

BANCO DE DADOS

Trabalho – Relatório

Curso:	Bacharelado em Engenharia de Software
Aluno(a):	Weydison dos Santos Andrade
RU:	4330561

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.

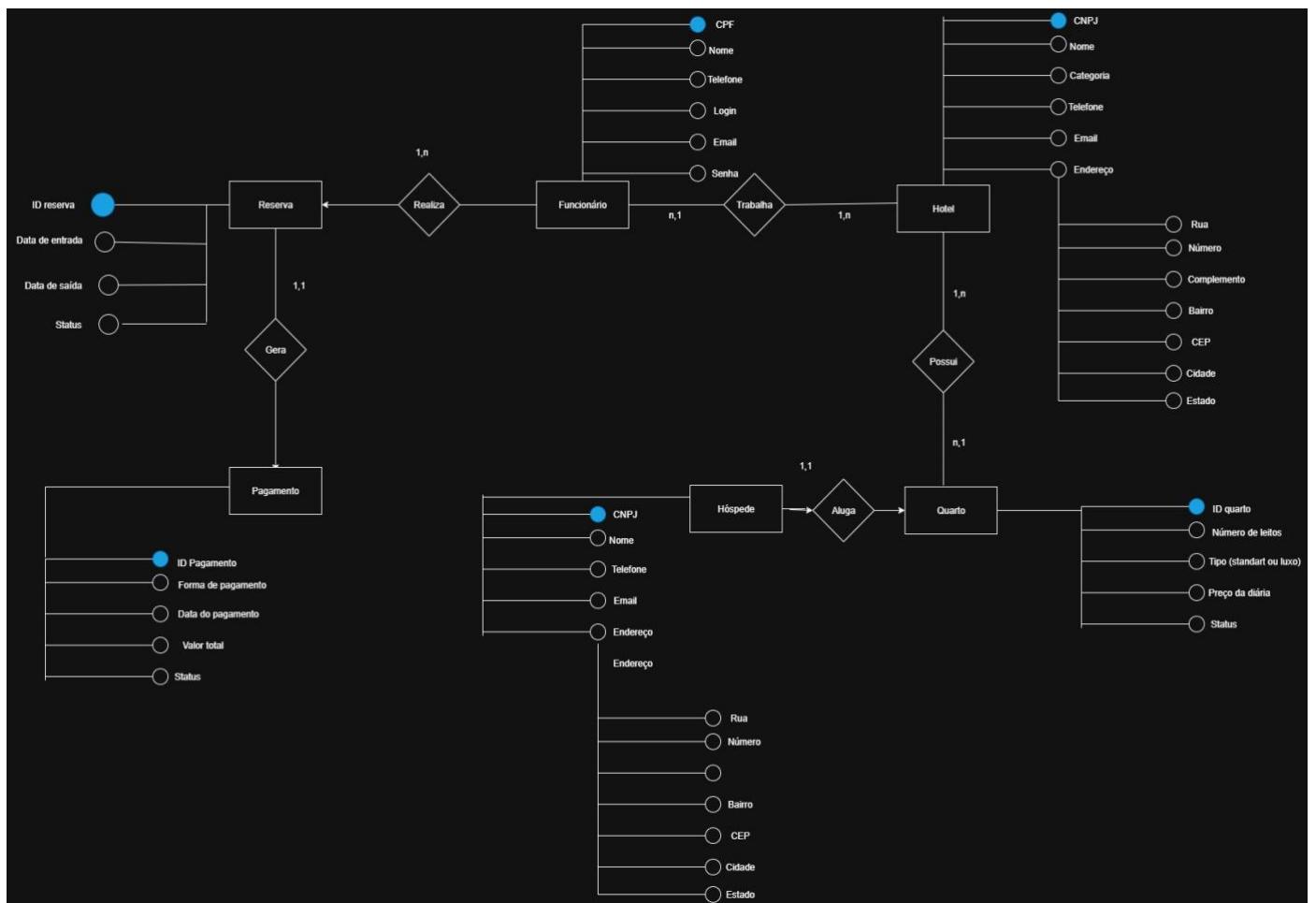
As regras de negócio são:

- 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:

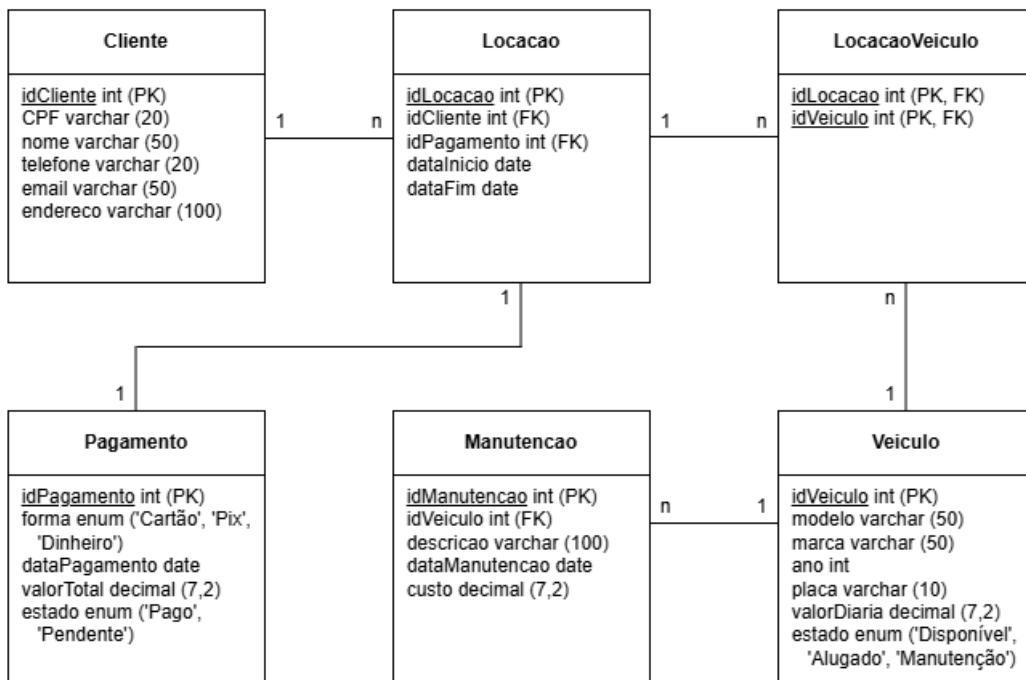
- O Modelo Entidade-Relacionamento (MER) deve considerar somente as regras de negócio dadas, não podendo ser criada nenhuma outra entidade ou atributo que não estejam nas regras de negócio;
- Em caso de haver entidade associativa, a mesma deve ser representada pela “Representação 1” (texto da Aula 1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



Fonte: Weydison Andrade

2. 2^a 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 (
    id int auto_increment primary key 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);
```

```
create table Locacao (  
idLocacao int auto_increment 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));
```

```
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));
```

```
create table Pagamento (  
idPagamento int not null primary key,  
forma enum('Cartão', 'Pix', 'Dinheiro') not null,  
dataPagamento date not null,  
valorTotal decimal(7,2) not null,  
estado enum('Pago', 'Pendente') not null);
```

```
create table Manutencao (  
idManutencao int not null primary key,  
idVeiculo int not null,  
descricao varchar(100) not null,  
dataManutencao date not null,
```

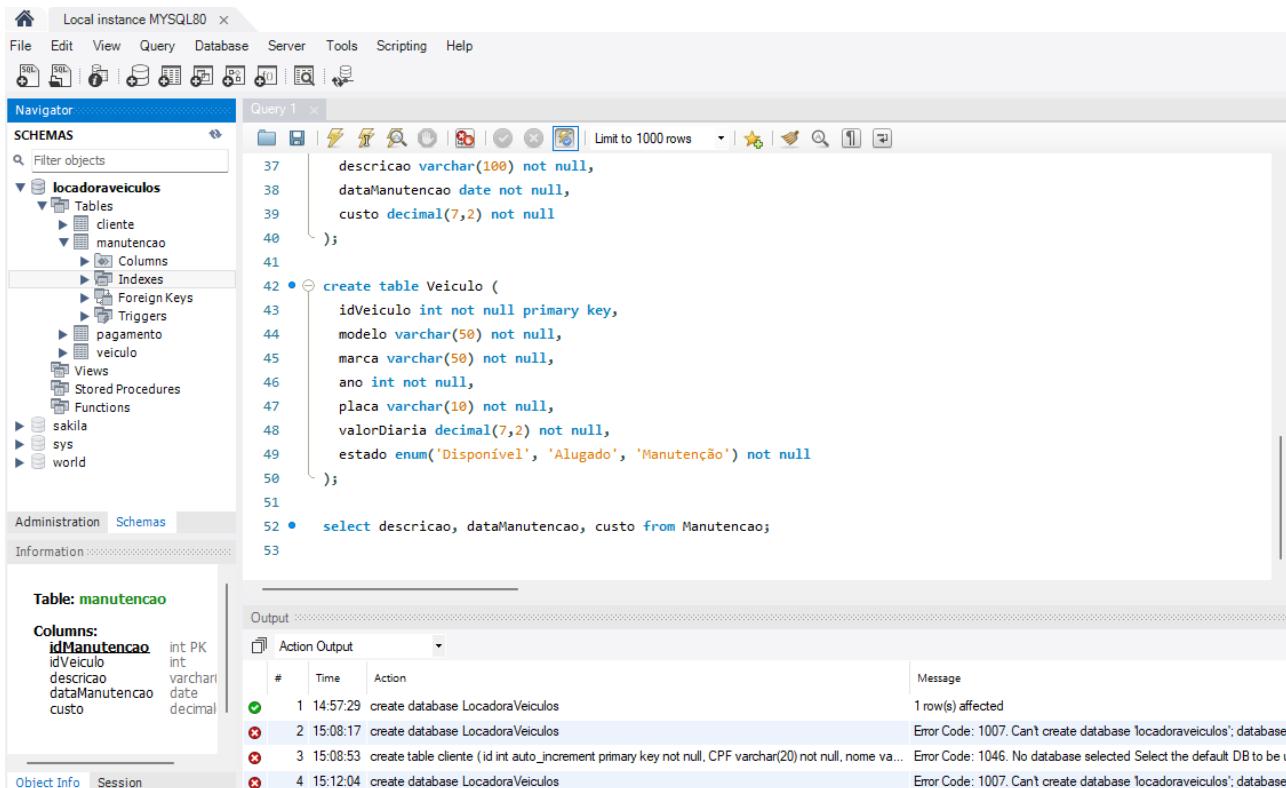
```
custo decimal(7,2) not null  
);  
  
create table Veiculo (  
    idVeiculo int not null primary key,  
    modelo varchar(50) not null,  
    marca varchar(50) not null,  
    ano int not null,  
    placa varchar(10) not null,  
    valorDiaria decimal(7,2) not null,  
    estado enum('Disponível', 'Alugado', 'Manutenção') not null  
);
```

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;
```

obs: Professora não consegui ver o resultado da consulta, essa é a tela printada da minha máquina após a consulta.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema for 'locadoraveiculos'. The 'manutencao' table is selected, showing its columns: idManutencao (int PK), idVeiculo (int), descricao (varchar), dataManutencao (date), and custo (decimal). The main area contains a SQL query window with the following code:

```

37     descricao varchar(100) not null,
38     dataManutencao date not null,
39     custo decimal(7,2) not null
40   );
41
42 •  create table Veiculo (
43     idVeiculo int not null primary key,
44     modelo varchar(50) not null,
45     marca varchar(50) not null,
46     ano int not null,
47     placa varchar(10) not null,
48     valorDiaria decimal(7,2) not null,
49     estado enum('Disponível', 'Alugado', 'Manutenção') not null
50   );
51
52 •  select descricao, dataManutencao, custo from Manutencao;
53

```

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	14:57:29	create database LocadoraVeiculos	1 row(s) affected
2	15:08:17	create database LocadoraVeiculos	Error Code: 1007. Can't create database 'locadoraveiculos'; database with the same name already exists
3	15:08:53	create table cliente (id int auto_increment primary key not null, CPF varchar(20) not null, nome va...)	Error Code: 1046. No database selected Select the default DB to be used
4	15:12:04	create database LocadoraVeiculos	Error Code: 1007. Can't create database 'locadoraveiculos'; database with the same name already exists

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 ValorTotalArrecadado from Pagamento where estado = 'Pago';

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
v.modelo,
v.marca,
count(l.idVeiculo) as
numero_de_alugueis
from
Veiculo as v join
Locacao as l on v.idVeiculo =
l.idVeiculo
group by
v.modelo, v.marca
order by
numero_de_alugueis desc;
```

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
c.nome as nome_cliente,
p.valorTotal as valor_devido
from
cliente as c join
Locacao as l on c.idClientes = l.idCliente
join
pagamento as p on l.idPagamento - p.idPagamento
where
p.estado = 'Pendente'
order by
c.nome asc;
```

