

BANCO DE DADOS

Trabalho – Relatório

Curso:	Análise e Desenvolvimento de Sistemas
Aluno(a):	Gabriel Salvador Bizaco Vieira
RU:	2587559

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.

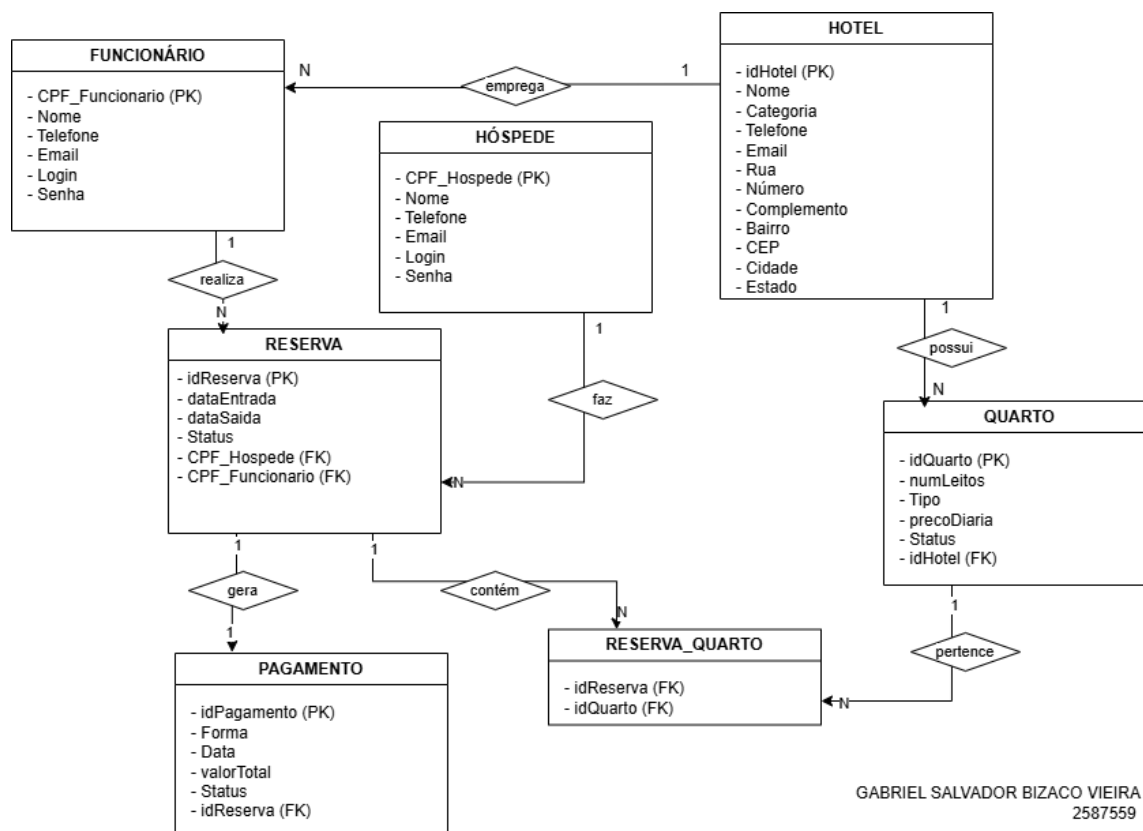
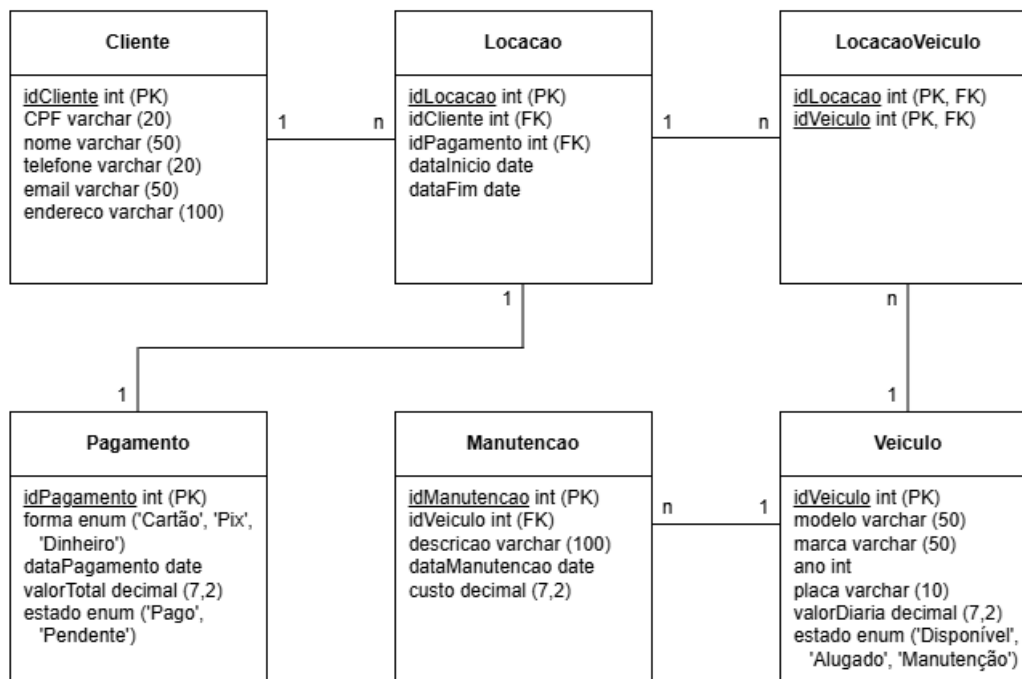


FIGURA 1 - MER

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*).

-- Criar o banco de dados e selecionar
CREATE DATABASE LocadoraVeiculos;
USE LocadoraVeiculos;

-- Tabela Cliente

```
CREATE TABLE Cliente (  
    idCliente INT PRIMARY KEY,  
    CPF VARCHAR(20) NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    telefone VARCHAR(20) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    endereco VARCHAR(200) NOT NULL  
);
```

-- Tabela Veiculo

```
CREATE TABLE Veiculo (  
    idVeiculo INT PRIMARY KEY,  
    modelo VARCHAR(50) NOT NULL,  
    marca VARCHAR(50) NOT NULL,  
    ano INT NOT NULL,  
    placa VARCHAR(10) NOT NULL,  
    valorDiaria DECIMAL(10,2) NOT NULL,  
    estado VARCHAR(20) NOT NULL  
);
```

-- Tabela Manutencao

```
CREATE TABLE Manutencao (  
    idManutencao INT PRIMARY KEY,  
    idVeiculo INT NOT NULL,  
    descricao VARCHAR(200) NOT NULL,  
    dataManutencao DATE NOT NULL,  
    custo DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (idVeiculo) REFERENCES Veiculo(idVeiculo)  
);
```

-- Tabela Pagamento

```
CREATE TABLE Pagamento (  
    idPagamento INT PRIMARY KEY,
```

```
forma VARCHAR(20) NOT NULL,  
dataPagamento DATE NOT NULL,  
valorTotal DECIMAL(10,2) NOT NULL,  
estado VARCHAR(20) NOT NULL  
);
```

-- Tabela Locacao

```
CREATE TABLE Locacao (  
    idLocacao INT PRIMARY KEY,  
    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 LocacaoVeiculo (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)  
);
```

Pontuação: 10 pontos.

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

-- Consulta 1

```
SELECT descricao, dataManutencao, custo  
FROM Manutencao;
```

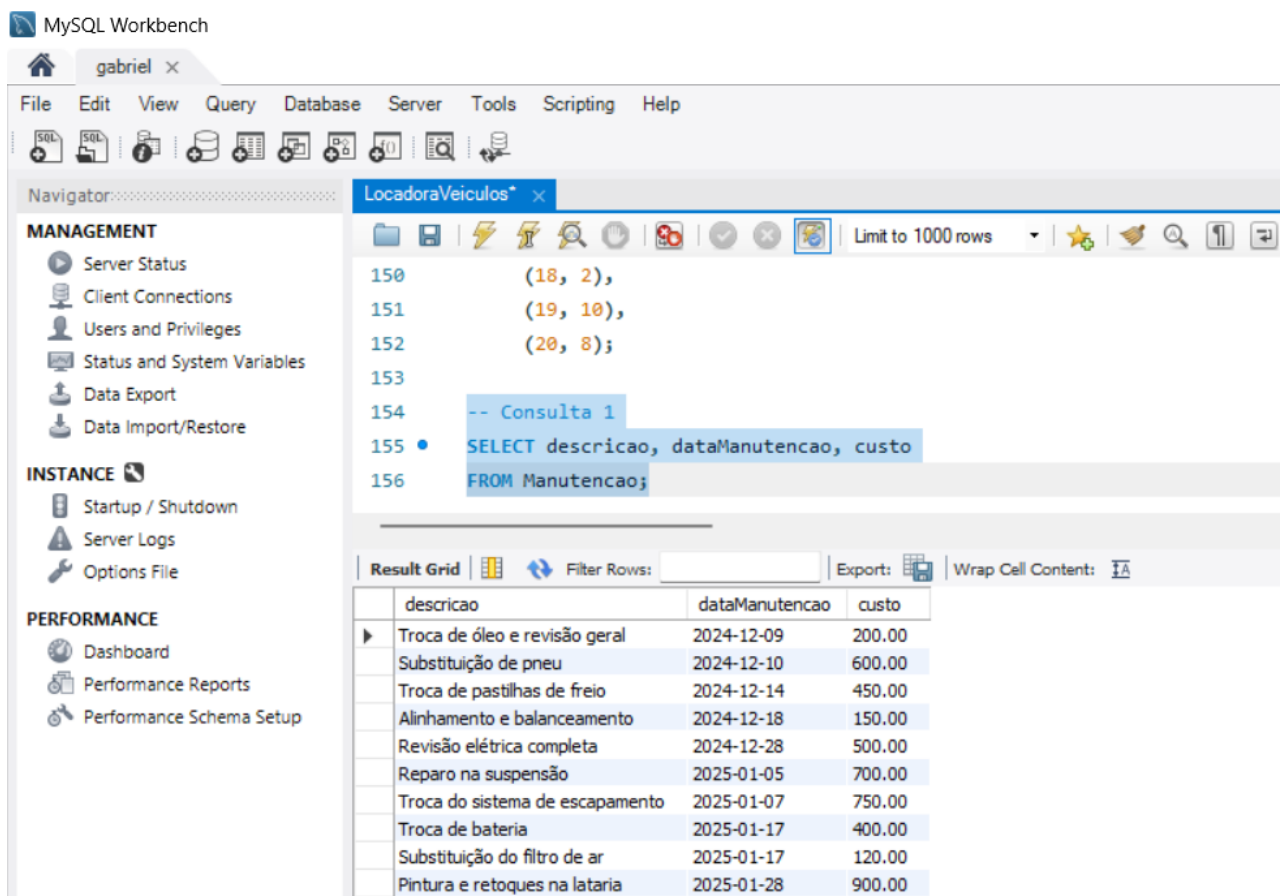


FIGURA 2 – Consulta 1

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.

-- Consulta 2

SELECT SUM(valorTotal) AS total_arrecadado
FROM Pagamento
WHERE estado = 'Pago';

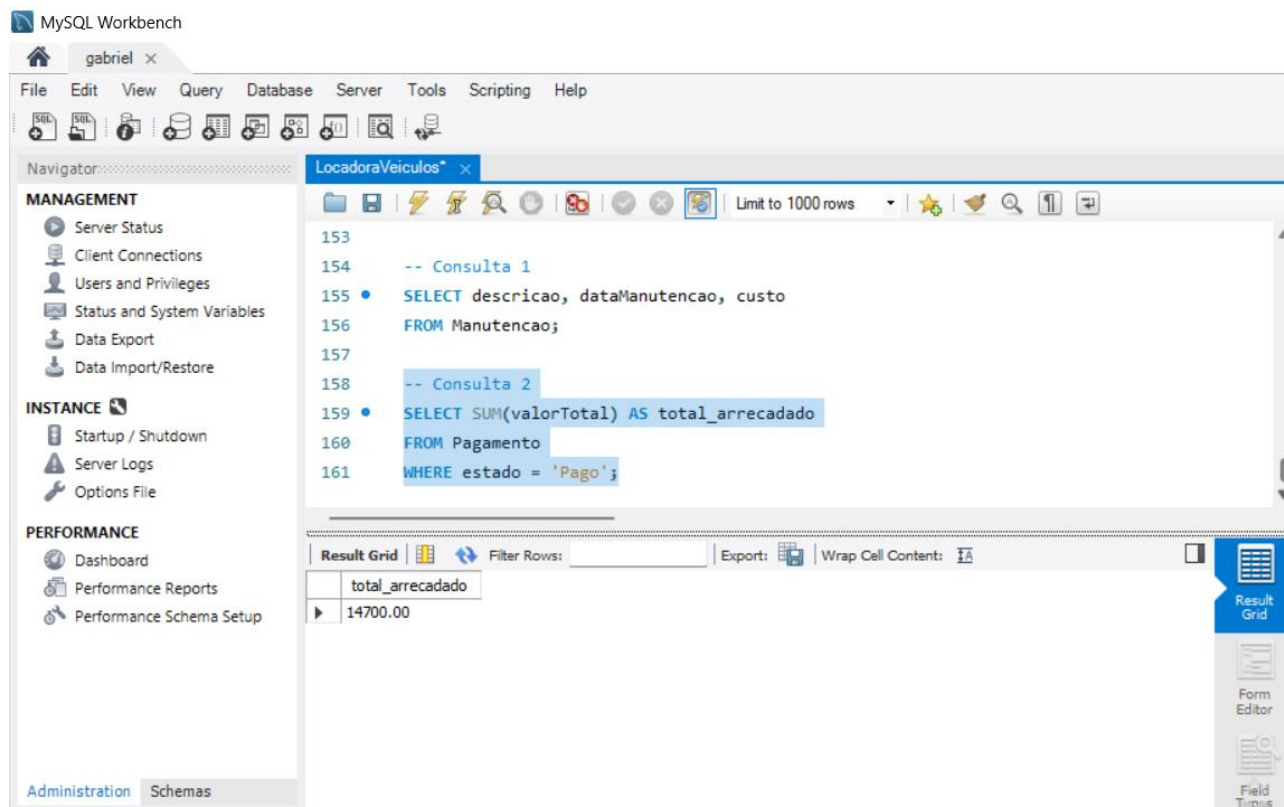


FIGURA 3 – Consulta 2

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 aluguéis.

Dica: Utilize a cláusula *group by*.

-- Consulta 3

```
SELECT V.modelo, V.marca, COUNT(*) AS total_locacoes  
FROM Veiculo V  
JOIN LocacaoVeiculo LV ON V.idVeiculo = LV.idVeiculo  
GROUP BY V.modelo, V.marca  
ORDER BY total_locacoes DESC;
```


The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' section with options like Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore. The 'INSTANCE' section includes Startup / Shutdown, Server Logs, and Options File. The 'PERFORMANCE' section includes Dashboard, Performance Reports, and Performance Schema Setup. The main editor window displays a SQL query for 'LocadoraVeiculos' database. The query is as follows:

```
160 FROM Pagamento
161 WHERE estado = 'Pago';
162
163 -- Consulta 3
164 SELECT V.modelo, V.marca, COUNT(*) AS total_locacoes
165 FROM Veiculo V
166 JOIN LocacaoVeiculo LV ON V.idVeiculo = LV.idVeiculo
167 GROUP BY V.modelo, V.marca
168 ORDER BY total_locacoes DESC;
```

The 'Result Grid' at the bottom shows the following data:

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

FIGURA 4 – Consulta 3

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*.

-- Consulta 4

```
SELECT C.nome, SUM(P.valorTotal) AS valor_devido  
FROM Cliente C  
JOIN Locacao L ON C.idCliente = L.idCliente  
JOIN Pagamento P ON L.idPagamento = P.idPagamento  
WHERE P.estado = 'Pendente'  
GROUP BY C.nome  
ORDER BY C.nome;
```

MySQL Workbench

gabriel x

File Edit View Query Database Server Tools Scripting Help

Navigator: LocadoraVeiculos* x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

```
166 JOIN LocacaoVeiculo LV ON V.idVeiculo = LV.idVeiculo
167 GROUP BY V.modelo, V.marca
168 ORDER BY total_locacoes DESC;
169
170 -- Consulta 4
171 SELECT C.nome, SUM(P.valorTotal) AS valor_devido
172 FROM Cliente C
173 JOIN Locacao L ON C.idCliente = L.idCliente
174 JOIN Pagamento P ON L.idPagamento = P.idPagamento
175 WHERE P.estado = 'Pendente'
176 GROUP BY C.nome
177 ORDER BY C.nome;
178
```

Result Grid

nome	valor_devido
João da Silva	880.00
Lucas Martins	2220.00
Pedro dos Santos	280.00

Result 4 x Read Only

FIGURA 5 – Consulta 4