

BANCO DE DADOS

Trabalho – Relatório

Curso:	CST ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
Aluno(a):	ALESSANDRO GOMES SOARES
RU:	4976113

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.

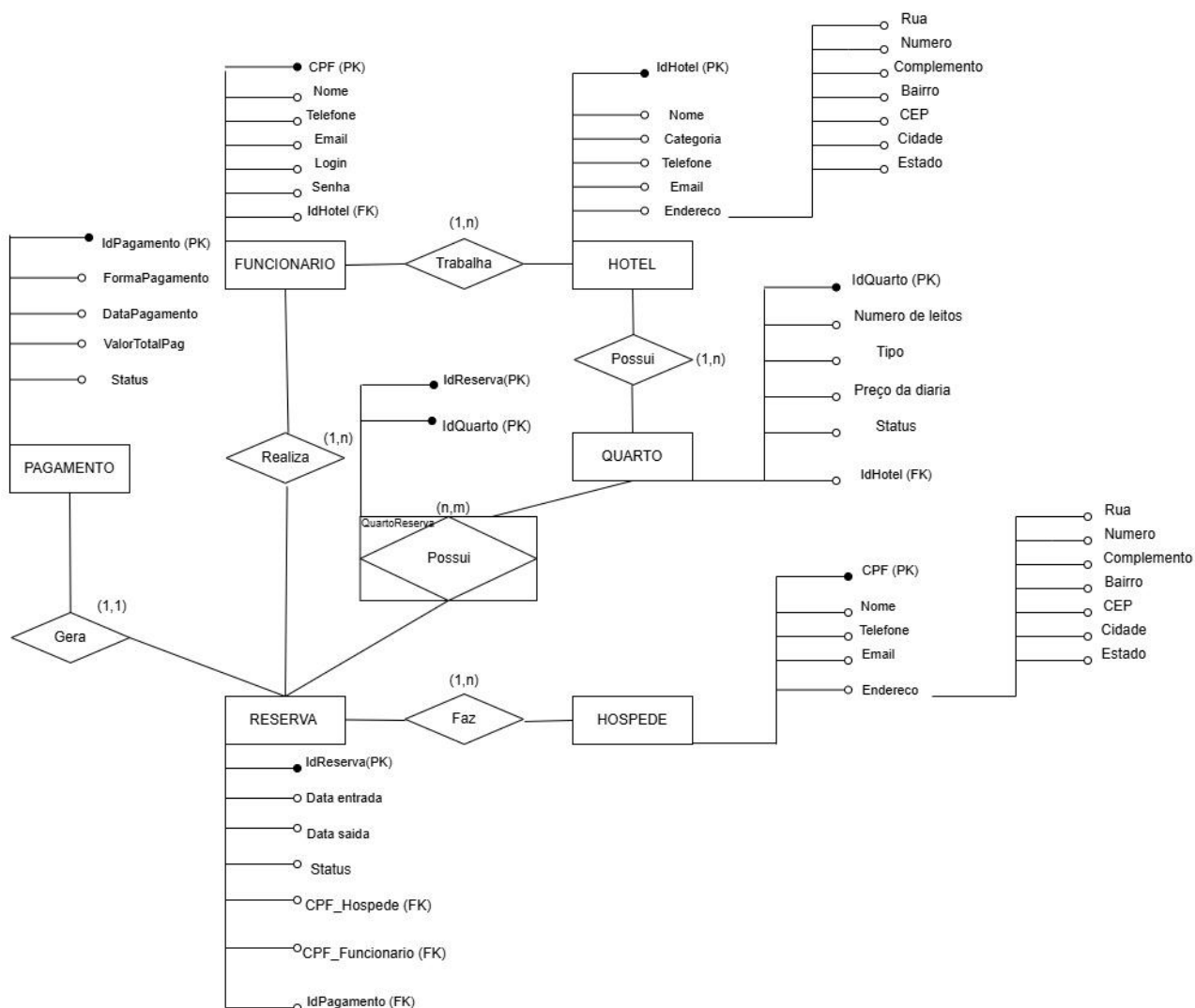
As regras de negócio são:

- 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.

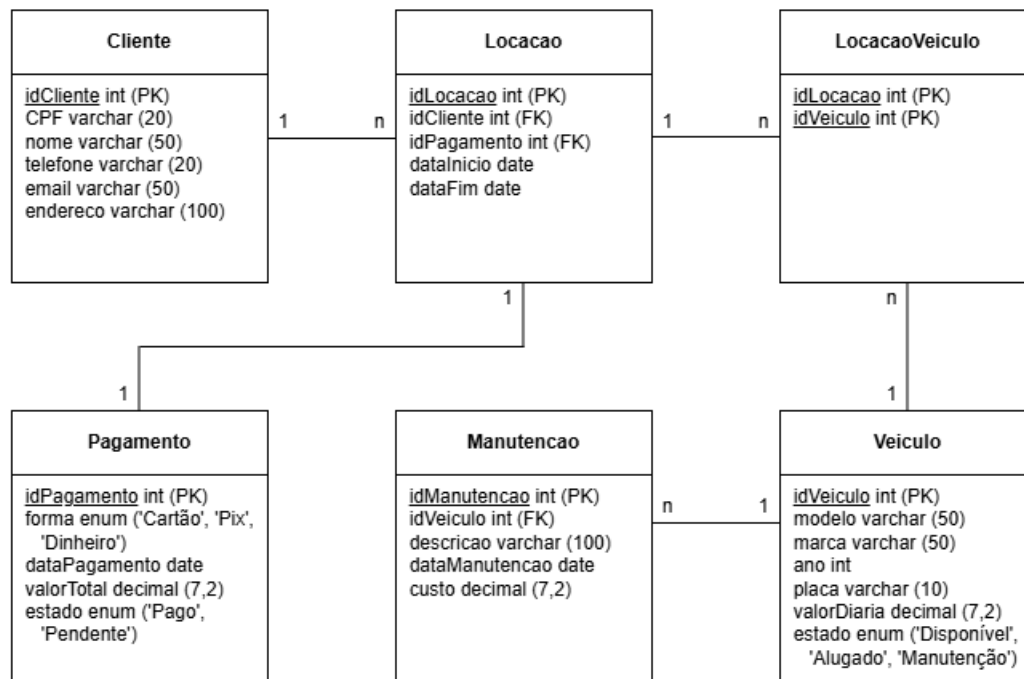
Importante:

- 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;

**CREATE TABLE Cliente (idCliente INT NOT NULL,
CPF VARCHAR(20) 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 Pagamento (  
idPagamento INT NOT NULL,  
forma ENUM('Cartao', 'Pix', 'Dinheiro') NOT NULL,  
dataPagamento DATE NOT NULL,  
valorTotal DECIMAL(7 , 2 ) NOT NULL,  
estado ENUM ('Pago','Pendente') NOT NULL,  
PRIMARY KEY (idPagamento)  
);
```

```
CREATE TABLE Veiculo (  
idVeiculo INT NOT NULL,  
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 ('Disponivel', 'Alugado', 'Manutenção') NOT NULL,  
PRIMARY KEY (idVeiculo)  
);
```

```
CREATE TABLE Locacao (  
idLocacao INT NOT NULL,  
idCliente INT NOT NULL,  
idPagamento INT NOT NULL,  
dataInicio DATE NOT NULL,  
dataFim DATE NOT NULL,
```

```
PRIMARY KEY (idLocacao),  
FOREIGN KEY (idCliente) REFERENCES Cliente (idCliente),  
FOREIGN KEY (idPagamento) REFERENCES Pagamento (idPagamento)  
);
```

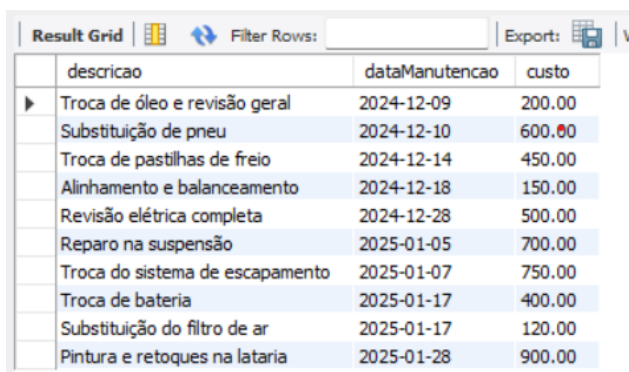
```
CREATE TABLE LocacaoVeiculo (  
idLocacao INT NOT NULL,  
idVeiculo INT NOT NULL,  
PRIMARY KEY (idLocacao, idVeiculo)  
);
```

```
CREATE TABLE Manutencao (  
idManutencao INT NOT NULL,  
idVeiculo INT NOT NULL,  
descricao VARCHAR(100) NOT NULL,  
dataManutencao DATE NOT NULL,  
custo DECIMAL(7 , 2 ) NOT NULL,  
PRIMARY KEY (idManutencao),  
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.

SELECT descricao, dataManutencao, custo FROM Manutencao;

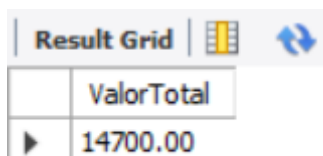


	descricao	dataManutencao	custo
▶	Troca de óleo e revisão geral	2024-12-09	200.00
	Substituição de pneu	2024-12-10	600.00
	Troca de pastilhas de freio	2024-12-14	450.00
	Alinhamento e balanceamento	2024-12-18	150.00
	Revisão elétrica completa	2024-12-28	500.00
	Reparo na suspensão	2025-01-05	700.00
	Troca 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

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.

```
SELECT SUM(valorTotal) as ValorTotal FROM Pagamento  
WHERE estado = 'Pago';
```



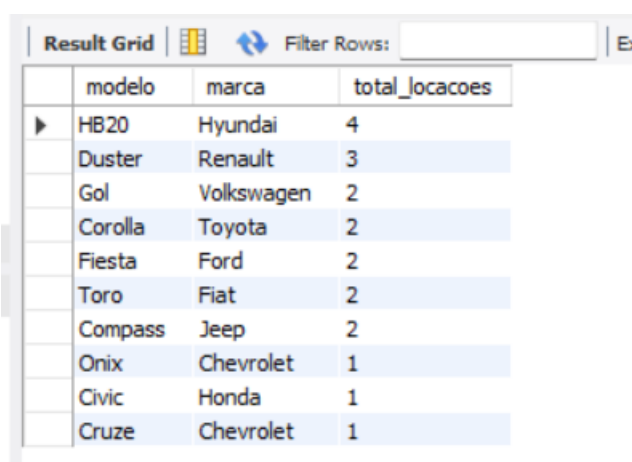
ValorTotal
14700.00

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*.

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



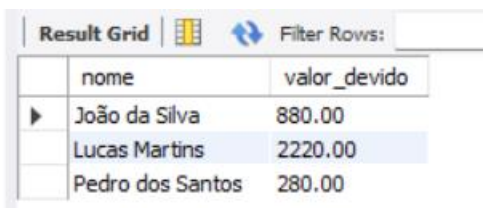
	modelo	marca	total_locacoes
▶	HB20	Hyundai	4
	Duster	Renault	3
	Gol	Volkswagen	2
	Corolla	Toyota	2
	Fiesta	Ford	2
	Toro	Fiat	2
	Compass	Jeep	2
	Onix	Chevrolet	1
	Civic	Honda	1
	Cruze	Chevrolet	1

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*.

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



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