

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA

Fundamentos de Banco de Dados – INF01145

Trabalho Prático – Etapa 2

Comandos de Definição de Dados (DDL)

Bruno Samuel Ardenghi Gonçalves — 550452

Leonardo Azzi Martins — 323721

Prof.^a Karin Becker

2025/1

1 Entidades

1.1 Tabela series

Uma série de livros contendo uma ou mais obras primárias explicitamente ordenadas e, opcionalmente, outras obras sem uma posição específica dentro da série.

```
CREATE TABLE series (  
    id SERIAL PRIMARY KEY,  
    title VARCHAR(100) NOT NULL  
);
```

1.2 Tabela works

Uma obra literária que engloba todas as suas edições.

```
CREATE TABLE works (  
    id SERIAL PRIMARY KEY,  
    title VARCHAR(100) NOT NULL,  
    first_publication_date DATE NOT NULL,  
    rating NUMERIC(2) CHECK (  
        rating BETWEEN 1 AND 5  
    )  
);
```

1.3 Tabela genres

Um gênero ou categoria de livros. É definido pelo seu nome e usado para agrupar obras com características similares.

```
CREATE TABLE genres (  
    id SERIAL PRIMARY KEY,  
    slug VARCHAR(50) NOT NULL UNIQUE,  
    label VARCHAR(50) NOT NULL,  
    description VARCHAR(2000)  
);
```

1.4 Tabela authors

Um autor com zero ou mais obras atribuídas a si. Usado para agrupar livros escritos pela mesma pessoa e fornecer informações sobre estas através de um perfil.

```
CREATE TABLE authors (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(150) NOT NULL,  
    picture VARCHAR(500),  
    birth_place VARCHAR(50),  
    birth_date DATE,  
    death_date DATE,  
    website VARCHAR(500),  
    biography VARCHAR(2000)  
);
```

1.5 Tabela editions

Uma edição ou instância de uma obra. Também implementa o relacionamento *Publication*, que atribui uma edição à obra a qual ela corresponde.

```
CREATE TABLE editions (  
    id SERIAL PRIMARY KEY,  
    title VARCHAR(150) NOT NULL,  
    page_count SMALLINT NOT NULL,  
    format VARCHAR(25) NOT NULL,  
    publication_date DATE NOT NULL,  
    publisher VARCHAR(50) NOT NULL,  
    language VARCHAR(50) NOT NULL,  
    cover VARCHAR(500),  
    summary VARCHAR(2000),  
    isbn CHAR(13),  
    asin CHAR(10),  
    -- Relacionamento Publication  
    work_id INTEGER NOT NULL REFERENCES works(id)  
);
```

1.6 Tabela users

Um usuário representa um perfil na plataforma, necessário para interagir com o site.

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    email VARCHAR(254) NOT NULL UNIQUE,  
    password VARCHAR(35) NOT NULL,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50),  
    picture VARCHAR(500)  
);
```

1.7 Tabela groups

Um grupo criado e composto por usuários, que podem ser moderadores ou membros.

```
CREATE TABLE groups (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    description VARCHAR(1000),  
    picture VARCHAR(500)  
);
```

1.8 Tabela lists

Uma lista de livros pública. Qualquer usuário pode adicionar livros à lista, assim como votar em obras já listadas.

```
CREATE TABLE lists (  
    id SERIAL PRIMARY KEY,  
    title VARCHAR(100) NOT NULL,  
    description VARCHAR(1000)  
);
```

1.9 Tabela `list_entries`

Representa uma obra específica dentro de uma lista, implementando também o relacionamento *Entry*, que a atribui à sua respectiva lista.

```
CREATE TABLE list_entries (  
  id SERIAL PRIMARY KEY,  
  vote_count INTEGER NOT NULL CHECK (vote_count >= 0),  
  -- Relacionamento Entry  
  list_id INTEGER NOT NULL REFERENCES lists(id),  
  edition_id INTEGER NOT NULL REFERENCES editions(id)  
);
```

1.10 Tabela `shelves`

Uma estante ou coleção de livros pública de um usuário.

```
CREATE TABLE shelves (  
  id SERIAL PRIMARY KEY,  
  slug VARCHAR(50) NOT NULL UNIQUE  
);
```

1.11 Tabela `quotes`

Uma citação atribuída a um autor por um usuário. Implementa o relacionamento *Attribution*, que formaliza a autoria da citação.

```
CREATE TABLE quotes (  
  id SERIAL PRIMARY KEY,  
  quote VARCHAR(500) NOT NULL,  
  -- Relacionamento Attribution  
  author_id INTEGER NOT NULL REFERENCES authors(id)  
);
```

2 Relacionamentos

2.1 Tabela positionings

Tabela de relacionamento que atribui uma obra a uma série, com ou sem uma posição específica.

```
CREATE TABLE positionings (  
    series_id INTEGER NOT NULL REFERENCES series(id),  
    work_id INTEGER NOT NULL REFERENCES works(id),  
    position NUMERIC(2),  
    PRIMARY KEY (series_id, work_id)  
);
```

2.2 Tabela categorizations

Tabela de relacionamento que atribui uma obra a um gênero.

```
CREATE TABLE categorizations (  
    work_id INTEGER NOT NULL REFERENCES works(id),  
    genre_id INTEGER NOT NULL REFERENCES genres(id),  
    PRIMARY KEY (work_id, genre_id)  
);
```

2.3 Tabela authorships

Tabela de relacionamento que atribui uma obra aos seus autores.

```
CREATE TABLE authorships (  
    work_id INTEGER NOT NULL REFERENCES works(id),  
    author_id INTEGER NOT NULL REFERENCES authors(id),  
    PRIMARY KEY (work_id, author_id)  
);
```

2.4 Tabela storages

Tabela de relacionamento que atribui uma edição de um livro a uma estante de um usuário.

```
CREATE TABLE storages (  
    edition_id INTEGER NOT NULL REFERENCES editions(id),  
    shelf_id INTEGER NOT NULL REFERENCES shelves(id),  
    PRIMARY KEY (edition_id, shelf_id)  
);
```

2.5 Tabela ownerships

Tabela de relacionamento que atribui uma estante ao seu usuário proprietário.

```
CREATE TABLE ownerships (  
    shelf_id INTEGER NOT NULL REFERENCES shelves(id),  
    user_id INTEGER NOT NULL REFERENCES users(id),  
    PRIMARY KEY (shelf_id, user_id)  
);
```

2.6 Tipo tracking_status

Define os possíveis status de rastreamento de leitura de um livro por um usuário.

```
CREATE TYPE tracking_status AS ENUM ('to-read', 'currently-  
reading', 'read');
```

2.7 Tabela trackings

Guarda o status e progresso de leitura de um usuário em relação a um livro.

```
CREATE TABLE trackings (  
    edition_id INTEGER NOT NULL REFERENCES editions(id),  
    user_id INTEGER NOT NULL REFERENCES users(id),  
    status tracking_status NOT NULL,  
    progress SMALLINT DEFAULT 0,  
    review VARCHAR(18800),  
    reading_period DATERANGE,  
    rating SMALLINT CHECK (  
        rating BETWEEN 1 AND 5  
    ),  
    PRIMARY KEY (edition_id, user_id)  
);
```

2.8 Tabela likes

Relacionamento criado quando um usuário curte uma citação.

```
CREATE TABLE likes (  
    quote_id INTEGER NOT NULL REFERENCES quotes(id),  
    user_id INTEGER NOT NULL REFERENCES users(id),  
    PRIMARY KEY (quote_id, user_id)  
);
```

2.9 Tabela memberships

Relacionamento que atribui um usuário a um grupo do qual ele faz parte.

```
CREATE TABLE memberships (  
    user_id INTEGER NOT NULL REFERENCES users(id),  
    group_id INTEGER NOT NULL REFERENCES groups(id),  
    PRIMARY KEY (user_id, group_id)  
);
```


2.10 Tabela moderations

Relacionamento que atribui a posição de moderador de um grupo a um usuário.

```
CREATE TABLE moderations (  
    user_id INTEGER NOT NULL REFERENCES users(id),  
    group_id INTEGER NOT NULL REFERENCES groups(id),  
    PRIMARY KEY (user_id, group_id)  
);
```

2.11 Tabela currently_readings

Relacionamento que atribui um livro como leitura atual de um grupo.

```
CREATE TABLE currently_readings (  
    edition_id INTEGER NOT NULL REFERENCES editions(id),  
    group_id INTEGER NOT NULL REFERENCES groups(id),  
    start_date DATE,  
    finish_date DATE,  
    PRIMARY KEY (edition_id, group_id)  
);
```

2.12 Tabela votes

Relacionamento criado quando um usuário vota em um livro em uma lista.

```
CREATE TABLE votes (  
    user_id INTEGER NOT NULL REFERENCES users(id),  
    list_entry_id INTEGER NOT NULL REFERENCES list_entries(  
        id),  
    PRIMARY KEY (user_id, list_entry_id)  
);
```