

# **Documentação técnica — Modelo de Banco de Dados com atributos de pessoas (requerente e requerido) + MER + Script de criação**

## Objetivo

- Modelar um banco relacional que suporte consultas ricas sobre acórdãos/decisões e permita explorar características das partes (requerente e requerido), incluindo:
  - gênero
  - profissão
  - estado civil
  - nacionalidade
  - estado de origem da demanda (UF de origem do processo)
  - estado associado à identidade (UF emissora do documento)
  - além dos metadados de acórdão previstos no layout do JSON já documentado (id, número do processo, classe, órgão julgador, ministro relator, ementa, tipo de decisão, data de julgamento, decisão, jurisprudência citada, notas, informações complementares, termos auxiliares, tese jurídica, tema, referências legislativas, acórdãos similares, data de publicação).

## Abordagem

- Relacional normalizado (PostgreSQL), com dimensões “fechadas” (gênero, estado civil, UF, país, papel da parte) e dimensões “abertas” (profissão com base em CBO opcional).
- Separação entre Processo e Acórdão (um processo pode ter múltiplos acórdãos/decisões ao longo do tempo).
- Partes associadas ao Processo com um papel (ex.: REQUERENTE, REQUERIDO) e atributos demográficos/socioeconômicos.
- Documento de identidade modelado separadamente para capturar a UF emissora (estado “na identidade”).
- Campos textuais dos acórdãos indexados para busca full-text.

## Resumo das entidades principais

- dim\_uf: estados brasileiros (UF).
- dim\_pais: países (nacionalidade).
- dim\_genero: classificação de gênero.
- dim\_estado\_civil: estado civil.
- dim\_profissao: catálogo de profissões (ex.: CBO).
- dim\_papel\_parte: papéis processuais (REQUIRENTE/REQUERIDO/...).
- classe\_processual: classe (sigla/descrição).
- orgao\_julgador, ministro: catálogos auxiliares.
- processo: identificação, classe, órgão, UF de origem da demanda.
- pessoa: dados da pessoa física/jurídica com demografia (gênero, estado civil, nacionalidade, profissão) e UF de residência/domicílio quando aplicável.
- pessoa\_documento: documentos de identificação (com UF emissora).

- processo\_parte: relação N:N entre processo e pessoa, por papel.
- acordao: metadados e textos do acórdão vinculados ao processo.

### Mapeamento do layout JSON → Modelo

- id → acordao.id\_externo (mantemos o id interno separado)
- numeroProcesso → processo.numero\_processo
- numeroRegistro → processo.numero\_registro
- siglaClasse → processo.sigla\_classe (FK para classe\_processual)
- descricaoClasse → classe\_processual.descricao
- nomeOrgaoJulgador → orgao\_julgador.nome (FK em processo)
- ministroRelator → ministro.nome (FK em acordao)
- ementa → acordao.ementa
- tipoDeDecisao → acordao.tipo\_decisao
- dataDecisao → acordao.data\_decisao
- decisao → acordao.decisao\_texto
- jurisprudenciaCitada → acordao.jurisprudencia\_citada (texto ou relacional posterior)
- notas → acordao.notas
- informacoesComplementares → acordao.informacoes\_complementares
- termosAuxiliares → acordao.termos\_auxiliares
- teseJuridica → acordao.tese\_juridica
- tema → acordao.tema
- referenciasLegislativas → acordao.referencias\_legislativas
- acordaosSimilares → acordao.acordaos\_similares
- dataPublicacao → acordao.data\_publicacao
- Novos pontos para exploração solicitados:
  - Gênero, profissão, estado civil, nacionalidade das partes → pessoa + FKs (dim\_genero, dim\_profissao, dim\_estado\_civil, dim\_pais) e vínculo pelo processo\_parte com papel REQUERENTE/REQUERIDO
  - Estado de origem da demanda → processo.uf\_origem\_demandra (FK dim\_uf)
  - Estado “na identidade” (UF emissora do documento da pessoa) → pessoa\_documento.uf\_emissor (FK dim\_uf)
  - Estado de residência/domicílio da pessoa (quando disponível) → pessoa.uf\_domicilio (FK dim\_uf) [opcional]

### Modelo Entidade-Relacionamento (MER) — descrição textual

- processo (1) — (N) acordao
- processo (1) — (N) processo\_parte — (N) pessoa (N)
- processo (N) — (1) classe\_processual
- processo (N) — (1) orgao\_julgador
- processo (N) — (1) dim\_uf [uf\_origem\_demandra]
- acordao (N) — (1) ministro
- pessoa (N) — (1) dim\_genero
- pessoa (N) — (1) dim\_estado\_civil
- pessoa (N) — (1) dim\_pais [nacionalidade]
- pessoa (N) — (1) dim\_profissao
- pessoa (1) — (N) pessoa\_documento (N) — (1) dim\_uf [uf\_emissor]

- processo\_parte (N) — (1) dim\_papel\_parte

Cardinalidades e chaves

- processo\_parte PK composta (processo\_id, pessoa\_id, papel\_id, dt\_inicio) para histórico; se não for necessário temporalidade, PK simples (processo\_id, pessoa\_id, papel\_id).
- Mantemos chaves técnicas (bigserial) e chaves naturais únicas (ex.: numero\_processo) com UNIQUE.

Script de criação (PostgreSQL 14+) — DDL Observação: ajuste nomes conforme sua convenção. Inclui índices úteis para exploração e busca. Extensões textuais são opcionais.

```
-- Opcional: extensões de busca
CREATE EXTENSION IF NOT EXISTS unaccent;
CREATE EXTENSION IF NOT EXISTS pg_trgm;

-- Dimensões pequenas
CREATE TABLE dim_uf (
    uf CHAR(2) PRIMARY KEY,
    nome VARCHAR(64) NOT NULL
);

CREATE TABLE dim_pais (
    iso_alpha2 CHAR(2) PRIMARY KEY,
    nome VARCHAR(128) NOT NULL
);

CREATE TABLE dim_genero (
    id SMALLSERIAL PRIMARY KEY,
    codigo VARCHAR(16) UNIQUE NOT NULL, -- ex.: M, F, NB, NI, ND
    descricao VARCHAR(64) NOT NULL
);

CREATE TABLE dim_estado_civil (
    id SMALLSERIAL PRIMARY KEY,
    codigo VARCHAR(16) UNIQUE NOT NULL, -- ex.: SOLTEIRO, CASADO,
    DIVORCIADO, VIUVO, UNIAO_ESTAVEL, ND
    descricao VARCHAR(64) NOT NULL
);

CREATE TABLE dim_profissao (
    id SERIAL PRIMARY KEY,
    cbo VARCHAR(10) UNIQUE,           -- código CBO, quando houver
    nome VARCHAR(128) NOT NULL
);

CREATE TABLE dim_papel_parte (
    id SMALLSERIAL PRIMARY KEY,
    codigo VARCHAR(32) UNIQUE NOT NULL, -- REQUERENTE, REQUERIDO,
    ASSISTENTE, LITISCONSORTE, TERCEIRO, etc.
    descricao VARCHAR(128)
);

-- Catálogos auxiliares do processo/acórdão
CREATE TABLE classe_processual (
    sigla VARCHAR(16) PRIMARY KEY,
```

```

        descricao VARCHAR(128) NOT NULL
    ) ;

CREATE TABLE orgao_julgador (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(256) UNIQUE NOT NULL
) ;

CREATE TABLE ministro (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(256) UNIQUE NOT NULL
) ;

-- Núcleo processual
CREATE TABLE processo (
    id BIGSERIAL PRIMARY KEY,
    numero_processo VARCHAR(50) UNIQUE NOT NULL,
    numero_registro VARCHAR(50),
    sigla_classe VARCHAR(16) NOT NULL REFERENCES classe_processual(sigla),
    orgao_julgador_id INT REFERENCES orgao_julgador(id),
    uf_origem_demanda CHAR(2) REFERENCES dim_uf(uf),
    criado_em TIMESTAMPTZ DEFAULT now(),
    atualizado_em TIMESTAMPTZ DEFAULT now()
) ;

CREATE INDEX idx_processo_classe ON processo(sigla_classe);
CREATE INDEX idx_processo_orgao ON processo(orgao_julgador_id);
CREATE INDEX idx_processo_uf_origem ON processo(uf_origem_demanda);
CREATE INDEX idx_processo_numero ON processo USING btree
(numero_processo);

-- Partes e documentos
CREATE TABLE pessoa (
    id BIGSERIAL PRIMARY KEY,
    nome VARCHAR(256) NOT NULL,
    genero_id SMALLINT REFERENCES dim_genero(id),
    estado_civil_id SMALLINT REFERENCES dim_estado_civil(id),
    nacionalidade_pais CHAR(2) REFERENCES dim_pais(iso_alpha2),
    profissao_id INT REFERENCES dim_profissao(id),
    uf_domicilio CHAR(2) REFERENCES dim_uf(uf), -- opcional: UF de
residência/domicílio
    tipo_pessoa VARCHAR(16) DEFAULT 'FISICA' CHECK (tipo_pessoa IN
('FISICA','JURIDICA')),
    criado_em TIMESTAMPTZ DEFAULT now(),
    atualizado_em TIMESTAMPTZ DEFAULT now()
) ;

CREATE INDEX idx_pessoa_nome_trgm ON pessoa USING gin (nome
gin_trgm_ops);
CREATE INDEX idx_pessoa_genero ON pessoa(genero_id);
CREATE INDEX idx_pessoa_estadocivil ON pessoa(estado_civil_id);
CREATE INDEX idx_pessoa_nacionalidade ON pessoa(nacionalidade_pais);
CREATE INDEX idx_pessoa_profissao ON pessoa(profissao_id);
CREATE INDEX idx_pessoa_ufdomicilio ON pessoa(uf_domicilio);

CREATE TABLE pessoa_documento (
    id BIGSERIAL PRIMARY KEY,
    pessoa_id BIGINT NOT NULL REFERENCES pessoa(id) ON DELETE CASCADE,
    tipo VARCHAR(32) NOT NULL, -- RG, RNE, CNH, PASSAPORTE, etc.
    numero VARCHAR(64) NOT NULL,

```

```

orgao_expedidor VARCHAR(64),
uf_emissor CHAR(2) REFERENCES dim_uf(uf), -- "estado na identidade"
data_emissao DATE,
UNIQUE (pessoa_id, tipo, numero)
);

CREATE INDEX idx_pessoa_documento_ufemissor ON
pessoa_documento(uf_emissor);

CREATE TABLE processo_parte (
    processo_id BIGINT NOT NULL REFERENCES processo(id) ON DELETE CASCADE,
    pessoa_id BIGINT NOT NULL REFERENCES pessoa(id) ON DELETE CASCADE,
    papel_id SMALLINT NOT NULL REFERENCES dim_papel_parte(id),
    dt_inicio DATE DEFAULT CURRENT_DATE,
    dt_fim DATE,
    observacoes TEXT,
    PRIMARY KEY (processo_id, pessoa_id, papel_id, dt_inicio),
    CHECK (dt_fim IS NULL OR dt_fim >= dt_inicio)
);

CREATE INDEX idx_processo_parte_papel ON processo_parte(papel_id);
CREATE INDEX idx_processo_parte_pessoa ON processo_parte(pessoa_id);

-- Acórdãos/decisões
CREATE TABLE acordao (
    id BIGSERIAL PRIMARY KEY,
    id_externo VARCHAR(64), -- mapeia o "id" do JSON, se aplicável
    processo_id BIGINT NOT NULL REFERENCES processo(id) ON DELETE CASCADE,
    ministro_relator_id INT REFERENCES ministro(id),
    tipo_decisao VARCHAR(64), -- monocrática, acórdão, etc.
    data_decisao DATE,
    data_publicacao DATE,
    ementa TEXT,
    decisao_texto TEXT,
    jurisprudencia_citada TEXT,
    notas TEXT,
    informacoes_complementares TEXT,
    termos_auxiliares TEXT,
    tese_juridica TEXT,
    tema VARCHAR(128),
    referencias_legislativas TEXT,
    acordaos_similares TEXT,
    criado_em TIMESTAMPTZ DEFAULT now(),
    atualizado_em TIMESTAMPTZ DEFAULT now()
);

-- Full-text indexes (opcionais)
CREATE INDEX idx_acordao_ementa_trgm ON acordao USING gin (ementa
gin_trgm_ops);
CREATE INDEX idx_acordao_decisao_trgm ON acordao USING gin
(decisao_texto gin_trgm_ops);

-- Gatilho simples para updated_at
CREATE OR REPLACE FUNCTION set_updated_at() RETURNS trigger AS $$
BEGIN
    NEW.atualizado_em = now();
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER tg_processo_updated_at
BEFORE UPDATE ON processo
FOR EACH ROW EXECUTE FUNCTION set_updated_at();

CREATE TRIGGER tg_pessoa_updated_at
BEFORE UPDATE ON pessoa
FOR EACH ROW EXECUTE FUNCTION set_updated_at();

CREATE TRIGGER tg_acordao_updated_at
BEFORE UPDATE ON acordao
FOR EACH ROW EXECUTE FUNCTION set_updated_at();

```

### Carga inicial mínima (dimensões)

- dim\_uf: listar UFs (AC, AL, ..., SP, TO) com nome.
- dim\_pais: carregar países (pelo menos BR, e principais nacionalidades de interesse).
- dim\_genero: por exemplo
  - M: Masculino
  - F: Feminino
  - NB: Não-binário
  - NI: Não informado
  - ND: Não declarado
- dim\_estado\_civil: SOLTEIRO, CASADO, DIVORCIADO, VIUVO, UNIAO\_ESTAVEL, ND
- dim\_papel\_parte: REQUERENTE, REQUERIDO, (expansível para ASSISTENTE, TERCEIRO, LITISCONSORTE, etc.)
- dim\_profissao: se houver CBO na origem, carregar catálogo; caso não, criar sob demanda.

### Exemplos de consultas de exploração (amostras)

#### 1. Taxa de decisões por UF de origem

```

SELECT u.uf, u.nome, COUNT(*) AS qtd_acordaos
FROM acordao a
JOIN processo p ON p.id = a.processo_id
LEFT JOIN dim_uf u ON u.uf = p.uf_origem_demannda
GROUP BY u.uf, u.nome
ORDER BY qtd_acordaos DESC;

```

#### 2. Perfil de gênero por papel (requerente/requerido)

```

SELECT dp.codigo AS papel, dg.codigo AS genero, COUNT(*) AS
qtd_pessoas
FROM processo_parte pp
JOIN dim_papel_parte dp ON dp.id = pp.papel_id
JOIN pessoa pe ON pe.id = pp.pessoa_id
LEFT JOIN dim_genero dg ON dg.id = pe.genero_id
GROUP BY dp.codigo, dg.codigo
ORDER BY dp.codigo, qtd_pessoas DESC;

```

#### 3. Distribuição por profissão e resultado (busca por palavras na decisão)

```

SELECT pr.nome AS profissao, COUNT(DISTINCT a.id) AS qtd_acordaos
FROM acordao a
JOIN processo p ON p.id = a.processo_id
JOIN processo_parte pp ON pp.processo_id = p.id
JOIN dim_papel_parte dpp ON dpp.id = pp.papel_id AND dpp.codigo =
'REQUERENTE'
JOIN pessoa pe ON pe.id = pp.pessoa_id
LEFT JOIN dim_profissao pr ON pr.id = pe.profissao_id
WHERE a.decisao_texto ILIKE '%improcedente%' -- exemplo simples
GROUP BY pr.nome
ORDER BY qtd_acordaos DESC NULLS LAST;

```

#### 4. Nacionalidade vs. UF emissora de documento

```

SELECT pa.nome AS nacionalidade, uf.uf AS uf_emissora, COUNT(*) AS qtd
FROM pessoa pe
LEFT JOIN dim_pais pa ON pa.iso_alpha2 = pe.nacionalidade_pais
JOIN pessoa_documento pd ON pd.pessoa_id = pe.id
LEFT JOIN dim_uf uf ON uf.uf = pd.uf_emissor
GROUP BY pa.nome, uf.uf
ORDER BY qtd DESC;

```

#### 5. Requerentes por estado civil em processos originados em SP

```

SELECT ec.descricao AS estado_civil, COUNT(DISTINCT pe.id) AS
qtd_requerentes
FROM processo p
JOIN dim_uf u ON u.uf = p.uf_origem_demanda AND u.uf = 'SP'
JOIN processo_parte pp ON pp.processo_id = p.id
JOIN dim_papel_parte dpp ON dpp.id = pp.papel_id AND dpp.codigo =
'REQUERENTE'
JOIN pessoa pe ON pe.id = pp.pessoa_id
LEFT JOIN dim_estado_civil ec ON ec.id = pe.estado_civil_id
GROUP BY ec.descricao
ORDER BY qtd_requerentes DESC NULLS LAST;

```

#### Boas práticas e considerações

- Normalização x performance:
  - Atributos categóricos controlados como dimensões (FKs) favorecem consistência e explorabilidade com filtros padronizados.
  - Para buscas full-text em ementa/decisão, trigram/unaccent ajudam no português com acentos e variações.
- Pessoas físicas x jurídicas:
  - O campo tipo\_pessoa permite evoluir o modelo para CNPJ, razão social, CNAE, etc., quando necessário.
- Identidade e privacidade:
  - A tabela pessoa\_documento permite registrar UF emissora sem expor dados sensíveis além do necessário para análises. Avalie mascaramento/criptografia quando houver números de documentos em produção.
- Temporalidade:
  - Se precisar histórico de alteração de papel da parte, mantenha dt\_inicio/dt\_fim em processo\_parte (já previsto). Se não precisar, simplifique a PK.

- Qualidade de dados:
  - Ao ingerir o JSON, normalize valores de gênero, estado civil e profissões mapeando sinônimos e abreviações para os códigos das dimensões.
  - Preencha uf\_origem\_demandas a partir do campo fonte/tribunal de origem ou metadado equivalente.

Como o MER fica (notação textual resumida)

- Processo [id] —< Acordao [id]
- Processo [id] —< Processo\_Parte [processo\_id, pessoa\_id, papel\_id, dt\_inicio] —> Pessoa [id]
- Processo — Classe\_Processual (N:1)
- Processo — Orgao\_Julgador (N:1)
- Processo — dim\_uf (uf\_origem\_demandas) (N:1)
- Acordao — Ministro (N:1)
- Pessoa — dim\_genero (N:1)
- Pessoa — dim\_estado\_civil (N:1)
- Pessoa — dim\_pais (nacionalidade) (N:1)
- Pessoa — dim\_profissao (N:1)
- Pessoa —< Pessoa\_Documento [uf\_emissor → dim\_uf] (N:1)

Próximos passos

- Se desejar, gero:
  - Inserts de carga inicial para UFs, países, gêneros, estados civis e papéis de parte.
  - Stored procedures para upsert das dimensões e ingestão do JSON.
  - Índices adicionais e materialized views para relatórios frequentes (ex.: perfis por UF/papel).