GUIA PRÁTICO DE MODELAGEM DE DADOS LOCADORA DE VEÍCULOS

Documento para alunos de modelagem de dados, SQL e SQLite
Baseado em levantamento simulado do analista de negócios

Objetivo pedagógico

Este documento foi escrito como se fosse uma entrega formal do analista de negócios para a equipe técnica. O objetivo é orientar o aluno, passo a passo, na criação de um banco de dados completo para uma locadora de veículos. O aluno deverá entender as entidades, as dependências, as chaves primárias simples e compostas, as chaves estrangeiras, as relações 1:N e N:M, a ordem correta de criação das tabelas e as operações de CRUD e pesquisa avançada em SQLite.

Observação didática importante: embora o SQLite seja flexível com tipos, neste material os campos foram especificados com tamanhos lógicos para treinar o aluno como se estivesse modelando também para bancos mais rígidos. Portanto, use VARCHAR(n) na maior parte dos campos textuais. Use TEXT apenas em observações, comentários e descrições extensas.

Cenário de negócio levantado pelo analista

- A empresa Locadora Prime Veículos atua com locação diária, semanal e mensal de veículos de passeio e utilitários.
- A empresa possui clientes pessoa física e pessoa jurídica, funcionários administrativos e operacionais, convênio com bancos para análise e aprovação de caução/crédito e múltiplos estacionamentos de apoio.
- Cada veículo pode entrar e sair do pátio diversas vezes. Toda movimentação deve ser rastreada, com data, hora, funcionário responsável, vaga de origem e vaga de destino.
- Toda locação deve registrar datas previstas e efetivas de retirada e devolução, funcionário atendente, cliente titular, veículo escolhido, situação da locação e banco responsável pela aprovação financeira quando aplicável.
- Na retirada do veículo deve existir inspeção de saída. No retorno, inspeção de entrada. Ambas devem permitir registrar quilometragem, combustível, avarias e observações.
- Uma locação pode ter condutores adicionais. Isso caracteriza relação N:M entre locações e clientes.

Entidades identificadas

- clientes
- funcionarios
- bancos
- estacionamentos
- vagas_estacionamento
- veiculos
- locacoes
- locacao_condutores
- inspecoes_saida
- inspecoes_entrada

- patio_movimentacoes

Mapa de dependências e cardinalidades

Origem	Destino	Cardinalidade	Implementação	Comentário didático
clientes	locacoes	1:N	locacoes.id_cliente -> clientes.id_cliente	Um cliente pode realizar várias locações.
funcionarios	locacoes	1:N	locacoes.id_funcionario_abertura -> funcionarios.id_funcionario	Um funcionário pode abrir várias locações.
bancos	locacoes	1:N	locacoes.id_banco_aprovacao -> bancos.id_banco	Um banco pode aprovar várias locações.
estacionamentos	vagas_estacionamento	1:N	vagas_estacionamento.id_estacionamento -> estacionamentos.id_estacionamento	Um estacionamento possui várias vagas.
veiculos	locacoes	1:N histórico	locacoes.id_veiculo -> veiculos.id_veiculo	Um veículo participa de várias locações ao longo do tempo, mas apenas uma locação ativa por vez.
locacoes	locacao_condutores	1:N	locacao_condutores.id_locacao -> locacoes.id_locacao	A locação possui zero ou vários condutores adicionais.
clientes	locacao_condutores	1:N	locacao_condutores.id_cliente_condutor -> clientes.id_cliente	Um cliente pode ser condutor adicional em várias locações.
locacoes	inspecoes_saida	1:1	inspecoes_saida.id_locacao UNIQUE	Uma locação deve ter no máximo uma inspeção de saída.
locacoes	inspecoes_entrada	1:1	inspecoes_entrada.id_locacao UNIQUE	Uma locação deve ter no máximo uma inspeção de entrada.
veiculos	patio_movimentacoes	1:N	patio_movimentacoes.id_veiculo -> veiculos.id_veiculo	Um veículo pode ter várias movimentações.
vagas_estacionamento	patio_movimentacoes	1:N	FK composta: (id_estacionamento, codigo_vaga)	A movimentação referencia a vaga de origem ou de destino por chave composta.
locacoes	clientes	N:M	Tabela associativa locacao_condutores	Relação N:M resolvida com chave composta (id_locacao, id_cliente_condutor).

Atenção: em tabelas associativas, a chave primária normalmente é composta pelas chaves estrangeiras. Isso força o aluno a entender dependência N:M e evita duplicidade lógica.

Ordem recomendada para criação das tabelas

1. 1. Ative o controle de integridade referencial no SQLite com PRAGMA foreign_keys = ON;.
2. 2. Crie primeiro as tabelas sem dependência: clientes, funcionarios, bancos e estacionamentos.
3. 3. Depois crie vagas_estacionamento, pois ela depende de estacionamentos e usa chave primária composta.
4. 4. Em seguida crie veiculos.
5. 5. Depois crie locacoes, pois ela depende de clientes, funcionarios, bancos e veiculos.
6. 6. Crie locacao_condutores após locacoes e clientes.
7. 7. Crie inspecoes_saida e inspecoes_entrada após locacoes e funcionarios.
8. 8. Por fim crie patio_movimentacoes, pois ela depende de veiculos, funcionarios, estacionamentos e vagas_estacionamento.
9. 9. Somente depois da estrutura pronta execute os INSERTs de carga inicial.
10. 10. Após inserir os dados, teste UPDATE, DELETE e SELECT com joins para validar a modelagem.

Dicionário de dados detalhado

Tabela: clientes

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_cliente	INTEGER	-	Não	PK	Identificador do cliente	Autoincrement
tipo_cliente	VARCHAR	2	Não	-	PF ou PJ	Usar CHECK
nome_razao	VARCHAR	150	Não	-	Nome completo ou razão social	
nome_fantasia	VARCHAR	150	Sim	-	Nome fantasia para PJ	
cpf_cnpj	VARCHAR	18	Não	UNIQUE	Documento principal	Formatado
rg_ie	VARCHAR	20	Sim	-	RG ou inscrição estadual	
cnh_numero	VARCHAR	20	Sim	-	Número da CNH	Obrigatório para condutor
cnh_categoria	VARCHAR	5	Sim	-	Categoria da CNH	
telefone	VARCHAR	20	Sim	-	Telefone principal	
email	VARCHAR	120	Sim	UNIQUE	E-mail principal	
logradouro	VARCHAR	120	Sim	-	Rua/avenida	
numero	VARCHAR	10	Sim	-	Número do endereço	
bairro	VARCHAR	60	Sim	-	Bairro	
cidade	VARCHAR	60	Sim	-	Cidade	

uf	VARCHAR	2	Sim	-	UF	
cep	VARCHAR	9	Sim	-	CEP	
status_cliente	VARCHAR	15	Não	-	ATIVO/ BLOQUEADO/ INATIVO	
observacoes	TEXT	-	Sim	-	Observações longas	Único campo textual longo

Tabela: funcionarios

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_funcionario	INTEGER	-	Não	PK	Identificador do funcionário	Autoincrement
nome	VARCHAR	120	Não	-	Nome do funcionário	
cpf	VARCHAR	14	Não	UNIQUE	CPF	
cargo	VARCHAR	60	Não	-	Cargo na empresa	
telefone	VARCHAR	20	Sim	-	Telefone	
email	VARCHAR	120	Sim	UNIQUE	E-mail	
matricula	VARCHAR	20	Não	UNIQUE	Matrícula interna	
status_funcionario	VARCHAR	15	Não	-	ATIVO/ INATIVO	
comentarios	TEXT	-	Sim	-	Comentários longos	TEXT permitido

Tabela: bancos

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_banco	INTEGER	-	Não	PK	Identificador do banco	Autoincrement
codigo_banco	VARCHAR	10	Não	UNIQUE	Código bancário	
nome_banco	VARCHAR	100	Não	-	Nome do banco	
contato_setor	VARCHAR	80	Sim	-	Contato responsável	
telefone	VARCHAR	20	Sim	-	Telefone	
email	VARCHAR	120	Sim	-	E-mail de contato	
status_banco	VARCHAR	15	Não	-	ATIVO/ INATIVO	
observacoes	TEXT	-	Sim	-	Observações	TEXT

					longas	permitido

Tabela: estacionamentos

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_estacionamento	INTEGER	-	Não	PK	Identificador do estacionamento	Autoincrement
nome_estacionamento	VARCHAR	100	Não	-	Nome da unidade/pátio	
tipo_estacionamento	VARCHAR	20	Não	-	MATRIZ/ FILIAL/APOIO	
logradouro	VARCHAR	120	Não	-	Endereço	
numero	VARCHAR	10	Sim	-	Número	
bairro	VARCHAR	60	Sim	-	Bairro	
cidade	VARCHAR	60	Não	-	Cidade	
uf	VARCHAR	2	Não	-	UF	
cep	VARCHAR	9	Sim	-	CEP	
capacidade_total	INTEGER	-	Não	-	Quantidade de vagas	
status_estacionamento	VARCHAR	15	Não	-	ATIVO/ INATIVO	
observacoes	TEXT	-	Sim	-	Observações do pátio	TEXT permitido

Tabela: vagas_estacionamento (chave composta)

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_estacionamento	INTEGER	-	Não	PK/FK	Estacionamento da vaga	Parte 1 da PK composta
codigo_vaga	VARCHAR	10	Não	PK	Código visível da vaga	Parte 2 da PK composta
setor	VARCHAR	30	Não	-	Setor interno	
tipo_vaga	VARCHAR	20	Não	-	PADRAO/ LAVAGEM/ MANUTENCAO/ENTREGA	
coberta	VARCHAR	1	Não	-	S ou N	
status_vaga	VARCHAR	15	Não	-	LIVRE/ OCUPADA/ BLOQUEADA	
observacoes	TEXT	-	Sim	-	Observações longas	TEXT permitido

Tabela: veiculos

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_veiculo	INTEGER	-	Não	PK	Identificador do veículo	Autoincrement
placa	VARCHAR	8	Não	UNIQUE	Placa	
renavam	VARCHAR	20	Não	UNIQUE	RENAVAM	
chassi	VARCHAR	25	Não	UNIQUE	Chassi	
marca	VARCHAR	40	Não	-	Marca	
modelo	VARCHAR	60	Não	-	Modelo	
ano_fabricacao	INTEGER	-	Não	-	Ano de fabricação	
ano_modelo	INTEGER	-	Não	-	Ano do modelo	
cor	VARCHAR	30	Não	-	Cor	
categoria	VARCHAR	30	Não	-	ECONOMICO/ SEDAN/SUV/ UTILITARIO	
quilometragem_atual	DECIMAL	10,1	Não	-	KM atual	
combustivel	VARCHAR	20	Não	-	Tipo de combustível	
status_veiculo	VARCHAR	20	Não	-	DISPONIVEL/ LOCADO/ MANUTENCAO/ HIGIENIZACAO	
id_estacionamento_atual	INTEGER	-	Sim	FK	Pátio atual	FK simples
codigo_vaga_atual	VARCHAR	10	Sim	-	Vaga atual	Comporá vínculo lógico com a vaga
observacoes	TEXT	-	Sim	-	Observações do veículo	TEXT permitido

Tabela: locacoes

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_locacao	INTEGER	-	Não	PK	Identificador da locação	Autoincrement
id_cliente	INTEGER	-	Não	FK	Cliente titular	
id_veiculo	INTEGER	-	Não	FK	Veículo locado	
id_funcionario_	INTEGER	-	Não	FK	Funcionário da	

abertura					abertura	
id_banco_aprovacao	INTEGER	-	Sim	FK	Banco de aprovação	
data_reserva	DATETIME	-	Não	-	Data e hora da reserva	
data_prev_retirada	DATETIME	-	Não	-	Retirada prevista	
data_prev_devolucao	DATETIME	-	Não	-	Devolução prevista	
data_retirada_efetiva	DATETIME	-	Sim	-	Retirada efetiva	
data_devolucao_efetiva	DATETIME	-	Sim	-	Devolução efetiva	
valor_diaria	DECIMAL	10,2	Não	-	Valor da diária	
quantidade_diarias	INTEGER	-	Não	-	Qtd. de diárias	
valor_previsto	DECIMAL	10,2	Não	-	Valor total previsto	
valor_fechado	DECIMAL	10,2	Sim	-	Valor final apurado	
status_locacao	VARCHAR	20	Não	-	RESERVADA/ ABERTA/ FINALIZADA/ CANCELADA/ ATRASADA	
observacoes	TEXT	-	Sim	-	Observações longas	TEXT permitido

Tabela: locacao_condutores (N:M com chave composta)

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_locacao	INTEGER	-	Não	PK/FK	Locação relacionada	Parte 1 da PK composta
id_cliente_condutor	INTEGER	-	Não	PK/FK	Cliente condutor adicional	Parte 2 da PK composta
ordem_condutor	INTEGER	-	Não	-	Ordem do condutor	1, 2, 3...
principal	VARCHAR	1	Não	-	S ou N	Evita mais de um principal
observacoes	TEXT	-	Sim	-	Observações	TEXT permitido

Tabela: inspecoes_saida

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações

id_inspecao_saida	INTEGER	-	Não	PK	Identificador da inspeção	Autoincrement
id_locacao	INTEGER	-	Não	FK/UNIQUE	Locação da inspeção	1:1 lógico
id_funcionario	INTEGER	-	Não	FK	Funcionário vistoriador	
data_hora_saida	DATETIME	-	Não	-	Momento da vistoria	
km_saida	DECIMAL	10,1	Não	-	Quilometragem de saída	
nivel_combustivel_saida	VARCHAR	15	Não	-	1/8, 1/4, 1/2, 3/4, 1/1	
status_lataria_saida	VARCHAR	20	Não	-	OK/AVARIAS	
status_pneus_saida	VARCHAR	20	Não	-	OK/ DESGASTE	
status_acessorios_saida	VARCHAR	20	Não	-	OK/ FALTANDO	
observacoes	TEXT	-	Sim	-	Detalhes longos	TEXT permitido

Tabela: inspecoes_entrada

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_inspecao_entrada	INTEGER	-	Não	PK	Identificador da inspeção	Autoincrement
id_locacao	INTEGER	-	Não	FK/UNIQUE	Locação da inspeção	1:1 lógico
id_funcionario	INTEGER	-	Não	FK	Funcionário vistoriador	
data_hora_entrada	DATETIME	-	Não	-	Momento da vistoria	
km_entrada	DECIMAL	10,1	Não	-	Quilometragem de entrada	
nivel_combustivel_entrada	VARCHAR	15	Não	-	1/8, 1/4, 1/2, 3/4, 1/1	
status_lataria_entrada	VARCHAR	20	Não	-	OK/AVARIAS	
status_pneus_entrada	VARCHAR	20	Não	-	OK/ DESGASTE	
status_acessorios_entrada	VARCHAR	20	Não	-	OK/ FALTANDO	
observacoes	TEXT	-	Sim	-	Detalhes longos	TEXT permitido

Tabela: patio_movimentacoes

Campo	Tipo	Tamanho	Nulo?	Chave	Descrição	Observações
id_movimentacao	INTEGER	-	Não	PK	Identificador da movimentação	Autoincrement
id_veiculo	INTEGER	-	Não	FK	Veículo movimentado	
id_funcionario	INTEGER	-	Não	FK	Funcionário responsável	
data_hora_movimentacao	DATETIME	-	Não	-	Data/hora do evento	
tipo_movimentacao	VARCHAR	20	Não	-	ENTRADA/ SAIDA/ TRANSFERENCIA/ ENTREGA/ RETORNO	
id_estacionamento_origem	INTEGER	-	Sim	-	Pátio de origem	Parte da FK composta
codigo_vaga_origem	VARCHAR	10	Sim	-	Vaga de origem	Parte da FK composta
id_estacionamento_destino	INTEGER	-	Sim	-	Pátio de destino	Parte da FK composta
codigo_vaga_destino	VARCHAR	10	Sim	-	Vaga de destino	Parte da FK composta
motivo	VARCHAR	60	Não	-	Motivo da movimentação	
observacoes	TEXT	-	Sim	-	Detalhes longos	TEXT permitido

Script base de criação das tabelas

```
PRAGMA foreign_keys = ON;
CREATE TABLE clientes (
    id_cliente INTEGER PRIMARY KEY AUTOINCREMENT,
    tipo_cliente VARCHAR(2) NOT NULL CHECK (tipo_cliente IN ('PF','PJ')),
    nome_razao VARCHAR(150) NOT NULL,
    nome_fantasia VARCHAR(150),
    cpf_cnpj VARCHAR(18) NOT NULL UNIQUE,
    rg_ie VARCHAR(20),
    cnh_numero VARCHAR(20),
    cnh_categoria VARCHAR(5),
    telefone VARCHAR(20),
    email VARCHAR(120) UNIQUE,
    logradouro VARCHAR(120),
    numero VARCHAR(10),
    bairro VARCHAR(60),
    cidade VARCHAR(60),
```

```
    uf VARCHAR(2),
    cep VARCHAR(9),
    status_cliente VARCHAR(15) NOT NULL,
    observacoes TEXT
);
CREATE TABLE funcionarios (
    id_funcionario INTEGER PRIMARY KEY AUTOINCREMENT,
    nome VARCHAR(120) NOT NULL,
    cpf VARCHAR(14) NOT NULL UNIQUE,
    cargo VARCHAR(60) NOT NULL,
    telefone VARCHAR(20),
    email VARCHAR(120) UNIQUE,
    matricula VARCHAR(20) NOT NULL UNIQUE,
    status_funcionario VARCHAR(15) NOT NULL,
    comentarios TEXT
);
CREATE TABLE bancos (
    id_banco INTEGER PRIMARY KEY AUTOINCREMENT,
    codigo_banco VARCHAR(10) NOT NULL UNIQUE,
    nome_banco VARCHAR(100) NOT NULL,
    contato_setor VARCHAR(80),
    telefone VARCHAR(20),
    email VARCHAR(120),
    status_banco VARCHAR(15) NOT NULL,
    observacoes TEXT
);
CREATE TABLE estacionamentos (
    id_estacionamento INTEGER PRIMARY KEY AUTOINCREMENT,
    nome_estacionamento VARCHAR(100) NOT NULL,
    tipo_estacionamento VARCHAR(20) NOT NULL,
    logradouro VARCHAR(120) NOT NULL,
    numero VARCHAR(10),
    bairro VARCHAR(60),
    cidade VARCHAR(60) NOT NULL,
    uf VARCHAR(2) NOT NULL,
    cep VARCHAR(9),
    capacidade_total INTEGER NOT NULL,
    status_estacionamento VARCHAR(15) NOT NULL,
    observacoes TEXT
);
CREATE TABLE vagas_estacionamento (
    id_estacionamento INTEGER NOT NULL,
    codigo_vaga VARCHAR(10) NOT NULL,
    setor VARCHAR(30) NOT NULL,
    tipo_vaga VARCHAR(20) NOT NULL,
    coberta VARCHAR(1) NOT NULL CHECK (coberta IN ('S','N')),
    status_vaga VARCHAR(15) NOT NULL,
    observacoes TEXT,
    PRIMARY KEY (id_estacionamento, codigo_vaga),
```

```
    FOREIGN KEY (id_estacionamento) REFERENCES estacionamentos(id_estacionamento)
);
CREATE TABLE veiculos (
    id_veiculo INTEGER PRIMARY KEY AUTOINCREMENT,
    placa VARCHAR(8) NOT NULL UNIQUE,
    renavam VARCHAR(20) NOT NULL UNIQUE,
    chassi VARCHAR(25) NOT NULL UNIQUE,
    marca VARCHAR(40) NOT NULL,
    modelo VARCHAR(60) NOT NULL,
    ano_fabricacao INTEGER NOT NULL,
    ano_modelo INTEGER NOT NULL,
    cor VARCHAR(30) NOT NULL,
    categoria VARCHAR(30) NOT NULL,
    quilometragem_atual DECIMAL(10,1) NOT NULL,
    combustivel VARCHAR(20) NOT NULL,
    status_veiculo VARCHAR(20) NOT NULL,
    id_estacionamento_atual INTEGER,
    codigo_vaga_atual VARCHAR(10),
    observacoes TEXT,
    FOREIGN KEY (id_estacionamento_atual) REFERENCES estacionamentos(id_estacionamento),
    FOREIGN KEY (id_estacionamento_atual, codigo_vaga_atual)
        REFERENCES vagas_estacionamento(id_estacionamento, codigo_vaga)
);
CREATE TABLE locacoes (
    id_locacao INTEGER PRIMARY KEY AUTOINCREMENT,
    id_cliente INTEGER NOT NULL,
    id_veiculo INTEGER NOT NULL,
    id_funcionario_abertura INTEGER NOT NULL,
    id_banco_aprovacao INTEGER,
    data_reserva DATETIME NOT NULL,
    data_prev_retirada DATETIME NOT NULL,
    data_prev_devolucao DATETIME NOT NULL,
    data_retirada_efetiva DATETIME,
    data_devolucao_efetiva DATETIME,
    valor_diaria DECIMAL(10,2) NOT NULL,
    quantidade_diarias INTEGER NOT NULL,
    valor_previsto DECIMAL(10,2) NOT NULL,
    valor_fechado DECIMAL(10,2),
    status_locacao VARCHAR(20) NOT NULL,
    observacoes TEXT,
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),
    FOREIGN KEY (id_veiculo) REFERENCES veiculos(id_veiculo),
    FOREIGN KEY (id_funcionario_abertura) REFERENCES funcionarios(id_funcionario),
    FOREIGN KEY (id_banco_aprovacao) REFERENCES bancos(id_banco)
);
CREATE TABLE locacao_condutores (
    id_locacao INTEGER NOT NULL,
    id_cliente_condutor INTEGER NOT NULL,
    ordem_condutor INTEGER NOT NULL,
    principal VARCHAR(1) NOT NULL CHECK (principal IN ('S','N')),
    observacoes TEXT,
```

```
    PRIMARY KEY (id_locacao, id_cliente_condutor),
    FOREIGN KEY (id_locacao) REFERENCES locacoes(id_locacao),
    FOREIGN KEY (id_cliente_condutor) REFERENCES clientes(id_cliente)
);
CREATE TABLE inspecoes_saida (
    id_inspecao_saida INTEGER PRIMARY KEY AUTOINCREMENT,
    id_locacao INTEGER NOT NULL UNIQUE,
    id_funcionario INTEGER NOT NULL,
    data_hora_saida DATETIME NOT NULL,
    km_saida DECIMAL(10,1) NOT NULL,
    nivel_combustivel_saida VARCHAR(15) NOT NULL,
    status_lataria_saida VARCHAR(20) NOT NULL,
    status_pneus_saida VARCHAR(20) NOT NULL,
    status_acessorios_saida VARCHAR(20) NOT NULL,
    observacoes TEXT,
    FOREIGN KEY (id_locacao) REFERENCES locacoes(id_locacao),
    FOREIGN KEY (id_funcionario) REFERENCES funcionarios(id_funcionario)
);
CREATE TABLE inspecoes_entrada (
    id_inspecao_entrada INTEGER PRIMARY KEY AUTOINCREMENT,
    id_locacao INTEGER NOT NULL UNIQUE,
    id_funcionario INTEGER NOT NULL,
    data_hora_entrada DATETIME NOT NULL,
    km_entrada DECIMAL(10,1) NOT NULL,
    nivel_combustivel_entrada VARCHAR(15) NOT NULL,
    status_lataria_entrada VARCHAR(20) NOT NULL,
    status_pneus_entrada VARCHAR(20) NOT NULL,
    status_acessorios_entrada VARCHAR(20) NOT NULL,
    observacoes TEXT,
    FOREIGN KEY (id_locacao) REFERENCES locacoes(id_locacao),
    FOREIGN KEY (id_funcionario) REFERENCES funcionarios(id_funcionario)
);
CREATE TABLE patio_movimentacoes (
    id_movimentacao INTEGER PRIMARY KEY AUTOINCREMENT,
    id_veiculo INTEGER NOT NULL,
    id_funcionario INTEGER NOT NULL,
    data_hora_movimentacao DATETIME NOT NULL,
    tipo_movimentacao VARCHAR(20) NOT NULL,
    id_estacionamento_origem INTEGER,
    codigo_vaga_origem VARCHAR(10),
    id_estacionamento_destino INTEGER,
    codigo_vaga_destino VARCHAR(10),
    motivo VARCHAR(60) NOT NULL,
    observacoes TEXT,
    FOREIGN KEY (id_veiculo) REFERENCES veiculos(id_veiculo),
    FOREIGN KEY (id_funcionario) REFERENCES funcionarios(id_funcionario),
    FOREIGN KEY (id_estacionamento_origem, codigo_vaga_origem)
        REFERENCES vagas_estacionamento(id_estacionamento, codigo_vaga),
    FOREIGN KEY (id_estacionamento_destino, codigo_vaga_destino)
        REFERENCES vagas_estacionamento(id_estacionamento, codigo_vaga)
);
```

Como o aluno deve construir o banco na prática

11. Passo 1: leia o cenário e destaque substantivos que viram tabelas. Exemplo: cliente, funcionário, veículo, locação, inspeção.
12. Passo 2: para cada tabela, marque qual é a chave primária. Se a tabela for associativa, verifique se a chave precisa ser dupla.
13. Passo 3: antes de digitar qualquer CREATE TABLE, desenhe o DER em papel ou ferramenta.
14. Passo 4: defina os campos obrigatórios e os opcionais. O aluno deve justificar o uso de NULL quando existir.
15. Passo 5: crie primeiro as tabelas-mãe. Isso evita erro de chave estrangeira.
16. Passo 6: crie as tabelas-filhas e confira se o tipo do campo FK é compatível com o tipo da PK referenciada.
17. Passo 7: execute INSERTs na mesma ordem das dependências.
18. Passo 8: faça UPDATEs para validar alteração de status e localização.
19. Passo 9: faça DELETEs controlados, sempre entendendo o impacto nas tabelas dependentes.
20. Passo 10: conclua com SELECTs simples, depois JOINs, agrupamentos, filtros por datas e consultas de auditoria.

Carga de teste e operações CRUD

A seguir há uma carga de exemplo com pelo menos 5 registros por tabela, além de comandos de UPDATE e DELETE. O aluno pode executar esses comandos para validar se o banco foi criado corretamente.

clientes - INSERT / UPDATE / DELETE

INSERT INTO clientes (id_cliente, tipo_cliente, nome_razao, nome_fantasia, cpf_cnpj, rg_ie, cnh_numero, cnh_categoria, telefone, email, logradouro, numero, bairro, cidade, uf, cep, status_cliente, observacoes) VALUES (1, 'PF', 'Ana Paula Gomes', NULL, '111.111.111-11', '22.333.444-5', '12345678900', 'B', '16999990001', 'ana@gmail.com', 'Rua A', '100', 'Centro', 'Ribeirão Preto', 'SP', '14000-000', 'ATIVO', 'Cliente frequente');

INSERT INTO clientes VALUES (2, 'PF', 'Bruno Henrique Lima', NULL, '222.222.222-22', '11.222.333-4', '22345678900', 'B', '16999990002', 'bruno@gmail.com', 'Rua B', '200', 'Campos Eliseos', 'Ribeirão Preto', 'SP', '14010-000', 'ATIVO', '');

INSERT INTO clientes VALUES (3, 'PJ', 'Comercial Delta Ltda', 'Delta', '12.345.678/0001-90', 'ISENTO', NULL, NULL, '1633334444', 'contato@delta.com', 'Av. Brasil', '300', 'Jardim Paulista', 'Ribeirão Preto', 'SP', '14020-000', 'ATIVO', 'Pessoa jurídica para frota');

INSERT INTO clientes VALUES (4, 'PF', 'Carla Regina Souza', NULL, '333.333.333-33', '44.555.666-7', '32345678900', 'AB', '16999990004', 'carla@gmail.com', 'Rua C', '40', 'Centro', 'Sertaozinho', 'SP', '14160-000', 'BLOQUEADO', 'Documentação em revisão');

INSERT INTO clientes VALUES (5, 'PF', 'Daniel Moreira', NULL, '444.444.444-44', '55.666.777-8', '42345678900', 'B', '16999990005', 'daniel@gmail.com', 'Rua D', '500', 'Ipiranga', 'Ribeirão Preto', 'SP', '14055-000', 'ATIVO', '');

UPDATE clientes SET status_cliente = 'INATIVO' WHERE id_cliente = 4;

DELETE FROM clientes WHERE id_cliente = 5;

funcionarios - INSERT / UPDATE / DELETE

INSERT INTO funcionarios VALUES (1, 'Marcos Silva', '101.101.101-01', 'Atendente', '16999110001', 'marcos@prime.com', 'MAT001', 'ATIVO', '');

INSERT INTO funcionarios VALUES (2, 'Paula Nunes', '202.202.202-02', 'Vistoriadora', '16999110002', 'paula@prime.com', 'MAT002', 'ATIVO', '');

INSERT INTO funcionarios VALUES (3, 'Ricardo Alves', '303.303.303-03', 'Patio', '16999110003', 'ricardo@prime.com', 'MAT003', 'ATIVO', '');

INSERT INTO funcionarios VALUES (4, 'Luciana Prado', '404.404.404-04', 'Financeiro', '16999110004', 'luciana@prime.com', 'MAT004', 'ATIVO', '');

INSERT INTO funcionarios VALUES (5, 'Sandro Melo', '505.505.505-05', 'Gerente', '16999110005', 'sandro@prime.com', 'MAT005', 'ATIVO', '');

UPDATE funcionarios SET cargo = 'Supervisor de Pátio' WHERE id_funcionario = 3;

DELETE FROM funcionarios WHERE id_funcionario = 5;

bancos - INSERT / UPDATE / DELETE

INSERT INTO bancos VALUES (1, '001', 'Banco do Brasil', 'Mesa Crédito', '1631110001', 'bb@banco.com', 'ATIVO', '');

INSERT INTO bancos VALUES (2, '237', 'Bradesco', 'Mesa Crédito', '1631110002', 'bradesco@banco.com', 'ATIVO', '');

INSERT INTO bancos VALUES (3, '341', 'Itaú', 'Mesa Crédito', '1631110003', 'itau@banco.com', 'ATIVO', '');

INSERT INTO bancos VALUES (4, '104', 'Caixa', 'Mesa Crédito', '1631110004', 'caixa@banco.com', 'ATIVO', '');

INSERT INTO bancos VALUES (5, '033', 'Santander', 'Mesa Crédito', '1631110005', 'santander@banco.com', 'ATIVO', '');

UPDATE bancos SET status_banco = 'INATIVO' WHERE id_banco = 4;

DELETE FROM bancos WHERE id_banco = 5;

estacionamentos - INSERT / UPDATE / DELETE

INSERT INTO estacionamentos VALUES (1, 'Matriz Centro', 'MATRIZ', 'Av. Central', '1000', 'Centro', 'Ribeirão Preto', 'SP', '14000-100', 80, 'ATIVO', '');

INSERT INTO estacionamentos VALUES (2, 'Patio Norte', 'FILIAL', 'Av. Norte', '2000', 'Lagoinha', 'Ribeirão Preto', 'SP', '14000-200', 50, 'ATIVO', '');

INSERT INTO estacionamentos VALUES (3, 'Patio Sul', 'FILIAL', 'Av. Sul', '3000', 'Vila Mariana', 'Ribeirão Preto', 'SP', '14000-300', 45, 'ATIVO', '');

INSERT INTO estacionamentos VALUES (4, 'Apoio Oficina', 'APOIO', 'Rua Oficina', '400', 'Industrial', 'Ribeirão Preto', 'SP', '14000-400', 25, 'ATIVO', '');

INSERT INTO estacionamentos VALUES (5, 'Apoio Aeroporto', 'APOIO', 'Rodovia Leste', '500', 'Aeroporto', 'Ribeirão Preto', 'SP', '14000-500', 30, 'ATIVO', '');

UPDATE estacionamentos SET capacidade_total = 55 WHERE id_estacionamento = 2;

DELETE FROM estacionamentos WHERE id_estacionamento = 5;

vagas_estacionamento - INSERT / UPDATE / DELETE

INSERT INTO vagas_estacionamento VALUES (1, 'A01', 'SETOR A', 'ENTREGA', 'S', 'LIVRE', '');

INSERT INTO vagas_estacionamento VALUES (1, 'A02', 'SETOR A', 'PADRAO', 'S', 'LIVRE', '');

INSERT INTO vagas_estacionamento VALUES (2, 'N01', 'SETOR N', 'PADRAO', 'N', 'LIVRE', '');

INSERT INTO vagas_estacionamento VALUES (2, 'N02', 'SETOR N', 'LAVAGEM', 'N', 'BLOQUEADA', 'Em limpeza');

INSERT INTO vagas_estacionamento VALUES (4, 'M01', 'OFICINA', 'MANUTENCAO', 'S', 'LIVRE', '');

UPDATE vagas_estacionamento SET status_vaga = 'OCUPADA' WHERE id_estacionamento = 1 AND codigo_vaga = 'A01';

DELETE FROM vagas_estacionamento WHERE id_estacionamento = 2 AND codigo_vaga = 'N02';

veiculos - INSERT / UPDATE / DELETE

```
INSERT INTO veiculos VALUES (1, 'ABC1D23', '10000000001', 'CHASSI00000000000000000001', 'Fiat', 'Mobi', 2023, 2024, 'Branco', 'ECONOMICO', 12500.0, 'Flex', 'DISPONIVEL', 1, 'A01', '');
INSERT INTO veiculos VALUES (2, 'DEF4G56', '10000000002', 'CHASSI00000000000000000002', 'Chevrolet', 'Onix', 2023, 2024, 'Prata', 'ECONOMICO', 18700.5, 'Flex', 'DISPONIVEL', 1, 'A02', '');
INSERT INTO veiculos VALUES (3, 'GHI7J89', '10000000003', 'CHASSI00000000000000000003', 'Jeep', 'Renegade', 2022, 2023, 'Preto', 'SUV', 30110.2, 'Flex', 'LOCADO', 2, 'N01', '');
INSERT INTO veiculos VALUES (4, 'JKL1M23', '10000000004', 'CHASSI00000000000000000004', 'Volkswagen', 'Saveiro', 2021, 2022, 'Branco', 'UTILITARIO', 45200.0, 'Flex', 'MANUTENCAO', 4, 'M01', '');
INSERT INTO veiculos VALUES (5, 'NOP4Q56', '10000000005', 'CHASSI00000000000000000005', 'Toyota', 'Corolla', 2024, 2024, 'Cinza', 'SEDAN', 8100.7, 'Flex', 'DISPONIVEL', 1, 'A02', '');
UPDATE veiculos SET status_veiculo = 'HIGIENIZACAO' WHERE id_veiculo = 2;
DELETE FROM veiculos WHERE id_veiculo = 5;
```

locacoes - INSERT / UPDATE / DELETE

```
INSERT INTO locacoes VALUES (1, 1, 1, 1, 1, '2026-03-01 08:00:00', '2026-03-02 09:00:00', '2026-03-05 09:00:00', '2026-03-02 09:15:00', NULL, 120.00, 3, 360.00, NULL, 'ABERTA', 'Cliente retirou com seguro básico');
INSERT INTO locacoes VALUES (2, 2, 2, 1, 2, '2026-03-01 10:00:00', '2026-03-03 08:00:00', '2026-03-04 08:00:00', '2026-03-03 08:10:00', '2026-03-04 08:05:00', 110.00, 1, 110.00, 110.00, 'FINALIZADA', '');
INSERT INTO locacoes VALUES (3, 3, 3, 1, 3, '2026-03-02 11:00:00', '2026-03-02 14:00:00', '2026-03-09 14:00:00', '2026-03-02 14:05:00', NULL, 220.00, 7, 1540.00, NULL, 'ABERTA', 'Contrato corporativo');
INSERT INTO locacoes VALUES (4, 1, 2, 1, NULL, '2026-03-04 09:00:00', '2026-03-05 10:00:00', '2026-03-06 10:00:00', NULL, NULL, 100.00, 1, 100.00, NULL, 'RESERVADA', '');
INSERT INTO locacoes VALUES (5, 4, 4, 1, 1, '2026-03-01 13:00:00', '2026-03-01 16:00:00', '2026-03-02 16:00:00', NULL, NULL, 150.00, 1, 150.00, NULL, 'CANCELADA', 'Cliente bloqueado posteriormente');
UPDATE locacoes SET status_locacao = 'ATRASADA' WHERE id_locacao = 1;
DELETE FROM locacoes WHERE id_locacao = 5;
```

locacao_condutores - INSERT / UPDATE / DELETE

```
INSERT INTO locacao_condutores VALUES (1, 1, 1, 'S', 'Titular dirige o veículo');
INSERT INTO locacao_condutores VALUES (1, 2, 2, 'N', 'Condutor adicional autorizado');
INSERT INTO locacao_condutores VALUES (2, 2, 1, 'S', '');
INSERT INTO locacao_condutores VALUES (3, 3, 1, 'S', 'Motorista do contrato corporativo');
INSERT INTO locacao_condutores VALUES (4, 1, 1, 'S', '');
UPDATE locacao_condutores SET ordem_condutor = 3 WHERE id_locacao = 1 AND id_cliente_condutor = 2;
DELETE FROM locacao_condutores WHERE id_locacao = 4 AND id_cliente_condutor = 1;
```

inspecoes_saida - INSERT / UPDATE / DELETE

```
INSERT INTO inspecoes_saida VALUES (1, 1, 2, '2026-03-02 09:10:00', 12500.0, '1/1', 'OK', 'OK', 'OK', 'Sem avarias aparentes');
INSERT INTO inspecoes_saida VALUES (2, 2, 2, '2026-03-03 08:00:00', 18700.5, '3/4', 'OK', 'OK', 'OK', '');
INSERT INTO inspecoes_saida VALUES (3, 3, 2, '2026-03-02 14:00:00', 30110.2, '1/1', 'OK', 'OK', 'OK', 'Contrato PJ');
INSERT INTO inspecoes_saida VALUES (4, 4, 2, '2026-03-05 09:40:00', 18700.5, '1/2', 'OK', 'OK', 'OK', 'Reserva ainda sem retirada oficial');
```

```
INSERT INTO inspecoes_saida VALUES (5, 5, 2, '2026-03-01 15:50:00', 45200.0, '1/2', 'AVARIAS', 'OK', 'OK', 'Risco no para-choque');
UPDATE inspecoes_saida SET observacoes = 'Sem avarias aparentes e veículo higienizado' WHERE id_inspecao_saida = 1;
DELETE FROM inspecoes_saida WHERE id_inspecao_saida = 5;
```

inspecoes_entrada - INSERT / UPDATE / DELETE

```
INSERT INTO inspecoes_entrada VALUES (1, 2, 2, '2026-03-04 08:05:00', 18820.9, '1/2', 'OK', 'OK', 'OK', 'Retorno normal');
INSERT INTO inspecoes_entrada VALUES (2, 5, 2, '2026-03-02 16:30:00', 45230.0, '1/4', 'AVARIAS', 'OK', 'OK', 'Locação cancelada no retorno');
INSERT INTO inspecoes_entrada VALUES (3, 1, 2, '2026-03-05 12:00:00', 12680.0, '3/4', 'OK', 'OK', 'OK', 'Exemplo de teste didático');
INSERT INTO inspecoes_entrada VALUES (4, 3, 2, '2026-03-09 14:10:00', 30890.0, '1/2', 'OK', 'DESGASTE', 'OK', 'Viagem longa');
INSERT INTO inspecoes_entrada VALUES (5, 4, 2, '2026-03-06 10:10:00', 18810.0, '1/2', 'OK', 'OK', 'FALTANDO', 'Sem tapete traseiro');
UPDATE inspecoes_entrada SET status_pneus_entrada = 'OK' WHERE id_inspecao_entrada = 4;
DELETE FROM inspecoes_entrada WHERE id_inspecao_entrada = 2;
```

patio_movimentacoes - INSERT / UPDATE / DELETE

```
INSERT INTO patio_movimentacoes VALUES (1, 1, 3, '2026-03-02 08:50:00', 'ENTREGA', 1, 'A01', 1, 'A01', 'Separação para retirada', '');
INSERT INTO patio_movimentacoes VALUES (2, 2, 3, '2026-03-03 07:40:00', 'ENTREGA', 1, 'A02', 1, 'A02', 'Separação para retirada', '');
INSERT INTO patio_movimentacoes VALUES (3, 3, 3, '2026-03-02 13:30:00', 'SAIDA', 2, 'N01', NULL, NULL, 'Veículo saiu para locação', '');
INSERT INTO patio_movimentacoes VALUES (4, 4, 3, '2026-03-01 12:00:00', 'TRANSFERENCIA', 1, 'A02', 4, 'M01', 'Envio para manutenção', '');
INSERT INTO patio_movimentacoes VALUES (5, 2, 3, '2026-03-04 08:20:00', 'RETORNO', NULL, NULL, 1, 'A02', 'Veículo retornou da locação', '');
UPDATE patio_movimentacoes SET motivo = 'Envio para manutenção preventiva' WHERE id_movimentacao = 4;
DELETE FROM patio_movimentacoes WHERE id_movimentacao = 5;
```

Pesquisas básicas e avançadas

> ***A ordem abaixo é proposital. O aluno deve executar na sequência para validar progressivamente a base.***

Filtro 1 - Clientes ativos por cidade

```
SELECT id_cliente, nome_razao, cidade, status_cliente FROM clientes WHERE status_cliente = 'ATIVO' AND cidade = 'Ribeirão Preto' ORDER BY nome_razao;
```

Validação esperada: Resultado esperado após a carga e deletes: listar Ana Paula Gomes, Bruno Henrique Lima e Comercial Delta Ltda.

Filtro 2 - Veículos disponíveis por estacionamento

```
SELECT e.nome_estacionamento, v.placa, v.marca, v.modelo, v.status_veiculo FROM veiculos v JOIN estacionamentos e ON e.id_estacionamento = v.id_estacionamento_atual WHERE v.status_veiculo = 'DISPONIVEL' ORDER BY e.nome_estacionamento, v.placa;
```

Validação esperada: Resultado esperado: pelo menos o veículo ABC1D23 deve aparecer na Matriz Centro, exceto se o aluno alterar status depois.

Filtro 3 - Locações em aberto ou atrasadas

```
SELECT l.id_locacao, c.nome_razao, v.placa, l.data_prev_devolucao, l.status_locacao FROM locacoes l JOIN clientes c ON c.id_cliente = l.id_cliente JOIN veiculos v ON v.id_veiculo = l.id_veiculo WHERE l.status_locacao IN ('ABERTA','ATRASADA') ORDER BY l.data_prev_devolucao;
```

Validação esperada: Resultado esperado: devem aparecer as locações 1 e 3; a locação 1 ficará marcada como ATRASADA após o UPDATE.

Filtro 4 - Histórico completo de uma locação

```
SELECT l.id_locacao, c.nome_razao, v.placa, s.data_hora_saida, e.data_hora_entrada, s.km_saida, e.km_entrada FROM locacoes l JOIN clientes c ON c.id_cliente = l.id_cliente JOIN veiculos v ON v.id_veiculo = l.id_veiculo LEFT JOIN inspecoes_saida s ON s.id_locacao = l.id_locacao LEFT JOIN inspecoes_entrada e ON e.id_locacao = l.id_locacao WHERE l.id_locacao = 2;
```

Validação esperada: Resultado esperado: retornar uma única linha com o cliente Bruno, veículo DEF4G56, km de saída e km de entrada.

Filtro 5 - Condutores adicionais de uma locação

```
SELECT lc.id_locacao, c.nome_razao, lc.ordem_condutor, lc.principal FROM locacao_condutores lc JOIN clientes c ON c.id_cliente = lc.id_cliente_condutor WHERE lc.id_locacao = 1 ORDER BY lc.ordem_condutor;
```

Validação esperada: Resultado esperado: dois registros antes do delete correspondente; depois do update, Bruno deverá ficar com ordem 3.

Filtro 6 - Ocupação de vagas por estacionamento

```
SELECT e.nome_estacionamento, COUNT(*) AS total_vagas, SUM(CASE WHEN v.status_vaga = 'OCUPADA' THEN 1 ELSE 0 END) AS vagas_ocupadas FROM estacionamentos e JOIN vagas_estacionamento v ON v.id_estacionamento = e.id_estacionamento GROUP BY e.id_estacionamento, e.nome_estacionamento ORDER BY e.nome_estacionamento;
```

Validação esperada: Resultado esperado: a Matriz Centro deve indicar pelo menos 2 vagas cadastradas e 1 vaga ocupada após o update da vaga A01.

Filtro 7 - Auditoria de movimentação do pátio

```
SELECT p.id_movimentacao, ve.placa, f.nome AS funcionario, p.tipo_movimentacao, p.data_hora_movimentacao, p.id_estacionamento_origem, p.codigo_vaga_origem, p.id_estacionamento_destino, p.codigo_vaga_destino, p.motivo FROM patio_movimentacoes p JOIN veiculos ve ON ve.id_veiculo = p.id_veiculo JOIN funcionarios f ON f.id_funcionario = p.id_funcionario ORDER BY p.data_hora_movimentacao;
```

Validação esperada: Resultado esperado: exibir a trilha cronológica das movimentações. Após o DELETE, a movimentação 5 não deve aparecer.

Filtro 8 - Valor previsto por cliente

```
SELECT c.nome_razao, COUNT(l.id_locacao) AS qtd_locacoes, SUM(l.valor_previsto) AS total_previsto FROM clientes c JOIN locacoes l ON l.id_cliente = c.id_cliente GROUP BY c.id_cliente, c.nome_razao ORDER BY total_previsto DESC;
```

Validação esperada: Resultado esperado: Comercial Delta Ltda deve aparecer com o maior total previsto entre as locações abertas na carga inicial.

Filtro 9 - Quilometragem rodada por locação

```
SELECT l.id_locacao, v.placa, (ie.km_entrada - isd.km_saida) AS km_rodados FROM locacoes l JOIN veiculos v ON v.id_veiculo = l.id_veiculo JOIN inspecoes_saida isd ON isd.id_locacao = l.id_locacao JOIN inspecoes_entrada ie ON ie.id_locacao = l.id_locacao WHERE ie.km_entrada >= isd.km_saida ORDER BY km_rodados DESC;
```

Validação esperada: Resultado esperado: a locação 3 tende a mostrar alta quilometragem em cenário de teste completo.

Filtro 10 - Relatório final da locadora

```
SELECT l.id_locacao, c.nome_razao, v.placa, e.nome_estacionamento, l.status_locacao, l.valor_previsto FROM locacoes l JOIN clientes c ON c.id_cliente = l.id_cliente JOIN veiculos v ON v.id_veiculo = l.id_veiculo LEFT JOIN estacionamentos e ON e.id_estacionamento = v.id_estacionamento_atual ORDER BY l.id_locacao;
```

Validação esperada: Resultado esperado: o aluno deve enxergar locação, cliente, veículo, pátio e valor previsto em uma visão consolidada.

Orientações finais ao aluno

- Não comece pelo SQL sem antes entender o negócio.
- Se houver relação N:M, não tente resolvê-la colocando vários valores em uma única coluna.
- Se a tabela usa chave composta, a chave estrangeira correspondente também deverá respeitar essa composição.
- Ao testar DELETE, verifique se existem registros dependentes. Em ambiente real, isso costuma exigir estratégia de negócio.
- Quando seu resultado estiver diferente do esperado, verifique a ordem de execução, os IDs utilizados e os efeitos dos UPDATEs e DELETEs.
- Como exercício complementar, o professor pode pedir índices adicionais, views, triggers e regras de auditoria.