# Padrões de Codificação de Banco de Dados para o Sistema AlugueLink

Padrões de Codificação de Banco de Dados para o Sistema AlugueLink

Este documento estabelece os padrões de codificação para o banco de dados do sistema AlugueLink. A adesão a estes padrões garante consistência, legibilidade, manutenibilidade e otimização do desempenho do banco de dados.

# 1. Convenções de Nomenclatura

## 1.1. Tabelas

Nomes de tabelas devem ser no plural, em snake\_case (minúsculas com sublinhados para separar palavras).

Devem ser descritivos e refletir o conteúdo da tabela.

Exemplos: locadores, imoveis, contratos\_aluguel, pagamentos.

## 1.2. Colunas

Nomes de colunas devem ser no singular, em snake\_case.

Devem ser descritivos e refletir o dado que armazenam.

Exemplos: nome, email, data\_cadastro, valor\_total.

## 1.3. Chaves Primárias (PK)

Devem ser nomeadas como id\_<nome\_da\_tabela\_singular>.

Exemplos: id\_locador, id\_imovel, id\_contrato\_aluguel.

## 1.4. Chaves Estrangeiras (FK)

Devem ser nomeadas como id\_<nome\_da\_tabela\_referenciada\_singular>.

Exemplos: id\_locador (em imoveis), id\_imovel (em contratos\_aluguel),

id\_inquilino (em contratos\_aluguel).

## 1.5. Índices

Índices devem ser prefixados com idx\_ seguido do nome da tabela e do nome

da coluna (ou colunas) indexada(s).

Exemplos: idx\_imoveis\_id\_locador, idx\_contratos\_aluguel\_id\_inquilino.

## 1.6. Restrições (Constraints)

Restrições de chave primária: pk\_<nome\_da\_tabela>.

Restrições de chave estrangeira: fk\_<tabela\_origem>\_<tabela\_destino>.

Restrições de unicidade: uq\_<nome\_da\_tabela>\_<nome\_da\_coluna>.

Exemplos: pk\_locadores, fk\_imoveis\_locadores, uq\_locadores\_cpf\_cnpj.

## 2. Tipos de Dados

Utilizar tipos de dados apropriados para minimizar o espaço de armazenamento

e otimizar o desempenho.

Inteiros: INT para IDs e números pequenos, BIGINT para números muito

grandes.

Strings: VARCHAR com tamanho adequado para textos variáveis, TEXT para

textos longos.

Datas e Horas: DATE para datas, TIME para horas, DATETIME ou TIMESTAMP

para data e hora.

Valores Monetários: DECIMAL(P,S) para valores monetários, especificando

precisão e escala (ex: DECIMAL(10,2)).

Booleanos: BOOLEAN ou TINYINT(1) (onde 0 é falso e 1 é verdadeiro).

## 3. Chaves Primárias e Estrangeiras

Todas as tabelas devem ter uma chave primária.

Chaves primárias devem ser AUTO\_INCREMENT (ou equivalente) para garantir

unicidade e facilidade de inserção.

Chaves estrangeiras devem ser definidas com ON DELETE e ON UPDATE para

garantir a integridade referencial.

ON DELETE CASCADE: Deleta registros dependentes automaticamente.

ON DELETE SET NULL: Define a FK como NULL se o registro pai for deletado

(se a coluna permitir NULL).

ON DELETE RESTRICT: Impede a exclusão do registro pai se houver

registros filhos dependentes.

ON UPDATE CASCADE: Atualiza a FK automaticamente se a PK pai for

atualizada.

## 4. Índices

Criar índices em colunas frequentemente usadas em cláusulas WHERE, JOIN,

ORDER BY e GROUP BY.

Chaves primárias e estrangeiras são automaticamente indexadas na maioria dos

SGBDs, mas verificar se o índice é eficiente.

Evitar a criação excessiva de índices, pois eles podem impactar o desempenho de

operações de escrita (INSERT, UPDATE, DELETE).

## 5. Normalização

O banco de dados deve seguir as formas normais (pelo menos 3FN) para evitar

redundância de dados e anomalias de atualização.

Considerar a desnormalização controlada para otimização de leitura em casos

específicos, documentando a justificativa.

## 6. Comentários

Adicionar comentários (COMMENT) em tabelas e colunas para descrever seu

propósito, especialmente para campos não óbvios ou regras de negócio

específicas.

## 7. Conjunto de Caracteres e Collation

Utilizar UTF-8 (ou utf8mb4 para suporte completo a emojis e caracteres

especiais) como conjunto de caracteres padrão para o banco de dados e tabelas.

Definir uma collation apropriada para o idioma, como utf8mb4\_unicode\_ci

para português, que é case-insensitive e acentuação-insensitive.

## 8. Segurança

Evitar o uso de root para conexões de aplicação.

Criar usuários com privilégios mínimos necessários para cada aplicação ou

serviço.

Utilizar senhas fortes e rotacioná-las regularmente.

## 9. Versionamento

Utilizar ferramentas de controle de versão Git para gerenciar scripts de

banco de dados (DDL e DML).

Manter um histórico de alterações do schema do banco de dados.

## 10. Exemplos de DDL (Data Definition Language)

CREATE TABLE locadores (

id\_locador INT AUTO\_INCREMENT PRIMARY KEY COMMENT 'Chave primária do

locador',

nome VARCHAR(255) NOT NULL COMMENT 'Nome completo do locador',

cpf\_cnpj VARCHAR(18) NOT NULL UNIQUE COMMENT 'CPF ou CNPJ do locador',

telefone VARCHAR(20) COMMENT 'Número de telefone do locador',

email VARCHAR(255) UNIQUE COMMENT 'Endereço de e-mail do locador'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE inquilinos (

id\_inquilino INT AUTO\_INCREMENT PRIMARY KEY COMMENT 'Chave primária do

inquilino',

nome VARCHAR(255) NOT NULL COMMENT 'Nome completo do inquilino',

cpf VARCHAR(14) NOT NULL UNIQUE COMMENT 'CPF do inquilino',

telefone VARCHAR(20) COMMENT 'Número de telefone do inquilino',

email VARCHAR(255) UNIQUE COMMENT 'Endereço de e-mail do inquilino'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE imoveis (

id\_imovel INT AUTO\_INCREMENT PRIMARY KEY COMMENT 'Chave primária do

imóvel',

id\_locador INT NOT NULL COMMENT 'Chave estrangeira para a tabela

locadores',

endereco VARCHAR(255) NOT NULL COMMENT 'Endereço completo do imóvel',

cidade VARCHAR(100) NOT NULL COMMENT 'Cidade onde o imóvel está

localizado',

estado VARCHAR(2) NOT NULL COMMENT 'Estado onde o imóvel está localizado

(UF)',

cep VARCHAR(10) COMMENT 'CEP do imóvel',

tipo\_imovel VARCHAR(50) COMMENT 'Tipo do imóvel (ex: apartamento, casa,

comercial)',

valor\_aluguel DECIMAL(10,2) NOT NULL COMMENT 'Valor do aluguel mensal do

imóvel',

disponivel BOOLEAN DEFAULT TRUE COMMENT 'Indica se o imóvel está disponível

para aluguel',

CONSTRAINT fk\_imoveis\_locadores FOREIGN KEY (id\_locador) REFERENCES

locadores(id\_locador) ON DELETE RESTRICT ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE contratos\_aluguel (

id\_contrato INT AUTO\_INCREMENT PRIMARY KEY COMMENT 'Chave primária do

contrato de aluguel',

id\_imovel INT NOT NULL COMMENT 'Chave estrangeira para a tabela imoveis',

id\_inquilino INT NOT NULL COMMENT 'Chave estrangeira para a tabela

inquilinos',

data\_inicio DATE NOT NULL COMMENT 'Data de início do contrato',

data\_fim DATE COMMENT 'Data de término do contrato (pode ser nula para

contratos ativos sem data final definida)',

valor\_mensal DECIMAL(10,2) NOT NULL COMMENT 'Valor mensal do aluguel

definido no contrato',

dia\_vencimento INT NOT NULL COMMENT 'Dia do mês para vencimento do

aluguel',

status\_contrato VARCHAR(50) DEFAULT 'Ativo' COMMENT 'Status atual do

contrato (ex: Ativo, Encerrado, Pendente)',

CONSTRAINT fk\_contratos\_aluguel\_imoveis FOREIGN KEY (id\_imovel) REFERENCES

imoveis(id\_imovel) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT fk\_contratos\_aluguel\_inquilinos FOREIGN KEY (id\_inquilino)

REFERENCES inquilinos(id\_inquilino) ON DELETE RESTRICT ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE pagamentos (

id\_pagamento INT AUTO\_INCREMENT PRIMARY KEY COMMENT 'Chave primária do

pagamento',

id\_contrato INT NOT NULL COMMENT 'Chave estrangeira para a tabela

contratos\_aluguel',

data\_pagamento DATE NOT NULL COMMENT 'Data em que o pagamento foi

realizado',

valor\_pago DECIMAL(10,2) NOT NULL COMMENT 'Valor pago',

referencia\_mes VARCHAR(7) NOT NULL COMMENT 'Mês de referência do pagamento

(formato YYYY-MM)',

status\_pagamento VARCHAR(50) DEFAULT 'Pendente' COMMENT 'Status do

pagamento (ex: Pago, Pendente, Atrasado)',

CONSTRAINT fk\_pagamentos\_contratos\_aluguel FOREIGN KEY (id\_contrato)

REFERENCES contratos\_aluguel(id\_contrato) ON DELETE RESTRICT ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

Referências

[1] SQL Style Guide. Disponível em: https://www.sqlstyle.guide/

[2] MySQL Documentation. Disponível em: https://dev.mysql.com/doc/