

R Notebook

Escola de Dança



dança

O objetivo deste *script* é fazer consultas utilizando as principais ferramentas da linguagem SQL. O banco de dados utilizado foi criado utilizando o PG Admin do PostgreSQL simulando uma escola de dança.

Os códigos com a criação do banco de dados estão disponíveis neste repositório para replicação.

As consultas foram realizadas na IDE do *R Studio*, mas os mesmos comandos podem ser executados utilizando a biblioteca PANDAS da linguagem Python.

O objetivo é responder perguntas de negócios por meio de consultas com a linguagem SQL com apoio da linguagem R. Serão respondidas as seguintes perguntas:

1. Quantas (os) alunas (os) avançados há na escola de dança?
2. Quantas (os) alunas (os) são do sexo feminino?
3. Quais as Unidades Federativas das (os) alunas (os)?
4. Quais os nomes das (os) professoras (es)?
5. Quais estilos de forró são ensinados na escola?
6. Quais os menores e maiores valores pagos por mês na escola?
7. Qual o valor médio que os alunos pagam por mês na escola?
8. Quais alunos pagam os maiores e menores valores por mês na escola?

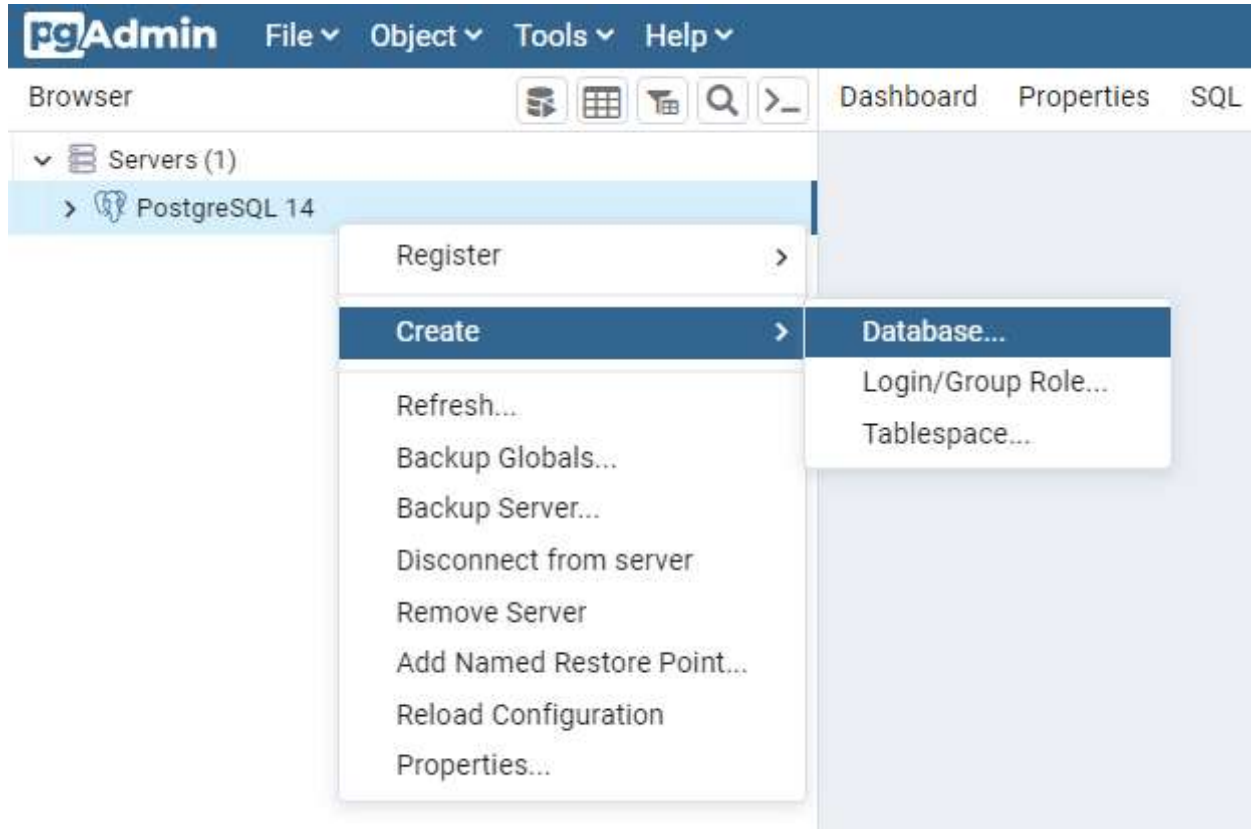
9. Qual o valor médio gasto por aluno por nome?

10. Quem tem maior média de gastos na escola: homens ou mulheres?

Vamos responder essas perguntas a seguir 🤖.

Criando o Banco de Dados no PostgreSQL

O primeiro passo é criar o banco de dados no PG Admin do PostgreSQL. Para criar o banco basta seguir o passo a passo das imagens abaixo:



dança

Create - Database

General Definition Security Parameters Advanced SQL

Database: dancaAP

Owner: postgres

Comment:

Close Reset Save

Basta clicar com o botão direito em **PostgreSQL 14**, depois selecionar a opção **Create** e depois selecionar **DataBase**. Por fim basta colocar o nome **dancasAP** e clicar em **Save**.

Caso você prefira criar via código, seguem abaixo os comandos:

Dashboard Properties SQL Statistics Dependencies Dependents

```

1  -- Database: dancaAP
2
3  -- DROP DATABASE IF EXISTS "dancaAP";
4
5  CREATE DATABASE "dancaAP"
6      WITH
7      OWNER = postgres
8      ENCODING = 'UTF8'
9      LC_COLLATE = 'Portuguese_Brazil.1252'
10     LC_CTYPE = 'Portuguese_Brazil.1252'
11     TABLESPACE = pg_default
12     CONNECTION LIMIT = -1;

```

dança

Alimentando o Banco de Dados

Neste projeto eu criei 5 tabelas: *alunos* (descrição dos alunos da escola de danças), *dancas* (para descrever as danças ensinadas na escola), *instrutores* (base com os instrutores das danças), *itensvenda* (os serviços e produtos vendidos na escola) e *turno* (os turnos em que ocorrem as aulas).

Como o objetivo deste *script* é fazer consultas, não serão detalhados os comandos para a criação das tabelas, se não ia ficar muito grande. Seguem abaixo os comandos para a criação das tabelas.

Dashboard Properties SQL Statistics Dependencies Dependents

```

1  -- Table: public.alunos
2
3  -- DROP TABLE IF EXISTS public.alunos;
4
5  CREATE TABLE IF NOT EXISTS public.alunos
6  (
7      idaluno integer NOT NULL DEFAULT nextval('idaluno'::regclass),
8      cliente character varying(50) COLLATE pg_catalog."default",
9      estado character varying(2) COLLATE pg_catalog."default",
10     sexo character(1) COLLATE pg_catalog."default",
11     status character varying(50) COLLATE pg_catalog."default",
12     CONSTRAINT alunos_pkey PRIMARY KEY (idaluno)
13 )
14
15 TABLESPACE pg_default;
16
17 ALTER TABLE IF EXISTS public.alunos
18     OWNER to postgres;

```

dança

Dashboard Properties SQL Statistics Dependencies Dependents

```

1  -- Table: public.dancas
2
3  -- DROP TABLE IF EXISTS public.dancas;
4
5  CREATE TABLE IF NOT EXISTS public.dancas
6  (
7      iddance integer NOT NULL DEFAULT nextval('iddance'::regclass),
8      forro character varying(100) COLLATE pg_catalog."default",
9      zouk character varying(100) COLLATE pg_catalog."default",
10     ventre character varying(100) COLLATE pg_catalog."default",
11     CONSTRAINT dancas_pkey PRIMARY KEY (iddance)
12 )
13
14 TABLESPACE pg_default;
15
16 ALTER TABLE IF EXISTS public.dancas
17     OWNER to postgres;

```

dança

Dashboard Properties SQL Statistics Dependencies Dependents

```
1  -- Table: public.instrutores
2
3  -- DROP TABLE IF EXISTS public.instrutores;
4
5  CREATE TABLE IF NOT EXISTS public.instrutores
6  (
7      idinstrutor integer NOT NULL DEFAULT nextval('idinstrutor'::regclass),
8      nome character varying(50) COLLATE pg_catalog."default",
9      CONSTRAINT instrutores_pkey PRIMARY KEY (idinstrutor)
10 )
11
12 TABLESPACE pg_default;
13
14 ALTER TABLE IF EXISTS public.instrutores
15     OWNER to postgres;
```

dança

Dashboard Properties SQL Statistics Dependencies Dependents

```
1  -- Table: public.alunos
2
3  -- DROP TABLE IF EXISTS public.alunos;
4
5  CREATE TABLE IF NOT EXISTS public.alunos
6  (
7      idaluno integer NOT NULL DEFAULT nextval('idaluno'::regclass),
8      cliente character varying(50) COLLATE pg_catalog."default",
9      estado character varying(2) COLLATE pg_catalog."default",
10     sexo character(1) COLLATE pg_catalog."default",
11     status character varying(50) COLLATE pg_catalog."default",
12     CONSTRAINT alunos_pkey PRIMARY KEY (idaluno)
13 )
14
15 TABLESPACE pg_default;
16
17 ALTER TABLE IF EXISTS public.alunos
18     OWNER to postgres;
```

dança

Dashboard Properties SQL Statistics Dependencies Dependents

```

1  -- Table: public.itensvenda
2
3  -- DROP TABLE IF EXISTS public.itensvenda;
4
5  CREATE TABLE IF NOT EXISTS public.itensvenda
6  (
7      iddance integer NOT NULL,
8      idaluno integer,
9      quantidade integer,
10     valorunitario numeric(10,2),
11     valortotal numeric(10,2),
12     desconto numeric(10,2),
13     CONSTRAINT itensvenda_pkey PRIMARY KEY (iddance),
14     CONSTRAINT itensvenda_idaluno_fkey FOREIGN KEY (idaluno)
15         REFERENCES public.alunos (idaluno) MATCH SIMPLE
16         ON UPDATE NO ACTION
17         ON DELETE CASCADE,
18     CONSTRAINT itensvenda_iddance_fkey FOREIGN KEY (iddance)
19         REFERENCES public.dancas (iddance) MATCH SIMPLE
20         ON UPDATE NO ACTION
21         ON DELETE RESTRICT
22 )
23
24 TABLESPACE pg_default;
25
26 ALTER TABLE IF EXISTS public.itensvenda
27     OWNER to postgres;

```

```

1  -- Table: public.turno
2
3  -- DROP TABLE IF EXISTS public.turno;
4
5  CREATE TABLE IF NOT EXISTS public.turno
6  (
7      idvenda integer NOT NULL DEFAULT nextval('iddance'::regclass),
8      manha character varying(50) COLLATE pg_catalog."default",
9      tarde character varying(50) COLLATE pg_catalog."default",
10     noite character varying(50) COLLATE pg_catalog."default",
11     CONSTRAINT turno_pkey PRIMARY KEY (idvenda)
12 )
13
14 TABLESPACE pg_default;
15
16 ALTER TABLE IF EXISTS public.turno
17     OWNER to postgres;

```

Alimentando as Tabelas

Agora que já criamos as tabelas no banco de dados então vamos colocar nosso tesouro nelas: **os dados**.

1. Tabela alunos

```
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(1,'Bruno Amante de Brasília', 'DF', 'M', 'Avançado');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(2,'Rock Lee', 'DF', 'M', 'Avançado');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(3,'Adriana Guedelha', 'RO', 'F', 'Intermediário');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(4,'Aida Dorneles', 'RN', 'F', 'Iniciante');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(5,'Márcio da Silva Força da Juventude', 'DF', 'M', 'Avançado');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(6,'Alberto Cezimbra', 'AM', 'M', 'Iniciante');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(7,'Alberto Monsanto', 'RN', 'M', 'Avançado');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(8,'Albino Canela', 'AC', 'M', 'Iniciante');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(9,'Alceste Varanda', 'RR', 'F', 'Iniciante');
INSERT INTO alunos(idaluno, cliente, estado, sexo, status) VALUES(10,'Alcides Carvalhais', 'RO', 'M', 'Iniciante');
```

2. Tabela dancas

```
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(1,'Classico', 'Basico', 'Avançado');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(2,'Universitário', 'Intermediario', 'Basico');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(3,'Classico', 'Basico', 'Avançado');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(4,'Classico', 'Basico', 'Avançado');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(5,'Classico', 'Basico', 'Avançado');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(6,'Classico', 'Basico', 'Avançado');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(7,'Classico', 'Basico', 'Avançado');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(8,'Classico', 'Basico', 'Basico');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(9,'Classico', 'Avancado', 'Avançado');
INSERT INTO dancas(iddance, forro, zouk, ventre) VALUES(10,'Classico', 'Basico', 'Avançado');
```

3. Tabela instrutores

```
INSERT INTO instrutores(idinstrutor, nome) VALUES(1,'Chuck Norris');
INSERT INTO instrutores(idinstrutor, nome) VALUES(2,'Bruce Lee');
INSERT INTO instrutores(idinstrutor, nome) VALUES(3,'Karin Abdul Jabar');
INSERT INTO instrutores(idinstrutor, nome) VALUES(4,'Mestre Miag');
INSERT INTO instrutores(idinstrutor, nome) VALUES(5,'Pai Mey');
```

4. Tabela itensvendas


```
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (1, 5, 1, 100, 100, 0.00);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (2, 6, 1, 97.75, 97.75, 0.98);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (1, 7, 1, 135.00, 135.00, 1.35);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (2, 4, 1, 150.40, 150.40, 1.50);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (3, 3, 2, 2966.00, 5932.00, 0.00);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (1, 2, 1, 155.00, 155.00, 0.00);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (2, 8, 1, 7820.85, 7820.85, 0.00);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (3, 6, 2, 2955.00, 5910.00, 59.10);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (1, 1, 1, 97.75, 97.75, 0.98);
INSERT INTO itensvenda(iddance, idaluno, quantidade, valorunitario, valortotal, desconto) VALUES (1, 10, 1, 8852.00, 8852.00, 0.00);
```

5. Turno

```
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (1, 4, 3, 'Manhã');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (1, 2, 2, 'Noite');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (2, 3, 4, 'Noite');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (1, 2, 5, 'Noite');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (1, 3, 6, 'Noite');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (4, 2, 7, 'Noite');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (2, 1, 8, 'Noite');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (1, 5, 9, 'Manhã');
INSERT INTO public.turno(iddance, idinstrutor, idaluno, horario) VALUES (1, 5, 10, 'Manhã');
```

O banco de dados já está criado e alimentado. Agora podemos conduzir as consultas e responder as perguntas de negócios utilizando a linguagem SQL. Vamos lá!

Introdução: Carregando os pacotes e conectando-se ao Banco de Dados

Neste caso vamos instalar os pacotes necessários e nos conectar ao banco de dados. Seguem abaixo os códigos e o acesso ao PostgreSQL.

```
# Pacotes para a analise

# 1. Instalando pacotes necessarios
# install.packages("RPostgreSQL") # instala o pacote RPostgreSQL
# install.packages("RPostgres") # Instala o pacote RPostgres
# install.packages("DBI") # Instala o pacote DBI

# 2. Carregando os pacotes necessario
library(RPostgreSQL) # acessando o banco de dados
```



```
## Warning: package 'RPostgreSQL' was built under R version 4.1.3
```

```
## Carregando pacotes exigidos: DBI
```

```
## Warning: package 'DBI' was built under R version 4.1.3
```

```
library(DBI) # organizando o banco  
library(RPostgres) # pacote mais antigo caso o RPostgreSQL nao funcione
```

```
## Warning: package 'RPostgres' was built under R version 4.1.3
```

```
library(tidyr) # pacote para organizacao dos dados
```

```
## Warning: package 'tidyr' was built under R version 4.1.1
```

```
library(dplyr) # pacote para acesso e organizacao e tratamento dos dados
```

```
##  
## 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(tinytex) # pacote para construir o portfolio na WEB
```

```
## Warning: package 'tinytex' was built under R version 4.1.1
```

```
# 3. Conectando o banco de dados  
conexao = dbConnect(RPostgres::Postgres(), dbname = "dancaAP",  
                    host = "localhost", port = 5432, user = "postgres",  
                    password = "Info@1234")
```

```
# 4. Acessando o banco de dados
```

```
dancas = "SELECT *
FROM dancas
"
```

```
alunos = "SELECT *
FROM alunos
"
```

```
instrutores = "SELECT *
FROM instrutores
"
```

```
itensvenda = "SELECT *
FROM itensvenda
"
```

```
turno = "SELECT *
FROM turno
"
```

```
dancas = dbGetQuery(conexao, dancas)
alunos = dbGetQuery(conexao, alunos)
instrutores = dbGetQuery(conexao, instrutores)
itensvenda = dbGetQuery(conexao, itensvenda)
turno = dbGetQuery(conexao, turno)
```

```
# 4.1. Visualizando partes do banco dancas
class(dancas) # classe dos dados
```

```
## [1] "data.frame"
```

```
dim(dancas) # dimensao do banco
```

```
## [1] 10 4
```

```
head(dancas) # primeiras linhas e colunas
```

	iddance <int>	forro <chr>	zouk <chr>	ventre <chr>
1	1	Classico	Basico	Avançado
2	2	Universitário	Intermediario	Basico
3	3	Classico	Basico	Avançado
4	4	Classico	Basico	Avançado
5	5	Classico	Basico	Avançado
6	6	Classico	Basico	Avançado
6 rows				

```
tail(dancas) # ultimas linhas e colunas
```

	iddance <int>	forro <chr>	zouk <chr>	ventre <chr>
5	5	Classico	Basico	Avançado
6	6	Classico	Basico	Avançado
7	7	Classico	Basico	Avançado
8	8	Classico	Basico	Basico
9	9	Classico	Avancado	Avançado
10	10	Classico	Basico	Avançado

6 rows

Agora vamos responder as perguntas de negócio.

1. Quantas (os) alunas (os) avançados há na escola de dança?

```
sql2 = "SELECT COUNT(alunos)
FROM alunos WHERE status = 'Avançado'"
```

```
nivel = dbGetQuery(conexao, sql2)
nivel
```

	count <int64>
	5

1 row

Como observado, existem 5 alunos no nível avançado na escolca.

2. Quais são as (os) alunas (os) são do sexo feminino?

```
sql3 = "SELECT cliente, sexo
FROM alunos WHERE sexo = 'F'
"
```

```
feminino = dbGetQuery(conexao, sql3)
feminino
```

cliente <chr>	sexo <chr>
Adriana Guedelha	F

cliente <chr>	sexo <chr>
Aida Dorneles	F
Alceste Varanda	F
Nicolle	F
4 rows	

Como observado, são 4 mulheres: a Adriana Gadelha, Aida Dorneles, Alceste Varanda e Nicolle.

3. Quais as Unidades Federativas das (os) alunas (os)?

```
sql4 = "SELECT cliente, estado  
FROM alunos  
"
```

```
uf = dbGetQuery(conexao, sql4)  
uf
```

cliente <chr>	estado <chr>
Bruno Amante de Brasília	DF
Rock Lee	DF
Adriana Guedelha	RO
Aida Dorneles	RN
Márcio da Silva Força da Juventude	DF
Alberto Cezimbra	AM
Alberto Monsanto	RN
Albino Canela	AC
Alceste Varanda	RR
Alcides Carvalhais	RO
1-10 of 11 rows	
Previous 1 2 Next	

3.1. Quais as (os) alunas (os) do DF?

```
sql4_1 = "SELECT cliente, estado  
FROM alunos WHERE estado = 'DF'  
"
```

```
df = dbGetQuery(conexao, sql4_1)  
df
```


cliente <chr>	estado <chr>
Bruno Amante de Brasília	DF
Rock Lee	DF
Márcio da Silva Força da Juventude	DF
3 rows	

Como observado, existem 3 alunos do DF e todos do sexo masculino: Bruno, Rock Lee e Márcio.

4. Quais os nomes das (os) professoras (es)?

```
sql5 = "SELECT *
FROM instrutores
"

nomes = dbGetQuery(conexao, sql5)
nomes
```

idinstrutor <int>	nome <chr>
1	Chuck Norris
2	Bruce Lee
3	Karin Abdul Jabar
4	Mestre Miag
5	Pai Mey
5 rows	

Os nomes dos intrutores são Chuck Norris, Bruce Lee, Karin Abdul Jabar, Mestre Miag e Pai Mey. É uma homenagem aos mestres de artes marciais e ex jogador de basquete Karin Abdul Jabar.

5. Quais estilos de forró são ensinados na escola?

```
sql6 = "SELECT DISTINCT forro
FROM dancas
"

estilos = dbGetQuery(conexao, sql6)
estilos
```

forro <chr>
Universitário

forro

<chr>

Classico

2 rows

No caso a escola oferta 2 estilos de forró: o clássico e universitário.

6. Quais os menores e maiores valores pagos por mês na escola?

```
sql7 = "SELECT MAX(valorunitario)
FROM itensvenda
"

maximo_pago = dbGetQuery(conexao, sql7)
maximo_pago
```

max

<dbl>

8852

1 row

```
sql7_1 = "SELECT MIN(valorunitario)
FROM itensvenda
"

minimo_pago = dbGetQuery(conexao, sql7_1)
minimo_pago
```

min

<dbl>

97.75

1 row

O valor máximo pago por mês é R\$ 8.852,00 e o mínimo de R\$ 97,75. Vamos descobrir quem são as (os) alunos que pagam estes valores.

```
sql7_2 = "SELECT alunos.cliente, itensvenda.valorunitario
FROM alunos JOIN itensvenda on alunos.idaluno = itensvenda.idaluno
ORDER BY itensvenda.valorunitario DESC
"

maximo_aluno = dbGetQuery(conexao, sql7_2)
maximo_aluno
```

cliente

<chr>

valorunitario

<dbl>

cliente <chr>	valorunitario <dbl>
Alcides Carvalhais	8852.00
Albino Canela	7820.85
Adriana Guedelha	2966.00
Alberto Cezimbra	2955.00
Rock Lee	155.00
Aida Dorneles	150.40
Alberto Monsanto	135.00
Márcio da Silva Força da Juventude	100.00
Bruno Amante de Brasília	97.75
Alberto Cezimbra	97.75
1-10 of 10 rows	

```
sql7_3 = "SELECT alunos.cliente, itensvenda.valorunitario
FROM alunos JOIN itensvenda on alunos.idaluno = itensvenda.idaluno
ORDER BY itensvenda.valorunitario
"
```

```
minimo_aluno = dbGetQuery(conexao, sql7_3)
minimo_aluno
```

cliente <chr>	valorunitario <dbl>
Alberto Cezimbra	97.75
Bruno Amante de Brasília	97.75
Márcio da Silva Força da Juventude	100.00
Alberto Monsanto	135.00
Aida Dorneles	150.40
Rock Lee	155.00
Alberto Cezimbra	2955.00
Adriana Guedelha	2966.00
Albino Canela	7820.85
Alcides Carvalhais	8852.00
1-10 of 10 rows	

Quem paga o maior valor mensal na academia é o Alcides Carvalhais e os menores são o Alberto Cezimbra e Bruno.

7. Qual o valor médio que os alunos pagam por mês na escola?

```
sql8 = "SELECT AVG(valorunitario)
from itensvenda
"

media = dbGetQuery(conexao, sql8)
media
```

	avg <dbl>
	2332.975

1 row

A média gasta por aluno foi de R\$ 2.332,98. Vamos analisar a mediana e comparar com a média

```
mediana = median(itensvenda$valorunitario)
mediana
```

```
## [1] 152.7
```

A mediana foi de R\$ 152,70, bem distante da média. Neste caso, há uma certa dispersão de pagamentos entre os clientes.

10. Quem tem maior média de gastos na escola: homens ou mulheres?

```
sql9 = "SELECT alunos.sexo, AVG(itensvenda.valorunitario)
FROM alunos JOIN itensvenda on alunos.idaluno = itensvenda.idaluno
GROUP BY alunos.sexo
"

media_sexo = dbGetQuery(conexao, sql9)
media_sexo
```

sexo <chr>	avg <dbl>
M	2526.669
F	1558.200

2 rows

Os homens gastam em média R\$ 2526,67 e as mulheres R\$ 1558,00. Neste caso, os homens tiveram um gasto maior na escola.