

BANCO DE DADOS

Trabalho – Relatório

Curso:	Análise e Desenvolvimento de Sistemas
Aluno(a):	Rodrigo P Alcantara
RU:	4844626

• 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:

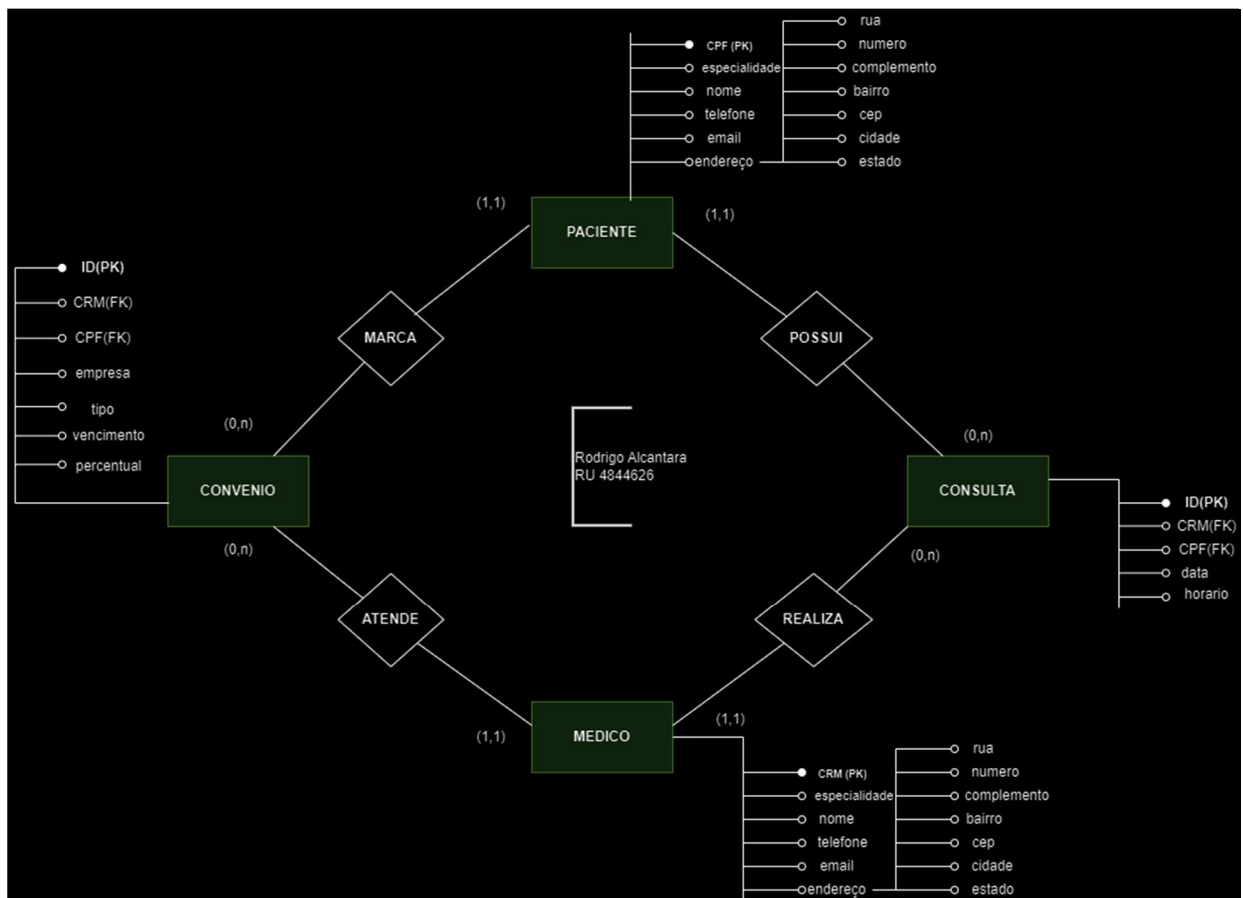
- Entidades;
- Atributos;
- Relacionamentos;
- Cardinalidades;
- Chaves primárias;
- Chaves estrangeiras.

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:

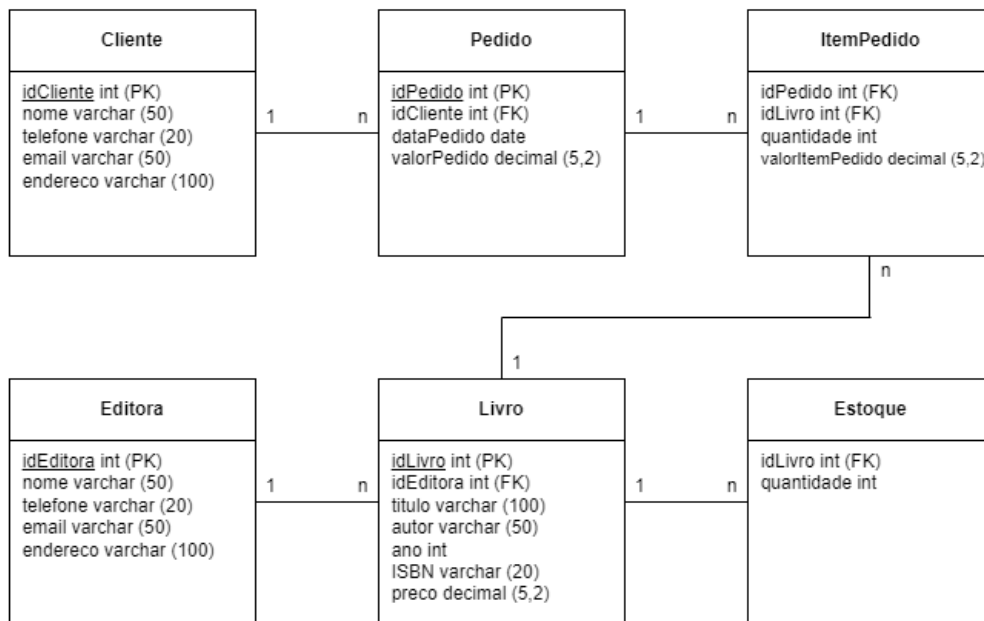
- Médico – Deverão ser armazenados os seguintes dados: CRM, especialidade, nome, telefone, e-mail e endereço, sendo o endereço composto por rua, número, complemento, bairro, CEP, cidade e estado;

- 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.



- **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.

- 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*).

```
create database Livraria;
```

```
use Livraria;
```

```
create table Cliente (
```

```
    idCliente int not null,  
    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 int not null,  
    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 Livro (
```

```
    idLivro int not null,  
    idEditora int not null,  
    titulo varchar (100) not null,  
    autor varchar (50) not null,  
    ano int not null,  
    ISBN varchar (20) not null,  
    preco decimal (5,2) not null,  
    primary key (idLivro),  
    foreign key (idEditora) references Editora (idEditora)
```

```
);
```

```
create table Estoque (
```

```
    idLivro int not null,  
    quantidade int not null,  
    foreign key (idLivro) references Livro (idLivro)
```

```
);
```

```

create table Pedido (
    idPedido int not null,
    idCliente int not null,
    dataPedido date not null,
    valorPedido decimal (5,2) not null,
    primary key (idPedido),
    foreign key (idCliente) references Cliente (idCliente)
);

create table itemPedido (
    quantidade int not null,
    idLivro int not null,
    idPedido int not null,
    valorItemPedido decimal (5,2) not null,
    foreign key (idPedido) references Pedido (idPedido),
    foreign key (idLivro) references Livro (idLivro)
);

```

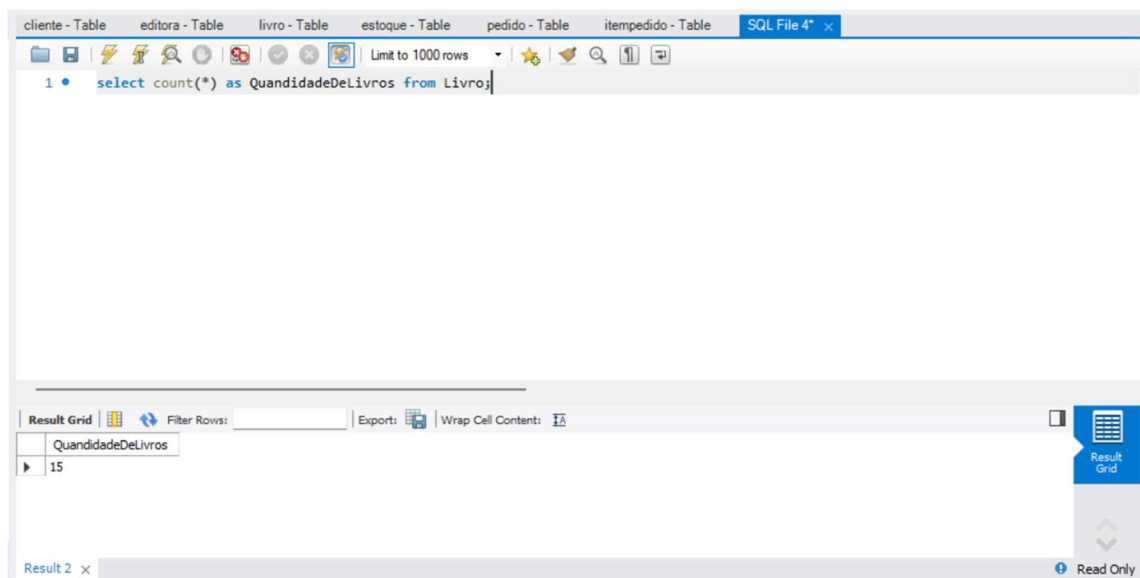
Pontuação: 10 pontos.

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

```

SELECT COUNT(*) AS QuantidadeDeLivros
FROM Livro;

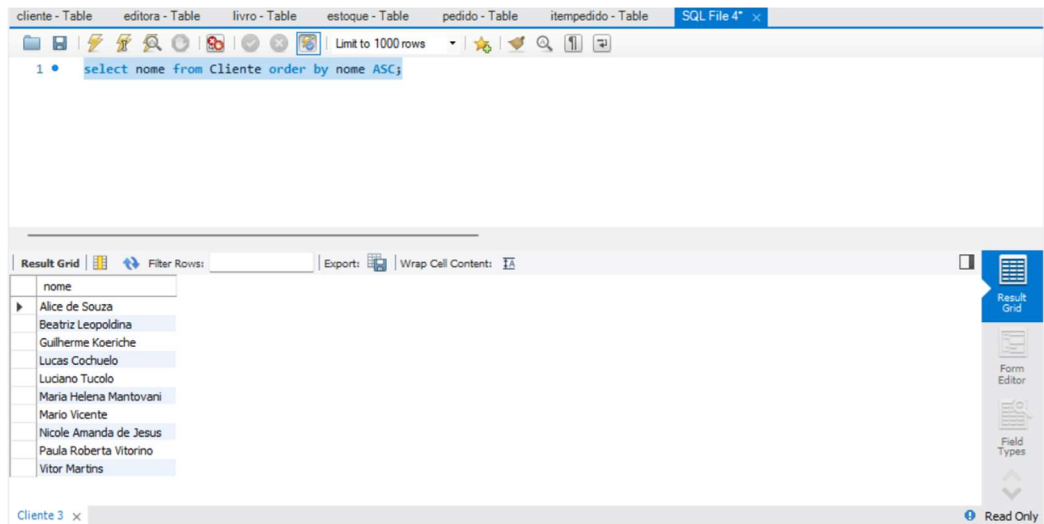
```



Pontuação: 10 pontos.

- 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;
```



The screenshot shows a database management interface with a SQL editor at the top containing the query: `select nome from Cliente order by nome ASC;`. Below the editor, the 'Result Grid' displays the results of the query. The results are as follows:

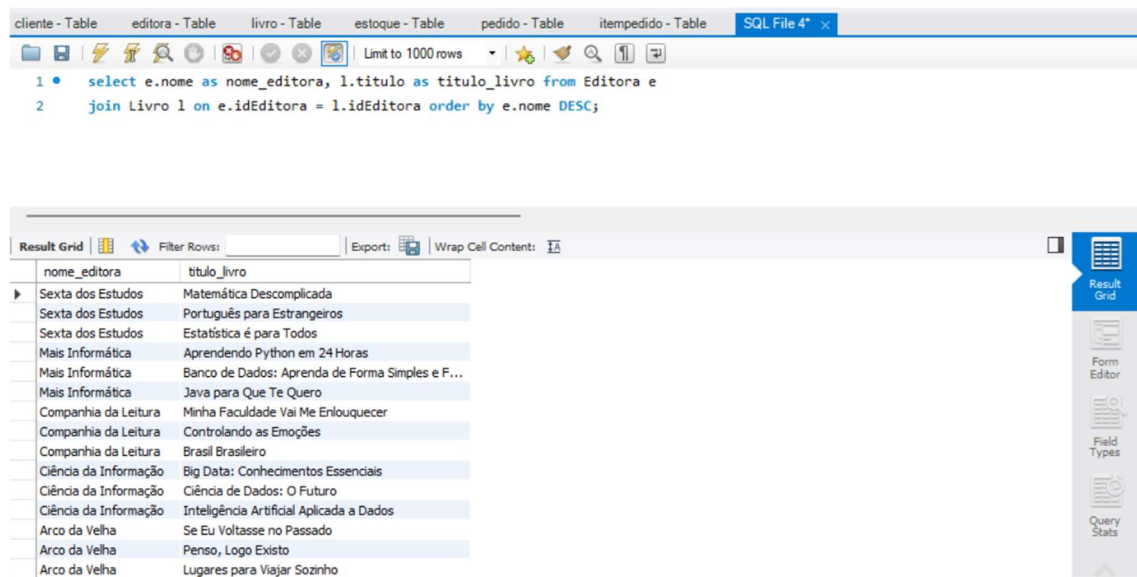
nome
Alice de Souza
Beatriz Leopoldina
Guilherme Koerich
Lucas Cochuelo
Luciano Tucolo
Maria Helena Mantovani
Mario Vicente
Nicole Amanda de Jesus
Paula Roberta Vitorino
Vitor Martins

The interface also includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Read Only' status indicator at the bottom right.

Pontuação: 10 pontos.

- 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 e.nome AS nome_editora, l.titulo AS titulo_livro
FROM Editora e
JOIN Livro l ON e.idEditora = l.idEditora
ORDER BY e.nome DESC;
```



The screenshot shows a database management tool interface. At the top, there are tabs for various tables: cliente - Table, editora - Table, livro - Table, estoque - Table, pedido - Table, itempedido - Table, and a SQL File 4*. Below the tabs is a toolbar with icons for file operations, a search icon, and a limit to 1000 rows. The main area displays a SQL query:

```
1 • select e.nome as nome_editora, l.titulo as titulo_livro from Editora e
2   join Livro l on e.idEditora = l.idEditora order by e.nome DESC;
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with two columns: 'nome_editora' and 'titulo_livro'. The table contains 15 rows of data, sorted by editor name in descending order. On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

nome_editora	titulo_livro
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: Aprenda de Forma Simples e F...
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.

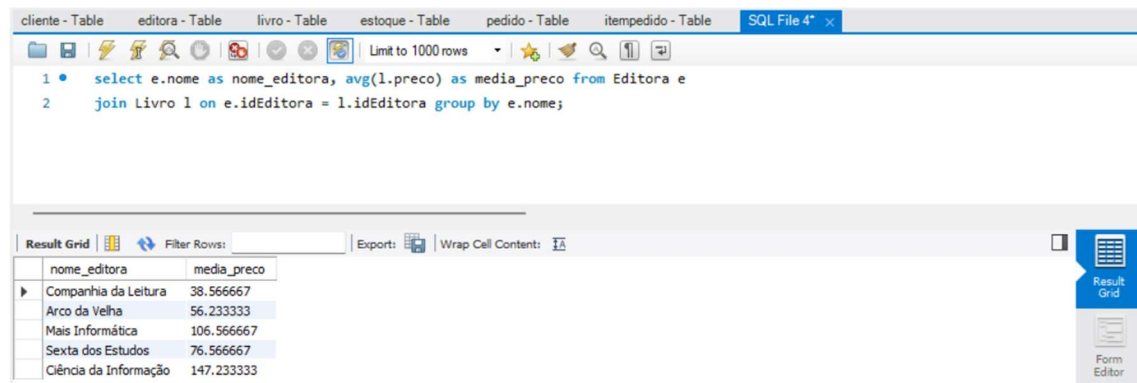
- Implemente uma consulta para listar o nome das editoras e a média de preço de seus respectivos livros. Para isso, utilize o comando *group by*.

```
SELECT e.nome AS nome_editora, AVG(l.preco) AS media_preco
```

```
FROM Editora e
```

```
JOIN Livro l ON e.idEditora = l.idEditora
```

```
GROUP BY e.nome;
```



The screenshot shows a database management interface with a tab labeled 'SQL File 4*'. The SQL query entered is:

```
1 • select e.nome as nome_editora, avg(l.preco) as media_preco from Editora e
2 join Livro l on e.idEditora = l.idEditora group by e.nome;
```

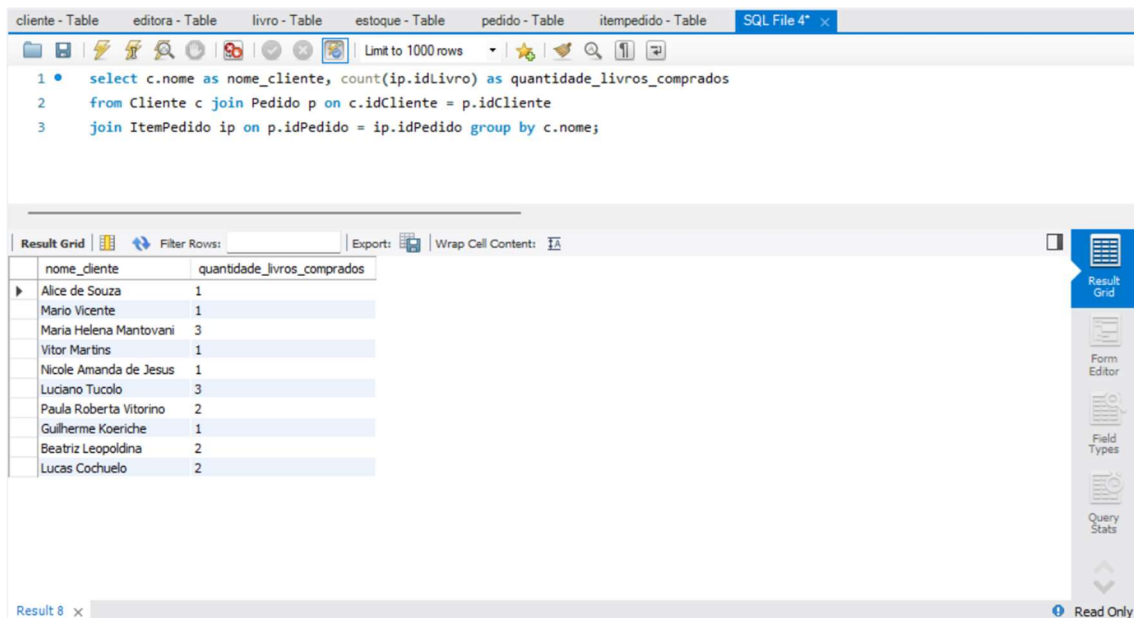
Below the query, the 'Result Grid' displays the following data:

nome_editora	media_preco
Companhia da Leitura	38.566667
Arco da Velha	56.233333
Mais Informática	106.566667
Sexta dos Estudos	76.566667
Ciência da Informação	147.233333

Pontuação: 10 pontos.

- 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      c.nome      AS      nome_cliente,      COUNT(ip.idLivro)      AS  
quantidade_livros_comprados  
FROM Cliente c  
JOIN Pedido p ON c.idCliente = p.idCliente  
JOIN ItemPedido ip ON p.idPedido = ip.idPedido  
GROUP BY c.nome;
```



The screenshot shows a database management tool interface. At the top, there are tabs for 'cliente - Table', 'editora - Table', 'livro - Table', 'estoque - Table', 'pedido - Table', and 'itempedido - Table'. The active tab is 'SQL File 4*'. Below the tabs is a toolbar with various icons. The main area displays an SQL query:

```
1 • select c.nome as nome_cliente, count(ip.idLivro) as quantidade_livros_comprados  
2 from Cliente c join Pedido p on c.idCliente = p.idCliente  
3 join ItemPedido ip on p.idPedido = ip.idPedido group by c.nome;
```

Below the query, there is a 'Result Grid' section. It shows a table with two columns: 'nome_cliente' and 'quantidade_livros_comprados'. The table contains 10 rows of data. On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom left, it says 'Result 8 x' and at the bottom right, it says 'Read Only'.

nome_cliente	quantidade_livros_comprados
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 Koerich	1
Beatriz Leopoldina	2
Lucas Cochuelo	2