

# Trabalho Prático do Módulo 3

**Entrega** 20 jun em 23:59      **Pontos** 25      **Perguntas** 10  
**Disponível** até 20 jun em 23:59      **Limite de tempo** Nenhum  
**Tentativas permitidas** 2

## Instruções

O Trabalho Prático do Módulo 3 está disponível!

### 1. Instruções para realizar o trabalho prático

Consulte a data de entrega no teste e em seu calendário.

Reserve um tempo para realizar a atividade, leia as orientações e enunciados com atenção. Em caso de dúvidas utilize o "Fórum de dúvidas do Trabalho Prático".

Para iniciá-lo clique em "Fazer teste". Você tem somente **uma** tentativa e não há limite de tempo definido para realizá-lo. Caso precise interromper a atividade, apenas deixe a página e, ao retornar, clique em "Retomar teste".

Clique em "Enviar teste" **somente** quando você concluí-lo. Antes de enviar confira todas as questões.

O gabarito será disponibilizado partir de sexta-feira, **19/06/2020**, às 21h.

Bons estudos!

### 2. O arquivo abaixo contém o enunciado do trabalho prático

[Enunciado Trabalho Prático - Módulo 3 - Bootcamp Analista de Dados.pdf](#) 

[Arquivos complementares - Módulo 3.zip](#)

[Tutorial de instalação do MySQL.pdf](#) 

## Histórico de tentativas

	Tentativa	Tempo	Pontuação
MANTIDO	<a href="#">Tentativa 2</a>	4 minutos	25 de 25
MAIS RECENTE	<a href="#">Tentativa 2</a>	4 minutos	25 de 25
	<a href="#">Tentativa 1</a>	122 minutos	22,5 de 25

⚠ As respostas corretas estarão disponíveis em 20 jun em 23:59.

Pontuação desta tentativa: **25** de 25

Enviado 19 jun em 17:55

Esta tentativa levou 4 minutos.

### Pergunta 1

2,5 / 2,5 pts

Considerando a atividade 1, analise os comandos abaixo e marque aquele que não pode ser usado para criação do esquema *bootcamp* e suas respectivas tabelas, ou para carga de dados das tabelas:

☐

```
CREATE TABLE `estado` (  
  `CodEstadoIBGE` int NOT NULL,  
  `NomeEstado` varchar(45) NOT NULL,  
  `SiglaEstado` char(2) NOT NULL,  
  `Regiao` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`CodEstadoIBGE`)  
);
```

☒

```
INSERT TABLE `cidade` (  
  `CodigoCompletoIBGE` varchar(45) NOT NULL,  
  `CodigoCidadeIBGE` varchar(10) NOT NULL,  
  `NomeCidade` varchar(150) NOT NULL,  
  `CodEstadoIBGE` int NOT NULL,  
  PRIMARY KEY (`CodigoCompletoIBGE`),  
  KEY `fk_Cidade_Estado_idx` (`CodEstadoIBGE`),  
  CONSTRAINT `fk_Cidade_Estado` FOREIGN KEY (`CodEstadoIBGE`)  
  REFERENCES `estado` (`CodEstadoIBGE`)  
);
```

☐

```
INSERT INTO `bootcamp`.`tipounidade` (`idTipoUnidade`, `dscTipoUnidade`) VALUES (1, 'Casa');
```

☐

```
CREATE SCHEMA IF NOT EXISTS `bootcamp` DEFAULT CHARACTER SET utf8 ;
```

### Pergunta 2

2,5 / 2,5 pts

Qual das opções não pode ser utilizada para recuperar todos os dados da tabela *estado*?

☐ `query <- "SELECT * FROM estado"`  
`resultado <- dbGetQuery(con,query)`  
`resultado`

☐ `query <- "SELECT CodEstadoIBGE, NomeEstado, SiglaEstado, Regiao FROM estado"`  
`resultado <- dbGetQuery(con,query)`  
`resultado`

☐ `resultado <- dbReadTable(con,"estado")`  
`resultado`

☒ `query <- "SELECT CodEstadoIBGE, NomeEstado, SiglaEstado, Regiao FROM estado"`  
`resultado <- dbSendQuery(con,query)`  
`resultado`

### Pergunta 3

2,5 / 2,5 pts

Considerando o programa abaixo, escrito na linguagem R, marque a opção correta que represente o resultado esperado para execução do comando da linha 13.

```
1 #Importa pacote RMariaDB se ele ainda não foi carregado
2 if(!"RMariaDB" %in% (.packages())){require(RMariaDB)}
3
4 #Conecta ao SGBD MySQL --> Banco de dados bootcamp
5 con <- dbConnect(MariaDB(), user = "root", password = "igti",
6                   dbname = "bootcamp", host = "localhost",serverTimezone='UTC')
7
8 resultado <- dbListTables(con)
9 resultado
10
11 resultado <- dbReadTable(con,"tipounidade")
12 resultado <- dbReadTable(con,"caracteristicasgerais")
13 resultado
14
15 resultado <- dbSendQuery(con,"INSERT INTO tipounidade(idTipoUnidade,dscTipoUnidade) VALUES(9,'Fazenda');")
16 resultado
17
18 query <- "SELECT * FROM caracteristicasgerais;"
19 resultado <- dbGetQuery(con,query)
20 resultado
```

☐ `'caracteristicageralimovel' 'caracteristicasgerais' 'cidade' 'estado' 'imovel' 'tipounidade'`

☐ `IdcaracteristicasGerais dsccaracteristicasGerais`

☐

```
<MariaDBResult>
SQL INSERT INTO tipounidade(idTipoUnidade,dscTipoUnidade) VALUES(9,'Fazenda');
ROWS Fetched: 0 [complete]
Changed: 1
```

idcaracteristicasGerais	dscCaracteristicasGerais
1	Portaria 24 horas
2	Elevador
3	Piscina

## Pergunta 4

2,5 / 2,5 pts

Considerando o programa abaixo, escrito na linguagem R, marque a opção INCORRETA.

```
1 #Importa pacote RMariaDB se ele ainda não foi carregado
2 if(!"RMariaDB" %in% (.packages())){require(RMariaDB)}
3
4 #Conecta ao SGBD MySQL --> Banco de dados bootcamp
5 con <- dbConnect(MariaDB(), user = "root", password = "igti",
6                 dbname = "bootcamp", host = "localhost",serverTimezone='UTC')
7
8 resultado <- dbListTables(con)
9 resultado
10
11 resultado <- dbReadTable(con,"tipounidade")
12 resultado <- dbReadTable(con,"caracteristicasgerais")
13 resultado
14
15 resultado <- dbSendQuery(con,"INSERT INTO tipounidade(idTipoUnidade,dscTipoUnidade) VALUES(9,'Fazenda');")
16 resultado
17
18 query <- "SELECT * FROM caracteristicasgerais;"
19 resultado <- dbGetQuery(con,query)
20 resultado
```

☐

O comando escrito nas linhas 5 e 6 serve para fazer a conexão do script com o banco de dados MySQL e pode ser usada também com o SGBD MariaDB.

☐

Na linha 8, o método dbListTables(con) é usado para consultar todas as tabelas existentes no banco de dados que foi conectado. A linha 9, exibe o resultado obtido que é:

```
'caracteristicageralimovel' 'caracteristicasgerais' 'cidade' 'estado' 'imovel' 'tipounidade'
```



O comando `dbGetQuery`, da linha 19, é usado para recuperar o resultado da query passada como parâmetro, e seu resultado é um dataframe com o resultado da consulta.



Na linhas 11 e 12, o comando `dbReadTable` é usado para ler o script de criação (comando DDL) da tabela que é passada como parâmetro.

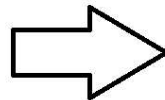
## Pergunta 5

2,5 / 2,5 pts

A tabela *caracteristicasgerais*, originalmente, possui apenas 3 características cadastradas. Para alterar esta tabela incluindo novas características, qual das opções abaixo não pode ser utilizada? Considere a linguagem R e que o arquivo “caracteristicasgerais.csv” está salvo no seguinte caminho: “C:\Bootcamp\Datasets\CSV”.

De

idcaracteristicasGerais	dscaracteristicasGerais
1	Portaria 24 horas
2	Elevador
3	Piscina
NULL	NULL



Para

idcaracteristicasGerais	dscaracteristicasGerais
1	Portaria 24 horas
2	Elevador
3	Piscina
4	Salão de festas
5	Área gourmet
6	Água individual
7	Gás canalizado
8	Aquecimento solar
9	Vaga coberta
10	Vaga livre
11	Armários na cozinha
12	Closet
13	Armários no quarto



```
filename <- "C:/Bootcamp/Datasets/CSV/caracteristicasgerais.csv"
insertdata <- read.csv(file=filename, header=TRUE, sep=",")
insertdata

dbWriteTable(con, value = insertdata, row.names = FALSE,
             name = "caracteristicasgerais", append = TRUE )
```



```
query <- "INSERT INTO caracteristicasGerais(idcaracteristicasGerais,dscaracteristicasGerais)
VALUES(4,'Salão de festas');"

dbWriteTable(con, value = query, row.names = FALSE,
             name = "caracteristicasgerais", append = TRUE )
```

```
id1 <- 5
desc1 <- 'Área gourmet'
id2 <- 6
desc2 <- 'Água individual'
id3 <- 7
desc3 <- 'Gás canalizado'
query <- paste("INSERT INTO caracteristicasGerais(idcaracteristicasGerais,dsccharacteristicasGerais)
VALUES(",id1,"","desc1,""),(",id2,"","desc2,""),(",id3,"","desc3,"");",sep='')
results <- dbSendQuery(con,query)
```

```
query <- "INSERT INTO caracteristicasGerais(idcaracteristicasGerais,dsccharacteristicasGerais)
VALUES(4,'Salão de festas');"
results <- dbSendQuery(con,query)
```

### Pergunta 6

2,5 / 2,5 pts

Na linguagem Python, usando o pacote mysql.connector, o comando usado para executar um comando no banco de dados MySQL é:

- ☐ mycursor.readlines("nome-da-tabela-desejado")
- ☐ csv.reader(con, query)
- ☐ print(query)
- ☒ mycursor.execute("comando-desejado")

### Pergunta 7

2,5 / 2,5 pts

Na linguagem Python, usando o pacote mysql.connector, é possível inserir múltiplas linhas em uma tabela por um só comando. Qual comando é este?

- ☐ Insert.many(query, values)
- ☐ dbWriteTable(query, values)
- ☐ mycursor.execute(query, values)

☒ mycursor.executemany(query, values)

### Pergunta 8

2,5 / 2,5 pts

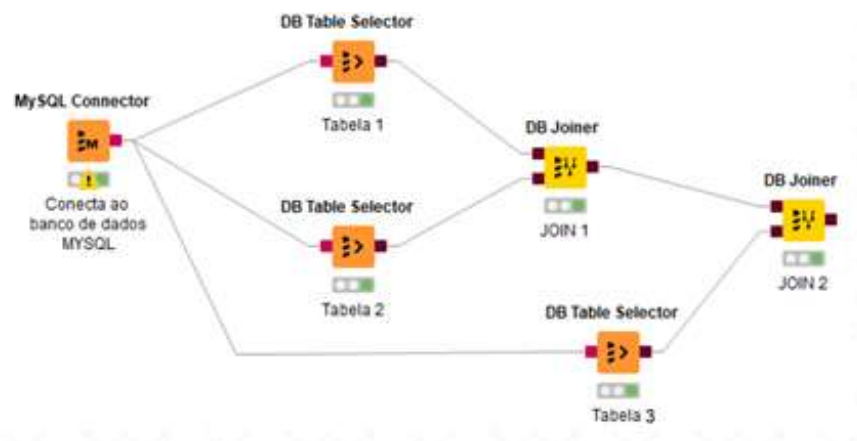
Qual dos comandos abaixo é usado para ler um arquivo texto não tabulado na linguagem R?

- ☒ read.delim(file)
- ☐ read.csv(file, sep=',')
- ☐ csv.reader(file, delimiter=',')
- ☐ read\_xlsx(file, header=TRUE)

### Pergunta 9

2,5 / 2,5 pts

Considere o workflow abaixo e a letra “C” da atividade 4. Marque a opção INCORRETA:



- ☐ O Join 1 pode ser entre as tabelas *caracteristicasgerais* e *caracteristicasimovel* ou *caracteristicasimovel* e *imóvel*.



As tabelas 1, 2 e 3 podem ser respectivamente as tabelas *caracteristicasgerais*, *imóvel* e *caracteristicasimovel*.



Se o Join 1 for entre as tabelas *caracteristicasgerais* e *caracteristicasimovel* , o Join 2 deve ser entre *caracteristicasimovel* e *imóvel*.

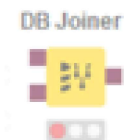


As tabelas 1, 2 e 3 podem ser respectivamente as tabelas *caracteristicasgerais*, *caracteristicasimovel* e *imóvel*.

### Pergunta 10

2,5 / 2,5 pts

Considere a necessidade de criar um workflow na plataforma Knime, para fazer uma consulta que exiba o nome da cidade e o nome seu respectivo estado. Qual nó NÃO é necessário neste workflow?



Pontuação do teste: **25** de 25