

## Desafio 8

2025-09-24

```
library(DBI)
```

```
## Warning: pacote 'DBI' foi compilado no R versão 4.5.1
```

```
library(RSQLite)
```

```
## Warning: pacote 'RSQLite' foi compilado no R versão 4.5.1
```

```
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(dbplyr)
```

```
##
```

```
## Anexando pacote: 'dbplyr'
```

```
## Os seguintes objetos são mascarados por 'package:dplyr':
```

```
##
```

```
##      ident, sql
```

```
library(tidyr)
```

```
library(stringr)
```

```
# -----  
# 1) Conectar ao banco  
#      (ajuste o caminho do arquivo se necessário)  
# -----  
db_path <- "C:/Users/pepem/OneDrive/Documentos/database.sqlite3"  
conn <- DBI::dbConnect(RSQLite::SQLite(), db_path)  
DBI::dbListTables(conn)
```

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

```
DBI::dbGetQuery(conn, "SELECT DISTINCT id, instructors.name FROM instructors
INNER JOIN teachings ON instructors.id = teachings.instructor_id
INNER JOIN sections ON teachings.section_uuid = sections.uuid
INNER JOIN subject_memberships ON sections.course_offering_uuid = subject_memberships.course_of.
INNER JOIN subjects ON subject_memberships.subject_code = subjects.code
WHERE subjects.abbreviation='STAT'")
```

##	id	name
## 1	4392906	MINJING TAO
## 2	393380	DONALD PORTER
## 3	4146030	SHENG WANG
## 4	5697552	KUNLING HUANG
## 5	4898014	DONG XIA
## 6	4613728	JENNIFER NGUYEN
## 7	5520811	SEULKEE YUN
## 8	5516726	DUZHE WANG
## 9	5696436	YONGFENG WU
## 10	4636315	JARED BROWN
## 11	5700058	YUCHEN ZHOU
## 12	4641200	VICTOR LUO
## 13	4884161	GUN WOONG PARK
## 14	5011312	BROOK LUERS
## 15	4728844	DUY NGUYEN
## 16	608997	JOHN GILLET
## 17	4161984	QUOC TRAN
## 18	4576059	YAoyao XU
## 19	4436368	JIE ZHANG
## 20	5015113	LONG PHAN
## 21	5161294	WESLEY CHANG
## 22	5329574	TONG LI
## 23	982957	SHANE HUBLER
## 24	4437355	FAN YANG
## 25	4731731	JU HEE CHO
## 26	4890283	JEEA CHOI
## 27	4885477	SHIXUE LIU
## 28	2602108	WEI-YIN LOH
## 29	4155990	BIN DAI
## 30	4145889	XIPEI YANG
## 31	2602050	DOUGLAS M. BATES
## 32	783769	MICHAEL GEORGE ILTIS
## 33	4845798	LIAM JOHNSTON
## 34	5498648	CRYSTAL CHEN
## 35	5693277	YIFAN MEI
## 36	5690182	RUI CHEN
## 37	4734274	JILI WANG
## 38	5017774	LUWAN ZHANG
## 39	5017655	GINA BENNINGER
## 40	5410648	JIE SONG
## 41	3791802	XIWEN MA

## 42	3456935	HAODA FU
## 43	3773167	DEYUAN JIANG
## 44	3973318	JINGJIANG PENG
## 45	3566378	ERICA LEE DEADMAN
## 46	3775809	AKICHIKA OZEKI
## 47	4280220	MENG SONG
## 48	5520701	HAO TENG
## 49	4897725	DONGGYU KIM
## 50	5332940	COLIN LONGHURST
## 51	361804	FREDERICK BOEHM
## 52	5518378	YONGSU LEE
## 53	5320625	GONZALO CONTADOR
## 54	5514514	HAO CHEN
## 55	4588103	XIN ZHANG
## 56	5632570	BO YANG
## 57	5898424	RUNGANG HAN
## 58	5068296	SEAN KENT
## 59	6043299	DEBRAJ DAS
## 60	5327139	YI LI
## 61	5887049	CLAIRE BOBST
## 62	5883206	CHAN PARK
## 63	5323628	SIJING LI
## 64	5017571	SHENGJI JIA
## 65	5903922	YUCHANG WU
## 66	3626007	ZHUANG WU
## 67	5170801	XIAOMAO LI
## 68	5321537	HYEBIN SONG
## 69	5326059	YOURAN QI
## 70	5177493	YOUNG MIN PARK
## 71	5186047	BIN ZHANG
## 72	4582107	QIURONG CUI
## 73	4733192	SEHO PARK
## 74	4285558	SHENG ZHANG
## 75	5168069	HUIKUN ZHANG
## 76	5162339	ALYSSA DIGILIO
## 77	5214858	YUQING XU
## 78	4538540	STEPHEN BERG
## 79	5326263	PEIGEN ZHOU
## 80	4721620	YAN CHEN
## 81	4460098	QIONG ZHANG
## 82	4278664	YOUNGDEOK HWANG
## 83	5010171	HAN CHEN
## 84	640407	NICHOLAS STEPHEN KEULER
## 85	3107260	ZHENGXIAO WU
## 86	4144479	YUJIN CHUNG
## 87	4590481	WENWEN ZHANG
## 88	4149787	PERLA REYES
## 89	3974100	QI TANG
## 90	3629214	CHIA-CHIEH LIN
## 91	4433317	JIALE XU
## 92	4285354	YANG ZHAO
## 93	3620612	XINWEI DENG
## 94	4133622	MING XIE
## 95	3783649	JINGCI MENG

## 96	143932	ISMOR FISCHER
## 97	4590715	JIAJIE CHEN
## 98	4138542	ZUOFENG SHANG
## 99	4435790	HAO ZHENG
## 100	4590559	XU XU
## 101	4436066	BIN ZHU
## 102	4432994	QUEFENG LI
## 103	3961448	YU QIU LIU
## 104	3981051	JUN LI
## 105	5204089	FAN GAO
## 106	4788841	VICTORIA MANSFIELD
## 107	5498092	ANQI SHI
## 108	5317373	KYLE HEBERT
## 109	5322566	XIUYU MA
## 110	4597046	CHENLIANG XU
## 111	3976099	CHANHAN HSU
## 112	4906458	YUANZHI LI
## 113	2602032	KAM-WAH TSUI
## 114	4430246	XINYU SONG
## 115	4879129	TIMOTHY IDOWU
## 116	4905505	XIAO NIE
## 117	4586558	CHEN CHENG
## 118	5003565	ANDREW LESLIE
## 119	4895244	YAOGUO XIE
## 120	4744350	KEVIN PACKARD
## 121	4586128	YI LIU
## 122	5322094	WENZHI CAO
## 123	4610245	JOSEPH DEUTSCH
## 124	5521838	LAN LUO
## 125	3727371	RYAN ZEA
## 126	2601091	ROBERT WARDROP
## 127	3981781	SANGBUM CHOI
## 128	4285528	XIN LI
## 129	3573852	SEUNGBONG HAN
## 130	3339239	ZHIGUO XIAO
## 131	2601613	RICHARD A. JOHNSON
## 132	4128750	RUI TANG
## 133	801445	IAN BRANSTAD RILEY
## 134	3484236	BO HUANG
## 135	4149843	HEATHER MARIE BRAZEAU
## 136	4286138	XU HE
## 137	495087	GARY HOWARD SCHROEDER
## 138	3781764	JUN ZHANG
## 139	3973813	CHENXI LI
## 140	3955531	SONA Z SWANSON
## 141	3791787	YUAN JIANG
## 142	3620552	JUNHEE HAN
## 143	4578330	LILUN DU
## 144	5170924	XIRAN WANG
## 145	5099221	ZIFENG ZHAO
## 146	5176022	CUIZE HAN
## 147	5170492	YING ZHANG
## 148	4880968	JOHN DAVIS
## 149	5018463	CHEN JING

## 150 4902482	JINGLAN LI
## 151 4285538	YI CHAI
## 152 4904255	THU LE
## 153 4958796	ALAN HUANG
## 154 4725072	HAOYANG FAN
## 155 5690247	BOWEN HU
## 156 5664494	TZU HSIANG HUNG
## 157 5567363	TUN LEE NG
## 158 4787911	ABIGAIL BENZINE
## 159 5656034	ALEXANDER COVINGTON
## 160 3376532	CHELSEY GREEN
## 161 5708167	MICHAEL KUTZLER
## 162 5693114	LILI ZHENG
## 163 5488796	REBECCA POST
## 164 5010560	LU YANG
## 165 5020061	RUOSI GUO
## 166 3668575	XUN ZHAO
## 167 3965653	MIN NIU
## 168 4603165	JIAN WU
## 169 3780817	WEI ZHENG
## 170 4726666	XINJIE HE
## 171 4292460	JIE WEI
## 172 4721578	CLAUDIA SOLIS LEMUS
## 173 4726703	XUEYAO CHEN
## 174 5302154	MANJUSHA KANCHARLA
## 175 4947463	MITCHELL PAUKNER
## 176 5258581	XIAOWU DAI
## 177 5906594	SHENG WANG
## 178 5592111	CHENGNING ZHANG
## 179 5862942	GINA OH
## 180 5192960	LUXI CAO
## 181 4953246	SONG WANG
## 182 4435446	LAM HO
## 183 4896939	TIEN VO
## 184 4598772	SHIZHEN WANG
## 185 5176905	SOHEIL SADEGHI
## 186 5144884	BINGYING XIE
## 187 4957688	GENG LI
## 188 4636145	JENNIFER BIRSTLER
## 189 4195000	MARIA KAMENETSKY
## 190 5478075	KYLE HERBET
## 191 4886987	JIANCHANG HU
## 192 4896172	TRAM TA
## 193 4733966	YUAN WANG
## 194 3777992	KAZUHIKO SHINKI
## 195 3622809	SIJIAN WANG
## 196 3613299	CHIEN-WEI CHEN
## 197 5852265	BEHZAD AALIPUR
## 198 4264265	DEREK BEAN
## 199 4140469	BEN ADAM HAALAND
## 200 2824494	ELOISA D CHAVAS
## 201 3800708	YALI WANG
## 202 3959251	VARSHA KULKARNI
## 203 3253489	SCOTT JOSEPH HETZEL

## 204 3575433	MIN JUNG LEE
## 205 3460189	FANG FANG
## 206 4199106	KRISTEN CYFFKA
## 207 3351080	TING-LI LIN
## 208 751075	BRET LARGET
## 209 4491860	KATHERINE GOODE
## 210 2602042	ERIK NORDHEIM
## 211 5523214	NING FAN
## 212 4646729	ANRU ZHANG
## 213 5214164	GARVESH RASKUTTI
## 214 5476019	KRISHNAKUMAR BALASUBRAMANIAN
## 215 4695492	YAZHEN WANG
## 216 4712098	CHAOYANG YU
## 217 5323691	YILIN ZHANG
## 218 4108525	ZHENGJUN ZHANG
## 219 4194670	XIUFENG SHAO
## 220 5167973	YUAN LI
## 221 4753018	BRET HANLON
## 222 3980675	LILI LAN
## 223 3955546	CECILE ANE
## 224 1426304	ZHANG CHUNMING
## 225 3443667	LEI XU
## 226 5514561	MUHONG GAO
## 227 4963378	AUGUST JENSEN
## 228 4591139	XIAO GUO
## 229 2601039	THOMAS G. KURTZ
## 230 3523865	NICHOLAS HENDERSON
## 231 4903489	CHENSHENG KUANG
## 232 4590738	TAERI UHM
## 233 3882760	NORBERT BINKIEWICZ
## 234 4929500	KARL ROHE
## 235 3095399	JUNGWON MUN
## 236 1126199	JOSEP GINEBRA
## 237 3781422	TAO YU
## 238 4286679	XINXIN YU
## 239 5323690	HAO ZHOU
## 240 4277002	LIE XIONG
## 241 3338237	QIAN ZHIGUANG
## 242 5515201	SHANG WU
## 243 5905014	SHAN LU
## 244 3122658	STEPHEN AARON STANHOPE
## 245 3103459	LANCINE KONATE
## 246 4886028	GUILHERME VIEIRA NUNES LUDWIG
## 247 5631398	PAUL SAVARIAPPAN
## 248 5015194	CUICUI QI
## 249 3404299	KJELL DOKSUM
## 250 5305675	DEREK NORTON
## 251 4261114	KEEGAN KORTHAUER
## 252 5012817	KAILEI CHEN
## 253 5023414	THEVAASIINEN CHANDERENG
## 254 3806328	JOHN KANE
## 255 4593965	SOKOL VAKO
## 256 5718599	QING LI
## 257 5514517	YOUNG LEE

## 258 4734607	HYUNSEUNG KANG
## 259 5694563	BEHZAD AALIPUR HAFSHEJANI
## 260 5170885	SHULEI WANG
## 261 5693629	ZHONGJIE YU
## 262 5514932	LIN QI
## 263 4286103	JIWEI ZHAO
## 264 3801422	SUNDUZ KELES
## 265 4588354	SHUYUN YE
## 266 3483534	HUI WANG
## 267 3978606	KEVIN HASEGAWA ENG
## 268 3343005	XIAODAN WEI
## 269 5167906	MARIA ROJO
## 270 3343055	RUIFENG XU
## 271 5644228	YONGJOON KIM
## 272 232828	NELLIE LAUGHLIN
## 273 270258	MICHAEL HOGAN
## 274 4957916	TIMOTHY IDOWU
## 275 4541610	KARL BROMAN
## 276 789363	REBECCA KOSCIK
## 277 5887858	RONGJUN ZHU
## 278 5906523	BI CHENG WU
## 279 5907599	YUNONG LIN
## 280 4588525	ZHIGENG GENG
## 281 4729481	XIAOPING FENG
## 282 4271924	SHUANG HUANG
## 283 4436002	GUANNAN SUN
## 284 4140397	JEE YEON KIM
## 285 5127444	GREGORY SHINAULT
## 286 3230527	TONGHAI YANG
## 287 3374880	TIMO SEPPALAINEN
## 288 4538519	BENEDEK VALKO
## 289 2602109	DAVID GRIFFEATH
## 290 4308515	DAVID ANDERSON
## 291 1515839	PHILIP WOOD
## 292 4601948	FLORIAN BERTRAND
## 293 3956063	JASON RICHARD SWANSON
## 294 3573775	ALEXANDER KISELEV
## 295 4752238	SAMUEL STECHMANN
## 296 4434130	WAI TONG FAN
## 297 3620114	JUN YIN
## 298 2600973	STEFFEN LEMPP
## 299 3987897	RUIFANG SONG
## 300 2601633	JAMES D KUELBS
## 301 129415	ANATOLE BECK
## 302 5620719	DONGHYUN LEE
## 303 3674168	ANDREJ ZLATOS
## 304 5846795	DANIELE CAPPELLETTI
## 305 6029366	MIHAELA IFRIM
## 306 5110390	SEBASTIEN ROCH
## 307 5354849	SCOTT HOTTOVY
## 308 1634418	JONATHON PETERSON
## 309 2601051	STEPHEN WAINGER
## 310 4680082	ALEXANDER FISH
## 311 4761200	GREGORIO MORENO-FLORES

## 312 3117835	SUKHENDU MEHROTRA
## 313 2600233	DIETRICH UHLENBROCK
## 314 3437432	LEV BORISOV
## 315 2600141	RICHARD A BRUALDI
## 316 2600893	ARNOLD MILLER
## 317 3467530	MATTHEW BALLARD
## 318 3661782	ALBRECHT KLEMM
## 319 2601093	PAUL M TERWILLIGER
## 320 5452538	STEVEN SAM
## 321 3127329	KEN ONO
## 322 4957895	ROBERT HARRON
## 323 4172260	YANNAN QIU
## 324 4896511	JOHN WILTSHIRE-GORDON
## 325 207204	KYUNGMAAN KIM
## 326 3855025	CHRISTOPHER WAGNER
## 327 967335	PAUL RATHOUZ
## 328 3382451	STEPHEN WRIGHT
## 329 4632211	BENJAMIN RECHT
## 330 1114818	MICHAEL FERRIS
## 331 3627934	JESSE THOMAS HOLZER
## 332 5167990	HUILIN HU
## 333 5451706	ALBERTO DEL PIA
## 334 2601512	ROBERT R MEYER
## 335 3127330	SHI JIN
## 336 333334	MARY LINDSTROM
## 337 3111842	ADIN-CRISTIAN ANDREI
## 338 5840930	MICHAEL LIOU
## 339 4724058	CHAOQUN MEI
## 340 340544	CHRISTINA M. KENDZIORSKI
## 341 893827	RONALD GANGNON
## 342 3780010	ROBERT WAYNE GREEN
## 343 4312759	COLE COOK
## 344 2601621	MURRAY CLAYTON
## 345 4633355	QI JIANG
## 346 321374	JUN ZHU
## 347 4952769	ANNE BRUCKNER
## 348 5520995	TING YE
## 349 1079018	JUN SHAO
## 350 3623432	HUAIBAO FENG
## 351 5591917	ZIJIAN NI
## 352 417071	THOMAS COOK
## 353 2601926	RICHARD J. CHAPPELL
## 354 3404298	MOO K CHUNG
## 355 2601916	MICHAEL NEWTON
## 356 2602055	BRIAN YANDELL
## 357 394810	MICHELLE HARRIS
## 358 2600689	NORMAN DRAPER
## 359 3010828	YU MENGANG
## 360 3077702	MING YUAN
## 361 3783301	FANGFANG WANG
## 362 4541633	JUNHO LEE
## 363 2600488	GRACE WAHBA
## 364 3011936	JASON P FINE
## 365 989414	MICHAEL RENE KOSOROK



```

## 366 4426932          YAJUAN SI
## 367 4726678          LU MAO
## 368 2601596          DAVID DEMETS
## 369 5515948          JAMES ANDERSON
## 370 5189535          EDWARD ERKER
## 371 4699427          SHUAI CHEN
## 372 2601674          MARI PALTA
## 373 4272783          GUANHUA CHEN
## 374 3884290  SRIKANTHMADHAVAN ARAVAMUTHAN
## 375 1505727          NATALIA DE LEON GATTI
## 376 1505756          GUILHERME ROSA
## 377 3457484          YANBING ZHENG
## 378 138086           CHRISTINE SORKNESS
## 379 245914           MARIAN R FISHER

```

```

DBI::dbGetQuery(conn, "
WITH stat_sections AS (
  SELECT s.uuid AS section_uuid, s.course_offering_uuid, s.number AS section_number
  FROM sections s
  JOIN subject_memberships sm ON s.course_offering_uuid = sm.course_offering_uuid
  JOIN subjects subj ON sm.subject_code = subj.code
  WHERE subj.abbreviation = 'STAT'
),
gpa_sections AS (
  SELECT
    g.course_offering_uuid,
    g.section_number,
    (4*a_count + 3.5*ab_count + 3*b_count + 2.5*bc_count + 2*c_count + 1*d_count + 0*f_count) AS numer,
    (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) AS denom
  FROM grade_distributions g
  JOIN stat_sections ss
    ON g.course_offering_uuid = ss.course_offering_uuid
   AND g.section_number      = ss.section_number
  WHERE (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
),
prof_gpa AS (
  SELECT t.instructor_id, i.name,
         SUM(gs.numer) * 1.0 / SUM(gs.denom) AS gpa
  FROM teachings t
  JOIN stat_sections ss ON t.section_uuid = ss.section_uuid
  JOIN gpa_sections gs
    ON gs.course_offering_uuid = ss.course_offering_uuid
   AND gs.section_number      = ss.section_number
  JOIN instructors i ON t.instructor_id = i.id
  GROUP BY t.instructor_id, i.name
)
SELECT *
FROM prof_gpa
ORDER BY gpa ASC
LIMIT 1;
")

```

```

##  instructor_id      name      gpa
## 1      2601633 JAMES D KUELBS 2.6375

```

```

DBI::dbGetQuery(conn, "
WITH stat_sections AS (
  SELECT s.uuid AS section_uuid, s.course_offering_uuid, s.number AS section_number
  FROM sections s
  JOIN subject_memberships sm ON s.course_offering_uuid = sm.course_offering_uuid
  JOIN subjects subj ON sm.subject_code = subj.code
  WHERE subj.abbreviation = 'STAT'
),
gpa_sections AS (
  SELECT
    g.course_offering_uuid,
    g.section_number,
    (4*a_count + 3.5*ab_count + 3*b_count + 2.5*bc_count + 2*c_count + 1*d_count + 0*f_count) AS numer,
    (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) AS denom
  FROM grade_distributions g
  JOIN stat_sections ss
    ON g.course_offering_uuid = ss.course_offering_uuid
   AND g.section_number      = ss.section_number
  WHERE (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
),
prof_gpa AS (
  SELECT t.instructor_id, i.name,
         SUM(gs.numer) * 1.0 / SUM(gs.denom) AS gpa
  FROM teachings t
  JOIN stat_sections ss ON t.section_uuid = ss.section_uuid
  JOIN gpa_sections gs
    ON gs.course_offering_uuid = ss.course_offering_uuid
   AND gs.section_number      = ss.section_number
  JOIN instructors i ON t.instructor_id = i.id
  GROUP BY t.instructor_id, i.name
)
SELECT *
FROM prof_gpa
ORDER BY gpa DESC
LIMIT 1;
")

```

```

##      instructor_id              name gpa
## 1      3884290 SRIKANTHMADHAVAN ARAVAMUTHAN    4

```

```

DBI::dbGetQuery(conn, "
WITH stat_offerings AS (
  SELECT co.uuid AS course_offering_uuid, c.uuid AS course_uuid, c.number, c.name
  FROM course_offerings co
  JOIN courses c ON co.course_uuid = c.uuid
  JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
  JOIN subjects subj ON sm.subject_code = subj.code
  WHERE subj.abbreviation = 'STAT'
),
gpa_sections AS (
  SELECT
    g.course_offering_uuid,
    (4*a_count + 3.5*ab_count + 3*b_count + 2.5*bc_count + 2*c_count + 1*d_count + 0*f_count) AS numer,

```

```

        (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) AS denom
    FROM grade_distributions g
    WHERE (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
),
course_gpa AS (
    SELECT so.course_uuid, so.number, so.name,
           SUM(gs.numer) * 1.0 / SUM(gs.denom) AS gpa
    FROM stat_offerings so
    JOIN gpa_sections gs ON so.course_offering_uuid = gs.course_offering_uuid
    GROUP BY so.course_uuid, so.number, so.name
)
SELECT *
FROM course_gpa
ORDER BY gpa ASC
LIMIT 1;
")

```

```

##                                course_uuid number
## 1 3ef3ba60-426e-39ad-a61f-17d381b193d6      431
##                                name      gpa
## 1 Introduction to the Theory of Probability 2.886374

```

```

DBI::dbGetQuery(conn, "
WITH stat_offerings AS (
    SELECT co.uuid AS course_offering_uuid, c.uuid AS course_uuid, c.number, c.name
    FROM course_offerings co
    JOIN courses c ON co.course_uuid = c.uuid
    JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
    JOIN subjects subj ON sm.subject_code = subj.code
    WHERE subj.abbreviation = 'STAT'
),
gpa_sections AS (
    SELECT
        g.course_offering_uuid,
        (4*a_count + 3.5*ab_count + 3*b_count + 2.5*bc_count + 2*c_count + 1*d_count + 0*f_count) AS numer,
        (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) AS denom
    FROM grade_distributions g
    WHERE (a_count + ab_count + b_count + bc_count + c_count + d_count + f_count) > 0
),
course_gpa AS (
    SELECT so.course_uuid, so.number, so.name,
           SUM(gs.numer) * 1.0 / SUM(gs.denom) AS gpa
    FROM stat_offerings so
    JOIN gpa_sections gs ON so.course_offering_uuid = gs.course_offering_uuid
    GROUP BY so.course_uuid, so.number, so.name
)
SELECT *
FROM course_gpa
ORDER BY gpa DESC
LIMIT 1;
")

```

```

##                                course_uuid number

```

```
## 1 126a81bd-34e1-35f8-82f4-5edb70c3ae8b    841
##                                           name gpa
## 1 Nonparametric Statistics and Machine Learning Methods    4
```

```
DBI::dbDisconnect(conn)
```