

Universidade Federal do Rio de Janeiro
Curso de Engenharia de Computação e Informação
Disciplina: Big Data

Projeto: Modelagem de Data Warehouse
Parte I – Modelagem de Banco de Dados Transacional (OLTP)

Integrante:

Rodrigo Lucas Pinto da Silva

DRE: 118055337

Repositório: https://github.com/Rodrigo-lpds/modelagem_sbd_oltp

Descrição do Projeto

1. Introdução e Visão Geral

Este documento descreve o projeto do banco de dados relacional transacional (OLTP) para a "Locadora de Veículos XYZ", uma das seis empresas que formam um consórcio para compartilhamento de pátios e integração de dados em um Data Warehouse (DW) unificado. O objetivo deste projeto é detalhar a estrutura e os componentes do banco de dados da Locadora XYZ, justificando a necessidade e a forma de extração de seus dados para fins analíticos e de BI (Business Intelligence) no DW.

2. Objetivos do Banco de Dados Transacional

O banco de dados transacional da Locadora XYZ foi projetado para:

- Suportar as operações diárias da empresa, incluindo gestão de frota, gerenciamento de pátios e vagas, cadastro de clientes e motoristas, processamento de reservas e controle de locações.
- Garantir a integridade, consistência e disponibilidade dos dados operacionais.
- Fornecer uma base de dados robusta e bem estruturada que sirva como fonte confiável para o processo de Extração, Transformação e Carga (ETL) ou Extração, Carga e Transformação (ELT) para o Data Warehouse integrado.
- Permitir o rastreamento detalhado das movimentações de veículos, ocupação de pátios e vagas, e interações com clientes.

3. Escopo do Banco de Dados

O modelo de dados abrange as seguintes áreas de negócio centrais para a operação da locadora e para a alimentação do DW:

- **Gestão de Frota:** Cadastro detalhado de Veiculos (placa, chassi, cor, ano, etc.), sua classificação em GrupoVeiculo (econômico, SUV, etc.), associação com Marca e Modelo, registro de Acessorios e histórico de ProntuarioManutencao.
- **Gestão de Pátios e Vagas:** Cadastro dos Patios da empresa e das Vagas individuais dentro de cada pátio, incluindo status da vaga (Livre, Ocupada, etc.) e o veículo ocupando esta vaga.
- **Gestão de Clientes:** Cadastro de Clientes (Pessoas Físicas e Jurídicas), incluindo dados para contato e faturamento, e de Motoristas autorizados, com detalhes da CNH.
- **Gestão de Reservas:** Registro de Reservas de veículos, especificando grupo, pátio de retirada/devolução e datas previstas.
- **Gestão de Locações:** Controle completo dos contratos de Locacao, desde a retirada até a devolução do veículo, incluindo dados do condutor, veículo, pátios envolvidos, proteções adicionais (LocacaoProtecao) e registro fotográfico do estado do veículo (FotoVeiculoEstado).
- **Gestão de Empresas Associadas:** A tabela Empresa permite identificar a qual empresa um veículo ou pátio pertence, crucial para a integração no DW.

4. Descrição do Modelo de Dados e Justificativa para Extração (Contexto ETL)

O banco de dados transacional foi modelado seguindo as fases conceitual, lógica e física (DDL ANSI SQL). A estrutura resultante é otimizada para transações (OLTP), mas contém informações que precisam ser extraídas para o ambiente OLAP (Online Analytical Processing) do Data Warehouse.

A seguir, detalha-se as principais entidades e por que seus dados são cruciais para extração:

- Empresa:
 - **Dados Relevantes para Extração:** id_empresa, nome_fantasia, cnpj.
 - **Justificativa ET:** Identifica a origem dos dados no DW, permitindo análises segmentadas por empresa ou a visão consolidada. Essencial para conformar dimensões como DimEmpresa.
- Patio:
 - **Dados Relevantes para Extração:** id_patio, nome, cidade, id_empresa_proprietaria.
 - **Justificativa ET:** Formará a base da dimensão DimPatio no DW. O id_empresa_proprietaria é crucial para o relatório de "origem" no "Controle de Pátio".

- **Vaga:**
 - **Dados Relevantes para Extração:** id_vaga, id_patio, codigo_vaga, status_vaga, tipo_vaga.
 - **Justificativa ET:** Embora o DW principal possa operar no nível de Pátio, dados de vagas podem ser usados para análises operacionais mais granulares ou para alimentar visões específicas de pátio dentro do DW. O status_vaga pode ser útil para calcular taxas de ocupação detalhadas.
- **GrupoVeiculo, Marca, Modelo:**
 - **Dados Relevantes para Extração:** Todos os atributos.
 - **Justificativa ET:** Estes formarão dimensões como DimGrupoVeiculo, DimMarca, DimModelo, que auxiliará a DimVeiculo ou fatos relacionados a veículos no DW.
- **Veiculo:**
 - **Dados Relevantes para Extração:** id_veiculo, placa, chassi, id_modelo, id_grupo, id_empresa_proprietaria, id_vaga_atual, status_operacional.
 - **Justificativa ET:** Fonte principal para a DimVeiculo. O id_empresa_proprietaria define a "origem" da frota. O id_vaga_atual (e, por inferência, Vaga.id_patio) informa a localização atual para o "Controle de Pátio".
- **Acessorio, VeiculoAcessorio:**
 - **Dados Relevantes para Extração:** Todos os atributos.
 - **Justificativa ET:** Podem ser usados para enriquecer a DimVeiculo com informações sobre acessórios, ou para análises específicas sobre a popularidade de acessórios.
- **Cliente:**
 - **Dados Relevantes para Extração:** id_cliente, tipo_cliente, nome_razao_social, cpf_cnpj, cidade_origem, estado_origem.
 - **Justificativa ET:** Base para a DimCliente. cidade_origem é necessária para o relatório de "Controle de Reservas" e "Grupos de Veículos mais Alugados". Implementação de Slowly Changing Dimensions (SCD) tipo 2 pode ser necessária para rastrear mudanças nos dados do cliente ao longo do tempo.
- **Motorista:**
 - **Dados Relevantes para Extração:** id_motorista, id_cliente_associado, cnh_numero, cnh_categoria, cnh_data_expiracao.
 - **Justificativa ET:** Base para a DimMotorista. Importante para rastrear quem conduziu o veículo.

- Reserva:
 - **Dados Relevantes para Extração:** id_reserva, id_cliente, id_grupo, id_patio_retirada, id_patio_devolucao, data_hora_retirada_prevista, data_hora_devolucao_prevista, status.
 - **Justificativa ET:** Fonte para uma tabela Fato de Reservas (FatoReserva)Necessária para o relatório de "Controle de Reservas" (quantitativo por grupo, pátio, tempo de retirada, duração, origem do cliente). Extração incremental baseada em data_criacao_reserva ou status.
- Locacao:
 - **Dados Relevantes para Extração:** Todos os atributos, especialmente id_locacao, id_cliente, id_motorista, id_veiculo, id_patio_retirada, id_patio_devolucao, data_hora_retirada_real, data_hora_devolucao_real, valor_cobrado_final.
 - **Justificativa ET:** É a principal fonte para a tabela FatoLocacao no DW.
 - id_patio_retirada e id_patio_devolucao são essenciais para a **análise de movimentação entre pátios (Cadeia de Markov)**.
 - Datas de retirada e devolução permitem calcular a duração da locação para o "Controle das Locações".
 - Links para diversas dimensões (DimCliente, DimMotorista, DimVeiculo, DimPatioRetirada, DimPatioDevolucao, DimTempo).
 - valor_cobrado_final é uma medida importante.
 - Extração incremental baseada em datas de transação ou status.
- Protecao, LocacaoProtecao:
 - **Dados Relevantes para Extração:** Todos os atributos.
 - **Justificativa ET:** Permite enriquecer a FatoLocacao com informações sobre proteções contratadas, ou formar uma tabela Fato separada para análise de serviços adicionais.
- ProntuarioManutencao:
 - **Dados Relevantes para Extração:** Todos os atributos.
 - **Justificativa ET:** Pode alimentar uma DimVeiculo com o estado de manutenção ou uma Fato de Manutenção para análises de custo e tempo de inatividade do veículo.
- FotoVeiculoEstado:
 - **Dados Relevantes para Extração:** Metadados das fotos (id_foto, id_locacao, id_veiculo, tipo_momento, data_hora_foto), não necessariamente as imagens em si para o DW principal.

- **Justificativa ET:** Os metadados podem ser úteis para auditoria ou para entender a frequência de registros fotográficos.

Dicionário de Dados do Modelo Relacional

Tabela: Empresa

Descrição: Armazena os dados cadastrais das empresas de locação que fazem parte do consórcio.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|----------------------------|--|
| id_empresa | INTEGER | PK | Identificador numérico único para cada empresa. |
| nome_fantasia | VARCHAR(100) | NOT NULL | Nome comercial (de fachada) da empresa. |
| razao_social | VARCHAR(150) | NOT NULL | Nome legal e oficial da empresa. |
| cnpj | VARCHAR(14) | NOT NULL, UNIQUE | Cadastro Nacional da Pessoa Jurídica, único por empresa. |

Tabela: `Patio`

Descrição: Armazena os dados dos pátios de estacionamento onde os veículos podem ser retirados ou devolvidos.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|--------------------------------------|--------------|---|--|
| <code>id_patio</code> | INTEGER | PK | Identificador numérico único para cada pátio. |
| <code>nome</code> | VARCHAR(100) | NOT NULL | Nome do pátio (Ex: "Aeroporto Santos Dumont"). |
| <code>endereço</code> | VARCHAR(255) | | Endereço completo do pátio. |
| <code>cidade</code> | VARCHAR(100) | | Cidade onde o pátio está localizado. |
| <code>id_empresa_proprietaria</code> | INTEGER | NOT NULL, FK -> <code>Empresa(id_empresa)</code> | Chave que referencia a empresa dona do pátio. |

Tabela: Vaga

Descrição: Representa um espaço de estacionamento individual dentro de um pátio.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|---|--|
| id_vaga | INTEGER | PK | Identificador numérico único para cada vaga. |
| id_patio | INTEGER | NOT NULL, FK -> Patio(id_patio) | Pátio ao qual a vaga pertence. |
| codigo_vaga | VARCHAR(20) | NOT NULL | Código alfanumérico da vaga dentro do pátio (Ex: "A01", "S2-10"). |
| status_vaga | VARCHAR(15) | DEFAULT 'Livre', CHECK (status_vaga IN ('Livre', 'Ocupada', 'Manutenção', 'Reservada')) | Status atual da vaga. |
| tipo_vaga | VARCHAR(50) | | Tipo da vaga (Ex: 'Normal', 'PCD', 'Veículo Elétrico', 'Coberta'). |

| | | | |
|--|--|---------------------------------------|---|
| | | UNIQUE (id_patio, codigo_vaga) | Garante que o código da vaga seja único dentro de cada pátio. |
|--|--|---------------------------------------|---|

Tabela: GrupoVeiculo

Descrição: Define as categorias ou grupos de veículos, que influenciam no preço e nas características da locação.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|-------------------|----------------|-------------------------|--|
| id_grupo | INTEGER | PK | Identificador numérico único para cada grupo. |
| nome | VARCHAR(50) | NOT NULL, UNIQUE | Nome do grupo (Ex: "Econômico", "SUV", "Premium"). |
| descricao | TEXT | | Descrição detalhada das características do grupo. |
| valor_diaria_base | NUMERIC(10, 2) | NOT NULL | Valor de referência para a diária de veículos deste grupo. |

Tabela: Marca

Descrição: Catálogo com as marcas (fabricantes) dos veículos da frota.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|-----------------|-----------|
|----------------|--------------|-----------------|-----------|

| | | | |
|----------|-------------|----------------------------|--|
| id_marca | INTEGER | PK | Identificador numérico único para cada marca. |
| nome | VARCHAR(50) | NOT NULL, UNIQUE | Nome da marca (Ex: "Fiat", "Chevrolet", "Toyota"). |

Tabela: Modelo

Descrição: Catálogo com os modelos específicos de veículos, associados a uma marca.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|--|---|
| id_modelo | INTEGER | PK | Identificador numérico único para cada modelo. |
| nome | VARCHAR(100) | NOT NULL | Nome do modelo (Ex: "Argo", "Onix", "Corolla"). |
| id_marca | INTEGER | NOT NULL, FK -> Marca(id_marca) | Chave que referencia a marca do veículo. |

Tabela: Veiculo**Descrição:** Registro individual de cada veículo pertencente à frota de uma das empresas.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|---------------------|--------------|---|---|
| id_veiculo | INTEGER | PK | Identificador numérico único para cada veículo. |
| placa | VARCHAR(7) | NOT NULL, UNIQUE | Placa de identificação do veículo. |
| chassi | VARCHAR(17) | NOT NULL, UNIQUE | Número de identificação do chassi do veículo. |
| cor | VARCHAR(30) | | Cor do veículo. |
| ano_fabricacao | INTEGER | | Ano de fabricação do veículo. |
| mecanizacao | VARCHAR(15) | CHECK (mecanizacao IN ('Manual', 'Automática')) | Tipo de câmbio do veículo. |
| tem_ar_condicionado | BOOLEAN | DEFAULT FALSE | Indica a presença de ar-condicionado. |

| | | | |
|-------------------------|-------------|---|--|
| status_operacional | VARCHAR(20) | DEFAULT 'Disponível', CHECK (status_operacional IN ('Disponível', 'Alugado', 'Em Manutenção', 'Reservado')) | Status atual do veículo. |
| id_modelo | INTEGER | NOT NULL, FK -> Modelo(id_modelo) | Referência ao modelo do veículo. |
| id_grupo | INTEGER | NOT NULL, FK -> GrupoVeiculo(id_grupo) | Referência ao grupo/categoria do veículo. |
| id_empresa_proprietaria | INTEGER | NOT NULL, FK -> Empresa(id_empresa) | Referência à empresa dona do veículo. |
| id_vaga_atual | INTEGER | FK -> Vaga(id_vaga) ON DELETE SET NULL | Vaga onde o veículo se encontra fisicamente (pode ser nulo). |

Tabela: Cliente

Descrição: Cadastro de clientes, podendo ser pessoas físicas (PF) ou jurídicas (PJ).

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|-----------------|-----------|
|----------------|--------------|-----------------|-----------|

| | | | |
|-------------------|--------------|--|--|
| id_cliente | INTEGER | PK | Identificador numérico único para cada cliente. |
| tipo_cliente | VARCHAR(2) | NOT NULL, CHECK (tipo_cliente IN ('PF', 'PJ')) | Distingue entre Pessoa Física e Pessoa Jurídica. |
| nome_razao_social | VARCHAR(150) | NOT NULL | Nome completo (PF) ou Razão Social (PJ). |
| cpf_cnpj | VARCHAR(14) | NOT NULL, UNIQUE | CPF ou CNPJ do cliente. |
| email | VARCHAR(100) | NOT NULL, UNIQUE | Endereço de e-mail para contato. |
| telefone | VARCHAR(20) | | Telefone para contato. |
| endereco_cobranca | VARCHAR(255) | | Endereço para faturamento. |
| cidade_origem | VARCHAR(100) | | Cidade de origem do cliente. |
| estado_origem | VARCHAR(2) | | Sigla do estado de origem do cliente. |

Tabela: Motorista

Descrição: Cadastro de pessoas habilitadas a conduzir os veículos, vinculadas a um cliente.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|-------------------------|---|
| id_motorista | INTEGER | PK | Identificador numérico único para cada motorista. |
| nome_completo | VARCHAR(150) | NOT NULL | Nome completo do motorista. |
| cpf | VARCHAR(11) | NOT NULL, UNIQUE | CPF do motorista. |
| cnh_numero | VARCHAR(20) | NOT NULL, UNIQUE | Número da Carteira Nacional de Habilitação. |
| cnh_categoria | VARCHAR(5) | NOT NULL | Categoria da CNH (A, B, C, etc.). |

| | | | |
|----------------------|---------|--|---|
| cnh_data_expiracao | DATE | NOT NULL | Data de validade da CNH. |
| id_cliente_associado | INTEGER | NOT NULL, FK -> Cliente(id_cliente) | Cliente ao qual o motorista está vinculado. |

Tabela: Protecao

Descrição: Catálogo de proteções e seguros adicionais oferecidos pela locadora.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|----------------|-------------------------|--|
| id_protecao | INTEGER | PK | Identificador único da proteção. |
| nome | VARCHAR(100) | NOT NULL, UNIQUE | Nome da proteção (Ex: "Seguro Total"). |
| descricao | TEXT | | Descrição detalhada da proteção. |
| valor_diaria | NUMERIC(10, 2) | NOT NULL | Valor da diária para esta proteção. |

Tabela: Acessorio

Descrição: Catálogo de acessórios que podem ser adicionados aos veículos ou locações.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|------------------|--|
| id_acessorio | INTEGER | PK | Identificador único do acessório. |
| nome | VARCHAR(100) | NOT NULL, UNIQUE | Nome do acessório (Ex: "Cadeira de Bebê"). |
| descricao | TEXT | | Descrição detalhada do acessório. |

Tabela: Reserva

Descrição: Registra as intenções de aluguel de veículos feitas pelos clientes.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|-----------------------------|--------------|---------------------------|------------------------------------|
| id_reserva | INTEGER | PK | Identificador único da reserva. |
| data_criacao_reserva | TIMESTAMP | DEFAULT CURRENT_TIMESTAMP | Data e hora da criação da reserva. |
| data_hora_retirada_prevista | TIMESTAMP | NOT NULL | Data e hora previstas |

| | | | |
|------------------------------|-------------|---|--|
| | | | para retirada do veículo. |
| data_hora_devolucao_prevista | TIMESTAMP | NOT NULL | Data e hora previstas para devolução do veículo. |
| status | VARCHAR(20) | DEFAULT 'Pendente', CHECK (status IN ('Confirmada', 'Pendente', 'Cancelada', 'Em Espera')) | Status atual da reserva. |
| id_cliente | INTEGER | NOT NULL, FK -> Cliente(id_cliente) | Cliente que fez a reserva. |
| id_grupo | INTEGER | NOT NULL, FK -> GrupoVeiculo(id_grupo) | Grupo de veículo reservado. |
| id_patio_retirada | INTEGER | NOT NULL, FK -> Patio(id_patio) | Pátio previsto para retirada. |
| id_patio_devolucao | INTEGER | NOT NULL, FK -> Patio(id_patio) | Pátio previsto para devolução. |

Tabela: Locacao

Descrição: Tabela transacional principal, que registra cada contrato de aluguel de veículo.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|--------------------------|----------------|---|--|
| id_locacao | INTEGER | PK | Identificador único da locação. |
| data_hora_retirada_real | TIMESTAMP | NOT NULL | Data e hora exatas da retirada. |
| data_hora_devolucao_real | TIMESTAMP | | Data e hora exatas da devolução. |
| km_saida | INTEGER | NOT NULL | Quilometragem do veículo na retirada. |
| km_chegada | INTEGER | | Quilometragem do veículo na devolução. |
| valor_cobrado_final | NUMERIC(10, 2) | | Valor final da locação após ajustes. |
| status | VARCHAR(15) | DEFAULT 'Ativa', CHECK (status IN ('Ativa', 'Finalizada', 'Cancelada')) | Status atual do contrato de locação. |

| | | | |
|--------------------|---------|--|--|
| id_reserva | INTEGER | FK -> Reserva(id_reserva) ON DELETE SET NULL | Referência à reserva que originou a locação (pode ser nulo). |
| id_cliente | INTEGER | NOT NULL, FK -> Cliente(id_cliente) | Cliente responsável pela locação. |
| id_motorista | INTEGER | NOT NULL, FK -> Motorista(id_motorista) | Condutor do veículo na locação. |
| id_veiculo | INTEGER | NOT NULL, FK -> Veiculo(id_veiculo) | Veículo específico alugado. |
| id_patio_retirada | INTEGER | NOT NULL, FK -> Patio(id_patio) | Pátio onde o veículo foi retirado. |
| id_patio_devolucao | INTEGER | FK -> Patio(id_patio) | Pátio onde o veículo foi devolvido. |

Tabela: VeiculoAcessorio

Descrição: Tabela associativa que relaciona Veículos com seus Acessórios (N:N).

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|-----------------|-----------|
|----------------|--------------|-----------------|-----------|

| | | | |
|--------------|---------|--|-----------------------------|
| id_veiculo | INTEGER | PK, FK -> Veiculo(id_veiculo) ON DELETE CASCADE | Identificador do veículo. |
| id_acessorio | INTEGER | PK, FK -> Acessorio(id_acessorio) ON DELETE CASCADE | Identificador do acessório. |

Tabela: LocacaoProtecao

Descrição: Tabela associativa que relaciona Locações com as Proteções contratadas (N:N).

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|------------------|----------------|---|--|
| id_locacao | INTEGER | PK, FK -> Locacao(id_locacao) ON DELETE CASCADE | Identificador da locação. |
| id_protecao | INTEGER | PK, FK -> Protecao(id_protecao) ON DELETE RESTRICT | Identificador da proteção. |
| valor_contratado | NUMERIC(10, 2) | NOT NULL | Valor da proteção no momento da contratação. |

Tabela: ProntuarioManutencao

Descrição: Registra o histórico de manutenções e serviços realizados nos veículos.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|--|---|
| id_prontuario | INTEGER | PK | Identificador único do registro de manutenção. |
| data_servico | DATE | NOT NULL | Data da realização do serviço. |
| descricao | TEXT | NOT NULL | Descrição do serviço realizado. |
| quilometragem | INTEGER | | Quilometragem do veículo no momento do serviço. |
| id_veiculo | INTEGER | NOT NULL, FK -> Veiculo(id_veiculo) | Veículo que recebeu o serviço. |

Tabela: FotoVeiculoEstado

Descrição: Armazena referências a fotos tiradas do veículo nos momentos de entrega e devolução.

| Nome da Coluna | Tipo de Dado | Chave/Restrição | Descrição |
|----------------|--------------|-----------------|------------------------------|
| id_foto | INTEGER | PK | Identificador único da foto. |

| | | | |
|----------------|--------------|--|---|
| url_foto | VARCHAR(255) | NOT NULL | Caminho ou URL para o arquivo da foto. |
| tipo_momento | VARCHAR(15) | NOT NULL, CHECK (tipo_momento IN ('Entrega', 'Devolução')) | Indica se a foto é da entrega ou devolução. |
| data_hora_foto | TIMESTAMP | NOT NULL | Data e hora em que a foto foi tirada. |
| id_locacao | INTEGER | NOT NULL, FK -> Locacao(id_locacao) | Locação à qual a foto está associada. |

Especificação das Restrições de Integridade

1. Restrição de Entidade (Entity Integrity)

Garante que cada registro em uma tabela seja único e não nulo em sua chave primária.

- **Implementação:** Através da definição de uma **Chave Primária (Primary Key - PK)** em todas as tabelas.
- **Exemplos:** id_vaga na tabela Vaga; id_locacao na tabela Locacao; a chave composta (id_veiculo, id_acessorio) em VeiculoAcessorio.

2. Restrição de Domínio (Domain Integrity)

Garante que os valores inseridos em uma coluna sejam válidos para seu domínio.

- **Implementação:**
 - **Tipos de Dados:** Definição precisa de tipos (INTEGER, VARCHAR, DATE, TIMESTAMP, NUMERIC, BOOLEAN).
 - **Restrição NOT NULL:** Campos essenciais não podem ser nulos (ex: Patio.nome, Veiculo.placa).

- **Restrição UNIQUE:** Garante unicidade para colunas ou conjuntos de colunas que não são PKs (ex: Empresa.cnpj, Vaga.id_patio, Vaga.codigo_vaga).
- **Restrição CHECK:** Valida se o valor da coluna está dentro de um conjunto permitido (ex: Vaga.status_vaga IN ('Livre', 'Ocupada', ...), Cliente.tipo_cliente IN ('PF', 'PJ')).
- **Restrição DEFAULT:** Define um valor padrão se nenhum for fornecido (ex: Vaga.status_vaga DEFAULT 'Livre').

3. Restrição de Integridade Referencial (Referential Integrity)

Mantém a consistência entre tabelas relacionadas através de Chaves Estrangeiras.

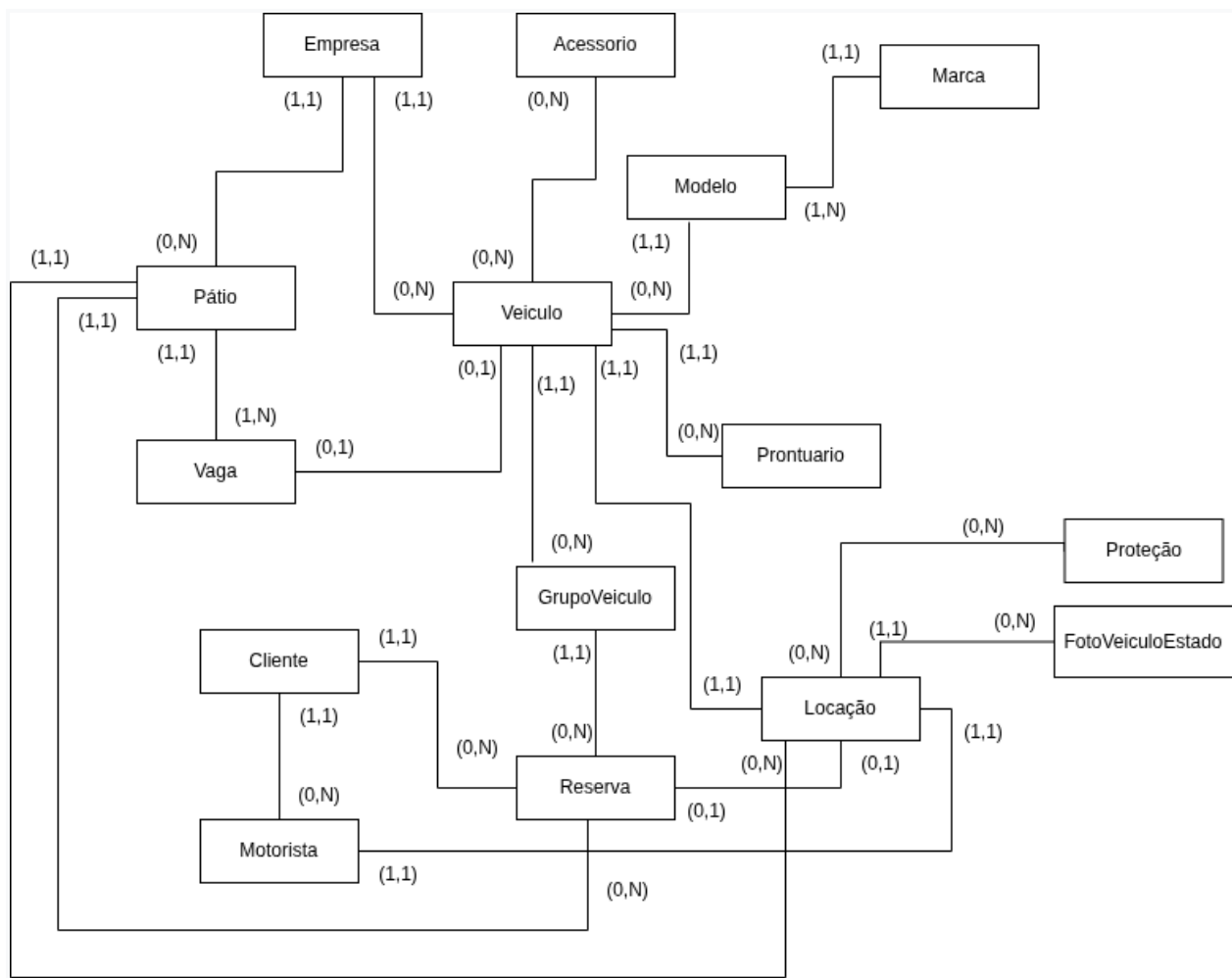
- **Implementação:** Uso de **Chaves Estrangeiras (Foreign Keys - FK)**.
- **Exemplos:**
 - Vaga.id_patio deve corresponder a um id_patio existente na tabela Patio.
 - Veiculo.id_vaga_atual deve corresponder a um id_vaga existente na tabela Vaga ou ser NULL.
- **Ações Referenciais ON DELETE / ON UPDATE:**
 - **ON DELETE SET NULL:** Usado em Veiculo.id_vaga_atual. Se uma vaga for excluída, o veículo associado terá id_vaga_atual definido como NULL. Similar para Locacao.id_reserva.
 - **ON DELETE CASCADE:** Usado em tabelas de ligação como VeiculoAcessorio e LocacaoProtecao. Se o registro pai (ex: Veiculo) for excluído, os registros filhos correspondentes são excluídos.
 - **ON DELETE RESTRICT (ou padrão do SGBD):** Impede a exclusão de um registro pai se houver filhos dependentes (ex: não deletar Marca se houver Modelo associado).

4. Restrições de Negócio (Business Rules)

Regras específicas do negócio, algumas implícitas no modelo, outras que podem ser reforçadas por CHECK constraints mais complexas, Triggers ou pela camada de aplicação.

- **Exemplos:**
 - A data_hora_devolucao_real em Locacao deve ser posterior à data_hora_retirada_real.
 - Um veículo com status_operacional 'Em Manutenção' não deveria estar associado a uma Locacao 'Ativa'.
 - A CNH de um Motorista não pode estar vencida no momento da Locacao.
 - Uma Vaga com status_vaga 'Ocupada' deve ter um Veiculo associado a ela (esta pode ser gerenciada pela aplicação ou triggers).

Modelagem



Script SQL/DDL do Modelo Físico do Banco de Dados

Modelo Físico (SQL/DDL)

Script DDL (Data Definition Language), em padrão ANSI SQL:

```
CREATE TABLE Empresa (  
    id_empresa INTEGER PRIMARY KEY,  
    nome_fantasia VARCHAR(100) NOT NULL,  
    razao_social VARCHAR(150) NOT NULL,  
    cnpj VARCHAR(14) NOT NULL UNIQUE  
);
```

```
CREATE TABLE Patio (  
    id_patio INTEGER PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    endereco VARCHAR(255),  
    cidade VARCHAR(100),  
    id_empresa_proprietaria INTEGER NOT NULL,  
    CONSTRAINT fk_patio_empresa FOREIGN KEY (id_empresa_proprietaria) REFERENCES  
Empresa(id_empresa)  
);
```

```
CREATE TABLE Vaga (  
    id_vaga INTEGER PRIMARY KEY,  
    id_patio INTEGER NOT NULL,  
    codigo_vaga VARCHAR(20) NOT NULL, -- Código da vaga dentro do pátio (Ex: A01, S2-10)  
    status_vaga VARCHAR(15) DEFAULT 'Livre' CHECK (status_vaga IN ('Livre', 'Ocupada', 'Manutenção',  
'Reservada')),  
    tipo_vaga VARCHAR(50), -- Opcional: Ex: 'Normal', 'PCD', 'Veículo Elétrico', 'Coberta'  
    CONSTRAINT fk_vaga_patio FOREIGN KEY (id_patio) REFERENCES Patio(id_patio),  
    CONSTRAINT uq_patio_codigo_vaga UNIQUE (id_patio, codigo_vaga) -- Garante que o código da vaga  
seja único dentro de cada pátio  
);
```

```
CREATE TABLE GrupoVeiculo (  
    id_grupo INTEGER PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL UNIQUE,  
    descricao TEXT,  
    valor_diaria_base NUMERIC(10, 2) NOT NULL  
);
```

```
CREATE TABLE Marca (  
    id_marca INTEGER PRIMARY KEY,
```



```
    nome VARCHAR(50) NOT NULL UNIQUE
);
```

```
CREATE TABLE Modelo (
    id_modelo INTEGER PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    id_marca INTEGER NOT NULL,
    CONSTRAINT fk_modelo_marca FOREIGN KEY (id_marca) REFERENCES Marca(id_marca)
);
```

```
CREATE TABLE Protecao (
    id_protecao INTEGER PRIMARY KEY,
    nome VARCHAR(100) NOT NULL UNIQUE,
    descricao TEXT,
    valor_diaria NUMERIC(10, 2) NOT NULL
);
```

```
CREATE TABLE Acessorio (
    id_acessorio INTEGER PRIMARY KEY,
    nome VARCHAR(100) NOT NULL UNIQUE,
    descricao TEXT
);
```

```
CREATE TABLE Veiculo (
    id_veiculo INTEGER PRIMARY KEY,
    placa VARCHAR(7) NOT NULL UNIQUE,
    chassi VARCHAR(17) NOT NULL UNIQUE,
    cor VARCHAR(30),
    ano_fabricacao INTEGER,
    mecanizacao VARCHAR(15) CHECK (mecanizacao IN ('Manual', 'Automática')),
    tem_ar_condicionado BOOLEAN DEFAULT FALSE,
    status_operacional VARCHAR(20) DEFAULT 'Disponível' CHECK (status_operacional IN ('Disponível',
'Alugado', 'Em Manutenção', 'Reservado')),
    id_modelo INTEGER NOT NULL,
    id_grupo INTEGER NOT NULL,
    id_empresa_proprietaria INTEGER NOT NULL,
    id_vaga_atual INTEGER, -- Veículo pode não estar em uma vaga específica (NULL)
    CONSTRAINT fk_veiculo_modelo FOREIGN KEY (id_modelo) REFERENCES Modelo(id_modelo),
    CONSTRAINT fk_veiculo_grupo FOREIGN KEY (id_grupo) REFERENCES GrupoVeiculo(id_grupo),
    CONSTRAINT fk_veiculo_empresa FOREIGN KEY (id_empresa_proprietaria) REFERENCES
Empresa(id_empresa),
    CONSTRAINT fk_veiculo_vaga FOREIGN KEY (id_vaga_atual) REFERENCES Vaga(id_vaga) ON
DELETE SET NULL -- Se uma vaga for removida, o veículo fica sem vaga, mas não é excluído.
);
```

```
CREATE TABLE Cliente (
```

```

id_cliente INTEGER PRIMARY KEY,
tipo_cliente VARCHAR(2) NOT NULL CHECK (tipo_cliente IN ('PF', 'PJ')),
nome_razao_social VARCHAR(150) NOT NULL,
cpf_cnpj VARCHAR(14) NOT NULL UNIQUE,
email VARCHAR(100) NOT NULL UNIQUE,
telefone VARCHAR(20),
endereco_cobranca VARCHAR(255),
cidade_origem VARCHAR(100),
estado_origem VARCHAR(2)
);

```

```

CREATE TABLE Motorista (
    id_motorista INTEGER PRIMARY KEY,
    nome_completo VARCHAR(150) NOT NULL,
    cpf VARCHAR(11) NOT NULL UNIQUE,
    cnh_numero VARCHAR(20) NOT NULL UNIQUE,
    cnh_categoria VARCHAR(5) NOT NULL,
    cnh_data_expiracao DATE NOT NULL,
    id_cliente_associado INTEGER NOT NULL,
    CONSTRAINT fk_motorista_cliente FOREIGN KEY (id_cliente_associado) REFERENCES
Cliente(id_cliente)
);

```

```

CREATE TABLE Reserva (
    id_reserva INTEGER PRIMARY KEY,
    data_criacao_reserva TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    data_hora_retirada_prevista TIMESTAMP NOT NULL,
    data_hora_devolucao_prevista TIMESTAMP NOT NULL,
    status VARCHAR(20) DEFAULT 'Pendente' CHECK (status IN ('Confirmada', 'Pendente', 'Cancelada', 'Em
Espera')),
    id_cliente INTEGER NOT NULL,
    id_grupo INTEGER NOT NULL,
    id_patio_retirada INTEGER NOT NULL,
    id_patio_devolucao INTEGER NOT NULL,
    CONSTRAINT fk_reserva_cliente FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente),
    CONSTRAINT fk_reserva_grupo FOREIGN KEY (id_grupo) REFERENCES GrupoVeiculo(id_grupo),
    CONSTRAINT fk_reserva_patio_retirada FOREIGN KEY (id_patio_retirada) REFERENCES
Patio(id_patio),
    CONSTRAINT fk_reserva_patio_devolucao FOREIGN KEY (id_patio_devolucao) REFERENCES
Patio(id_patio)
);

```

```

CREATE TABLE Locacao (
    id_locacao INTEGER PRIMARY KEY,
    data_hora_retirada_real TIMESTAMP NOT NULL,

```

```

data_hora_devolucao_real TIMESTAMP,
km_saida INTEGER NOT NULL,
km_chegada INTEGER,
valor_cobrado_final NUMERIC(10, 2),
status VARCHAR(15) DEFAULT 'Ativa' CHECK (status IN ('Ativa', 'Finalizada', 'Cancelada')),
id_reserva INTEGER,
id_cliente INTEGER NOT NULL,
id_motorista INTEGER NOT NULL,
id_veiculo INTEGER NOT NULL,
id_patio_retirada INTEGER NOT NULL,
id_patio_devolucao INTEGER,
CONSTRAINT fk_locacao_reserva FOREIGN KEY (id_reserva) REFERENCES Reserva(id_reserva) ON
DELETE SET NULL,
CONSTRAINT fk_locacao_cliente FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente),
CONSTRAINT fk_locacao_motorista FOREIGN KEY (id_motorista) REFERENCES
Motorista(id_motorista),
CONSTRAINT fk_locacao_veiculo FOREIGN KEY (id_veiculo) REFERENCES Veiculo(id_veiculo),
CONSTRAINT fk_locacao_patio_retirada FOREIGN KEY (id_patio_retirada) REFERENCES
Patio(id_patio),
CONSTRAINT fk_locacao_patio_devolucao FOREIGN KEY (id_patio_devolucao) REFERENCES
Patio(id_patio)
);

```

```

CREATE TABLE VeiculoAcessorio (
    id_veiculo INTEGER NOT NULL,
    id_acessorio INTEGER NOT NULL,
    CONSTRAINT pk_veiculo_acessorio PRIMARY KEY (id_veiculo, id_acessorio),
    CONSTRAINT fk_va_veiculo FOREIGN KEY (id_veiculo) REFERENCES Veiculo(id_veiculo) ON
DELETE CASCADE,
    CONSTRAINT fk_va_acessorio FOREIGN KEY (id_acessorio) REFERENCES Acessorio(id_acessorio)
ON DELETE CASCADE
);

```

```

CREATE TABLE LocacaoProtecao (
    id_locacao INTEGER NOT NULL,
    id_protecao INTEGER NOT NULL,
    valor_contratado NUMERIC(10, 2) NOT NULL,
    CONSTRAINT pk_locacao_protecao PRIMARY KEY (id_locacao, id_protecao),
    CONSTRAINT fk_lp_locacao FOREIGN KEY (id_locacao) REFERENCES Locacao(id_locacao) ON
DELETE CASCADE,
    CONSTRAINT fk_lp_protecao FOREIGN KEY (id_protecao) REFERENCES Protecao(id_protecao) ON
DELETE RESTRICT
);

```

```

CREATE TABLE ProntuarioManutencao (

```

```
id_prontuario INTEGER PRIMARY KEY,  
data_servico DATE NOT NULL,  
descricao TEXT NOT NULL,  
quilometragem INTEGER,  
id_veiculo INTEGER NOT NULL,  
CONSTRAINT fk_prontuario_veiculo FOREIGN KEY (id_veiculo) REFERENCES Veiculo(id_veiculo)  
);
```

```
CREATE TABLE FotoVeiculoEstado (  
id_foto INTEGER PRIMARY KEY,  
url_foto VARCHAR(255) NOT NULL,  
tipo_momento VARCHAR(15) NOT NULL CHECK (tipo_momento IN ('Entrega', 'Devolução')),  
data_hora_foto TIMESTAMP NOT NULL,  
id_locacao INTEGER NOT NULL,  
CONSTRAINT fk_foto_locacao FOREIGN KEY (id_locacao) REFERENCES Locacao(id_locacao)  
);
```