

Documentação

Entidade / Colunas Principais / Observações

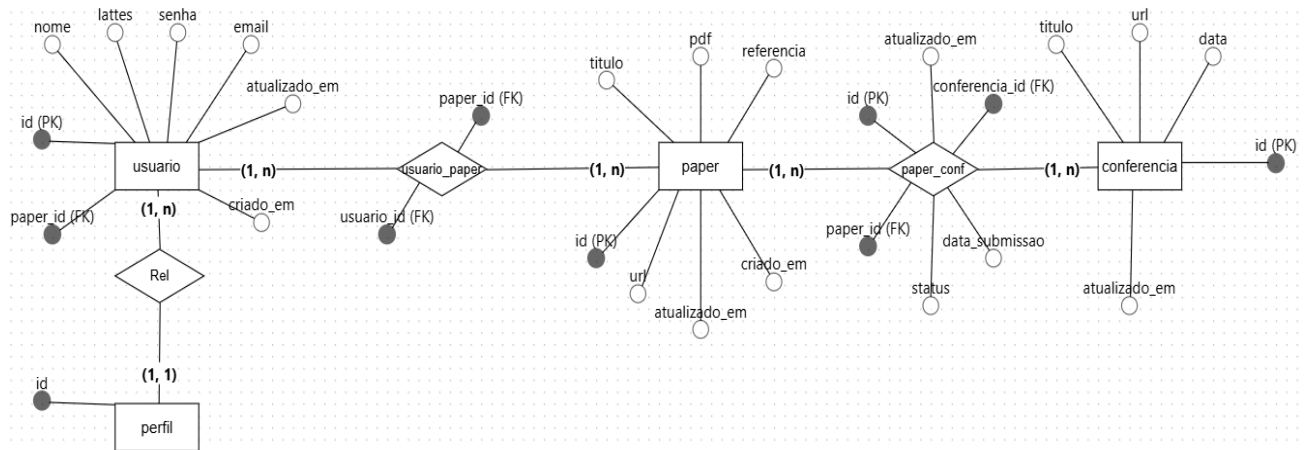
Entidade	Colunas Principais	Observações / Cardinalidade
perfil	id (PK), nome	1 perfil para N usuários
usuario	id (PK), nome, senha, email, lattes, perfil_id (FK), criado_em, atualizado_em	N usuários para 1 perfil Relaciona com muitos papers via usuario_paper
paper	id (PK), titulo, url, pdf, referencia, criado_em, atualizado_em	N papers podem ter N autores e N conferências
usuario_paper	usuario_id (FK), paper_id (FK)	Relação N:N entre usuário e paper (vários autores por paper)
conferencia	id (PK), titulo, url, data, atualizado_em	N conferências podem receber N papers
paper_conf	id (PK), paper_id (FK), conferencia_id (FK), status, data_submissao, atualizado_em	Relação N:N entre paper e conferência, info por submissão

Resumo das Cardinalidades

Relacionamento	Cardinalidade	Tabela Intermediária
perfil - usuario	1 : N	-
usuario - paper	N : N	usuario_paper
paper - conferencia	N : N	paper_conf
paper_conf - status (campo)	1 : 1 (por submissão)	-

Estrutura Visual das Tabelas

Tabela	Colunas
perfil	id (PK), nome
usuario	id (PK), nome, senha, email, lattes, perfil_id (FK), criado_em, atualizado_em
paper	id (PK), titulo, url, pdf, referencia, criado_em, atualizado_em
usuario_paper	usuario_id (FK), paper_id (FK), PRIMARY KEY (usuario_id, paper_id)
conferencia	id (PK), titulo, url, data, atualizado_em
paper_conf	paper_id (FK), conferencia_id (FK), status (aceito, em avaliacao, reprovado), data_submissao, atualizado_em



-- PERFIL

```

CREATE TABLE perfil (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    deletado_em TIMESTAMP
);
  
```

-- USUARIO

```

CREATE TABLE usuario (
    id SERIAL PRIMARY KEY,
  
```

```
nome VARCHAR(100) NOT NULL,  
senha VARCHAR(128) NOT NULL,  
email VARCHAR(100) NOT NULL UNIQUE,  
lattes VARCHAR(255),  
perfil_id INTEGER NOT NULL REFERENCES perfil(id),  
deletado_em TIMESTAMP  
);
```

-- PAPER

```
CREATE TABLE paper (  
    id SERIAL PRIMARY KEY,  
    titulo VARCHAR(255) NOT NULL,  
    url VARCHAR(255),  
    pdf BYTEA, -- Guarda o arquivo binário do PDF  
    referencia TEXT,  
    deletado_em TIMESTAMP  
);
```

-- USUARIO_PAPER (N:N entre usuário e paper)

```
CREATE TABLE usuario_paper (  
    usuario_id INTEGER NOT NULL REFERENCES usuario(id),  
    paper_id INTEGER NOT NULL REFERENCES paper(id),  
    deletado_em TIMESTAMP,  
    PRIMARY KEY (usuario_id, paper_id)  
);
```

-- CONFERENCIA

```
CREATE TABLE conferencia (  
    id SERIAL PRIMARY KEY,  
    titulo VARCHAR(255) NOT NULL,  
    url VARCHAR(255),  
    data DATE,  
    deletado_em TIMESTAMP  
);
```

-- PAPER_CONF (N:N entre paper e conferencia, com status e info por submissão)

```
CREATE TABLE paper_conf (  
    id SERIAL PRIMARY KEY,  
    paper_id INTEGER NOT NULL REFERENCES paper(id),  
    conferencia_id INTEGER NOT NULL REFERENCES conferencia(id),  
    status VARCHAR(30) CHECK (status IN ('em avaliacao', 'aceito', 'reprovado')) NOT NULL,  
    data_submissao DATE,  
    deletado_em TIMESTAMP  
);
```

-- VIEWS para mostrar apenas registros não deletados (soft delete)

```
CREATE OR REPLACE VIEW v_perfil AS  
SELECT * FROM perfil WHERE deletado_em IS NULL;
```

```
CREATE OR REPLACE VIEW v_usuario AS  
SELECT * FROM usuario WHERE deletado_em IS NULL;
```

```
CREATE OR REPLACE VIEW v_paper AS  
SELECT * FROM paper WHERE deletado_em IS NULL;
```

```
CREATE OR REPLACE VIEW v_usuario_paper AS  
SELECT * FROM usuario_paper WHERE deletado_em IS NULL;
```

```
CREATE OR REPLACE VIEW v_conferencia AS  
SELECT * FROM conferencia WHERE deletado_em IS NULL;
```

```
CREATE OR REPLACE VIEW v_paper_conf AS  
SELECT * FROM paper_conf WHERE deletado_em IS NULL;
```