



Projeto Final - Módulo Banco de Dados (Postgres) Sistema de Gerenciamento de Biblioteca

Discente: Cristiano Neves

Docentes: Ana Beatriz Ferraz, Bruna Leal, Kamila Antunes e Renata Wro.

Sistema de Gerenciamento de Biblioteca

1. Objetivo

Este projeto tem como objetivo criar o design de um banco de dados para uma biblioteca fictícia. A finalidade principal do desenvolvimento deste projeto é a avaliação dos alunos no módulo de banco de dados - PostgreSQL, no âmbito do curso de DevOps oferecido pela Ada Tech em parceria com a Nuclea, sob a instrução do professor Cristiano Neves.

O escopo do projeto envolve o desenvolvimento da estrutura do banco de dados para um sistema de gerenciamento de biblioteca. Esse sistema permitirá que os usuários realizem pesquisas e reservem livros, que os funcionários efetuem empréstimos e devoluções, e que os administradores mantêm o controle do acervo da biblioteca.

2. Funcionalidades principais

As principais funcionalidades do sistema de gerenciamento de biblioteca, são:

- **Cadastro de usuários:** os usuários devem se cadastrar no sistema fornecendo informações como nome, endereço, número de telefone e endereço de e-mail.
- **Gerenciamento do acervo:** os administradores do sistema podem adicionar, atualizar e excluir informações sobre livros (incluindo autor, ISBN, editora e número de cópias disponíveis), deve ser possível identificar individualmente cada exemplar de um mesmo livro.
- **Empréstimo e devolução:** os funcionários da biblioteca podem registrar empréstimos, associando um exemplar de livro a um usuário e uma data de empréstimo e prazo de devolução. Os funcionários também podem registrar devoluções, atualizando o status do livro emprestado e registrando a data de devolução.
- **Pesquisa de livros:** os usuários podem pesquisar livros por títulos, autor, gênero, ISBN, palavras chaves, etc. Os resultados da pesquisa devem mostrar informações sobre o livro e sua disponibilidade.
- **Reserva de livros:** os usuários podem reservar livros que estão atualmente emprestados por outros usuários. Quando um livro reservado se tornar disponível, será possível o sistema notificar o usuário que o reservou. Quando for efetuar o empréstimo de um livro deve ser possível verificar se o mesmo se encontra reservado para outros usuários que devem ter prioridade no empréstimo do mesmo.



Projeto Final - Módulo Banco de Dados (Postgres)

Sistema de Gerenciamento de Biblioteca

3. Cenários e requisitos

a. Cadastro de usuários:

- i. Os usuários devem poder se cadastrar no sistema, fornecendo informações como nome, endereço, número de telefone e endereço de e-mail.
- ii. O sistema deve validar a unicidade do endereço de e-mail para evitar últimos cadastrados com o mesmo e-mail.
- iii. Deve ser possível que os usuários atualizem suas informações pessoais, como endereço ou número de telefone.

b. Gerenciamento de acervo:

- i. Os administradores do sistema devem ter a capacidade de adicionar, atualizar e excluir informações sobre livros.
- ii. Cada exemplar de um livro deve ser identificado individualmente para controle de disponibilidade.
- iii. O sistema deve manter informações detalhadas sobre cada livro, incluindo autor, ISBN, editora e número de cópias disponíveis.

c. Empréstimo e devolução:

- i. Os funcionários da biblioteca devem ser capazes de registrar empréstimos, associando um exemplar de livro a um usuário e especificando a data de empréstimo e o prazo de devolução.
- ii. O sistema deve calcular automaticamente a data de devolução com base no prazo estabelecido.
- iii. Os funcionários também devem ser capazes de registrar devoluções, atualizando o status do livro emprestado e registrando a data de devolução.
- iv. Deve haver um mecanismo para lidar com atrasos nas devoluções, incluindo cálculo de multas, se aplicável.

d. Pesquisa de livros:

- i. Os usuários devem poder pesquisar livros por diferentes critérios, como título, autor, gênero, ISBN e palavras-chaves.
- ii. Os resultados da pesquisa devem mostrar informações detalhadas sobre o livro, incluindo disponibilidade (se está emprestado ou não) e a localização na biblioteca.

e. Reserva de livros:

- i. Os usuários devem ser capazes de reservar livros que estão atualmente emprestados por outros usuários.
- ii. O sistema deve manter uma lista de reservas para cada livro, registrando os usuários que reservaram e a ordem de reservas.

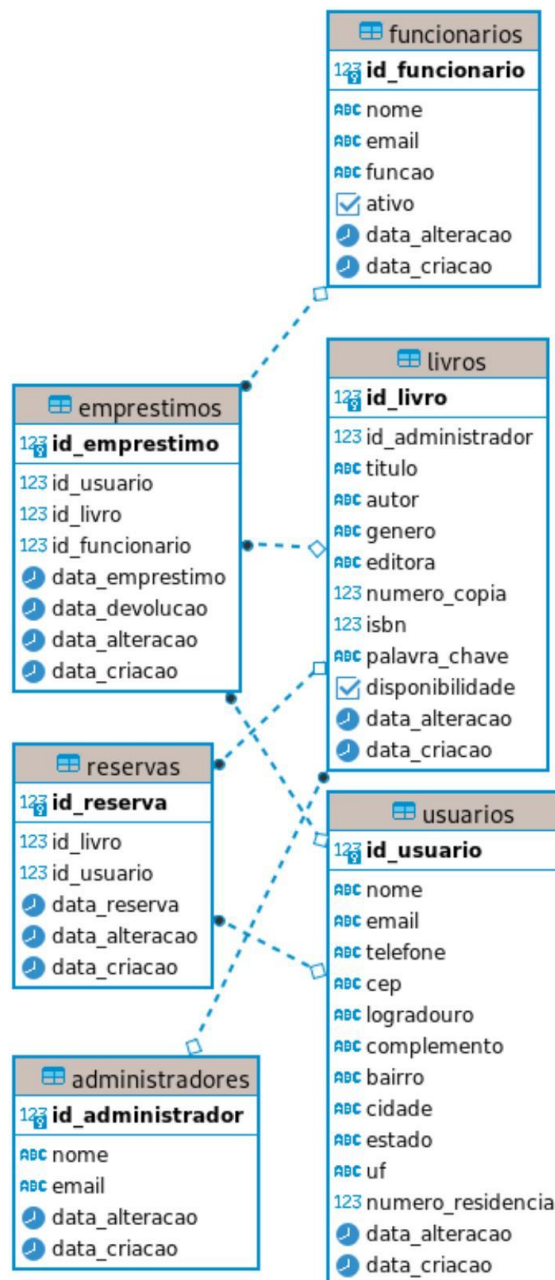


Projeto Final - Módulo Banco de Dados (Postgres)

Sistema de Gerenciamento de Biblioteca

- iii. Quando um livro reservado se tornar disponível, o sistema deve notificar o usuário que o reservou.
- iv. Durante o processo de empréstimo, o sistema deve verificar se o livro está reservado para outros usuários e priorizar o empréstimo para o usuário que fez a reserva.

4. Diagrama DER



5. Scripts



Projeto Final - Módulo Banco de Dados (Postgres) Sistema de Gerenciamento de Biblioteca

```
CREATE DATABASE "biblioteca";
```

```
CREATE TABLE IF NOT EXISTS usuarios(  
  id_usuario SERIAL PRIMARY KEY,  
  nome CHARACTER VARYING(100) NOT NULL,  
  email CHARACTER VARYING(50) NOT NULL UNIQUE,  
  telefone CHARACTER VARYING(15) NOT NULL,  
  cep CHARACTER VARYING(9) NOT NULL,  
  logradouro CHARACTER VARYING(100) NOT NULL,  
  complemento CHARACTER VARYING(100),  
  bairro CHARACTER VARYING(45) NOT NULL,  
  cidade CHARACTER VARYING(45) NOT NULL,  
  estado CHARACTER VARYING(30) NOT NULL,  
  uf CHARACTER(2) NOT NULL,  
  numero_residencia INT NOT null,  
  data_alteracao TIMESTAMP,  
  data_criacao TIMESTAMP DEFAULT now()  
);
```

```
CREATE TABLE IF NOT EXISTS administradores(  
  id_administrador SERIAL PRIMARY KEY,  
  nome CHARACTER VARYING(100) NOT NULL,  
  email CHARACTER VARYING(50) NOT NULL UNIQUE,  
  data_alteracao TIMESTAMP,  
  data_criacao TIMESTAMP DEFAULT now()  
);
```

```
CREATE TABLE IF NOT EXISTS funcionarios(  
  id_funcionario SERIAL PRIMARY KEY,  
  nome CHARACTER VARYING(100) NOT NULL,  
  email CHARACTER VARYING(50) NOT NULL UNIQUE,  
  funcao CHARACTER VARYING(100) NOT NULL,  
  ativo BOOLEAN DEFAULT true,  
  data_alteracao TIMESTAMP,  
  data_criacao TIMESTAMP DEFAULT now()  
);
```



Projeto Final - Módulo Banco de Dados (Postgres)

Sistema de Gerenciamento de Biblioteca

```
CREATE TABLE IF NOT EXISTS livros(  
  id_livro SERIAL PRIMARY KEY,  
  id_administrador INT,  
  titulo CHARACTER VARYING(50) NOT NULL,  
  autor CHARACTER VARYING(100) NOT NULL,  
  genero CHARACTER VARYING(50) NOT NULL,  
  editora CHARACTER VARYING(50) NOT NULL,  
  numero_copia INTEGER DEFAULT 1,  
  isbn INT NOT NULL UNIQUE,  
  palavra_chave CHARACTER VARYING(20) NOT NULL,  
  disponibilidade BOOLEAN NOT null,  
  data_alteracao TIMESTAMP,  
  data_criacao TIMESTAMP DEFAULT now(),  
  
  FOREIGN KEY (id_administrador) REFERENCES administradores(id_administrador)  
);
```

```
CREATE TABLE IF NOT EXISTS emprestimos(  
  id_emprestimo SERIAL PRIMARY KEY,  
  id_usuario INT,  
  id_livro INT,  
  id_funcionario INT,  
  data_emprestimo TIMESTAMP NOT NULL,  
  data_devolucao TIMESTAMP NOT NULL,  
  data_alteracao TIMESTAMP,  
  data_criacao TIMESTAMP DEFAULT now(),  
  
  FOREIGN KEY (id_livro) REFERENCES livros(id_livro),  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),  
  FOREIGN KEY (id_funcionario) REFERENCES funcionarios(id_funcionario)  
);
```

```
CREATE TABLE if NOT EXISTS reservas(  
  id_reserva SERIAL PRIMARY KEY,  
  id_livro INT,  
  id_usuario INT,  
  data_reserva TIMESTAMP,  
  data_alteracao TIMESTAMP,  
  data_criacao TIMESTAMP DEFAULT now(),  
  
  FOREIGN KEY (id_livro) REFERENCES livros(id_livro),  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)  
);
```



Projeto Final - Módulo Banco de Dados (Postgres)

Sistema de Gerenciamento de Biblioteca

```
**INSERTS NAS TABELAS****

-- Cria usuário
INSERT INTO public.usuarios(nome, email, telefone, cep, logradouro, complemento, bairro, cidade, estado, uf, numero_residencia)
VALUES ('Wagner Nascimento', 'wagner.nascimento@gmail.com', '9953124250', '00000-000', 'Rua da Alegria', 'apto 12', 'Campestre', 'Santo André', 'São Paulo', 'UF', 111);

commit;

-- Cria Administrador
INSERT INTO public.administradores(nome, email)
VALUES ('Filipe Santos', 'filipsants@gmail.com');

commit;

-- Cria funcionário
INSERT INTO public.funcionarios(nome, email, funcao, ativo)
VALUES ('Bruna Leal', 'bruna.leal@gmail.com', 'Bibliotecaria', TRUE);

commit;
```

```
**CRIAR LIVROS****

INSERT INTO public.livros (id_administrador, titulo, autor, genero, editora, numero_copia, isbn, palavra_chave, disponibilidade)
VALUES (1, 'Dom Casmurro', 'Machado de Assis', 'Romance', 'Editora A', 5, 1234567890, 'Brasil', true);

commit;

INSERT INTO public.livros (id_administrador, titulo, autor, genero, editora, numero_copia, isbn, palavra_chave, disponibilidade)
VALUES (1, 'O Senhor dos Anéis', 'J.R.R. Tolkien', 'Fantasia', 'Editora B', 3, 9876540, 'Fantasia', false);

commit;
```

```
**CRIAR UMA RESERVA****

INSERT INTO public.reservas (id_livro, id_usuario, data_reserva)
VALUES (2, 1, now());

commit;
```

```
**CRIAR UM EMPRÉSTIMO****

INSERT INTO emprestimos (id_usuario, id_livro, id_funcionario, data_emprestimo, data_devolucao)
VALUES (1, 1, 1, '2023-10-07 14:30:00', '2023-10-14 14:30:00');

commit;
```

```
**CONSULTAS DAS TABELAS****

/* Dados do usuário */

select *
from public.usuarios

/* Dados do Administrador que cadastrou o livro */

select l.id_livro,
       l.titulo,
       l.autor,
       l.genero,
       a.id_administrador,
       a.nome
from livros l
join administradores a on (l.id_administrador = a.id_administrador)

/* Dados de Empréstimos feitos pela Funcionária Bruna Leal */

select f.id_funcionario,
       f.nome,
       e.id_emprestimo,
       e.data_emprestimo,
       e.data_devolucao,
       l.id_livro,
       l.titulo,
       l.autor,
       l.genero
from funcionarios f
join emprestimos e on (f.id_funcionario = e.id_funcionario)
join livros l on (e.id_livro = l.id_livro)

/* Busca de livros por condicionais */

select *
from public.livros
where titulo like '%Dom%'
and autor = 'Machado de Assis'
and genero = 'Romance'
and isbn = 1234567890
and numero_copia = 5
```