

## BANCO DE DADOS

### Trabalho – Relatório

<b>Curso:</b>	Análise e Desenvolvimento de sistemas
<b>Aluno(a):</b>	Antonio Dias Reducino
<b>RU:</b>	4654528

#### 1. 1ª Etapa – Modelagem

**Pontuação:** 25 pontos.

Dadas as regras de negócio abaixo listadas, referentes ao estudo de caso de uma Clínica Médica, elabore o Modelo Entidade-Relacionamento (MER), isto é, o modelo conceitual.

O Modelo Entidade-Relacionamento (MER) deve contemplar os seguintes itens:

- Entidades;
- Atributos;
- Relacionamentos;
- Cardinalidades;
- Chaves primárias;
- Chaves estrangeiras.

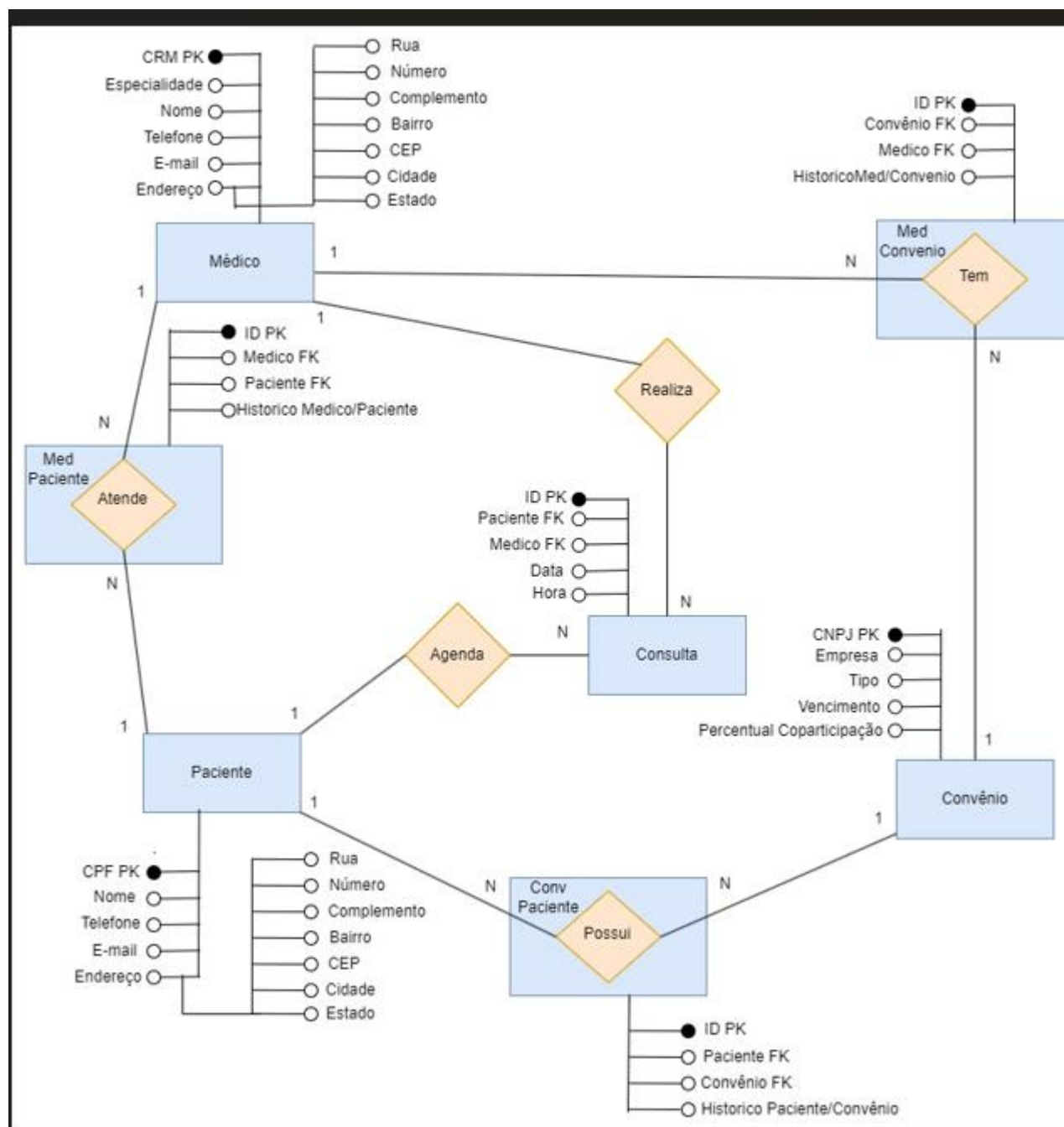
Uma Clínica Médica necessita controlar os dados das consultas realizadas. Para isso, contratou um profissional de Banco de Dados, a fim de modelar o Banco de Dados que armazenará os dados das consultas.

As regras de negócio são:

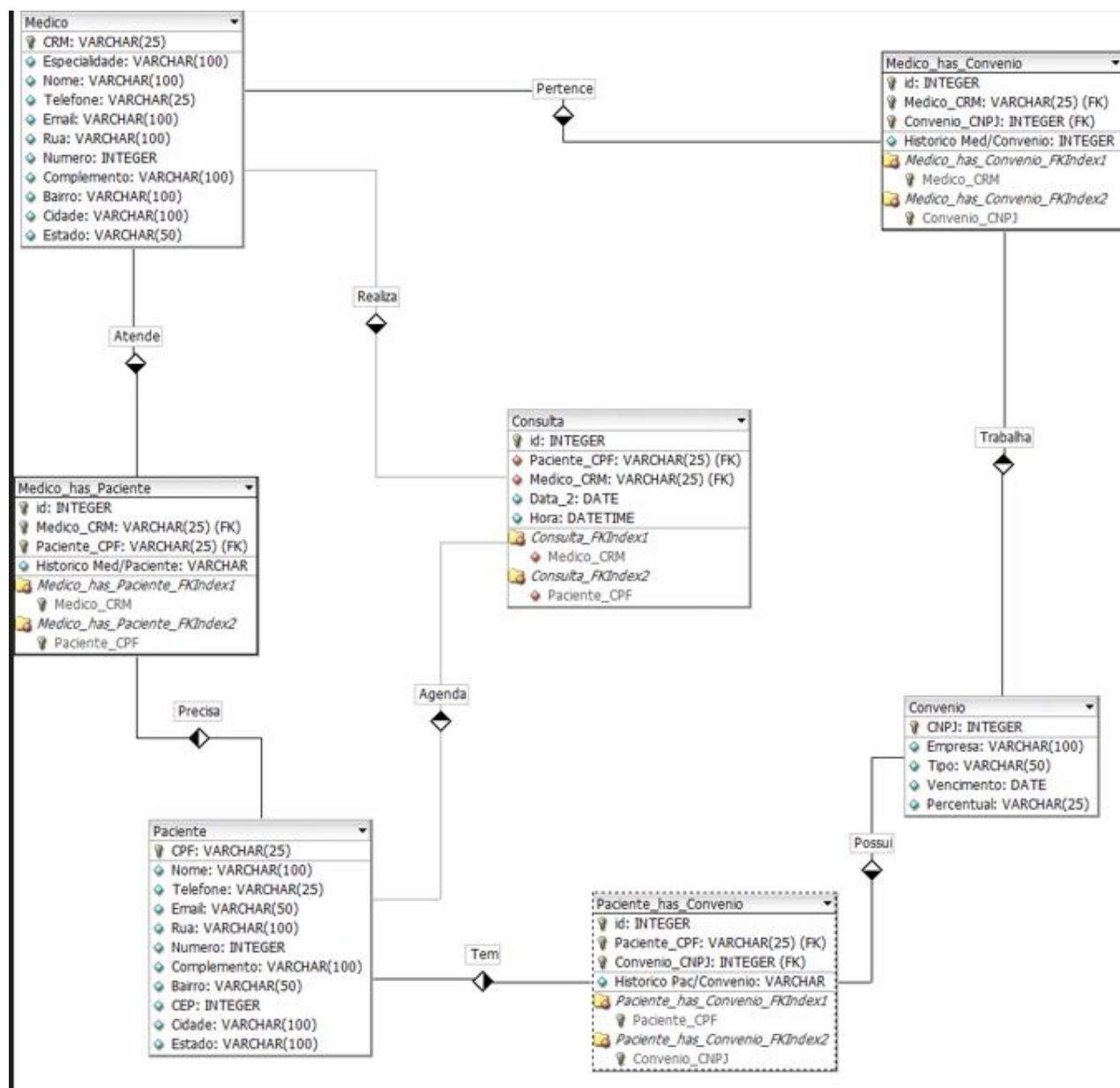
- Médico – Deverão ser armazenados os seguintes dados: CRM, especialidade, nome, telefone, e-mail e endereço, sendo o endereço composto por rua, número, complemento, bairro, CEP, cidade e estado;

- Consulta – Deverão ser armazenados os seguintes dados: identificação da consulta, data e horário;
- Paciente – Deverão ser armazenados os seguintes dados: CPF, nome, telefone, e-mail e endereço, sendo o endereço composto por rua, número, complemento, bairro, CEP, cidade e estado;
- Convênio – Deverão ser armazenados os seguintes dados: identificação do convênio, empresa, tipo, vencimento e percentual de coparticipação;
- Um médico pode realizar zero ou várias consultas, assim como zero ou várias consultas podem ser realizadas por um médico;
- Um médico pode atender zero ou vários convênios, assim como zero ou vários convênios podem ser atendidos por um médico;
- Um paciente pode marcar zero ou várias consultas, assim como zero ou várias consultas podem ser marcadas por um paciente;
- Um paciente pode possuir zero ou vários convênios, assim como zero ou vários convênios podem pertencer a um paciente.

MER:

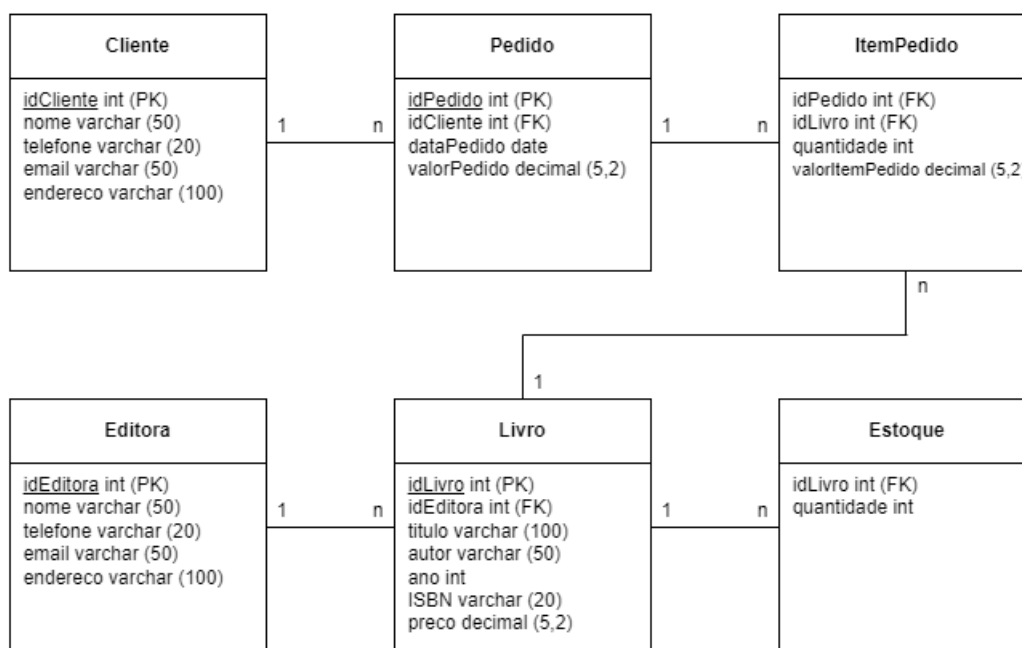


Relacional:



## 2. 2ª Etapa – Implementação

Considere o seguinte Modelo Relacional (modelo lógico), referente ao estudo de caso de uma Livraria:

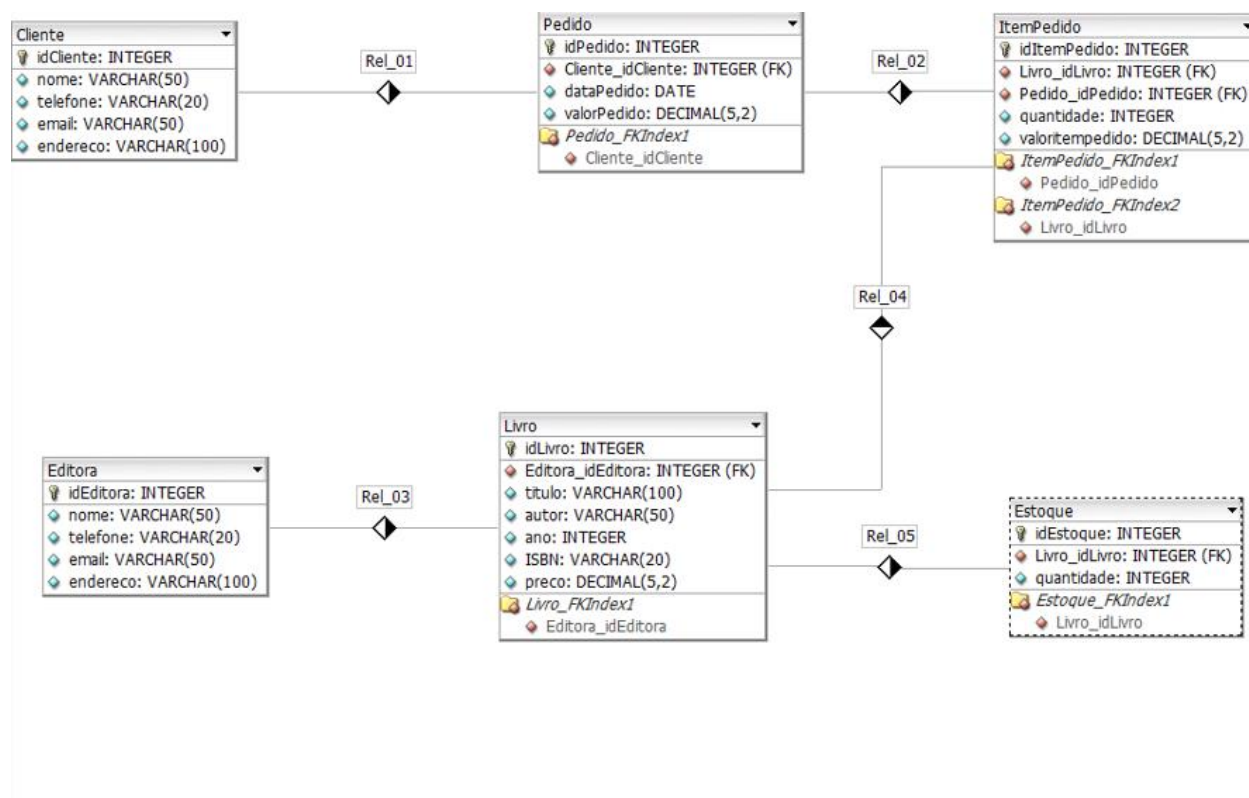


Com base no Modelo Relacional dado e utilizando a *Structured Query Language* (SQL), no MySQL Workbench, implemente o que se pede.

**Observação:** Para testar o Banco de Dados após a implementação, utilize os comandos contidos no arquivo “Trabalho – Populando o Banco de Dados” para popular as tabelas. Tal arquivo contém todos os comandos de inserção dos dados (fictícios) necessários para a realização dos testes.

**Pontuação:** 25 pontos.

1. Implemente um Banco de Dados chamado “Livraria”. Após, implemente as tabelas, conforme o Modelo Relacional dado, observando as chaves primárias e as chaves estrangeiras. Todos os campos, de todas as tabelas, não podem ser nulos (*not null*).



-- Criando o Banco de dados --

Create database Livraria;

Use livraria;

```
CREATE TABLE Cliente (  
    idCliente INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(50) NOT NULL,  
    telefone VARCHAR(20) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    endereco VARCHAR(100) NOT NULL,  
    PRIMARY KEY(idCliente)  
);
```

```
CREATE TABLE Editora (  
    idEditora INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(50) NOT NULL,  
    telefone VARCHAR(20) NOT NULL,  
    email VARCHAR(50) NOT NULL,
```

```
endereco VARCHAR(100) NOT NULL,  
PRIMARY KEY(idEditora)  
);
```

```
CREATE TABLE Estoque (  
    idEstoque INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    Livro_idLivro INTEGER UNSIGNED NOT NULL,  
    quantidade INTEGER UNSIGNED NOT NULL,  
    PRIMARY KEY(idEstoque),  
    INDEX Estoque_FKIndex1(Livro_idLivro)  
);
```

```
CREATE TABLE ItemPedido (  
    idItemPedido INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    Livro_idLivro INTEGER UNSIGNED NOT NULL,  
    Pedido_idPedido INTEGER UNSIGNED NOT NULL,  
    quantidade INTEGER UNSIGNED NOT NULL,  
    valoritempedido DECIMAL(5,2) NOT NULL,  
    PRIMARY KEY(idItemPedido),  
    INDEX ItemPedido_FKIndex1(Pedido_idPedido),  
    INDEX ItemPedido_FKIndex2(Livro_idLivro)  
);
```

```
CREATE TABLE Livro (  
    idLivro INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    Editora_idEditora INTEGER UNSIGNED NOT NULL,  
    titulo VARCHAR(100) NOT NULL,  
    autor VARCHAR(50) NOT NULL,  
    ano INTEGER UNSIGNED NOT NULL,  
    ISBN VARCHAR(20) NOT NULL,  
    preco DECIMAL(5,2) NOT NULL,  
    PRIMARY KEY(idLivro),  
    INDEX Livro_FKIndex1(Editora_idEditora)
```

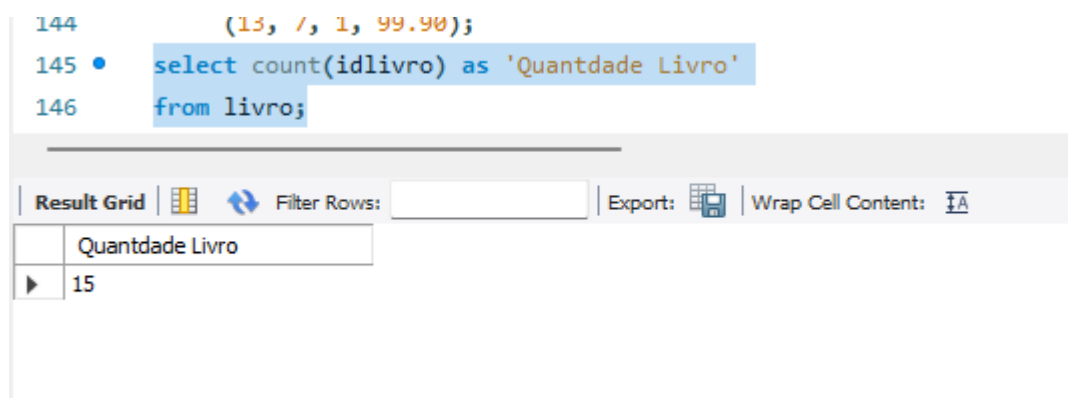
);

```
CREATE TABLE Pedido (  
    idPedido INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    Cliente_idCliente INTEGER UNSIGNED NOT NULL,  
    dataPedido DATE NOT NULL,  
    valorPedido DECIMAL(5,2) NOT NULL,  
    PRIMARY KEY(idPedido),  
    INDEX Pedido_FKIndex1(Cliente_idCliente)  
);
```

**Pontuação:** 10 pontos.

2. Implemente uma consulta para listar o quantitativo de livros cadastrados, independentemente da editora.

```
select count(idlivro) as 'Quantdade Livro'  
from livro;
```



The screenshot shows a SQL query editor with the following code:

```
144      (13, /, 1, 99.90);  
145 • select count(idlivro) as 'Quantdade Livro'  
146 from livro;
```

Below the editor is a toolbar with options: Result Grid, Filter Rows, Export, and Wrap Cell Content. Below the toolbar is a result grid showing the output of the query:

Quantdade Livro
15



**Pontuação:** 10 pontos.

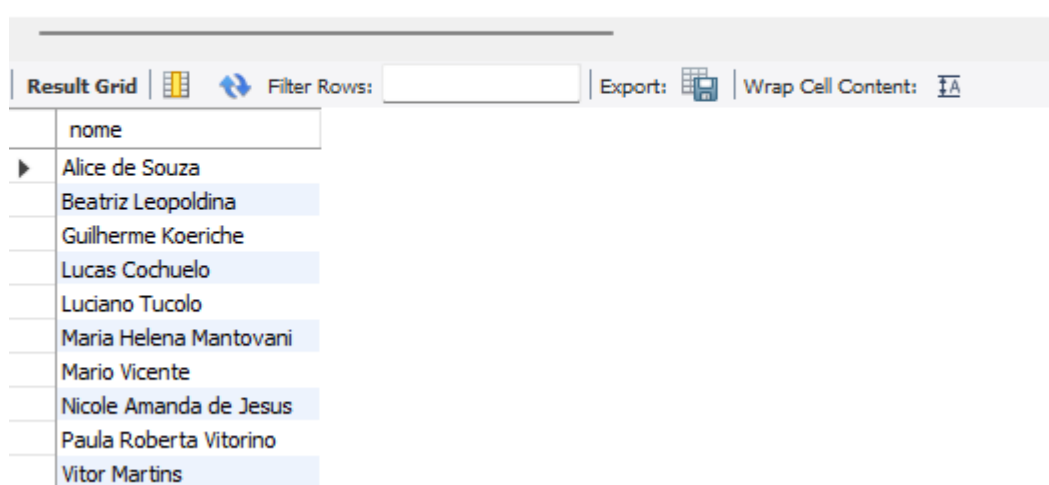
3. Implemente uma consulta para listar o nome dos clientes cadastrados. A listagem deve ser mostrada em ordem crescente.

`select nome from Cliente order by nome asc;`

148 • `select nome from Cliente order by nome asc;`

149

150



nome
Alice de Souza
Beatriz Leopoldina
Guilherme Koeriche
Lucas Cochuelo
Luciano Tucolo
Maria Helena Mantovani
Mario Vicente
Nicole Amanda de Jesus
Paula Roberta Vitorino
Vitor Martins

**Pontuação:** 10 pontos.

4. Implemente uma consulta para listar o nome de todas as editoras e os títulos de seus respectivos livros. A listagem deve ser mostrada em ordem decrescente pelo nome das editoras.

`select Editora.nome as NomeEditora, Livro.titulo as TituloLivro  
from Editora  
join Livro on Editora.idEditora = Livro.Editora_idEditora  
order by Editora.nome desc;`

```
150 • select Editora.nome as NomeEditora, Livro.titulo as TituloLivro
151 from Editora
152 join Livro on Editora.idEditora = Livro.Editora_idEditora
153 order by Editora.nome desc;
154
155
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	NomeEditora	TituloLivro		
▶	Sexta dos Estudos	Matemática Descomplicada		
	Sexta dos Estudos	Português para Estrangeiros		
	Sexta dos Estudos	Estatística é para Todos		
	Mais Informática	Aprendendo Python em 24 Horas		
	Mais Informática	Banco de Dados: Aprendendo Python em 24 Horas		
	Mais Informática	Java para Que Te Quero		
	Companhia da Leitura	Minha Faculdade Vai Me Enlouquecer		
	Companhia da Leitura	Controlando as Emoções		
	Companhia da Leitura	Brasil Brasileiro		
	Ciência da Informação	Big Data: Conhecimentos Essenciais		
	Ciência da Informação	Ciência de Dados: O Futuro		
	Ciência da Informação	Inteligência Artificial Aplicada a Dados		
	Arco da Velha	Se Eu Voltasse no Passado		
	Arco da Velha	Penso, Logo Existo		
	Arco da Velha	Lugares para Viajar Sozinho		

**Pontuação:** 10 pontos.

5. Implemente uma consulta para listar o nome das editoras e a média de preço de seus respectivos livros. Para isso, utilize o comando *group by*.

```
select Editora.nome as NomeEditora, round(avg(Livro.preco), 2) as MediaPreco
from Editora
join Livro on Editora.idEditora = Livro.Editora_idEditora
group by Editora.nome;
```

```
156 • select Editora.nome as NomeEditora, round(avg(Livro.preco), 2) as MediaPreco
157 from Editora
158 join Livro on Editora.idEditora = Livro.Editora_idEditora
159 group by Editora.nome;
160
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
NomeEditora	MediaPreco		
Companhia da Leitura	38.57		
Arco da Velha	56.23		
Mais Informática	106.57		
Sexta dos Estudos	76.57		
Ciência da Informação	147.23		

**Pontuação:** 10 pontos.

6. Implemente uma consulta para listar o nome de todos os clientes e a quantidade de livros comprados pelos mesmos. Para isso, utilize o comando *group by*.

```
select Cliente.nome as NomeCliente, COUNT(ItemPedido.Livro_idLivro) as
QuantidadeLivrosComprados
from Cliente
join Pedido on Cliente.idCliente = Pedido.Cliente_idCliente
join ItemPedido on Pedido.idPedido = ItemPedido.Pedido_idPedido
group by Cliente.idCliente;
```

```
163 • select Cliente.nome as NomeCliente, COUNT(ItemPedido.Livro_idLivro) as QuantidadeLivrosComprados
164 from Cliente
165 join Pedido on Cliente.idCliente = Pedido.Cliente_idCliente
166 join ItemPedido on Pedido.idPedido = ItemPedido.Pedido_idPedido
167 group by Cliente.idCliente;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
NomeCliente	QuantidadeLivrosComprados		
Alice de Souza	1		
Mario Vicente	1		
Maria Helena Mantovani	3		
Vitor Martins	1		
Nicole Amanda de Jesus	1		
Luciano Tucolo	3		
Paula Roberta Vitorino	2		
Guilherme Koeriche	1		
Beatriz Leopoldina	2		
Lucas Cochuelo	2		