

BANCO DE DADOS

Trabalho – Relatório

Curso:	Análise e desenvolvimento de sistemas
Aluno(a):	Ellen Cristina Alves Soares
RU:	4498919

1. 1ª Etapa – Modelagem

Pontuação: 30 pontos.

Dadas as regras de negócio abaixo listadas, referentes ao estudo de caso de uma Rede de Hotéis, 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 Rede de Hotéis necessita controlar os dados dos funcionários, das unidades, dos quartos, dos hóspedes, das reservas e dos pagamentos. Para isso, contratou um profissional de Banco de Dados, a fim de modelar o Banco de Dados que armazenará todos os dados.

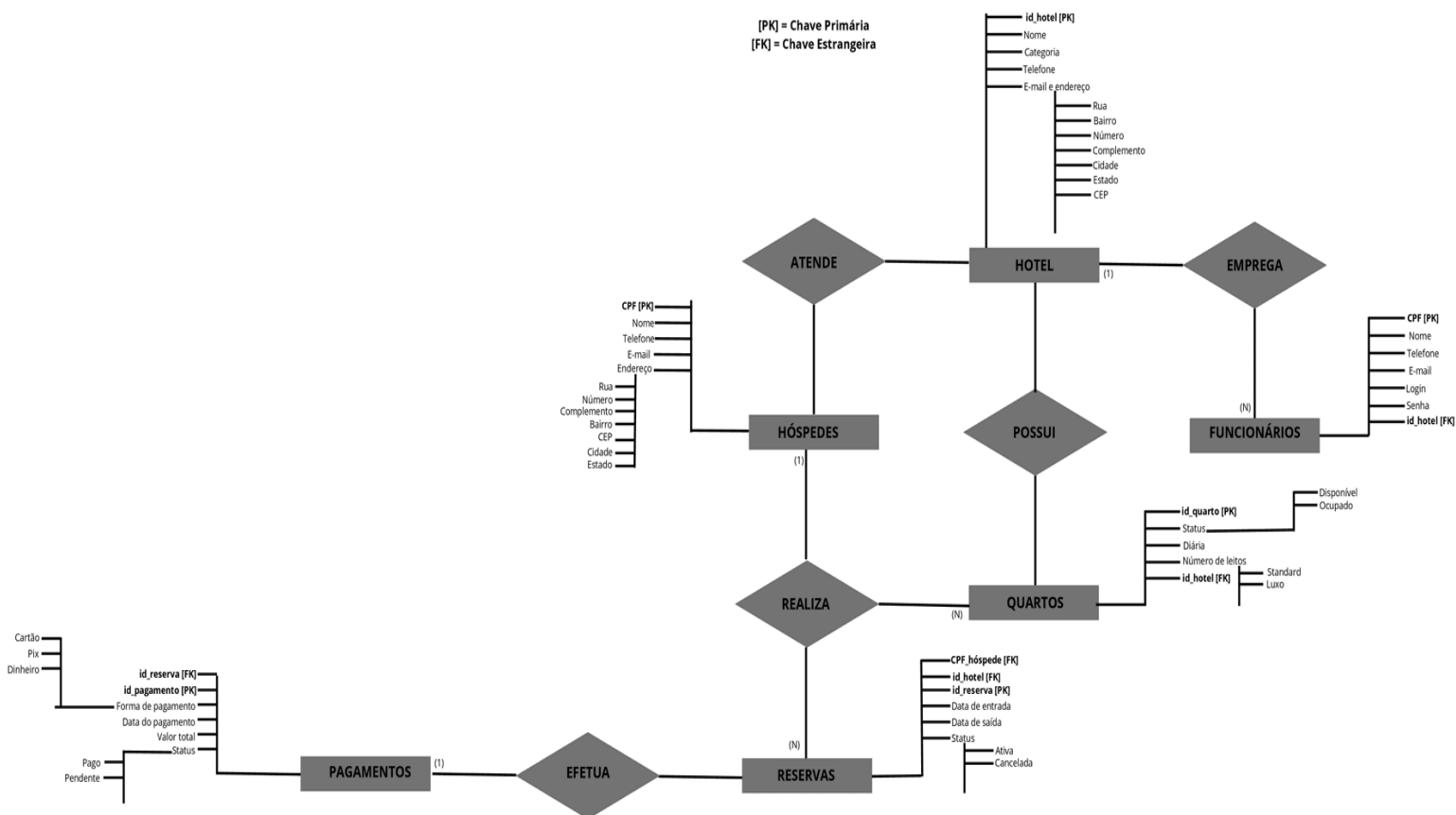
As regras de negócio são:

- Funcionário – Deverão ser armazenados os seguintes dados: CPF, nome, telefone, e-mail, login e senha;

- Hotel – Deverão ser armazenados os seguintes dados: identificação do hotel, nome, categoria, telefone, e-mail e endereço, sendo o endereço composto por rua, número, complemento, bairro, CEP, cidade e estado;
- Quarto – Deverão ser armazenados os seguintes dados: identificação do quarto, número de leitos, tipo (*standard*, luxo ou suíte), preço da diária e *status* (disponível, ocupado ou manutenção);
- Hóspede – 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;
- Reserva – Deverão ser armazenados os seguintes dados: identificação da reserva, data de entrada, data de saída e *status* (ativa, cancelada ou concluída);
- Pagamento – Deverão ser armazenados os seguintes dados: identificação do pagamento, forma de pagamento (cartão, pix ou dinheiro), data do pagamento, valor total e *status* (pago ou pendente);
- Um hotel possui um ou vários quartos;
- Um ou vários funcionários trabalham em um hotel;
- Um funcionário realiza uma ou várias reservas;
- Um ou vários quartos fazem parte de uma ou várias reservas;
- Um hóspede pode fazer uma ou várias reservas;
- Uma reserva gera um pagamento.

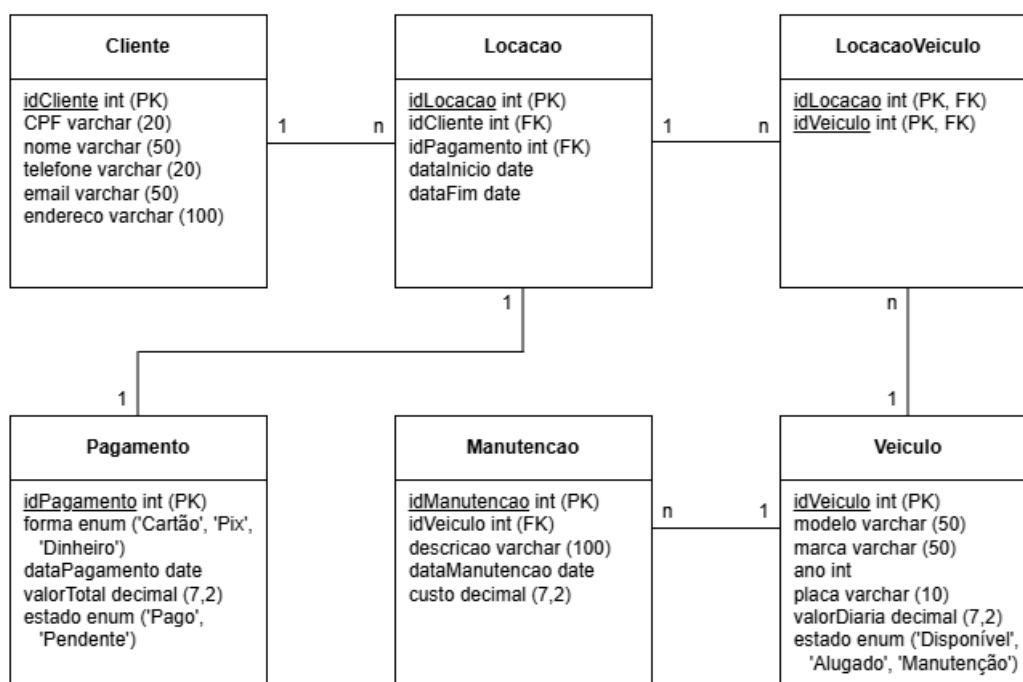
Importante:

- O Modelo Entidade-Relacionamento (MER) deve considerar somente as regras de negócio dadas, não podendo ser criada nenhuma outra entidade ou atributo que não estejam nas regras de negócio;
- Em caso de haver entidade associativa, a mesma deve ser representada pela “Representação 1” (texto da Aula 1 – Fundamentos de Banco de Dados, Figura 25);
- Em caso de haver cardinalidade (1,1), a chave estrangeira deve fazer parte da entidade que possui o maior número de chaves estrangeiras.



2. 2ª Etapa – Implementação

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



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

Importante: 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: 30 pontos.

1. Implemente um Banco de Dados chamado “LocadoraVeiculos”. 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 LocadoraVeiculos;
USE LocadoraVeiculos;

-- TABELA CLIENTE
CREATE TABLE Cliente (
    idCliente INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    telefone VARCHAR(20) NOT NULL,
    email VARCHAR(100) NOT NULL,
    cpf VARCHAR(20) NOT NULL,
    endereco VARCHAR(100) NOT NULL
);

-- TABELA VEICULO
CREATE TABLE Veiculo (
    idVeiculo INT PRIMARY KEY AUTO_INCREMENT,
    modelo VARCHAR(50) NOT NULL,
    marca VARCHAR(50) NOT NULL,
    ano INT NOT NULL,
    placa VARCHAR(10) NOT NULL,
    valorDiaria DECIMAL(7,2) NOT NULL,
    estado ENUM('Disponível', 'Alugado', 'Manutenção') NOT NULL
);

-- TABELA PAGAMENTO
CREATE TABLE Pagamento (
    idPagamento INT PRIMARY KEY AUTO_INCREMENT,
    forma ENUM('Cartão', 'Pix', 'Dinheiro') NOT NULL,
    dataPagamento DATE NOT NULL,
    valorTotal DECIMAL(7,2) NOT NULL,
    estado ENUM('Pago', 'Pendente') NOT NULL
);

-- TABELA LOCACAO
```

```
CREATE TABLE Locacao (  
    idLocacao INT PRIMARY KEY AUTO_INCREMENT,  
    idCliente INT NOT NULL,  
    idPagamento INT NOT NULL,  
    dataInicio DATE NOT NULL,  
    dataFim DATE NOT NULL,  
    FOREIGN KEY (idCliente) REFERENCES Cliente(idCliente),  
    FOREIGN KEY (idPagamento) REFERENCES Pagamento(idPagamento)  
);  
  
-- TABELA LOCACAO VEICULO (TABELA ASSOCIATIVA)  
CREATE TABLE LocacaoVeiculo (  
    idLocacao INT NOT NULL,  
    idVeiculo INT NOT NULL,  
    PRIMARY KEY (idLocacao, idVeiculo),  
    FOREIGN KEY (idLocacao) REFERENCES Locacao(idLocacao),  
    FOREIGN KEY (idVeiculo) REFERENCES Veiculo(idVeiculo)  
);  
  
-- TABELA MANUTENCAO  
CREATE TABLE Manutencao (  
    idManutencao INT PRIMARY KEY AUTO_INCREMENT,  
    idVeiculo INT NOT NULL,  
    descricao VARCHAR(100) NOT NULL,  
    dataManutencao DATE NOT NULL,  
    custo DECIMAL(7,2) NOT NULL,  
    FOREIGN KEY (idVeiculo) REFERENCES Veiculo(idVeiculo)  
);  
  
INSERT INTO Cliente (idCliente, cpf, nome, telefone, email, endereco)  
VALUES  
    (1, '045.159.785-10', 'João da Silva', '(41) 99854-5648',  
    'joao.s@email.com.br', 'Rua das Flores, 3578, Curitiba - PR'),  
    (2, '147.963.789-02', 'Maria de Souza', '(21) 99763-1213',  
    'maria.s@email.com.br', 'Rua Vicente Machado, 10503, apto. 1007, Rio de  
    Janeiro - RJ'),
```

```
(3, '357.456.309-07', 'Carlos Mendes', '(85) 98754-2050',  
'carlos.m@email.com.br', 'Rua Osvaldo Cruz, 578, Fortaleza - CE'),  
(4, '569.428.308.12', 'Ana Oliveira', '(11) 99527-3721',  
'ana.o@email.com.br', 'Avenida Ipiranga, 1246, apto. 201, São Paulo -  
SP'),  
(5, '096.429.761-05', 'Pedro dos Santos', '(92) 98418-3141',  
'pedro.s@email.com.br', 'Rua Venezuela, 649, Manaus - AM'),  
(6, '548.302.980-13', 'Fernanda Lima', '(51) 99234-5458',  
'fernanda.l@email.com.br', 'Avenida Uruguai, 3152, apto. 2202, Porto  
Alegre - RS'),  
(7, '947.652.415-09', 'Ricardo Alves', '(65) 98953-7828',  
'ricardo.a@email.com.br', 'Rua dos Açudes, 1029, Cuiabá - MT'),  
(8, '012.359.678-15', 'Juliana Costa', '(63) 99315-6264',  
'juliana.c@email.com.br', 'Avenida Brasil, 953, apto. 709, Palmas - TO'),  
(9, '254.635.485-05', 'Lucas Martins', '(71) 99264-3585',  
'lucas.m@email.com.br', 'Rua dos Baianos, 12549, Salvador - BA'),  
(10, '785.126.497-72', 'Beatriz Rocha', '(69) 98767-1545',  
'beatriz.r@email.com.br', 'Rua Independente, 209, Porto Velho - RO');  
  
INSERT INTO Veiculo (idVeiculo, modelo, marca, ano, placa, valorDiaria,  
estado) VALUES  
(1, 'Onix', 'Chevrolet', 2021, 'ABC-1D34', 150.00, 'Disponível'),  
(2, 'HB20', 'Hyundai', 2022, 'XYZ-5E78', 180.00, 'Alugado'),  
(3, 'Gol', 'Volkswagen', 2020, 'DEF-9W12', 140.00, 'Alugado'),  
(4, 'Civic', 'Honda', 2023, 'GHI-3F56', 250.00, 'Disponível'),  
(5, 'Corolla', 'Toyota', 2022, 'JKL-7I90', 240.00, 'Alugado'),  
(6, 'Fiesta', 'Ford', 2019, 'MNO-2P45', 130.00, 'Manutenção'),  
(7, 'Cruze', 'Chevrolet', 2021, 'PQR-6A89', 200.00, 'Disponível'),  
(8, 'Duster', 'Renault', 2022, 'STU-1Z34', 220.00, 'Alugado'),  
(9, 'Toro', 'Fiat', 2023, 'VWX-5Q78', 260.00, 'Manutenção'),  
(10, 'Compass', 'Jeep', 2024, 'YZA-9S12', 280.00, 'Disponível');  
  
INSERT INTO Pagamento (idPagamento, forma, dataPagamento, valorTotal,  
estado) VALUES  
(1, 'Cartão', '2024-12-01', 360.00, 'Pago'),  
(2, 'Cartão', '2024-12-01', 140.00, 'Pago'),  
(3, 'Pix', '2024-12-05', 480.00, 'Pago'),  
(4, 'Cartão', '2024-12-10', 1100.00, 'Pago'),  
(5, 'Dinheiro', '2024-12-11', 1350.00, 'Pago'),
```

```
(6, 'Cartão', '2024-12-15', 2210.00, 'Pago'),  
(7, 'Pix', '2024-12-20', 1080.00, 'Pago'),  
(8, 'Pix', '2024-12-23', 2200.00, 'Pago'),  
(9, 'Cartão', '2024-12-26', 500.00, 'Pago'),  
(10, 'Pix', '2024-12-28', 840.00, 'Pago'),  
(11, 'Pix', '2025-01-02', 520.00, 'Pago'),  
(12, 'Dinheiro', '2025-01-05', 900.00, 'Pago'),  
(13, 'Cartão', '2025-01-05', 260.00, 'Pago'),  
(14, 'Cartão', '2025-01-09', 1120.00, 'Pago'),  
(15, 'Pix', '2025-01-12', 600.00, 'Pago'),  
(16, 'Cartão', '2025-01-13', 1680.00, 'Pendente'),  
(17, 'Pix', '2025-01-16', 1040.00, 'Pago'),  
(18, 'Pix', '2025-01-18', 540.00, 'Pendente'),  
(19, 'Cartão', '2025-01-19', 280.00, 'Pendente'),  
(20, 'Pix', '2025-01-21', 880.00, 'Pendente');  
  
INSERT INTO Locacao (idLocacao, idCliente, idPagamento, dataInicio,  
dataFim) VALUES  
(1, 1, 1, '2024-12-01', '2024-12-03'),  
(2, 2, 2, '2024-12-01', '2024-12-02'),  
(3, 3, 3, '2024-12-05', '2024-12-07'),  
(4, 1, 4, '2024-12-10', '2024-12-15'),  
(5, 4, 5, '2024-12-11', '2024-12-20'),  
(6, 2, 6, '2024-12-15', '2025-01-01'),  
(7, 5, 7, '2024-12-20', '2024-12-26'),  
(8, 6, 8, '2024-12-23', '2025-01-02'),  
(9, 1, 9, '2024-12-26', '2024-12-28'),  
(10, 3, 10, '2024-12-28', '2025-01-03'),  
(11, 7, 11, '2025-01-02', '2025-01-04'),  
(12, 4, 12, '2025-01-05', '2025-01-10'),  
(13, 6, 13, '2025-01-05', '2025-01-07'),  
(14, 8, 14, '2025-01-09', '2025-01-13'),  
(15, 7, 15, '2025-01-12', '2025-01-15'),  
(16, 9, 16, '2025-01-13', '2025-01-20'),  
(17, 10, 17, '2025-01-16', '2025-01-20'),  
(18, 9, 18, '2025-01-18', '2025-01-21'),  
(19, 5, 19, '2025-01-19', '2025-01-20'),  
(20, 1, 20, '2025-01-21', '2025-01-25');
```



```
INSERT INTO LocacaoVeiculo (idLocacao, idVeiculo) VALUES
```

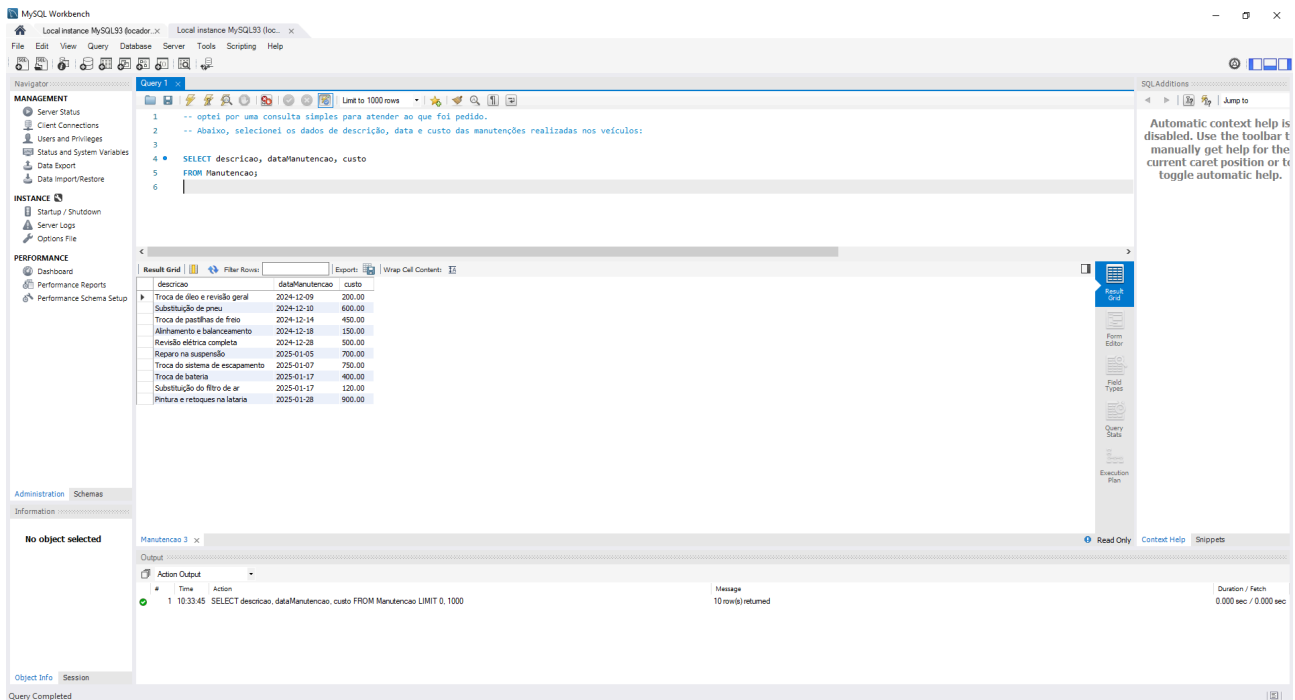
```
(1, 2),  
(2, 3),  
(3, 5),  
(4, 8),  
(5, 1),  
(6, 6),  
(7, 2),  
(8, 8),  
(9, 4),  
(10, 3),  
(11, 9),  
(12, 2),  
(13, 6),  
(14, 10),  
(15, 7),  
(16, 5),  
(17, 9),  
(18, 2),  
(19, 10),  
(20, 8);
```

```
INSERT INTO Manutencao (idManutencao, idVeiculo, descricao,  
dataManutencao, custo) VALUES
```

```
(1, 1, 'Troca de óleo e revisão geral', '2024-12-09', 200.00),  
(2, 1, 'Substituição de pneu', '2024-12-10', 600.00),  
(3, 3, 'Troca de pastilhas de freio', '2024-12-14', 450.00),  
(4, 5, 'Alinhamento e balanceamento', '2024-12-18', 150.00),  
(5, 5, 'Revisão elétrica completa', '2024-12-28', 500.00),  
(6, 8, 'Reparo na suspensão', '2025-01-05', 700.00),  
(7, 10, 'Troca do sistema de escapamento', '2025-01-07', 750.00),  
(8, 6, 'Troca de bateria', '2025-01-17', 400.00),  
(9, 6, 'Substituição do filtro de ar', '2025-01-17', 120.00),  
(10, 9, 'Pintura e retoques na lataria', '2025-01-28', 900.00);
```

Pontuação: 10 pontos.

2. Implemente uma consulta para listar a descrição, a data e o custo de todas as manutenções realizadas nos veículos.



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 -- optei por uma consulta simples para atender ao que foi pedido.  
2 -- Abaixo, selecionei os dados de descrição, data e custo das manutenções realizadas nos veículos:  
3  
4 SELECT descricao, dataManutencao, custo  
5 FROM Manutencao;  
6
```

The Results Grid shows the following data:

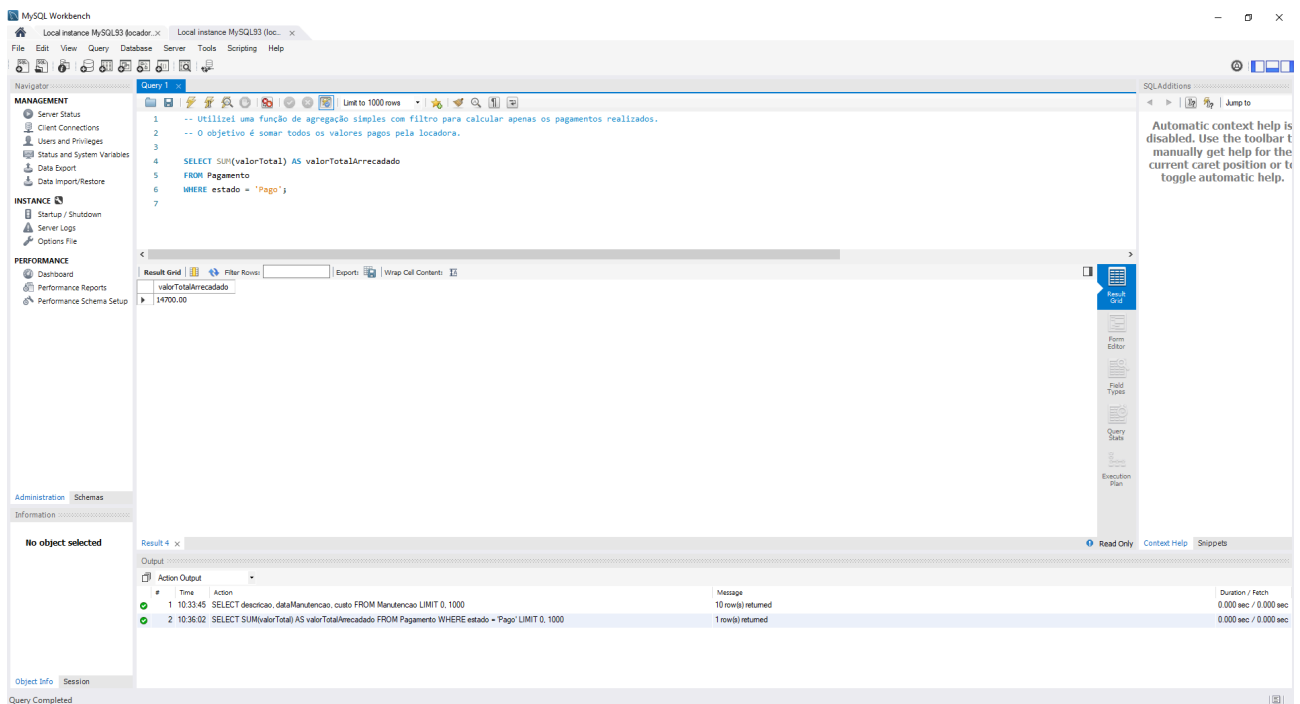
descricao	dataManutencao	custo
Troca de óleo e revisão geral	2024-12-09	200.00
Substituição de pneu	2024-12-10	600.00
Troca de pastilhas de freio	2024-12-14	450.00
Alinhamento e balanceamento	2024-12-18	150.00
Revisão elétrica completa	2024-12-28	500.00
Reparo na suspensão	2025-01-05	700.00
Troca do sistema de escapamento	2025-01-07	750.00
Troca de bateria	2025-01-17	400.00
Substituição do filtro de ar	2025-01-17	120.00
Pintura e retoques na lataria	2025-01-28	900.00

The Output tab shows the following message:

```
1 10:33:45 SELECT descricao, dataManutencao, custo FROM Manutencao LIMIT 0, 1000  
10 row(s) returned  
Duration: 0.000 sec / 0.000 sec
```

Pontuação: 10 pontos.

3. Implemente uma consulta para listar o valor total arrecadado pela locadora. Lembre-se que pagamentos “pendentes” não fazem parte da soma.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 -- Utilizei uma função de agregação simples com filtro para calcular apenas os pagamentos realizados.  
2 -- O objetivo é somar todos os valores pagos pela locadora.  
3  
4 SELECT SUM(valorTotal) AS valorTotalArrecadado  
5 FROM Pagamento  
6 WHERE estado = 'Pago';  
7
```

The Results grid shows the output of the query:

valorTotalArrecadado
14700.00

The bottom panel shows the execution log with the following entries:

Time	Action	Message	Duration / Fetch
1 10:33:45	SELECT descricao, dataManutencao, custo FROM Manutencao LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
2 10:36:02	SELECT SUM(valorTotal) AS valorTotalArrecadado FROM Pagamento WHERE estado = 'Pago' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Pontuação: 10 pontos.

4. Implemente uma consulta para listar o modelo e a marca dos veículos, bem como o número de vezes que cada um foi locado. A listagem deve ser mostrada em ordem decrescente pelo número de alugueis.

Dica: Utilize a cláusula *group by*.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- precisei juntar as tabelas e agrupar os dados por veículo.
1 SELECT
2   V.modelo,
3   V.marca,
4   COUNT(LV.idVeiculo) AS totalLocacoes
5 FROM
6   Veiculo V
7 JOIN
8   LocacaoVeiculo LV ON V.idVeiculo = LV.idVeiculo
9 GROUP BY
10  V.idVeiculo, V.modelo, V.marca
11 ORDER BY
12   totalLocacoes DESC;
13
14
```

The Results tab displays the following data:

modelo	marca	totalLocacoes
HB20	Hyundai	4
Duster	Renault	3
Gol	Volkswagen	2
Corolla	Toyota	2
Fiesta	Ford	2
Yoro	Fiat	2
Compass	Jeep	2
Onix	Chevrolet	1
Civic	Honda	1
Cruze	Chevrolet	1

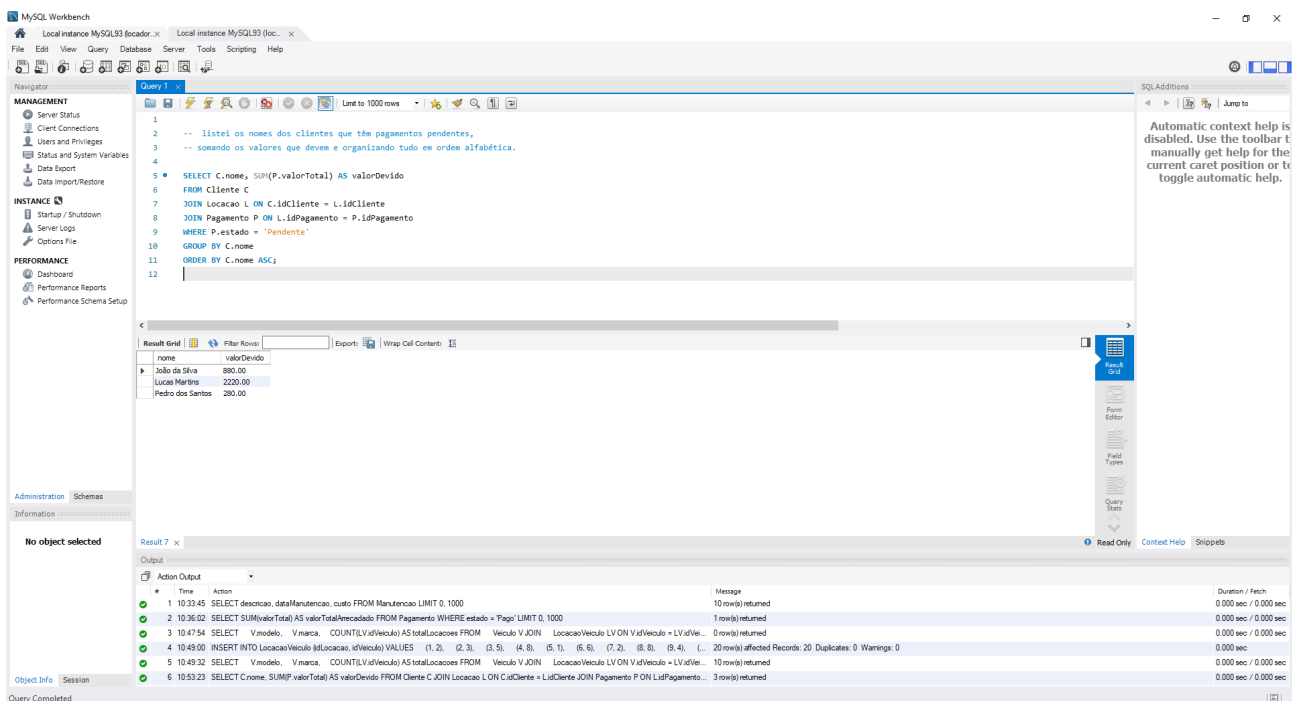
The Output tab shows the execution log with the following messages:

```
1 10:33:45 SELECT descricao, dataManutencao, custo FROM Manutencao LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
2 10:36:02 SELECT SUM(valorTotal) AS valorTotalArrecado FROM Pagamento WHERE estado = 'Pago' LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
3 10:47:54 SELECT V.modelo, V.marca, COUNT(LV.idVeiculo) AS totalLocacoes FROM Veiculo V JOIN LocacaoVeiculo LV ON V.idVeiculo = LV.idVe... 0 row(s) returned 0.000 sec / 0.000 sec
4 10:49:00 INSERT INTO LocacaoVeiculo (idLocacao, idVeiculo) VALUES (1, 2), (2, 3), (3, 5), (4, 8), (5, 1), (6, 6), (7, 2), (8, 8), (9, 4), ... 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0 0.000 sec
5 10:49:32 SELECT V.modelo, V.marca, COUNT(LV.idVeiculo) AS totalLocacoes FROM Veiculo V JOIN LocacaoVeiculo LV ON V.idVeiculo = LV.idVe... 10 row(s) returned 0.000 sec / 0.000 sec
```

Pontuação: 10 pontos.

5. Implemente uma consulta para listar o nome dos clientes que possuem pagamento “pendente”, bem como o valor devido por eles. A listagem deve ser mostrada em ordem alfabética crescente pelo nome dos clientes.

Dica: Utilize a cláusula *group by*.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 -- liste os nomes dos clientes que têm pagamentos pendentes,  
2 -- somando os valores que devem e organizando tudo em ordem alfabética.  
3  
4  
5 SELECT C.nome, SUM(P.valorTotal) AS valorDevido  
6 FROM Cliente C  
7 JOIN Locacao L ON C.idCliente = L.idCliente  
8 JOIN Pagamento P ON L.idPagamento = P.idPagamento  
9 WHERE P.estado = "Pendente"  
10 GROUP BY C.nome  
11 ORDER BY C.nome ASC;  
12
```

The Results tab shows the following data:

nome	valorDevido
João da Silva	880.00
Lucas Martins	2220.00
Pedro dos Santos	200.00

The Output tab shows the execution log with the following entries:

Time	Action	Message	Duration / Fetch
1 10:33:45	SELECT descrecao, dataManutencao, custo FROM Manutencao LIMIT 0, 1000		0.000 sec / 0.000 sec
2 10:36:02	SELECT SUM(valorTotal) AS valorTotalRecebido FROM Pagamento WHERE estado = 'Pago' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
3 10:47:54	SELECT V.modelo, V.marca, COUNT(LV.idVeiculo) AS totalLocacoes FROM Veiculo V JOIN Locacao/Veiculo LV ON V.idVeiculo = LV.idVeiculo	0 row(s) returned	0.000 sec / 0.000 sec
4 10:49:00	INSERT INTO Locacao/Veiculo (idLocacao, idVeiculo) VALUES (1, 2), (2, 3), (3, 5), (4, 8), (5, 1), (6, 6), (7, 2), (8, 8), (9, 4), (...)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
5 10:49:32	SELECT V.modelo, V.marca, COUNT(LV.idVeiculo) AS totalLocacoes FROM Veiculo V JOIN Locacao/Veiculo LV ON V.idVeiculo = LV.idVeiculo	10 row(s) returned	0.000 sec / 0.000 sec
6 10:53:23	SELECT C.nome, SUM(P.valorTotal) AS valorDevido FROM Cliente C JOIN Locacao L ON C.idCliente = L.idCliente JOIN Pagamento P ON L.idPagamento	3 row(s) returned	0.000 sec / 0.000 sec