

Sumário

Diagrama Entidade-Relacionamento (DER).....	4
Discussão sobre a Normalização.....	6
Justificativa das Escolhas.....	7
Descrição das Tabelas e Atributos.....	10

Gestão de Locadora

Integrantes:

Giovana Rafaela da Silva

Maria Eduarda Claro

Pedro Henrique Dias de Paula Santos

Tobias Perassi Alquezar

Professor:

André Cassulino Araújo Souza

Disciplinas:

Banco de Dados

Ciência de Dados

Sorocaba, 2025

15/06/2025

Introdução:

A organização e o gerenciamento de dados são fundamentais para o bom funcionamento de qualquer empresa que lida com um grande volume de informações. No contexto das locadoras de veículos, por exemplo, é essencial manter o controle preciso sobre os veículos disponíveis, os clientes atendidos, as reservas realizadas e os pagamentos efetuados. Com isso em vista, este projeto de Banco de Dados tem como foco o desenvolvimento de um sistema de gestão para uma locadora, com o objetivo de otimizar o armazenamento, a recuperação e a manipulação dessas informações de forma eficiente, segura e estruturada.

A escolha do tema justifica-se pela necessidade real de informatização e automação dos processos administrativos em empresas do ramo de locação de veículos, que muitas vezes ainda utilizam métodos manuais ou planilhas desorganizadas. Nesse sentido, o projeto visa modelar um banco de dados relacional capaz de atender às principais demandas operacionais da locadora, oferecendo uma base sólida para futuras integrações com sistemas de front-end e relatórios gerenciais.

O principal objetivo do projeto é desenvolver e documentar a estrutura de um banco de dados relacional completo, incluindo o levantamento de requisitos, o modelo entidade-relacionamento (MER), o modelo lógico e a implementação física das tabelas. O sistema proposto deve permitir o gerenciamento eficiente de clientes, veículos, contratos de locação, formas de pagamento e histórico de transações, contribuindo assim para a modernização dos processos internos da empresa.

A base de dados desenvolvida é composta por cinco tabelas principais: clientes, endereco, filmes, funcionarios e locacao. A estrutura proposta foi modelada com foco na normalização, integridade relacional e facilidade de uso, visando representar com fidelidade o funcionamento operacional de uma locadora de filmes.

Modelagem Conceitual:

O sistema de gestão para a locadora foi modelado com base em um Diagrama Entidade-Relacionamento (DER), que visa representar as principais entidades envolvidas no processo de locação de filmes, bem como seus respectivos atributos e relacionamentos.

Entidades e Atributos

1. Cliente

- ID_Cliente (chave primária)
- CPF
- Nome
- Telefone
- Email
- ID_Endereço (chave estrangeira)

2. Endereço

- ID_Endereço (chave primária)
- Rua
- Número
- Bairro
- Cidade
- CEP

3. Locação

- ID_Locação (chave primária)
- ID_Cliente (chave estrangeira)
- ID_Filme (chave estrangeira)

- ID_Funcionário (chave estrangeira)
- Data Início
- Data Fim
- Valor

4. Filmes

- ID_Filmes (chave primária)
- Títulos
- Disponível
- Ano
- Categoria

5. Funcionário

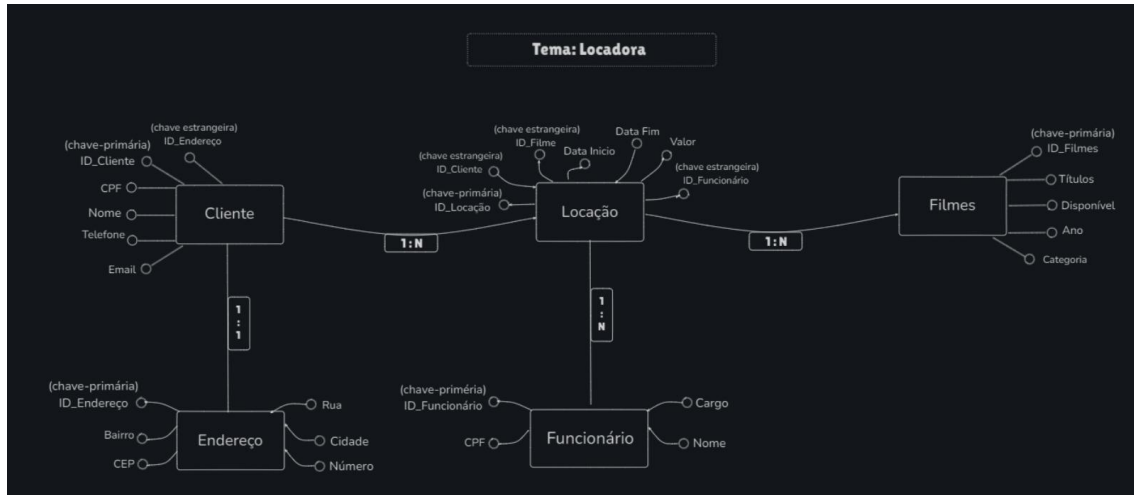
- ID_Funcionário (chave primária)
- CPF
- Nome
- Cargo

Relacionamentos

- Um Cliente está associado a um Endereço (relação 1:1).
- Um Cliente pode realizar múltiplas Locações (relação 1:N).
- Um Filme pode estar presente em várias Locações, mas cada locação refere-se a um Filme específico (relação 1:N).
- Um Funcionário pode registrar várias Locações, mas cada locação é registrada por um Funcionário (relação 1:N).

Diagrama Entidade-Relacionamento (DER)

O diagrama abaixo representa graficamente o modelo lógico do banco de dados, ilustrando as entidades envolvidas, seus atributos e os relacionamentos entre elas:



Modelagem Lógica

A partir do Diagrama Entidade-Relacionamento (DER) apresentado, foi realizado o processo de transformação para o modelo relacional, que consiste na criação de tabelas correspondentes às entidades e a definição das chaves primárias e estrangeiras. Cada entidade foi convertida em uma tabela, onde os atributos tornaram-se colunas, respeitando as restrições impostas pelos relacionamentos.

Chaves Primárias e Estrangeiras

- As chaves primárias (PK) foram atribuídas a atributos únicos que identificam de forma exclusiva os registros de cada tabela.

- As chaves estrangeiras (FK) foram utilizadas para garantir a integridade referencial entre as tabelas, estabelecendo ligações entre registros relacionados. Exemplo: ID_Endereço em Cliente referencia a tabela Endereço.

Discussão sobre a Normalização

A normalização é um processo fundamental no projeto de banco de dados, cujo objetivo é eliminar redundâncias e evitar anomalias de inserção, atualização ou exclusão. Neste projeto, as entidades foram normalizadas até a Terceira Forma Normal (3FN), conforme descrito a seguir:

Primeira Forma Normal (1FN)

- Todas as tabelas possuem atributos atômicos (isto é, indivisíveis).
- Não há grupos repetitivos, e cada campo possui apenas um valor por registro.

Segunda Forma Normal (2FN)

- Todas as tabelas estão na 1FN e todos os atributos não-chave dependem totalmente da chave primária.
- Não existem dependências parciais, pois todas as tabelas possuem chaves primárias simples.

Terceira Forma Normal (3FN)

- Todas as tabelas estão na 2FN e não há dependências transitivas (ou seja, nenhum atributo não-chave depende de outro atributo não-chave).
- Exemplo: o endereço foi separado da tabela Cliente, evitando que os dados do endereço fiquem duplicados para clientes com o mesmo CEP ou cidade.

Justificativa das Escolhas

A escolha por normalizar o banco até a 3FN baseou-se no equilíbrio entre desempenho e integridade dos dados. A 3FN permite maior organização e consistência, o que facilita futuras manutenções e ampliações no sistema, além de otimizar a estrutura para consultas relacionais mais seguras e confiáveis.

Além disso, a estrutura criada atende perfeitamente ao escopo de um sistema de locadora, separando entidades como Filmes, Clientes, Funcionários e Endereços, e organizando os relacionamentos de forma lógica e escalável.

Estrutura do Banco de Dados

-- Criação do banco de dados

```
CREATE DATABASE locadora;
```

-- Criação do schema

```
CREATE SCHEMA gerenciamento;
```

-- Tabela de clientes

```
CREATE TABLE gerenciamento.clientes (
```

```
    id_cliente SERIAL PRIMARY KEY,
```

```
    endereco VARCHAR(200) NOT NULL,
```

```
    cpf BIGINT NOT NULL,
```

```
    telefone VARCHAR(20) NOT NULL,
```

```
email VARCHAR(100) NOT NULL

);

-- Tabela de funcionários

CREATE TABLE gerenciamento.funcionarios (

    id_funcionario SERIAL PRIMARY KEY,

    cpf BIGINT NOT NULL,

    cargo VARCHAR(50) NOT NULL,

    nome VARCHAR(100) NOT NULL

);
```

```
-- Tabela de filmes

CREATE TABLE gerenciamento.filmes (

    id_filme SERIAL PRIMARY KEY,

    titulo VARCHAR(100) NOT NULL,

    disponivel BOOLEAN NOT NULL,

    ano INT NOT NULL,

    categoria VARCHAR(50) NOT NULL

);
```


-- Tabela de endereço

```
CREATE TABLE gerenciamento.endereco (  
  
    id_endereco SERIAL PRIMARY KEY,  
  
    bairro VARCHAR(100) NOT NULL,  
  
    cep BIGINT NOT NULL,  
  
    rua VARCHAR(100) NOT NULL,  
  
    cidade VARCHAR(100) NOT NULL,  
  
    numero INT NOT NULL  
  
);
```

-- Tabela de locações

```
CREATE TABLE gerenciamento.locacao (  
  
    id_locacao SERIAL PRIMARY KEY,  
  
    data_inicio DATE NOT NULL,  
  
    data_final DATE NOT NULL,  
  
    valor DECIMAL(10,2) NOT NULL,  
  
    id_cliente INT,  
  
    id_filme INT,  
  
    id_funcionario INT,  
  
    id_endereco INT,
```

```

        FOREIGN          KEY          (id_cliente)          REFERENCES
gerenciamento.clientes(id_cliente),

        FOREIGN KEY (id_filme) REFERENCES gerenciamento.filmes(id_filme),

        FOREIGN          KEY          (id_funcionario)          REFERENCES
gerenciamento.funcionarios(id_funcionario),

        FOREIGN          KEY          (id_endereco)          REFERENCES
gerenciamento.endereco(id_endereco)

);

```

Descrição das Tabelas e Atributos

1. clientes

Armazena os dados dos clientes da locadora.

- id_cliente: Identificador único do cliente (PK).
- endereco: Endereço completo do cliente.
- cpf: Cadastro de Pessoa Física (único por cliente).
- telefone: Contato do cliente.
- email: E-mail para comunicações.

2. funcionarios

Armazena informações dos funcionários da locadora.

- id_funcionario: Identificador único do funcionário (PK).
- cpf: Cadastro de Pessoa Física do funcionário.
- cargo: Função exercida na empresa.
- nome: Nome completo.

3. filmes

Contém os dados dos filmes disponíveis para locação.

- id_filme: Identificador único do filme (PK).
- titulo: Nome do filme.
- disponivel: Indica se o filme está disponível para locação.
- ano: Ano de lançamento.
- categoria: Gênero ou tipo do filme.

4. endereco

Tabela separada para armazenar os endereços (relacionada à locação e potencialmente aos clientes).

- id_endereco: Identificador único do endereço (PK).
- bairro, cep, rua, cidade, numero: Componentes do endereço.

5. locacao

Representa uma locação feita por um cliente.

- id_locacao: Identificador da locação (PK).
- data_inicio, data_final: Datas de início e fim da locação.
- valor: Valor cobrado pela locação.
- id_cliente: FK referenciando o cliente.
- id_filme: FK referenciando o filme alugado.
- id_funcionario: FK referenciando o funcionário responsável.
- id_endereco: FK para o local de entrega ou retirada (opcional dependendo do sistema).

DML e DQL

```
INSERT INTO gerenciamento.endereço (bairro, cep, rua, cidade, numero)
VALUES
('Centro', 12345678, 'Rua das Flores', 'São Paulo', 123),
('Jardim', 87654321, 'Avenida Principal', 'Rio de Janeiro', 456);
```

```
INSERT INTO gerenciamento.clientes (endereço, cpf, telefone, email) VALUES
('Rua das Flores, 123, Centro, São Paulo', 12345678901, '(11) 98765-4321',
'pedro1@email.com'),
('Avenida Principal, 456, Jardim, Rio de Janeiro', 98765432109, '(21) 91234-
5678', 'duda2@email.com');
```

```
INSERT INTO gerenciamento.funcionarios (cpf, cargo, nome) VALUES
(11425235644, 'Atendente', 'Giovana'),
(53461677388, 'Gerente', 'Duda');
```

```
INSERT INTO gerenciamento.filmes (titulo, disponivel, ano, categoria) VALUES
('O Senhor dos Anéis', TRUE, 2001, 'Fantasia'),
('Matrix', TRUE, 1999, 'Ficção Científica');
```

```
INSERT INTO gerenciamento.locacao (data_inicio, data_final, valor, id_cliente,
id_filme, id_funcionario, id_endereço) VALUES
('2025-05-16', '2025-05-20', 15.00, 1, 1, 1, 1), -- Cliente 1 aluga "O Senhor dos
Anéis" com funcionário 1
('2025-05-16', '2025-05-18', 12.00, 2, 2, 2, 2); -- Cliente 2 aluga "Matrix" com
funcionário 2
```

-- Listar todas as locações com nome do cliente e título do filme

SELECT

l.id_locacao,
c.nome AS nome_cliente,
f.titulo AS filme,
l.data_inicio,
l.data_final,
l.valor

FROM gerenciamento.locacao l

INNER JOIN gerenciamento.clientes c ON l.id_cliente = c.id_cliente

INNER JOIN gerenciamento.filmes f ON l.id_filme = f.id_filme;

-- Ver locações com nome do funcionário responsável

SELECT

l.id_locacao,
func.nome AS funcionario,
l.data_inicio,
l.valor

FROM gerenciamento.locacao l

INNER JOIN gerenciamento.funcionarios func ON l.id_funcionario =
func.id_funcionario;

-- Ver endereço completo da locação, com cidade e rua

SELECT

l.id_locacao,
e.cidade,
e.rua,
e.numero

FROM gerenciamento.locacao l

INNER JOIN gerenciamento.endereco e ON l.id_endereco = e.id_endereco;

DCL e DTL (Bonificação)

```
CREATE USER admin_user WITH PASSWORD 'admin123';
```

```
CREATE USER normal_user WITH PASSWORD 'normal123';
```

```
GRANT ALL PRIVILEGES ON SCHEMA gerenciamento TO admin_user; --  
Coloca privilégios de criar tabelas, funções etc, porém ainda precisa de  
permissões para manipulá-las
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA gerenciamento TO  
admin_user; --Garante privilégios para o usuário ter controle total dentro das  
tabelas que estão dentro do esquema gerenciamento
```

```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA gerenciamento  
TO admin_user; --Fornece permissões sobre todas as sequências existentes em  
gerenciamento
```

```
GRANT USAGE ON SCHEMA gerenciamento TO normal_user; --Concede ao  
usuário leitura_user a permissão de uso do esquema
```

```
GRANT SELECT ON ALL TABLES IN SCHEMA gerenciamento TO  
normal_user; --Concede ao usuário leitura_user a permissão de leitura (somente  
consulta) em todas as tabelas
```

```
-- Garante que permissões sejam propagadas para futuras tabelas
```

```
ALTER DEFAULT PRIVILEGES IN SCHEMA gerenciamento
```

```
    GRANT SELECT ON TABLES TO normal_user;
```

```
GRANT INSERT ON ALL TABLES IN SCHEMA gerenciamento TO normal_user;
```

```
REVOKE INSERT ON ALL TABLES IN SCHEMA gerenciamento FROM  
normal_user;
```

```
BEGIN;
```

```
SAVEPOINT antes_inserir_filme;
```

```
INSERT INTO gerenciamento.filmes (titulo, genero)
VALUES ('O Senhor dos Anéis', 'Fantasia');
```

```
UPDATE gerenciamento.filmes SET disponivel = TRUE
WHERE titulo = 'O Senhor dos Anéis'
```

```
IF (SELECT COUNT(*) FROM gerenciamento.filmes WHERE titulo = 'O Senhor
dos Anéis') > 1 THEN
    ROLLBACK TO SAVEPOINT antes_inserir_filme;
ELSE

    COMMIT;
END IF;
```

Conclusão

O projeto teve início com a construção do Diagrama Entidade-Relacionamento (DER), que serviu como base para identificar as entidades, seus atributos e os relacionamentos entre elas. A partir do DER, foi realizada a transformação para o modelo relacional, com definição clara das tabelas, chaves primárias e

estrangeiras, respeitando os princípios da normalização até a Terceira Forma Normal (3FN) para evitar redundâncias e garantir a integridade dos dados.

Foram desenvolvidos scripts DDL para a criação do banco de dados e suas tabelas, seguidos pelos scripts DML para inserção, atualização e exclusão de dados. Por fim, elaboraram-se consultas DQL utilizando SELECT e INNER JOIN para ilustrar a recuperação e cruzamento de dados de forma eficiente.

- A importância do modelo ER como base sólida para a estruturação do banco de dados.
- Como utilizar chaves primárias e estrangeiras para manter relacionamentos consistentes entre tabelas.
- O papel da normalização na prevenção de inconsistências e duplicidades.
- A prática com comandos SQL (DDL, DML e DQL) reforçou a sintaxe e a lógica relacional.
- A criação de relacionamentos complexos (ex: locações com cliente, funcionário, filme e endereço) demonstrou a importância da modelagem correta desde o início.

1. Separação de endereço em uma tabela única para clientes e locações:
 - O campo endereco em clientes é textual. Uma melhoria seria fazer clientes referenciar diretamente a tabela endereco, unificando e evitando redundância.
2. Adição de controle de devolução de filmes:
 - Incluir um campo data_devolucao em locacao para registrar quando o filme foi devolvido.
3. Criação de uma tabela de categorias:
 - Em vez de armazenar a categoria como texto, criar uma tabela categorias e fazer referência com id_categoria.
4. Validações de integridade mais robustas:

- Restrições de unicidade para CPF e e-mail em clientes e funcionários.

5. Interface de acesso ou aplicação web:

- Desenvolver uma interface para facilitar o uso por usuários finais, utilizando um front-end simples ou sistema em PHP/Node.js.

6. Histórico de alterações e auditoria:

- Adicionar tabelas ou triggers para registrar alterações em registros importantes, como locações e devoluções.