SENAI – GASPAR RICARDO JUNIOR

BANCO DE DADOS

GIOVANA RAFAELA DA SILVA

MARIA EDUARDA CLARO

PEDRO HENRIQUE DIAS DE PAULA SANTOS

TOBIAS PERASSI ALQUEZAR

PROFESSOR: ANDRÉ CASSULINO ARAÚJO SOUZA

DISCIPLINA: BANCO DE DADOS

9 DE JUNHO DE 2025+

Sumário

1. Introdução 3
2. Modelagem Conceitual 3
3. Modelagem Lógica 5
4. Discussão sobre a Normalização 5
5. Justificativa das Escolhas 6
6. Estrutura do Banco de Dados 6

1. 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 filmes, por exemplo, é essencial manter o controle preciso sobre os filmes disponíveis, os clientes atendidos e as reservas realizadas. 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 empreendimentos do ramo de locação de filmes, 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, filmes e locação, contribuindo assim para a modernização dos processos internos da empresa.

A base de dados desenvolvida é composta por cinco tabelas principais: clientes, endereço, filmes, funcionários e locação. 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.

2. 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

Cliente:

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

Endereço:

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

Locação:

- ID_Locação (chave primária)
- ID_Cliente (chave estrangeira)
- ID_Filme (chave estrangeira)
- ID_Funcionário (chave estrangeira)
- Data Início
- Data Fim
- Valor

Filmes:

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

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).

3. 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.

4. 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.

5. 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.

6. 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));

DML e DQL

INSERT INTO gerenciamento.endereco (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 (endereco, 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_endereco) VALUES ('2025-05-16', '2025-05-20', 15.00, 1, 1, 1, 1), ('2025-05-16', '2025-05-18', 12.00, 2, 2, 2, 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

CREATE USER admin_user WITH PASSWORD 'admin123'; CREATE USER normal_user WITH PASSWORD 'normal123';

GRANT ALL PRIVILEGES ON SCHEMA gerenciamento TO admin_user; GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA gerenciamento TO admin_user; GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA gerenciamento TO admin_user;

GRANT USAGE ON SCHEMA gerenciamento TO normal_user; GRANT SELECT ON ALL TABLES IN SCHEMA gerenciamento TO normal_user;

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;