

Big Data - Modelagem SBD OLTP

Grupo

Alice Duarte Faria Ribeiro - DRE 122058907

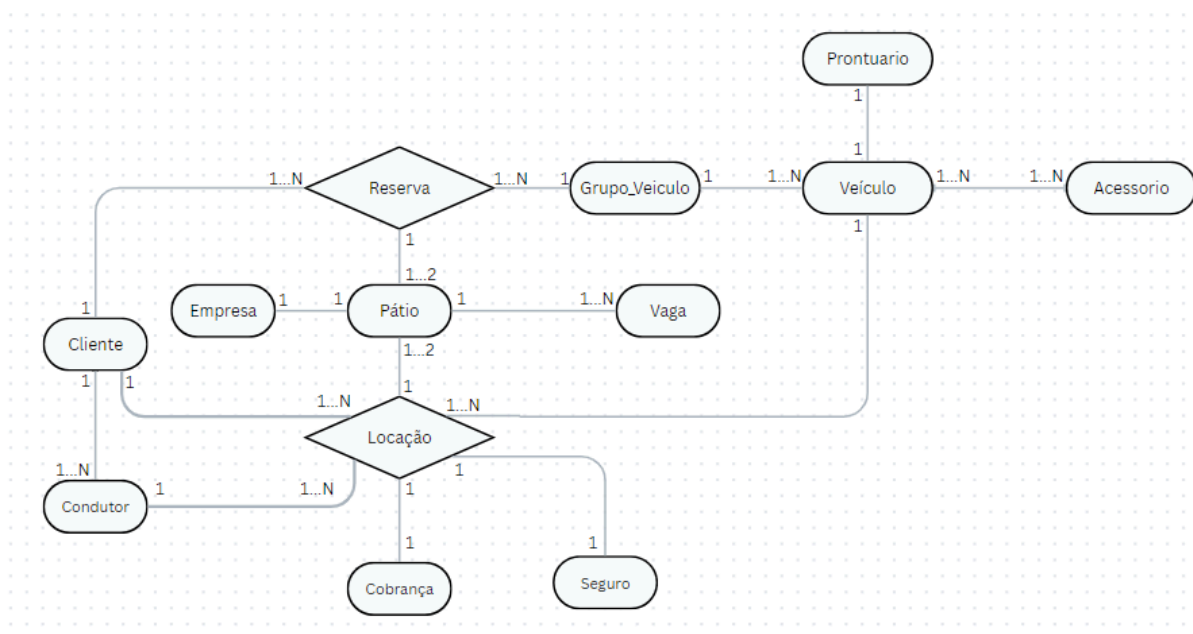
Beatriz Farias do Nascimento – DRE 122053127

Gustavo do Amaral Roxo Pereira - DRE 122081146

Link do github

<https://github.com/alicedfr/Big-Data-P2>

1. Modelo Conceitual



1.1. Descrição das entidades

Empresa: Representa cada uma das seis empresas de locação de veículos.

Pátio: Representa os locais físicos onde os veículos são retirados e devolvidos, com identificação de vagas.

Vaga: Espaços específicos dentro de um pátio.

Veículo: A frota de automóveis disponíveis para locação, com suas características detalhadas.

Grupo Veículo: Categorias ou classes de veículos para fins de precificação e luxo.

Acessório: Atributo adicional dos veículos (ex: ar-condicionado, cadeirinha).

Prontuário: Registro do histórico de manutenção e conservação do veículo.

Cliente: Pode ser Pessoa Física (PF) ou Pessoa Jurídica (PJ).

Condutor: A pessoa que efetivamente dirige o veículo, com detalhes de CNH.

Reserva: O registro de uma intenção de locação de veículo.

Locação: O contrato de aluguel de veículo propriamente dito.

Seguro: Representa diferentes *planos* de seguro que a locadora oferece, onde um plano pode incluir várias *coberturas*. O plano (ex: Básico, Completo, Premium) tem um conjunto predefinido de coberturas (ex: roubo, colisões, vidros).

Cobrança: O registro dos valores devidos e pagos pela locação.

1.2. Descrição dos relacionamentos

Uma **Empresa** opera um **Pátio**.

Um **Pátio** possui várias **Vagas**.

Um **Veículo** pertence a um **Grupo Veículo** e pode ter vários **Acessórios**.

Um **Acessório** pode pertencer a vários **Veículos**.

Um **Veículo** tem um **Prontuário Veículo**.

Um **Cliente** (PF ou PJ) pode fazer várias **Reservas** e **Locações**.

Uma **Reserva** é para um **Grupo Veículo** (inicialmente) e é associada a um **Cliente** e a um (ou dois) **Pátio(s)**.

Uma **Locação** é feita por um **Cliente**, utiliza um **Veículo**, tem um **Condutor**, está associada a um (ou dois) **Pátio(s)**, e pode ter um **Seguro**.

Uma **Cobrança** está associada a uma **Locação**.

3. Modelo Lógico

EMPRESA

- id_empresa (PK) - int
- nome_empresa - str(100)
- cnpj - str(14). Exemplo: XXXXXXXXXXXXXXXX
- endereço - str(255). Exemplo: Rua xxxxxx, Número xxx, Complemento xxxx
- telefone - str(13). Exemplo: 5521999999999

PATIO

- id_patio (PK) - int
- id_empresa (FK) - int
- nome_patio - str
- endereço - str(255)
- capacidade_vagas - int

VAGA

- id_vaga (PK) - int
- id_patio (FK) - int
- codigo_vaga - str

- status_vaga (Livre, Ocupada, Manutencao) - str

GRUPO_VEICULO

- id_grupo_veiculo (PK) - int
- nome_grupo - str
- descricao - str(255)
- valor_diaria_base - numeric

VEICULO

- id_veiculo (PK) - int
- id_grupo_veiculo (FK) - int
- id_patio_atual (FK - indica onde o veículo está atualmente) - int
- placa (Unique): str(7). Modelo antigo (ABC1234) ou novo (ABC1D23) de placa do Brasil.
- chassi (Unique): str(17). Exemplo: 9BWZZZ376TH004508
- marca: str
- modelo: str
- ano_fabricacao: int. A partir de 1900.
- cor: str
- tipo_mecanizacao (Manual, Automatica) - str
- quilometragem_atual: numeric
- url_foto: str
- status_veiculo (Disponivel, Alugado, Em Manutencao, Indisponivel) - str

ACESSORIO

- id_acessorio (PK) - int
- nome_acessorio: str
- descricao: str

VEICULO_ACESSORIO (Tabela Associativa)

- id_veiculo (PK, FK) - int
- id_acessorio (PK, FK) - int

PRONTUARIO

- id_prontuario (PK) - int
- id_veiculo (FK, Unique) - int
- data_ultima_revisao - date. Formato: DD/MM/YYYY
- quilometragem_ultima_revisao - numeric
- observacoes - str

CLIENTE

- id_cliente (PK) - int
- tipo_cliente (PF, PJ) - str(2)
- nome_razao_social - str

- cpf (Unique) - str(11)
- cnpj (Unique) - str(17)
- endereco - str
- telefone - str
- email - str

CONDUTOR

- id_condutor (PK) - int
- id_cliente (FK) - int
- nome_completo - str
- numero_cnh (Unique) - str(11)
- categoria_cnh - str(2). Exemplo: A, B, AB.
- data_expiracao_cnh - date. Formato: DD/MM/YYYY
- data_nascimento - date. Formato: DD/MM/YYYY

RESERVA

- id_reserva (PK) - int
- id_cliente (FK) - int
- id_grupo_veiculo (FK) - int
- id_patio_retirada_previsto (FK) - int
- data_hora_reserva - datetime. Formato: DD/MM/YYYY HH:mm
- data_hora_retirada_prevista - datetime. Formato: DD/MM/YYYY HH:mm
- data_hora_devolucao_prevista - datetime. Formato: DD/MM/YYYY HH:mm
- status_reserva (Confirmada, Cancelada, Em Espera) - str

LOCAÇÃO

- id_locacao (PK) - int
- id_cliente (FK) - int
- id_veiculo (FK) - int
- id_condutor (FK) - int
- id_patio_retirada_real (FK) - int
- id_patio_devolucao_prevista (FK) - int
- id_patio_devolucao_real (FK, Nullable) - int
- data_hora_retirada_real - datetime Formato: DD/MM/YYYY HH:mm
- data_hora_devolucao_prevista - datetime. Formato: DD/MM/YYYY HH:mm
- data_hora_devolucao_real (Nullable) - datetime. Formato: DD/MM/YYYY HH:mm
- quilometragem_retirada - numeric
- quilometragem_devolucao (Nullable) - numeric
- valor_total_previsto - numeric
- valor_total_final (Nullable) - numeric
- status_locacao (Ativa, Concluida, Cancelada) - str

SEGURO

- id_seguro (PK) - int
- nome_seguro - str
- descricao - str
- valor_diario - numeric

COBRANCA

- id_cobranca (PK) - int
- id_locacao (FK, Unique) - int
- data_cobranca - date. Formato: DD/MM/YYYY
- valor_base - numeric
- valor_multas_taxas - numeric
- valor_seguro - numeric
- valor_descontos - numeric
- valor_final_cobranca - numeric
- status_pagamento (Pendente, Pago, Cancelado) - str
- data_vencimento - date. Formato: DD/MM/YYYY
- data_pagamento (Nullable) - date. Formato: DD/MM/YYYY

4. Dicionário de Dados do Modelo do Banco de Dados

Tabela	Coluna	Tipo de Dado	Restrições de Integridade	Descrição
EMPRESA	id_empresa	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único da empresa.
	nome_empresa	VARCHAR(100)	NOT NULL, UNIQUE	Nome completo da empresa.
	cnpj	VARCHAR(14)	NOT NULL, UNIQUE	CNPJ da empresa.
	endereco	VARCHAR(255)	NOT NULL	Endereço completo da sede da empresa.
	telefone	VARCHAR(13)		Telefone de contato da empresa
PATIO	id_patio	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do pátio.
	id_empresa	INT	FOREIGN KEY REFERENCES EMPRESA(id_empresa), NOT NULL	Identificador da empresa proprietária do pátio.
	nome_patio	VARCHAR(100)	NOT NULL, UNIQUE	Nome do pátio (ex: "Pátio Aeroporto Galeão").
	endereco	VARCHAR(255)	NOT NULL	Endereço do pátio.

	capacidade_vagas	INT	NOT NULL, CHECK (capacidade_vagas > 0)	Número total de vagas disponíveis no pátio.
VAGA	id_vaga	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único da vaga.
	id_patio	INT	FOREIGN KEY REFERENCES PATIO(id_patio), NOT NULL	Identificador do pátio ao qual a vaga pertence.
	codigo_vaga	VARCHAR(20)	NOT NULL, UNIQUE (id_patio, codigo_vaga)	Código alfanumérico da vaga (ex: "A01", "B-10").
	status_vaga	VARCHAR(20)	NOT NULL, DEFAULT 'Livre', CHECK (status_vaga IN ('Livre', 'Ocupada', 'Manutencao'))	Status atual da vaga.
GRUPO_VEICULO	id_grupo_veiculo	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do grupo de veículo.
	nome_grupo	VARCHAR(50)	NOT NULL, UNIQUE	Nome do grupo/categoria de veículo (ex: "Econômico", "SUV", "Luxo").
	descricao	VARCHAR(255)		Descrição detalhada do grupo.
	valor_diaria_base	DECIMAL(10, 2)	NOT NULL, CHECK (valor_diaria_base > 0)	Valor base da diária de aluguel para veículos deste grupo.
VEICULO	id_veiculo	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do veículo.
	id_grupo_veiculo	INT	FOREIGN KEY REFERENCES GRUPO_VEICULO(id_grupo_veiculo), NOT NULL	Identificador do grupo ao qual o veículo pertence.
	id_patio_atual	INT	FOREIGN KEY REFERENCES PATIO(id_patio)	Identificador do pátio onde o veículo está atualmente. Pode ser NULL se o veículo estiver em trânsito.
	placa	VARCHAR(7)	NOT NULL, UNIQUE	Placa do veículo.
	chassi	VARCHAR(17)	NOT NULL, UNIQUE	Número do chassi do veículo.
	marca	VARCHAR(50)	NOT NULL	Marca do veículo.
	modelo	VARCHAR(50)	NOT NULL	Modelo do veículo.
	ano_fabricacao	INT	NOT NULL, CHECK (ano_fabricacao > 1900)	Ano de fabricação do veículo.
	cor	VARCHAR(30)		Cor do veículo.
	tipo_mecanizacao	VARCHAR(20)	NOT NULL, CHECK (tipo_mecanizacao IN ('Manual', 'Automatica'))	Tipo de transmissão do veículo.
	quilometragem_atual	DECIMAL(10, 2)	NOT NULL, CHECK (quilometragem_atual >= 0)	Quilometragem atual do veículo.
	url_foto	VARCHAR(255)		Url da foto do veículo.

	status_veiculo	VARCHAR(20)	NOT NULL, DEFAULT 'Disponivel', CHECK (status_veiculo IN ('Disponivel', 'Alugado', 'Em Manutencao', 'Indisponivel'))	Status atual de disponibilidade do veículo.
ACESSORIO	id_acessorio_caracteristica	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do acessório ou característica.
	nome_acessorio	VARCHAR(100)	NOT NULL, UNIQUE	Nome do acessório ou característica (ex: "Ar-Condicionado", "Cadeirinha").
	descricao	VARCHAR(255)		Descrição detalhada do acessório ou característica.
VEICULO_ACESSORIO	id_veiculo	INT	PRIMARY KEY, FOREIGN KEY REFERENCES VEICULO(id_veiculo), NOT NULL	Identificador do veículo.
	id_acessorio	INT	PRIMARY KEY, FOREIGN KEY REFERENCES ACESSORIO_CARACTERISTICA(id_acessorio_caracteristica), NOT NULL	Identificador do acessório.
PRONTUARIO	id_prontuario	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do prontuário.
	id_veiculo	INT	FOREIGN KEY REFERENCES VEICULO(id_veiculo), NOT NULL, UNIQUE	Identificador do veículo ao qual o prontuário pertence.
	data_ultima_revisao	DATE		Data da última revisão do veículo.
	quilometragem_ultima_revisao	DECIMAL(10, 2)	CHECK (quilometragem_ultima_revisao >= 0)	Quilometragem do veículo na última revisão.
	observacoes	TEXT		Campo para observações gerais sobre o estado de conservação e revisões.
CLIENTE	id_cliente	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do cliente.
	tipo_cliente	VARCHAR(2)	NOT NULL, CHECK (tipo_cliente IN ('PF', 'PJ'))	Tipo de cliente (Pessoa Física ou Pessoa Jurídica).
	nome_razao_social	VARCHAR(100)	NOT NULL	Nome completo (PF) ou Razão Social (PJ) do cliente.
	cpf_cnpj	VARCHAR(17)	NOT NULL, UNIQUE	CPF (PF) ou CNPJ (PJ) do cliente.
	endereço	VARCHAR(255)	NOT NULL	Endereço completo do cliente.
	telefone	VARCHAR(20)		Telefone de contato do cliente.

	email	VARCHAR(100)	UNIQUE	E-mail do cliente.
CONDUTOR	id_condutor	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do condutor.
	id_cliente	INT	FOREIGN KEY REFERENCES CLIENTE(id_cliente)	Identificador do Cliente.
	nome_completo	VARCHAR(100)	NOT NULL	Nome completo do condutor.
	numero_cnh	VARCHAR(11)	NOT NULL, UNIQUE	Número da Carteira Nacional de Habilitação (CNH).
	categoria_cnh	VARCHAR(10)	NOT NULL	Categoria da CNH (ex: "B", "AB").
	data_expiracao_cnh	DATE	NOT NULL	Data de expiração da CNH.
	data_nascimento	DATE	NOT NULL	Data de nascimento do condutor.
RESERVA	id_reserva	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único da reserva.
	id_cliente	INT	FOREIGN KEY REFERENCES CLIENTE(id_cliente), NOT NULL	Identificador do cliente que fez a reserva.
	id_grupo_veiculo	INT	FOREIGN KEY REFERENCES GRUPO_VEICULO(id_grupo_veiculo), NOT NULL	Identificador do grupo de veículo reservado.
	id_patio_retirada_previsto	INT	FOREIGN KEY REFERENCES PATIO(id_patio), NOT NULL	Identificador do pátio onde o cliente pretende retirar o veículo.
	data_hora_reserva	DATETIME	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Data e hora em que a reserva foi feita.
	data_hora_retirada_prevista	DATETIME	NOT NULL, CHECK (data_hora_retirada_prevista > data_hora_reserva)	Data e hora prevista para a retirada do veículo.
	data_hora_devolucao_prevista	DATETIME	NOT NULL, CHECK (data_hora_devolucao_prevista > data_hora_retirada_prevista)	Data e hora prevista para a devolução do veículo.
	status_reserva	VARCHAR(20)	NOT NULL, DEFAULT 'Confirmada', CHECK (status_reserva IN ('Confirmada', 'Cancelada', 'Em Espera'))	Status atual da reserva.
LOCACAO	id_locacao	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único da locação.
	id_cliente	INT	FOREIGN KEY	Identificador do cliente locatário.

		REFERENCES CLIENTE(id_cliente), NOT NULL	
id_veiculo	INT	FOREIGN KEY REFERENCES VEICULO(id_veiculo), NOT NULL	Identificador do veículo alugado.
id_condutor	INT	FOREIGN KEY REFERENCES CONDUTOR(id_condutor), NOT NULL	Identificador do condutor principal do veículo.
id_patio_retirada _real	INT	FOREIGN KEY REFERENCES PATIO(id_patio), NOT NULL	Identificador do pátio real de retirada do veículo.
id_patio_devoluc ao_prevista	INT	FOREIGN KEY REFERENCES PATIO(id_patio), NOT NULL	Identificador do pátio previsto para a devolução do veículo.
id_patio_devoluc ao_real	INT	FOREIGN KEY REFERENCES PATIO(id_patio)	Identificador do pátio real de devolução do veículo (NULL até a devolução).
data_hora_retira da_real	DATETIME	NOT NULL	Data e hora real da retirada do veículo.
data_hora_devol ucao_prevista	DATETIME	NOT NULL, CHECK (data_hora_devolucao_previ sta > data_hora_retirada_real)	Data e hora prevista para a devolução do veículo.
data_hora_devol ucao_real	DATETIME	CHECK (data_hora_devolucao_real >= data_hora_retirada_real)	Data e hora real da devolução do veículo (NULL até a devolução).
quilometragem_r etirada	DECIMAL(10, 2)	NOT NULL, CHECK (quilometragem_retirada >= 0)	Quilometragem do veículo no momento da retirada.
quilometragem_ devolucao	DECIMAL(10, 2)	CHECK (quilometragem_devolucao >= quilometragem_retirada)	Quilometragem do veículo no momento da devolução (NULL até a devolução).
valor_total_previ sto	DECIMAL(10, 2)	NOT NULL, CHECK (valor_total_previsto >= 0)	Valor total previsto da locação.
valor_total_final	DECIMAL(10, 2)	CHECK (valor_total_final >= 0)	Valor total final da locação, após ajustes (NULL até a conclusão).
status_locacao	VARCHAR(20)	NOT NULL, DEFAULT 'Ativa', CHECK (status_locacao IN ('Ativa', 'Concluida', 'Cancelada'))	Status atual da locação.
SEGURO	id_seguro	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único do seguro.
	nome_seguro	VARCHAR(100) NOT NULL, UNIQUE	Nome do plano de seguro (ex:

				Básico, Completo, Premium)
	descricao	VARCHAR(255)		Descrição detalhada da(s) cobertura(s) do seguro (ex: roubo, colisões, vidros).
	valor_diario	DECIMAL(10, 2)	NOT NULL, CHECK (valor_diario >= 0)	Valor diário do plano de seguro.
COBRANCA	id_cobranca	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Identificador único da cobrança.
	id_locacao	INT	FOREIGN KEY REFERENCES LOCACAO(id_locacao), NOT NULL, UNIQUE	Identificador da locação à qual a cobrança se refere.
	data_cobranca	DATE	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Data em que a cobrança foi gerada.
	valor_base	DECIMAL(10, 2)	NOT NULL, CHECK (valor_base >= 0)	Valor base da locação antes de multas, taxas e proteções.
	valor_multas_taxas	DECIMAL(10, 2)	NOT NULL, DEFAULT 0.00, CHECK (valor_multas_taxas >= 0)	Valor de multas, taxas por atraso, ou outras.
	valor_seguro	DECIMAL(10, 2)	NOT NULL, DEFAULT 0.00, CHECK (valor_seguro >= 0)	Valor das proteções adicionais contratadas.
	valor_descontos	DECIMAL(10, 2)	NOT NULL, DEFAULT 0.00, CHECK (valor_descontos >= 0)	Valor de descontos aplicados.
	valor_final_cobranca	DECIMAL(10, 2)	NOT NULL, CHECK (valor_final_cobranca >= 0)	Valor total final da cobrança.
	status_pagamento	VARCHAR(20)	NOT NULL, DEFAULT 'Pendente', CHECK (status_pagamento IN ('Pendente', 'Pago', 'Cancelado'))	Status do pagamento da cobrança.
	data_vencimento	DATE	NOT NULL, CHECK (data_vencimento >= data_cobranca)	Data de vencimento da cobrança.
	data_pagamento	DATE	CHECK (data_pagamento >= data_cobranca)	Data em que o pagamento foi realizado (NULL se não pago).

5. Modelo Físico

```
CREATE DATABASE locadora_veiculos;
```

```
USE locadora_veiculos;
```

```
-- Criação da tabela EMPRESA
```

```

CREATE TABLE EMPRESA (
    id_empresa INT PRIMARY KEY AUTO_INCREMENT,
    nome_empresa VARCHAR(100) NOT NULL UNIQUE,
    cnpj VARCHAR(14) NOT NULL UNIQUE,
    endereco VARCHAR(255) NOT NULL,
    telefone VARCHAR(13)
);

-- Criação da tabela PATIO
CREATE TABLE PATIO (
    id_patio INT PRIMARY KEY AUTO_INCREMENT,
    id_empresa INT NOT NULL,
    nome_patio VARCHAR(100) NOT NULL UNIQUE,
    endereco VARCHAR(255) NOT NULL,
    capacidade_vagas INT NOT NULL CHECK (capacidade_vagas > 0),
    FOREIGN KEY (id_empresa) REFERENCES EMPRESA(id_empresa)
);

-- Criação da tabela VAGA
CREATE TABLE VAGA (
    id_vaga INT PRIMARY KEY AUTO_INCREMENT,
    id_patio INT NOT NULL,
    codigo_vaga VARCHAR(20) NOT NULL,
    status_vaga VARCHAR(20) NOT NULL DEFAULT 'Livre' CHECK (status_vaga IN
('Livre', 'Ocupada', 'Manutencao')),
    FOREIGN KEY (id_patio) REFERENCES PATIO(id_patio),
    UNIQUE (id_patio, codigo_vaga) -- Garante que o código da vaga seja único
dentro de um pátio
);

-- Criação da tabela GRUPO_VEICULO
CREATE TABLE GRUPO_VEICULO (
    id_grupo_veiculo INT PRIMARY KEY AUTO_INCREMENT,
    nome_grupo VARCHAR(50) NOT NULL UNIQUE,
    descricao VARCHAR(255),
    valor_diaria_base DECIMAL(10, 2) NOT NULL CHECK (valor_diaria_base > 0)
);

-- Criação da tabela VEICULO
CREATE TABLE VEICULO (
    id_veiculo INT PRIMARY KEY AUTO_INCREMENT,
    id_grupo_veiculo INT NOT NULL,
    id_patio_atual INT, -- Pode ser NULL se o veículo estiver em trânsito ou fora de
um pátio específico

```

```

placa VARCHAR(7) NOT NULL UNIQUE,
chassi VARCHAR(17) NOT NULL UNIQUE,
marca VARCHAR(50) NOT NULL,
modelo VARCHAR(50) NOT NULL,
ano_fabricacao INT NOT NULL CHECK (ano_fabricacao > 1900),
cor VARCHAR(30),
tipo_mecanizacao VARCHAR(20) NOT NULL CHECK (tipo_mecanizacao IN
('Manual', 'Automatica')),
quilometragem_atual DECIMAL(10, 2) NOT NULL CHECK (quilometragem_atual
>= 0),
url_foto VARCHAR(255),
status_veiculo VARCHAR(20) NOT NULL DEFAULT 'Disponivel' CHECK
(status_veiculo IN ('Disponivel', 'Alugado', 'Em Manutencao', 'Indisponivel')),
FOREIGN KEY (id_grupo_veiculo) REFERENCES
GRUPO_VEICULO(id_grupo_veiculo),
FOREIGN KEY (id_patio_atual) REFERENCES PATIO(id_patio)
);

```

-- Criação da tabela ACESSORIO

```

CREATE TABLE ACESSORIO (
    id_acessorio INT PRIMARY KEY AUTO_INCREMENT,
    nome_acessorio VARCHAR(100) NOT NULL UNIQUE,
    descricao VARCHAR(255)
);

```

-- Criação da tabela associativa VEICULO_ACESSORIO

```

CREATE TABLE VEICULO_ACESSORIO (
    id_veiculo INT NOT NULL,
    id_acessorio INT NOT NULL,
    PRIMARY KEY (id_veiculo, id_acessorio),
    FOREIGN KEY (id_veiculo) REFERENCES VEICULO(id_veiculo),
    FOREIGN KEY (id_acessorio) REFERENCES ACESSORIO(id_acessorio)
);

```

-- Criação da tabela PRONTUARIO

```

CREATE TABLE PRONTUARIO (
    id_prontuario INT PRIMARY KEY AUTO_INCREMENT,
    id_veiculo INT NOT NULL UNIQUE,
    data_ultima_revisao DATE,
    quilometragem_ultima_revisao DECIMAL(10, 2) CHECK
(kilometragem_ultima_revisao >= 0),
observacoes VARCHAR(255),
FOREIGN KEY (id_veiculo) REFERENCES VEICULO(id_veiculo)
);

```

-- Criação da tabela CLIENTE

```
CREATE TABLE CLIENTE (  
    id_cliente INT PRIMARY KEY AUTO_INCREMENT,  
    tipo_cliente VARCHAR(2) NOT NULL CHECK (tipo_cliente IN ('PF', 'PJ')),  
    nome_razao_social VARCHAR(100) NOT NULL,  
    cpf VARCHAR(11) UNIQUE,  
    cnpj VARCHAR(17) UNIQUE,  
    endereco VARCHAR(255) NOT NULL,  
    telefone VARCHAR(13),  
    email VARCHAR(100) UNIQUE,  
    CHECK (cpf IS NOT NULL OR cnpj IS NOT NULL)  
);
```

-- Criação da tabela CONDUTOR

```
CREATE TABLE CONDUTOR (  
    id_condutor INT PRIMARY KEY AUTO_INCREMENT,  
    id_cliente INT NOT NULL,  
    nome_completo VARCHAR(100) NOT NULL,  
    numero_cnh VARCHAR(11) NOT NULL UNIQUE,  
    categoria_cnh VARCHAR(2) NOT NULL,  
    data_expiracao_cnh DATE NOT NULL,  
    data_nascimento DATE NOT NULL,  
    FOREIGN KEY (id_cliente) REFERENCES CLIENTE(id_cliente)  
);
```

-- Criação da tabela RESERVA

```
CREATE TABLE RESERVA (  
    id_reserva INT PRIMARY KEY AUTO_INCREMENT,  
    id_cliente INT NOT NULL,  
    id_grupo_veiculo INT NOT NULL,  
    id_patio_retirada_previsto INT NOT NULL,  
    data_hora_reserva DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    data_hora_retirada_prevista DATETIME NOT NULL,  
    data_hora_devolucao_prevista DATETIME NOT NULL,  
    status_reserva VARCHAR(20) NOT NULL DEFAULT 'Confirmada' CHECK  
(status_reserva IN ('Confirmada', 'Cancelada', 'Em Espera')),  
    FOREIGN KEY (id_cliente) REFERENCES CLIENTE(id_cliente),  
    FOREIGN KEY (id_grupo_veiculo) REFERENCES  
GRUPO_VEICULO(id_grupo_veiculo),  
    FOREIGN KEY (id_patio_retirada_previsto) REFERENCES PATIO(id_patio)  
);
```

-- Trigger para BEFORE INSERT na tabela RESERVA

DELIMITER //

```
CREATE TRIGGER trg_reserva_before_insert
BEFORE INSERT ON RESERVA
FOR EACH ROW
BEGIN
    -- Verifica se data_hora_retirada_prevista é maior que data_hora_reserva
    IF NEW.data_hora_retirada_prevista <= NEW.data_hora_reserva THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Erro: A data e hora de retirada prevista deve ser
posterior à data e hora da reserva.';
    END IF;

    -- Verifica se data_hora_devolucao_prevista é maior que
data_hora_retirada_prevista
    IF NEW.data_hora_devolucao_prevista <= NEW.data_hora_retirada_prevista
THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Erro: A data e hora de devolução prevista deve ser
posterior à data e hora de retirada prevista.';
    END IF;
END;
//
```

DELIMITER ;

-- Trigger para BEFORE UPDATE na tabela RESERVA

DELIMITER //

```
CREATE TRIGGER trg_reserva_before_update
BEFORE UPDATE ON RESERVA
FOR EACH ROW
BEGIN
    -- Verifica se data_hora_retirada_prevista é maior que data_hora_reserva
    IF NEW.data_hora_retirada_prevista <= NEW.data_hora_reserva THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Erro: A data e hora de retirada prevista deve ser
posterior à data e hora da reserva.';
    END IF;

    -- Verifica se data_hora_devolucao_prevista é maior que
data_hora_retirada_prevista
```

```

    IF NEW.data_hora_devolucao_prevista <= NEW.data_hora_retirada_prevista
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Erro: A data e hora de devolução prevista deve ser
posterior à data e hora de retirada prevista.';
    END IF;
END;
//

DELIMITER ;

-- Criação da tabela LOCACAO
CREATE TABLE LOCACAO (
    id_locacao INT PRIMARY KEY AUTO_INCREMENT,
    id_reserva INT UNIQUE, -- Pode ser NULL para locações diretas (sem reserva
prévia)
    id_cliente INT NOT NULL,
    id_veiculo INT NOT NULL,
    id_condutor INT NOT NULL,
    id_patio_retirada_real INT NOT NULL,
    id_patio_devolucao_prevista INT NOT NULL,
    id_patio_devolucao_real INT, -- Pode ser NULL até a devolução
data_hora_retirada_real DATETIME NOT NULL,
    data_hora_devolucao_prevista DATETIME NOT NULL, -- CHECK constraint
removed here
    data_hora_devolucao_real DATETIME, -- Pode ser NULL até a devolução
quilometragem_retirada DECIMAL(10, 2) NOT NULL CHECK
(quilometragem_retirada >= 0),
    quilometragem_devolucao DECIMAL(10, 2), -- CHECK constraint removed here
valor_total_previsto DECIMAL(10, 2) NOT NULL CHECK (valor_total_previsto >=
0),
    valor_total_final DECIMAL(10, 2) CHECK (valor_total_final >= 0), -- NULL até a
conclusão da locação
    status_locacao VARCHAR(20) NOT NULL DEFAULT 'Ativa' CHECK
(status_locacao IN ('Ativa', 'Concluida', 'Cancelada')),
    FOREIGN KEY (id_reserva) REFERENCES RESERVA(id_reserva),
    FOREIGN KEY (id_cliente) REFERENCES CLIENTE(id_cliente),
    FOREIGN KEY (id_veiculo) REFERENCES VEICULO(id_veiculo),
    FOREIGN KEY (id_condutor) REFERENCES CONDUTOR(id_condutor),
    FOREIGN KEY (id_patio_retirada_real) REFERENCES PATIO(id_patio),
    FOREIGN KEY (id_patio_devolucao_prevista) REFERENCES PATIO(id_patio),
    FOREIGN KEY (id_patio_devolucao_real) REFERENCES PATIO(id_patio)
);

```

```
-- Trigger para BEFORE INSERT na tabela LOCACAO  
DELIMITER //
```

```
CREATE TRIGGER trg_locacao_before_insert  
BEFORE INSERT ON LOCACAO  
FOR EACH ROW  
BEGIN  
    IF NEW.data_hora_devolucao_prevista <= NEW.data_hora_retirada_real THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Erro: A data e hora de devolução prevista deve ser  
posterior à data e hora de retirada real.';  
    END IF;  
  
    IF NEW.data_hora_devolucao_real IS NOT NULL AND  
NEW.data_hora_devolucao_real < NEW.data_hora_retirada_real THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Erro: A data e hora de devolução real não pode ser  
anterior à data e hora de retirada real.';  
    END IF;  
  
    IF NEW.quilometragem_devolucao IS NOT NULL AND  
NEW.quilometragem_devolucao < NEW.quilometragem_retirada THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Erro: A quilometragem de devolução não pode ser  
menor que a quilometragem de retirada.';  
    END IF;  
END;  
//
```

```
-- Trigger para BEFORE UPDATE na tabela LOCACAO  
CREATE TRIGGER trg_locacao_before_update  
BEFORE UPDATE ON LOCACAO  
FOR EACH ROW  
BEGIN  
    IF NEW.data_hora_devolucao_prevista <= NEW.data_hora_retirada_real THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Erro: A data e hora de devolução prevista deve ser  
posterior à data e hora de retirada real.';  
    END IF;  
  
    IF NEW.data_hora_devolucao_real IS NOT NULL AND  
NEW.data_hora_devolucao_real < NEW.data_hora_retirada_real THEN  
        SIGNAL SQLSTATE '45000'
```



```
        SET MESSAGE_TEXT = 'Erro: A data e hora de devolução real não pode ser  
anterior à data e hora de retirada real.';  
    END IF;
```

```
    IF NEW.quilometragem_devolucao IS NOT NULL AND  
NEW.quilometragem_devolucao < NEW.quilometragem_retirada THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Erro: A quilometragem de devolução não pode ser  
menor que a quilometragem de retirada.';  
    END IF;  
END;  
//
```

```
DELIMITER ;
```

```
-- Criação da tabela SEGURO
```

```
CREATE TABLE SEGURO (  
    id_seguro INT PRIMARY KEY AUTO_INCREMENT,  
    nome_seguro VARCHAR(100) NOT NULL UNIQUE,  
    descricao VARCHAR(255),  
    valor_diario DECIMAL(10, 2) NOT NULL CHECK (valor_diario >= 0)  
);
```

```
-- Criação da tabela COBRANCA
```

```
CREATE TABLE COBRANCA (  
    id_cobranca INT PRIMARY KEY AUTO_INCREMENT,  
    id_locacao INT NOT NULL UNIQUE,  
    data_cobranca DATE NOT NULL DEFAULT (CURRENT_DATE),  
    valor_base DECIMAL(10, 2) NOT NULL CHECK (valor_base >= 0),  
    valor_multas_taxas DECIMAL(10, 2) NOT NULL DEFAULT 0.00 CHECK  
(valor_multas_taxas >= 0),  
    valor_seguro DECIMAL(10, 2) NOT NULL DEFAULT 0.00 CHECK (valor_seguro  
>= 0),  
    valor_descontos DECIMAL(10, 2) NOT NULL DEFAULT 0.00 CHECK  
(valor_descontos >= 0),  
    valor_final_cobranca DECIMAL(10, 2) NOT NULL CHECK (valor_final_cobranca  
>= 0),  
    status_pagamento VARCHAR(20) NOT NULL DEFAULT 'Pendente' CHECK  
(status_pagamento IN ('Pendente', 'Pago', 'Cancelado')),  
    data_vencimento DATE NOT NULL,  
    data_pagamento DATE,  
    FOREIGN KEY (id_locacao) REFERENCES LOCACAO(id_locacao)  
);
```

-- Trigger para BEFORE INSERT na tabela COBRANCA

DELIMITER //

CREATE TRIGGER trg_cobranca_before_insert

BEFORE INSERT ON COBRANCA

FOR EACH ROW

BEGIN

-- Rule: data_vencimento >= data_cobranca (comparing dates only)

IF NEW.data_vencimento < DATE(NEW.data_cobranca) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = 'Erro: A data de vencimento não pode ser anterior à data da cobrança.';

END IF;

-- Rule: data_pagamento (if not null) >= data_cobranca

IF NEW.data_pagamento IS NOT NULL AND NEW.data_pagamento < NEW.data_cobranca THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = 'Erro: A data de pagamento não pode ser anterior à data da cobrança.';

END IF;

END;

//

DELIMITER ;

-- Trigger para BEFORE UPDATE na tabela COBRANCA

DELIMITER //

CREATE TRIGGER trg_cobranca_before_update

BEFORE UPDATE ON COBRANCA

FOR EACH ROW

BEGIN

-- Rule: data_vencimento >= data_cobranca (comparing dates only)

IF NEW.data_vencimento < DATE(NEW.data_cobranca) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = 'Erro: A data de vencimento não pode ser anterior à data da cobrança.';

END IF;

-- Rule: data_pagamento (if not null) >= data_cobranca

```
    IF NEW.data_pagamento IS NOT NULL AND NEW.data_pagamento <
NEW.data_cobranca THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Erro: A data de pagamento não pode ser anterior à
data da cobrança.';
    END IF;
END;
//

DELIMITER ;
```