

Padrão de Nomenclatura - Engenharia de Dados

Versão: 2.0 - APENAS PREFIXOS

Data: 2025-10-13

Princípios Fundamentais

SEMPRE

- Use **snake_case** (minúsculas com underscore)
- Use **APENAS PREFIXOS** para qualificar campos
- Seja **descritivo e claro**
- Evite **abreviações** obscuras
- Adicione **comentários** em colunas importantes

NUNCA

- Não use sufixos (_id, _sk, _nome, _data) 
 - Não use CamelCase ou PascalCase
 - Não use espaços ou caracteres especiais
 - Não use palavras reservadas SQL
 - Não use acentos ou cedilha
-

Prefixos para Tabelas

Prefixo	Contexto	Exemplo
<code>dim_</code>	Dimensão (Star Schema)	<code>dim_cliente</code> , <code>dim_produto</code>
<code>fato_</code>	Fato (Star Schema)	<code>fato_vendas</code> , <code>fato_estoque</code>
<code>raw_</code>	Camada Raw/Bronze	<code>raw_cliente</code>
<code>stg_</code>	Camada Staging	<code>stg_cliente</code>
<code>agg_</code>	Agregação	<code>agg_vendas_diarias</code>
<code>rpt_</code>	Relatório	<code>rpt_vendas_regiao</code>
<code>tmp_</code>	Temporária	<code>tmp_carga</code>
<code>hist_</code>	Histórico	<code>hist_cliente</code>
<code>aud_</code>	Auditória	<code>aud_acesso</code>

Prefixo	Contexto	Exemplo
(sem prefixo)	Core/Produção	<code>cliente</code> , <code>produto</code>

🔑 Prefixos para Campos

Chaves

Prefixo	Tipo	Uso	Exemplo
<code>id</code>	Primary Key	Identificador único	<code>id_cliente</code> , <code>id_produto</code>
<code>fk</code>	Foreign Key	Referência (tabelas ER)	<code>fk_cliente</code> , <code>fk_vendedor</code>
<code>sk</code>	Surrogate Key	Chave artificial (dimensões)	<code>sk_cliente</code> , <code>sk_produto</code>
<code>nk</code>	Natural Key	Chave natural do sistema origem	<code>nk_cliente</code> , <code>nk_produto</code>

Exemplos:

```
sql

-- Tabela ER: pedido
id_pedido      -- PK
fk_cliente     -- FK
fk_vendedor    -- FK

-- Dimensão: dim_cliente
sk_cliente     -- Surrogate Key
nk_cliente     -- Natural Key (ID origem)
```

Datas e Timestamps

Prefixo	Tipo	Exemplo
<code>dt</code>	DATE	<code>dt_criacao</code> , <code>dt_nascimento</code> , <code>dt_vencimento</code>
<code>ts</code>	TIMESTAMP	<code>ts_criacao</code> , <code>ts_atualizacao</code> , <code>ts_processamento</code>

Booleanos (Flags)

Prefixo	Uso	Exemplo
<code>fl</code>	Flag/Boolean	<code>fl_ativo</code> , <code>fl_pago</code> , <code>fl_disponivel</code>
<code>is</code>	Alternativa (internacional)	<code>is_ativo</code> , <code>is_premium</code>

Valores e Medidas

Prefixo	Tipo	Exemplo
vl_	Valor monetário	(vl_total), (vl_desconto), (vl_frete)
qt_	Quantidade/Contagem	(qt_estoque), (qt_vendida), (qt_itens)
nr_	Número sequencial	(nr_pedido), (nr_nota_fiscal), (nr_versao)
pc_	Percentual	(pc_desconto), (pc_margem), (pc_comissao)
md_	Medida física	(md_peso_kg), (md_altura_cm), (md_volume_m3)

Textos e Descrições

Prefixo	Tipo	Exemplo
nm_	Nome/Identificador	(nm_cliente), (nm_produto), (nm_marca)
tx_	Texto geral	(tx_email), (tx_telefone), (tx_endereco)
ds_	Descrição	(ds_produto), (ds_observacao), (ds_detalhada)
cd_	Código alfanumérico	(cd_sku), (cd_cpf), (cd_rastreio)

Status e Categorias

Prefixo	Tipo	Exemplo
st_	Status	(st_pedido), (st_pagamento), (st_entrega)
tp_	Tipo	(tp_pagamento), (tp_entrega), (tp_cliente)
ct_	Categoria	(ct_produto), (ct_segmento), (ct_risco)

Auditoria

Prefixo	Tipo	Exemplo
aud_	Campos de auditoria	(aud_usuario_criacao), (aud_ip_origem)

Estruturas Padrão

Modelagem Tradicional (ER)

```
CREATE TABLE cliente (
    -- Chaves
    id_cliente BIGINT PRIMARY KEY,
    
    -- Dados
    nm_cliente STRING NOT NULL,
    tx_email STRING NOT NULL,
    cd_cpf STRING,
    dt_nascimento DATE,
    
    -- Status
    fl_ativo BOOLEAN DEFAULT TRUE,
    
    -- Métricas
    qt_pedidos INT DEFAULT 0,
    vl_total_gasto DECIMAL(12,2) DEFAULT 0,
    
    -- Auditoria
    ts_criacao TIMESTAMP NOT NULL,
    ts_atualizacao TIMESTAMP,
    aud_usuario_criacao STRING
);
```

```
CREATE TABLE pedido (
    -- Chaves
    id_pedido BIGINT PRIMARY KEY,
    fk_cliente BIGINT NOT NULL,
    fk_vendedor BIGINT,
    
    -- Identificação
    nr_pedido STRING NOT NULL,
    
    -- Datas
    dt_pedido DATE NOT NULL,
    ts_criacao TIMESTAMP NOT NULL,
    
    -- Status
    st_pedido STRING NOT NULL,
    
    -- Valores
    vl_subtotal DECIMAL(12,2) NOT NULL,
    vl_desconto DECIMAL(12,2) DEFAULT 0,
    vl_total DECIMAL(12,2) NOT NULL
);
```

Modelagem Dimensional (Star Schema)

Dimensão:

sql

```
CREATE TABLE dim_cliente (
    -- Chaves
    sk_cliente BIGINT PRIMARY KEY,
    nk_cliente BIGINT NOT NULL,

    -- Atributos
    nm_cliente STRING,
    tx_email STRING,
    cd_cpf STRING,
    ct_segmento STRING,

    -- Geografia (hierarquia)
    tx_cidade STRING,
    tx_estado STRING,
    ct_regiao STRING,

    -- SCD Type 2
    nr_versao INT NOT NULL,
    dt_inicio_vigencia DATE NOT NULL,
    dt_fim_vigencia DATE,
    fl_registro_atual BOOLEAN NOT NULL,

    -- Auditoria
    ts_carga TIMESTAMP NOT NULL
);
```

Fato:

sql

```

CREATE TABLE fato_vendas (
    -- Chaves (FKs para dimensões)
    sk_cliente BIGINT NOT NULL,
    sk_produto BIGINT NOT NULL,
    sk_tempo INT NOT NULL,
    sk_vendedor BIGINT NOT NULL,

    -- Chave degenerada
    nr_pedido STRING,

    -- Medidas aditivas
    qt_vendida INT NOT NULL,
    vl_bruto DECIMAL(12,2) NOT NULL,
    vl_desconto DECIMAL(12,2) NOT NULL,
    vl_liquido DECIMAL(12,2) NOT NULL,
    vl_custo DECIMAL(12,2) NOT NULL,

    -- Medidas não-aditivas
    vl_unitario DECIMAL(10,2) NOT NULL,
    pc_desconto DECIMAL(5,2),
    pc_margem DECIMAL(5,2),

    -- Auditoria
    ts_carga TIMESTAMP NOT NULL
);

```

Dimensão Tempo:

sql

```

CREATE TABLE dim_tempo (
    sk_tempo INT PRIMARY KEY,
    dt_completa DATE NOT NULL,

    -- Ano
    nr_ano INT NOT NULL,
    nr_ano_mes INT NOT NULL,

    -- Mes
    nr_mes INT NOT NULL,
    nm_mes STRING NOT NULL,
    nm_mes_abrev STRING NOT NULL,

    -- Dia
    nr_dia INT NOT NULL,
    nr_dia_ano INT NOT NULL,
    nr_dia_semana INT NOT NULL,
    nm_dia_semana STRING NOT NULL,
    nm_dia_semana_abrev STRING NOT NULL,

    -- Flags
    fl_final_semana BOOLEAN NOT NULL,
    fl_feriado BOOLEAN NOT NULL,
    fl_dia_util BOOLEAN NOT NULL,
    nm_feriado STRING
);

```

🔍 Índices e Constraints

Índices

sql

FORMATO: idx_<tabela>_<coluna(s)>

```

CREATE INDEX idx_cliente_tx_email ON cliente(tx_email);
CREATE INDEX idx_pedido_fk_cliente ON pedido(fk_cliente);
CREATE UNIQUE INDEX unq_cliente_cd_cpf ON cliente(cd_cpf);

```

Constraints

sql

FORMATO: <tipo> _ <tabela> _ <descritor>

-- Primary Key

pk_cliente

pk_produto

-- Foreign Key

fk_pedido_cliente

fk_item_pedido_produto

-- Check

chk_produto_vl_positivo

chk_pedido_dt_valida

-- Unique

unq_cliente_cd_cpf

unq_produto_cd_sku

Tabela Resumo de Prefixos

Prefixo	Significado	Exemplo
id_	Primary Key	id_cliente
fk_	Foreign Key	fk_cliente
sk_	Surrogate Key	sk_cliente
nk_	Natural Key	nk_cliente
dt_	Date	dt_criacao
ts_	Timestamp	ts_criacao
fl_	Boolean/Flag	fl_ativo
vl_	Valor monetário	vl_total
nm_	Nome	nm_cliente
tx_	Texto	tx_email
ds_	Descrição	ds_produto
cd_	Código	cd_cpf
qt_	Quantidade	qt_estoque
nr_	Número	nr_pedido
pc_	Percentual	pc_desconto
md_	Medida física	md_peso_kg
st_	Status	st_pedido

Prefixo	Significado	Exemplo
tp_	Tipo	tp_pagamento
ct_	Categoria	ct_produto
aud_	Auditória	aud_usuario

✓ Checklist de Validação

Tabelas

- Nome em snake_case?
- Prefixo correto para a camada?
- Nome descritivo e claro?
- Evitou palavras reservadas SQL?

Colunas

- Nome em snake_case?
- Possui prefixo correto?
- NÃO possui sufixo? ✗
- PKs usam `id_<entidade>`?
- FKs usam `fk_<tabela_referenciada>`?
- SKs usam `sk_<dimensao>`?
- Datas usam `dt` ou `ts`?
- Valores monetários usam `vl`?
- Booleanos usam `fl` ou `is`?
- Tem comentário se necessário?

🎯 Comparação: Sufixo vs Prefixo

✗ Com Sufixo	✓ Com Prefixo
<code>cliente_id</code>	<code>id_cliente</code>
<code>data_criacao</code>	<code>dt_criacao</code>
<code>valor_total</code>	<code>vl_total</code>
<code>nome_cliente</code>	<code>nm_cliente</code>
<code>ativo_flag</code>	<code>fl_ativo</code>
<code>quantidade_estoque</code>	<code>qt_estoque</code>
<code>numero_pedido</code>	<code>nr_pedido</code>

Benefícios do Padrão

Vantagens de usar APENAS PREFIXOS:

- **Agrupamento natural:** Campos do mesmo tipo ficam juntos alfabeticamente
- **Identificação rápida:** Tipo do campo é imediato
- **Autocompletar eficiente:** IDEs agrupam por prefixo
- **Padrão único:** Sem ambiguidade
- **Leitura consistente:** Todos os campos seguem o mesmo padrão

Benefícios gerais:

- Código legível e manutenível
 - Onboarding rápido
 - Menos erros e confusões
 - Documentação implícita
 - Queries SQL mais claras
-

Lembre-se: Consistência é mais importante que perfeição. Escolha este padrão e siga-o em todo o projeto!