

TODOS OS SCRIPTS SQL USADOS PARA CRIAR O BANCO DE DADOS DO SISTEMA

```
-- Banco de Dados: bd_biblioteca_online
CREATE DATABASE bd_biblioteca_online
WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'pt-BR'
    LC_CTYPE = 'pt-BR'
    LOCALE_PROVIDER = 'libc'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;
COMMENT ON DATABASE bd_biblioteca_online
    IS 'Banco de dados do projeto Sistema de Biblioteca Online';
```

```
-- Tabela de Livros.
CREATE TABLE Livros (
    id SERIAL PRIMARY KEY,
    titulo VARCHAR(255) NOT NULL,
    autor VARCHAR(255) NOT NULL,
    isbn VARCHAR(20) UNIQUE,
    ano_publicacao INTEGER,
    editora VARCHAR(100),
    quantidade_total INTEGER NOT NULL,
    quantidade_disponivel INTEGER NOT NULL,
    descricao TEXT,
    numero_paginas INTEGER,
    genero VARCHAR(100));
```

```
-- Tabela de Usuários.
CREATE TABLE Usuarios (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    tipo VARCHAR(50) NOT NULL CHECK (tipo IN ('aluno', 'professor', 'pesquisador',
'administrador')),
    email VARCHAR(255) UNIQUE,
    telefone VARCHAR(20) NOT NULL,
    password VARCHAR(255) NOT NULL,
    last_login TIMESTAMP NULL,
    punido_ate DATE
```

);

-- Tabela de Exemplares (Cópias de Livros Físicos).

```
CREATE TABLE Exemplares (  
  id SERIAL PRIMARY KEY,  
  id_livro INTEGER REFERENCES Livros(id),  
  codigo_barras VARCHAR(100) UNIQUE,  
  status VARCHAR(50) CHECK (status IN ('disponivel', 'emprestado', 'reservado',  
  'indisponivel', 'perdido')) DEFAULT 'disponivel'  
);
```

-- Tabela de Recursos Digitais.

```
CREATE TABLE RecursosDigitais (  
  id SERIAL PRIMARY KEY,  
  titulo VARCHAR(255) NOT NULL,  
  tipo VARCHAR(50) NOT NULL CHECK (tipo IN ('e-book', 'periodico', 'tese',  
  'dissertacao', 'base_de_dados')),  
  url TEXT NOT NULL,  
  data_disponibilidade DATE,  
  descricao TEXT,  
  numero_paginas INTEGER,  
  genero VARCHAR(100)  
);
```

-- Tabela de Reservas.

```
CREATE TABLE Reservas (  
  id SERIAL PRIMARY KEY,  
  id_exemplar INTEGER REFERENCES Exemplares(id),  
  id_usuario INTEGER REFERENCES Usuarios(id),  
  data_reserva TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
  status VARCHAR(20)  
  CHECK (status IN ('pendente', 'aguardando_confirmacao', 'cancelada', 'finalizada'))  
  NULL;  
);
```

-- Tabela de Empréstimos.

```
CREATE TABLE Emprestimos (  
  id SERIAL PRIMARY KEY,  
  id_exemplar INTEGER REFERENCES Exemplares(id),  
  id_usuario INTEGER REFERENCES Usuarios(id),  
  data_emprestimo DATE NOT NULL DEFAULT CURRENT_DATE,  
  data_devolucao_prevista DATE NOT NULL,
```

```
data_devolucao_real DATE,  
status VARCHAR(20)  
CHECK (status IN ('pendente', 'ativo', 'concluido', 'cancelado'))  
NULL;  
);
```

```
-- Tabela de Listas de Leitura.  
CREATE TABLE ListasLeitura (  
id SERIAL PRIMARY KEY,  
nome VARCHAR(255) NOT NULL,  
descricao TEXT,  
id_usuario INTEGER REFERENCES Usuarios(id)  
);
```

```
-- Tabela de Livros em Listas de Leitura (relação muitos-para-muitos).  
CREATE TABLE LivrosListasLeitura (  
id_lista INTEGER REFERENCES ListasLeitura(id),  
id_livro INTEGER REFERENCES Livros(id),  
PRIMARY KEY (id_lista, id_livro)  
);
```

```
-- Tabela de Recursos Digitais em Listas de Leitura (relação muitos-para-muitos).  
CREATE TABLE RecursosDigitaisListasLeitura (  
id_lista INTEGER REFERENCES ListasLeitura(id),  
id_recurso_digital INTEGER REFERENCES RecursosDigitais(id),  
PRIMARY KEY (id_lista, id_recurso_digital)  
);
```

```
-- Tabela de Histórico de Empréstimos.  
CREATE TABLE HistoricoEmprestimos (  
id SERIAL PRIMARY KEY,  
id_exemplar INTEGER REFERENCES Exemplares(id),  
id_usuario INTEGER REFERENCES Usuarios(id),  
data_emprestimo DATE NOT NULL,  
data_devolucao_prevista DATE NOT NULL,  
data_devolucao_real DATE NOT NULL,  
data_registro_historico TIMESTAMP WITH TIME ZONE DEFAULT  
CURRENT_TIMESTAMP  
);
```

```
-- Tabela de Favoritos.  
CREATE TABLE Favoritos (  
id SERIAL PRIMARY KEY,  
id_usuario INTEGER REFERENCES Usuarios(id),
```

```

id_livro INTEGER REFERENCES Livros(id),
id_recurso_digital INTEGER REFERENCES RecursosDigitais(id),
data_favoritado TIMESTAMP WITH TIME ZONE DEFAULT
CURRENT_TIMESTAMP,
UNIQUE (id_usuario, id_livro, id_recurso_digital),
CONSTRAINT fk_favoritos_livro FOREIGN KEY (id_livro) REFERENCES Livros(id)
ON DELETE CASCADE,
CONSTRAINT fk_favoritos_recurso_digital FOREIGN KEY (id_recurso_digital)
REFERENCES RecursosDigitais(id) ON DELETE CASCADE,
CONSTRAINT fk_favoritos_usuario FOREIGN KEY (id_usuario) REFERENCES
Usuarios(id) ON DELETE CASCADE,
15
CONSTRAINT check_favorito_item CHECK (id_livro IS NOT NULL OR
id_recurso_digital IS NOT NULL),
CONSTRAINT check_favorito_exclusivo CHECK (NOT (id_livro IS NOT NULL AND
id_recurso_digital IS NOT NULL))
);

```

-- Tabela de Notificações

```

CREATE TABLE notificacoes (
    id SERIAL PRIMARY KEY,
    usuario_id INTEGER NOT NULL REFERENCES usuarios(id) ON DELETE
CASCADE,
mensagem TEXT NOT NULL,
lida BOOLEAN DEFAULT FALSE,
data_criacao TIMESTAMP WITH TIME ZONE DEFAULT
CURRENT_TIMESTAMP
);

```