

Enunciado do Exercício: Aluguel de locadora

Uma locadora de carros deseja criar um sistema simples para gerenciar seus aluguéis, com foco no cadastro de clientes, veículos e no controle dos contratos de aluguel. Para simplificar o sistema, as informações serão organizadas em no máximo três tabelas, utilizando as melhores práticas de normalização.

Requisitos:

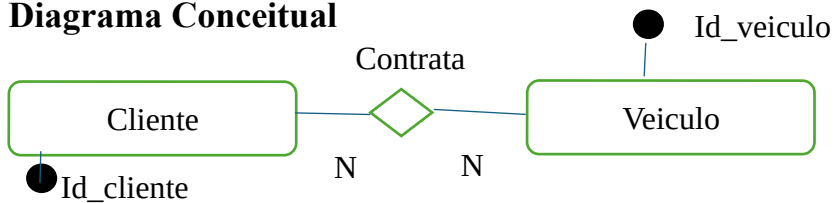
1. Clientes: Cada cliente tem um identificador único (ID), nome, e CPF.
2. Veículos: Cada veículo tem um identificador único (placa), modelo e fabricante.
3. Um Cliente pode alugar vários veículos ao longo do tempo, mas cada contrato envolve apenas um cliente e um veículo. Cada contrato de aluguel registra a data do aluguel, data de devolução, o valor total do aluguel e deve estar relacionado a um cliente e a um veículo.

4. Banco de dados é SQLite

Pergunta do Exercício:

- I. Monte o DER para este sistema (Conceitual).
- II. Forneça o projeto lógico das tabelas.
- III. Cuide para que as relações do projeto lógico estejam normalizadas na 1FN, 2FN e 3FN.
- IV. Explique por que cada tabela do modelo lógico está normalizada.

Diagrama Conceitual



Um cliente pode alugar vários veículos ao longo do tempo e um veículo pode ser alugado por vários clientes ao longo do tempo. Só que um cliente pode alugar um carro por vez.

Projeto Lógico

Cliente: (ID_Cliente, Nome, CPF)

Veículo: (Placa, Modelo, Fabricante)

Contrato de Aluguel: (ID_Contrato, Data_Aluguel, Data_Devolução, Valor_Aluguel, ID_Cliente, Placa) esta tabela resolve o relacionamento N para N. Perceba que contrato tem um tempo de vida e um cliente pode ter vários contratos, se usarmos a id_cliente e id_veiculo como chaves de únicas de contrato um cliente só vai poder realizar um único alugue. Assim, vamos incluir uma chave única id_contrato e as chaves de cliente e veículo serão definidas como chave estrangeira. Na prática o uso de chave composta não é desejado quando se desenvolve um sistema.

Cliente

- **ID_Cliente** int (PK)
- Nome text
- CPF int

Veículo

- **Placa** text (PK)
- Modelo text
- Fabricante text

Contrato

- **ID_Contrato** int (PK)
- Dia_aluguel int
- Mês_aluguel int
- Ano_aluguel int
- dia_devolucao int
- mês_devolucao int
- ano_devolucao int
- Valor_Total real
- ID_Cliente int (FK para cliente)
- Placa text (FK para veículo)

1ª Forma Normal (1FN):

- Elimina valores repetidos ou múltiplos valores em uma única coluna. As tabelas já estão na 1ª FN, pois todos os atributos contêm valores atômicos (únicos e indivisíveis).

2ª Forma Normal (2FN):

- Uma tabela está na 2ª FN se estiver na 1ª FN e todos os atributos que não são chaves dependem inteiramente da chave primária.
- **Cliente:** Todos os atributos (Nome, CPF) dependem unicamente de **ID_Cliente**.
- **Veículo:** Todos os atributos (Modelo, Fabricante) dependem unicamente de **Placa**.
- **Contrato:** Todos os atributos (Data_Aluguel, Data_Devolucao, Valor_Total) dependem de **ID_Contrato**, e as chaves estrangeiras (ID_Cliente, Placa) estão corretamente relacionadas.

3ª Forma Normal (3FN):

- Para estar na 3ª FN, a tabela deve estar na 2ª FN e não deve haver dependência transitiva (ou seja, nenhum atributo não chave deve depender de outro atributo não chave).
- As tabelas **Cliente**, **Veículo** e **Contrato** não possuem dependências transitivas, pois cada atributo depende diretamente de suas chaves primárias.