Laboratório 07 - SQLite - Parte II

Desafio 08

Rafael Gomes Carneiro - RA185462

SQLite

O **SQLite** é uma biblioteca embutida em um processo único que é autônomo, não necessita de servidor e exige zero de configuração. O código é de domínio público e seu uso é gratuito em qualquer tipo de utilização. Trata-se de um sistema SQL completo, capaz de utilizar múltiplas tabelas, índices, gatilhos e visões. Ele funciona, virtualmente, em qualquer plataforme (incluindo móveis) e pesa cerca de 600KB.

Objetivos

Ao fim deste laboratório, você deverá ser capaz de:

- Conectar-se a um banco de dados do tipo SQLite utilizando o R como interface;
- Explorar quais são as tabelas disponíveis no referido banco de dados;
- Identificar quais são as colunas de uma dada tabela existente no banco de dados;
- Realizar pesquisas simples;
- Extrair registros do banco de dados e armazená-las em objetos do R;
- Realizar algumas pesquisas complexas, utilizando `WHERE`, `INNER JOIN`.
- Criar novas tabelas no banco de dados.

Fonte do problema:

Estes dados são de uma iniciativa de publicidade de desempenho de alunos e professores nas Universidades Americanas. A UW-Madison disponibilizou os seus dados online (no Office of the Registrar: https://registrar.wisc.edu/), e alguém trouxe os dados para uma competição no Kaggle (https://www.kaggle.com/Madgrades/uw-madison-courses). Nesta atividades, examinaremos disciplinas oferecidas no assunto de Estatística pela referida Universidade. O esquema do banco de dados é apresentado na figura abaixo.

Observações

- Esta tarefa não deve ser realizada no site jupyter.ime.unicamp.br ;
- Você deve realizar o download dos dados para o computador que estiver utilizando e, então, iniciar a atividade;
- Recomenda-se a utilização dos seguintes pacotes:
  - `RSQLite`
- Toda a atividade deve ser realizada utilizando-se apenas de SQLite (i.e., não utilize `dbplyr`);

Atividade

1. Baixe o arquivo `uwmadison.sqlite3`. Conecte-se a ele usando o pacote `RSQLite`, armazenando a conexão em uma variável `conn`.
2. Quem são e quantos são os professores que lecionaram disciplinas cujo tópico era estatística (`subjects.abbreviation='STAT'`);
3. O GPA americano é definido numa escala de 0 a 4, em que A = 4, AB = 3.5, B = 3, BC = 2.5, C = 2, D = 1 e F = 0. Determinando a nota média de cada oferecimento pela ponderação da quantidade de alunos em cada extrato com os valores numéricos de cada conceito, indique (no que se referente a disciplinas no assunto de estatística):

- Quem é o professor mais difícil?
- Quem é o professor mais fácil?
- Qual é a disciplina mais difícil?
- Qual é a disciplina mais fácil?

4. Desconecte do banco de dados.

```r
library(DBI)
library(RSQLite)
library(dplyr)
```

```r
# caminho local do arquivo .sqlite3
conn <- dbConnect(SQLite(), "../dados/database.sqlite3")

# ver tabelas disponíveis
dbListTables(conn)
```

```
 [1] "course_offerings"    "courses"             "grade_distributions"
 [4] "instructors"         "rooms"               "schedules"
 [7] "sections"            "subject_memberships" "subjects"
[10] "teachings"
```

```r
query <- "
SELECT i.name AS professor
FROM instructors i
JOIN teachings t ON i.id = t.instructor_id
JOIN sections s ON t.section_uuid = s.uuid
JOIN course_offerings co ON s.course_offering_uuid = co.uuid
JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
JOIN subjects sub ON sm.subject_code = sub.code
WHERE sub.abbreviation = 'STAT'
GROUP BY i.id, i.name
ORDER BY i.name
"

# Executar a consulta
professores_stat <- dbGetQuery(conn, query)

# Mostrar os nomes
professores_stat
```

```
                     professor
1                ABIGAIL BENZINE
2            ADIN-CRISTIAN ANDREI
3                  AKICHIKA OZEKI
4                      ALAN HUANG
5                ALBERTO DEL PIA
6                 ALBRECHT KLEMM
```

7 ALEXANDER COVINGTON
8 ALEXANDER FISH
9 ALEXANDER KISELEV
10 ALYSSA DIGILIO
11 ANATOLE BECK
12 ANDREJ ZLATOS
13 ANDREW LESLIE
14 ANNE BRUCKNER
15 ANQI SHI
16 ANRU ZHANG
17 ARNOLD MILLER
18 AUGUST JENSEN
19 BEHZAD AALIPUR
20 BEHZAD AALIPUR HAFSHEJANI
21 BEN ADAM HAALAND
22 BENEDEK VALKO
23 BENJAMIN RECHT
24 BI CHENG WU
25 BIN DAI
26 BIN ZHANG
27 BIN ZHU
28 BINGYING XIE
29 BO HUANG
30 BO YANG
31 BOWEN HU
32 BRET HANLON
33 BRET LARGET
34 BRIAN YANDELL
35 BROOK LUERS
36 CECILE ANE
37 CHAN PARK
38 CHANHAN HSU
39 CHAOQUN MEI
40 CHAOYANG YU
41 CHELSEY GREEN
42 CHEN CHENG
43 CHEN JING
44 CHENGNING ZHANG
45 CHENLIANG XU
46 CHENSHENG KUANG
47 CHENXI LI
48 CHIA-CHIEH LIN
49 CHIEN-WEI CHEN

50 CHRISTINA M. KENDZIORSKI
51 CHRISTINE SORKNESS
52 CHRISTOPHER WAGNER
53 CLAIRE BOBST
54 CLAUDIA SOLIS LEMUS
55 COLE COOK
56 COLIN LONGHURST
57 CRYSTAL CHEN
58 CUICUI QI
59 CUIZE HAN
60 DANIELE CAPPELLETTI
61 DAVID ANDERSON
62 DAVID DEMETS
63 DAVID GRIFFEATH
64 DEBRAJ DAS
65 DEREK BEAN
66 DEREK NORTON
67 DEYUAN JIANG
68 DIETRICH UHLENBROCK
69 DONALD PORTER
70 DONG XIA
71 DONGGYU KIM
72 DONGHYUN LEE
73 DOUGLAS M. BATES
74 DUY NGUYEN
75 DUZHE WANG
76 EDWARD ERKER
77 ELOISA D CHAVAS
78 ERICA LEE DEADMAN
79 ERIK NORDHEIM
80 FAN GAO
81 FAN YANG
82 FANG FANG
83 FANGFANG WANG
84 FLORIAN BERTRAND
85 FREDERICK BOEHM
86 GARVESH RASKUTTI
87 GARY HOWARD SCHROEDER
88 GENG LI
89 GINA BENNINGER
90 GINA OH
91 GONZALO CONTADOR
92 GRACE WAHBA

93 GREGORIO MORENO-FLORES
94 GREGORY SHINAULT
95 GUANHUA CHEN
96 GUANNAN SUN
97 GUILHERME ROSA
98 GUILHERME VIEIRA NUNES LUDWIG
99 GUN WOONG PARK
100 HAN CHEN
101 HAO CHEN
102 HAO TENG
103 HAO ZHENG
104 HAO ZHOU
105 HAODA FU
106 HAOYANG FAN
107 HEATHER MARIE BRAZEAU
108 HUAIBAO FENG
109 HUI WANG
110 HUIKUN ZHANG
111 HUILIN HU
112 HYEBIN SONG
113 HYUNSEUNG KANG
114 IAN BRANSTAD RILEY
115 ISMOR FISCHER
116 JAMES ANDERSON
117 JAMES D KUELBS
118 JARED BROWN
119 JASON P FINE
120 JASON RICHARD SWANSON
121 JEE YEON KIM
122 JEEA CHOI
123 JENNIFER BIRSTLER
124 JENNIFER NGUYEN
125 JESSE THOMAS HOLZER
126 JIAJIE CHEN
127 JIALE XU
128 JIAN WU
129 JIANCHANG HU
130 JIE SONG
131 JIE WEI
132 JIE ZHANG
133 JILI WANG
134 JINGCI MENG
135 JINGJIANG PENG

136 JINGLAN LI
137 JIWEI ZHAO
138 JOHN DAVIS
139 JOHN GILLETT
140 JOHN KANE
141 JOHN WILTSHIRE-GORDON
142 JONATHON PETERSON
143 JOSEP GINEBRA
144 JOSEPH DEUTSCH
145 JU HEE CHO
146 JUN LI
147 JUN SHAO
148 JUN YIN
149 JUN ZHANG
150 JUN ZHU
151 JUNGWON MUN
152 JUNHEE HAN
153 JUNHO LEE
154 KAILEI CHEN
155 KAM-WAH TSUI
156 KARL BROMAN
157 KARL ROHE
158 KATHERINE GOODE
159 KAZUHIKO SHINKI
160 KEEGAN KORTHAUER
161 KEN ONO
162 KEVIN HASEGAWA ENG
163 KEVIN PACKARD
164 KJELL DOKSUM
165 KRISHNAKUMAR BALASUBRAMANIAN
166 KRISTEN CYFFKA
167 KUNLING HUANG
168 KYLE HEBERT
169 KYLE HERBET
170 KYUNGMANN KIM
171 LAM HO
172 LAN LUO
173 LANCINE KONATE
174 LEI XU
175 LEV BORISOV
176 LIAM JOHNSTON
177 LIE XIONG
178 LILI LAN

179 LILI ZHENG
180 LILUN DU
181 LIN QI
182 LONG PHAN
183 LU MAO
184 LU YANG
185 LUWAN ZHANG
186 LUXI CAO
187 MANJUSHA KANCHARLA
188 MARI PALTA
189 MARIA KAMENETSKY
190 MARIA ROJO
191 MARIAN R FISHER
192 MARY LINDSTROM
193 MATTHEW BALLARD
194 MENG SONG
195 MICHAEL FERRIS
196 MICHAEL GEORGE ILTIS
197 MICHAEL HOGAN
198 MICHAEL KUTZLER
199 MICHAEL LIOU
200 MICHAEL NEWTON
201 MICHAEL RENE KOSOROK
202 MICHELLE HARRIS
203 MIHAELA IFRIM
204 MIN JUNG LEE
205 MIN NIU
206 MING XIE
207 MING YUAN
208 MINJING TAO
209 MITCHELL PAUKNER
210 MOO K CHUNG
211 MUHONG GAO
212 MURRAY CLAYTON
213 NATALIA DE LEON GATTI
214 NELLIE LAUGHLIN
215 NICHOLAS HENDERSON
216 NICHOLAS STEPHEN KEULER
217 NING FAN
218 NORBERT BINKIEWICZ
219 NORMAN DRAPER
220 PAUL M TERWILLIGER
221 PAUL RATHOUZ

PAUL SAVARIAPPAN
PEIGEN ZHOU
PERLA REYES
PHILIP WOOD
QI JIANG
QI TANG
QIAN ZHIGUANG
QING LI
QIONG ZHANG
QIURONG CUI
QUEFENG LI
QUOC TRAN
REBECCA KOSCIK
REBECCA POST
RICHARD A BRUALDI
RICHARD A. JOHNSON
RICHARD J. CHAPPELL
ROBERT HARRON
ROBERT R MEYER
ROBERT WARDROP
ROBERT WAYNE GREEN
RONALD GANGNON
RONGJUN ZHU
RUI CHEN
RUI TANG
RUIFANG SONG
RUIFENG XU
RUNGANG HAN
RUOSI GUO
RYAN ZEA
SAMUEL STECHMANN
SANGBUM CHOI
SCOTT HOTTOVY
SCOTT JOSEPH HETZEL
SEAN KENT
SEBASTIEN ROCH
SEHO PARK
SEULKEE YUN
SEUNGBONG HAN
SHAN LU
SHANE HUBLER
SHANG WU
SHENG WANG

265 SHENG WANG
266 SHENG ZHANG
267 SHENGJI JIA
268 SHI JIN
269 SHIXUE LIU
270 SHIZHEN WANG
271 SHUAI CHEN
272 SHUANG HUANG
273 SHULEI WANG
274 SHUYUN YE
275 SIJIAN WANG
276 SIJING LI
277 SOHEIL SADEGHI
278 SOKOL VAKO
279 SONA Z SWANSON
280 SONG WANG
281 SRIKANTHMADHAVAN ARAVAMUTHAN
282 STEFFEN LEMPP
283 STEPHEN AARON STANHOPE
284 STEPHEN BERG
285 STEPHEN WAINGER
286 STEPHEN WRIGHT
287 STEVEN SAM
288 SUKHENDU MEHROTRA
289 SUNDUZ KELES
290 TAERI UHM
291 TAO YU
292 THEVAASIINEN CHANDERENG
293 THOMAS COOK
294 THOMAS G. KURTZ
295 THU LE
296 TIEN VO
297 TIMO SEPPALAINEN
298 TIMOTHY IDOWU
299 TIMOTHY IDOWU
300 TING YE
301 TING-LI LIN
302 TONG LI
303 TONGHAI YANG
304 TRAM TA
305 TUN LEE NG
306 TZU HSIANG HUNG
307 VARSHA KULKARNI

308 VICTOR LUO
309 VICTORIA MANSFIELD
310 WAI TONG FAN
311 WEI ZHENG
312 WEI-YIN LOH
313 WENWEN ZHANG
314 WENZHI CAO
315 WESLEY CHANG
316 XIAO GUO
317 XIAO NIE
318 XIAODAN WEI
319 XIAOMAO LI
320 XIAOPING FENG
321 XIAOWU DAI
322 XIN LI
323 XIN ZHANG
324 XINJIE HE
325 XINWEI DENG
326 XINXIN YU
327 XINYU SONG
328 XIPEI YANG
329 XIRAN WANG
330 XIUFENG SHAO
331 XIUYU MA
332 XIWEN MA
333 XU HE
334 XU XU
335 XUEYAO CHEN
336 XUN ZHAO
337 YAJUAN SI
338 YALI WANG
339 YAN CHEN
340 YANBING ZHENG
341 YANG ZHAO
342 YANNAN QIU
343 YAOGUO XIE
344 YAOYAO XU
345 YAZHEN WANG
346 YI CHAI
347 YI LI
348 YI LIU
349 YIFAN MEI
350 YILIN ZHANG

```
351                   YING ZHANG
352                  YONGFENG WU
353                 YONGJOON KIM
354                   YONGSU LEE
355                    YOUNG LEE
356               YOUNG MIN PARK
357              YOUNGDEOK HWANG
358                    YOURAN QI
359                  YU MENGGANG
360                   YU QIU LIU
361                   YUAN JIANG
362                      YUAN LI
363                    YUAN WANG
364                   YUANZHI LI
365                   YUCHANG WU
366                  YUCHEN ZHOU
367                  YUJIN CHUNG
368                   YUNONG LIN
369                    YUQING XU
370               ZHANG CHUNMING
371               ZHENGJUN ZHANG
372                 ZHENGXIAO WU
373                 ZHIGENG GENG
374                  ZHIGUO XIAO
375                  ZHONGJIE YU
376                    ZHUANG WU
377                  ZIFENG ZHAO
378                    ZIJIAN NI
379                 ZUOFENG SHANG
```

```r
# Contar quantos professores distintos
n_professores <- nrow(professores_stat)
n_professores
```

```
[1] 379
```

```r
query_gpa_prof <- "
SELECT
    i.name AS professor,
    SUM(
        (a_count * 4.0) +
        (ab_count * 3.5) +
```

```
        (b_count * 3.0) +
        (bc_count * 2.5) +
        (c_count * 2.0) +
        (d_count * 1.0) +
        (f_count * 0.0)
    ) * 1.0 /
    SUM(
        a_count + ab_count + b_count + bc_count + c_count + d_count + f_count
    ) AS gpa_medio
FROM grade_distributions g
JOIN sections s ON g.section_number = s.number AND g.course_offering_uuid = s.course_offering
JOIN course_offerings co ON s.course_offering_uuid = co.uuid
JOIN teachings t ON s.uuid = t.section_uuid
JOIN instructors i ON t.instructor_id = i.id
JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
JOIN subjects sub ON sm.subject_code = sub.code
WHERE sub.abbreviation = 'STAT'
GROUP BY i.name
HAVING SUM(a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
ORDER BY gpa_medio ASC
LIMIT 1
"

# GPA médio ponderado do professor mais difícil
professor_mais_dificil <- dbGetQuery(conn, query_gpa_prof)

professor_mais_dificil
```

```
       professor gpa_medio
1 JAMES D KUELBS    2.6375
```

```
query_gpa_facil <- "
SELECT
    i.name AS professor,
    SUM(
        (a_count * 4.0) +
        (ab_count * 3.5) +
        (b_count * 3.0) +
        (bc_count * 2.5) +
        (c_count * 2.0) +
        (d_count * 1.0) +
```

```
        (f_count * 0.0)
    ) * 1.0 /
    SUM(
        a_count + ab_count + b_count + bc_count + c_count + d_count + f_count
    ) AS gpa_medio
FROM grade_distributions g
JOIN sections s ON g.section_number = s.number AND g.course_offering_uuid = s.course_offering
JOIN course_offerings co ON s.course_offering_uuid = co.uuid
JOIN teachings t ON s.uuid = t.section_uuid
JOIN instructors i ON t.instructor_id = i.id
JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
JOIN subjects sub ON sm.subject_code = sub.code
WHERE sub.abbreviation = 'STAT'
GROUP BY i.name
HAVING SUM(a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
ORDER BY gpa_medio DESC
LIMIT 1
"

# GPA médio ponderado do professor mais fácil
professor_mais_facil <- dbGetQuery(conn, query_gpa_facil)

professor_mais_facil
```

```
  professor gpa_medio
1 YAJUAN SI         4
```

```
query_gpa_disciplina <- "
SELECT
    c.name AS course_name,
    SUM(
        (a_count * 4.0) +
        (ab_count * 3.5) +
        (b_count * 3.0) +
        (bc_count * 2.5) +
        (c_count * 2.0) +
        (d_count * 1.0) +
        (f_count * 0.0)
    ) * 1.0 /
    SUM(
        a_count + ab_count + b_count + bc_count + c_count + d_count + f_count
```

```
    ) AS gpa_medio
FROM grade_distributions g
JOIN sections s ON g.section_number = s.number AND g.course_offering_uuid = s.course_offering
JOIN course_offerings co ON s.course_offering_uuid = co.uuid
JOIN courses c ON co.course_uuid = c.uuid
JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
JOIN subjects sub ON sm.subject_code = sub.code
WHERE sub.abbreviation = 'STAT'
GROUP BY c.name
HAVING SUM(a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
ORDER BY gpa_medio ASC
LIMIT 1
"

# GPA médio ponderado da disciplina mais difícil
disciplina_mais_dificil <- dbGetQuery(conn, query_gpa_disciplina)

disciplina_mais_dificil
```

```
                              course_name gpa_medio
1 Introduction to the Theory of Probability  2.886374
```

```
query_gpa_disciplina_facil <- "
SELECT
    c.name AS course_name,
    SUM(
        (a_count * 4.0) +
        (ab_count * 3.5) +
        (b_count * 3.0) +
        (bc_count * 2.5) +
        (c_count * 2.0) +
        (d_count * 1.0) +
        (f_count * 0.0)
    ) * 1.0 /
    SUM(
        a_count + ab_count + b_count + bc_count + c_count + d_count + f_count
    ) AS gpa_medio
FROM grade_distributions g
JOIN sections s ON g.section_number = s.number AND g.course_offering_uuid = s.course_offering
JOIN course_offerings co ON s.course_offering_uuid = co.uuid
JOIN courses c ON co.course_uuid = c.uuid
```

```
JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
JOIN subjects sub ON sm.subject_code = sub.code
WHERE sub.abbreviation = 'STAT'
GROUP BY c.name
HAVING SUM(a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
ORDER BY gpa_medio DESC
LIMIT 1
"

# GPA médio ponderado da disciplina mais fácil
disciplina_mais_facil <- dbGetQuery(conn, query_gpa_disciplina_facil)

disciplina_mais_facil
```

```
                     course_name gpa_medio
1 Sample Survey Theory and Method         4
```

```
dbDisconnect(conn)
```