON albums.albumid=tracks.albumid')
dbGetQuery(db, sql)[1:5,]
```

```
##   TrackId                                        Name
## 1       1 For Those About To Rock (We Salute You)
## 2       6                    Put The Finger On You
## 3       7                          Let's Get It Up
## 4       8                         Inject The Venom
## 5       9                               Snowballed
##                                Title
## 1 For Those About To Rock We Salute You
## 2 For Those About To Rock We Salute You
## 3 For Those About To Rock We Salute You
## 4 For Those About To Rock We Salute You
## 5 For Those About To Rock We Salute You
```

A sintaxe é análoga para `LEFT JOIN`

Funções de Agregação

- `AVG`: `AVG([ALL | DISTINCT] expressao)` calcula a média de todos os valores não-nulos ou dos valores distintos;
- `COUNT`: `COUNT([ALL | DISTINCT] expressao)` realiza a contagem de todos registros;
- `MAX`, `MIN`, `SUM` funcionam de maneira análoga às funções anteriores;