BANCO DE DADOS

Trabalho - Relatório

Curso:	CST ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - DISTÂNCIA
Aluno(a):	Kaíque Marques Santos
RU:	5230113

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.

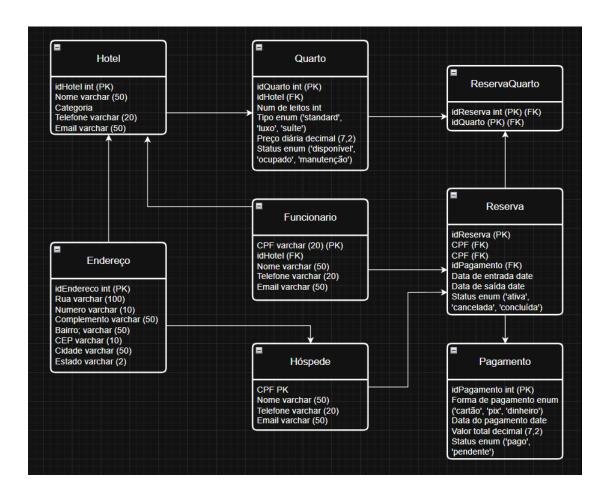
As regras de negócio são:

- 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:

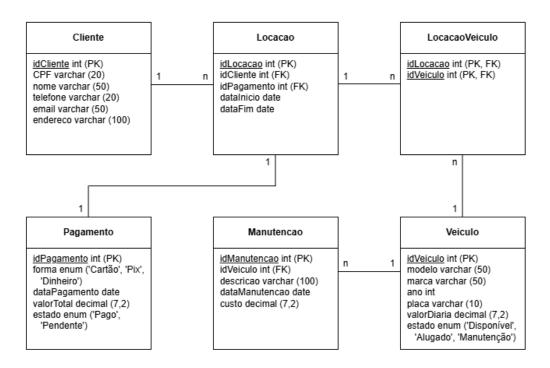
- O Modelo Entidade-Relacionamento (MER) deve considerar somente as regras de negócio dadas, não podendo ser criada nenhuma outra entidade ou atributo que não estejam nas regras de negócio;
- Em caso de haver entidade associativa, a mesma deve ser representada pela "Representação 1" (texto da Aula 1 – 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.



• 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.

 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 IF NOT EXISTS Locadora Veiculos; USE Locadora Veiculos;

CREATE TABLE Cliente (

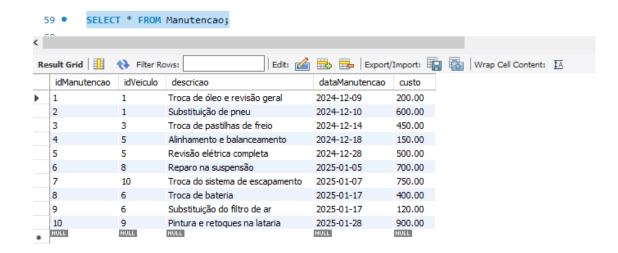
```
idCliente INT NOT NULL PRIMARY KEY,
  CPF VARCHAR(14) NOT NULL,
  nome VARCHAR(100) NOT NULL,
  telefone VARCHAR(20) NOT NULL,
  email VARCHAR(100) NOT NULL,
  endereco VARCHAR(200) NOT NULL
);
CREATE TABLE Veiculo (
  idVeiculo INT NOT NULL PRIMARY KEY,
  modelo VARCHAR(50) NOT NULL,
  marca VARCHAR(50) NOT NULL,
  ano INT NOT NULL,
  placa VARCHAR(10) NOT NULL,
  valorDiaria DECIMAL(10,2) NOT NULL,
  estado VARCHAR(20) NOT NULL
);
CREATE TABLE Manutencao (
  idManutencao INT NOT NULL 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)
);
CREATE TABLE Pagamento (
  idPagamento INT NOT NULL PRIMARY KEY,
  forma VARCHAR(20) NOT NULL,
  dataPagamento DATE NOT NULL,
```

```
valorTotal DECIMAL(10,2) NOT NULL,
  estado VARCHAR(20) NOT NULL
);
CREATE TABLE Locacao (
  idLocacao INT NOT NULL PRIMARY KEY,
  idCliente INT NOT NULL,
  idPagamento INT NOT NULL,
  datalnicio DATE NOT NULL,
  dataFim DATE NOT NULL,
  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),
  FOREIGN KEY (idLocacao) REFERENCES Locacao(idLocacao),
  FOREIGN KEY (idVeiculo) REFERENCES Veiculo(idVeiculo)
);
```

Pontuação: 10 pontos.

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

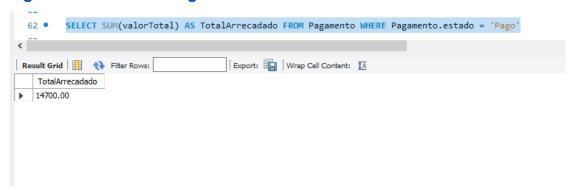
SELECT * FROM Manutencao;



Pontuação: 10 pontos.

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 TotalArrecadado FROM Pagamento WHERE Pagamento.estado = 'Pago'

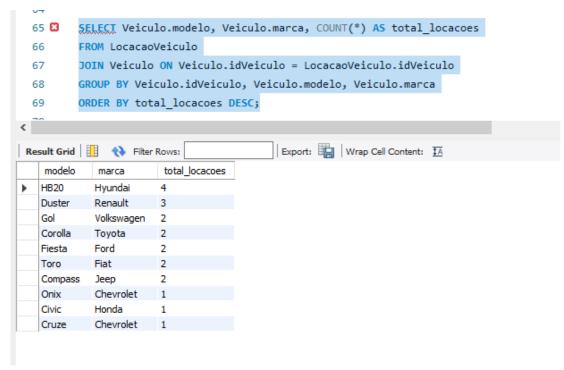


Pontuação: 10 pontos.

 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(*) AS total_locacoes FROM LocacaoVeiculo

JOIN Veiculo ON Veiculo.idVeiculo = LocacaoVeiculo.idVeiculo GROUP BY Veiculo.idVeiculo, Veiculo.modelo, Veiculo.marca ORDER BY total_locacoes DESC;



Pontuação: 10 pontos.

 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_pendente FROM Cliente

JOIN Locacao ON Cliente.idCliente = Locacao.idCliente

JOIN Pagamento ON Locacao.idPagamento = Pagamento.idPagamento WHERE Pagamento.estado = 'Pendente'

GROUP BY Cliente.nome

ORDER BY Cliente.nome ASC;

