Scripts SQL

Postgre SQL

Tabelas

```sql
--Schema do Projeto
CREATE schema Biblioteca;

--Tabela Editora
CREATE TABLE Editora (
  id_editora INT PRIMARY KEY,
  nome_editora VARCHAR(30) NOT NULL,
  logradouro VARCHAR(20),
  cep INT NOT NULL,
  numero INT,
  bairro VARCHAR(40),
  cidade VARCHAR(40),
  estado VARCHAR(15)
);

--Localização
CREATE TABLE Localizacao (
  id_localizacao INT PRIMARY KEY,
  andar INT NOT NULL,
  setor INT NOT NULL,
  corredor INT NOT NULL,
  estante INT NOT NULL,
  prateleira INT NOT NULL
);

--Tabela Leitor
CREATE TABLE Leitor (
  id_leitor INT PRIMARY KEY,
  nome_leitor VARCHAR(30) NOT NULL,
  telefone VARCHAR(15) NOT NULL,
  data_nascimento DATE NOT NULL,
  cpf INT NOT NULL
);

--Tabela Email_Leitor
CREATE TABLE Email_Leitor (
  id_email_leitor INT PRIMARY KEY,
  email_leitor VARCHAR(60) NOT NULL,
  id_leitor INT NOT NULL,
  FOREIGN KEY (id_leitor) REFERENCES Leitor(id_leitor)
);
```

```sql
CREATE TYPE mood AS ENUM ('Ativa', 'Suspensa', 'Expirada');

--Tabela Carteira da Bibliotecca
CREATE TABLE Carteira_da_Biblioteca (
  id_carteira INT PRIMARY KEY,
  validade DATE,
  status VARCHAR(10), --'Ativa', 'Suspensa', 'Expirada'
  id_leitor INT NOT NULL,
  FOREIGN KEY (id_leitor) REFERENCES Leitor(id_leitor)
);

--Tabela Livro
CREATE TABLE Livro (
  id_livro INT PRIMARY KEY,
  titulo_livro VARCHAR(40) NOT NULL,
  nome_autor VARCHAR(35) NOT NULL,
  ano_publicacao INT NOT NULL,
  id_editora INT NOT NULL,
  FOREIGN KEY (id_editora) REFERENCES Editora(id_editora)
);

--Tabela Livro_Leitor
CREATE TABLE Livro_Leitor (
  id_livro_leitor INT PRIMARY KEY,
  id_livro INT NOT NULL,
  id_leitor INT NOT NULL,
  FOREIGN KEY (id_livro) REFERENCES Livro(id_livro),
  FOREIGN KEY (id_leitor) REFERENCES Leitor(id_leitor)
);

--Tabela Gênero do Livro
CREATE TABLE Genero_Livro (
  id_genero_livro INT PRIMARY KEY,
  genero_livro VARCHAR(50) NOT NULL,
  id_livro INT NOT NULL,
  FOREIGN KEY (id_livro) REFERENCES Livro(id_livro)
  );

CREATE TYPE state AS ENUM ('Novo', 'Bom', 'Regular', 'Danificado');

--Tabela Exemplar
CREATE TABLE Exemplar (
  id_exemplar INT PRIMARY KEY,
  estado_de_conservacao VARCHAR(12) NOT NULL, --'Novo', 'Bom', 'Regular', 'Danificado'
  codigo_de_barras VARCHAR(15),
  id_livro INT NOT NULL,
  id_localizacao INT NOT NULL,
  FOREIGN KEY (id_livro) REFERENCES Livro(id_livro),
  FOREIGN KEY (id_localizacao) REFERENCES Localizacao(id_localizacao)
);

--Tabela Empréstimo
```

```sql
CREATE TABLE Emprestimo (
  id_emprestimo INT PRIMARY KEY,
  data_emprestimo DATE NOT NULL,
  data_devolucao DATE,
  id_exemplar INT NOT NULL,
  id_carteira INT NOT NULL,
  FOREIGN KEY (id_exemplar) REFERENCES Exemplar(id_exemplar),
  FOREIGN KEY (id_carteira) REFERENCES Carteira_da_Biblioteca(id_carteira)
);
```

Inserindo Dados

```sql
--Tabela Editora

INSERT INTO Editora (
    id_editora,
    nome_editora,
    logradouro,
    numero,
    cep,
    bairro,
    cidade,
    estado
)
SELECT
    g AS id_editora,
    'Editora ' || g AS nome_editora,
    'Rua ' || g AS logradouro,
    (g % 500) AS numero,  -- valores de 1 a 500
    (10000000 + g) AS cep,
    'Bairro ' || g AS bairro,
    'Cidade ' || (g % 500) AS cidade,
    CASE (g % 27)
        WHEN 0 THEN 'AC'
        WHEN 1 THEN 'AL'
        WHEN 2 THEN 'AP'
        WHEN 3 THEN 'AM'
        WHEN 4 THEN 'BA'
        WHEN 5 THEN 'CE'
        WHEN 6 THEN 'DF'
        WHEN 7 THEN 'ES'
        WHEN 8 THEN 'GO'
        WHEN 9 THEN 'MA'
        WHEN 10 THEN 'MT'
        WHEN 11 THEN 'MS'
        WHEN 12 THEN 'MG'
        WHEN 13 THEN 'PA'
        WHEN 14 THEN 'PB'
        WHEN 15 THEN 'PR'
```

```
        WHEN 16 THEN 'PE'
        WHEN 17 THEN 'PI'
        WHEN 18 THEN 'RJ'
        WHEN 19 THEN 'RN'
        WHEN 20 THEN 'RS'
        WHEN 21 THEN 'RO'
        WHEN 22 THEN 'RR'
        WHEN 23 THEN 'SC'
        WHEN 24 THEN 'SP'
        WHEN 25 THEN 'SE'
        ELSE 'TO'
    END AS estado
FROM generate_series(1, 500) g;

--*****************************************************
--Tabela Localização

INSERT INTO Localizacao (
    id_localizacao,
    andar,
    setor,
    corredor,
    estante,
    prateleira
)
SELECT
    g AS id_localizacao,
    (g % 5) + 1 AS andar,        -- valores de 1 a 6
    (g % 10) + 1 AS setor,       -- valores de 1 a 10
    (g % 20) + 1 AS corredor,    -- valores de 1 a 20
    (g % 30) + 1 AS estante,     -- valores de 1 a 30
    (g % 6) + 1 AS prateleira    -- valores de 1 a 6
FROM generate_series(1, 500) g;

--*********************************************************
--Tabela Leitor

INSERT INTO Leitor (
    id_leitor,
    nome_leitor,
    telefone,
    data_nascimento,
    cpf
)
SELECT
    g AS id_leitor,

    -- Nome real brasileiro aleatório dentre 120 opções
    (ARRAY[
        'Miguel Souza','Arthur Almeida','Heitor Freitas','Theo Cardoso','Davi Farias',
        'Bernardo Siqueira','Samuel Ribeiro','Gabriel Castro','Isaac Fernandes','Lucas
Pereira',
        'Rafael Santos','Matheus Costa','Vinícius Moura','Enzo Gabriel','Pedro
Henrique',
```

```
        'João Pedro','Caio Vasconcelos','Gustavo Martins','Eduardo Teixeira','Felipe
Carvalho',
        'Daniel Rodrigues','Henrique Monteiro','Alexandre Pires','Rodrigo
Antunes','Leonardo Rezende',
        'Diego Tavares','Renato Silveira','André Moura','Marcelo Campos','Bruno
Araújo',
        'Igor Batista','Murilo Nascimento','Tiago Lemos','Paulo Henrique','Álvaro
Assis',
        'César Andrade','Otávio Neves','Fábio Couto','Roberto Mendes','Jorge Lima',
        'Ana Clara','Maria Eduarda','Mariana Rocha','Júlia Melo','Laura Ribeiro',
        'Alice Barbosa','Valentina Duarte','Giovanna Fonseca','Sofia
Almeida','Isabella Martins',
        'Beatriz Azevedo','Yasmin Barros','Heloísa Torres','Lívia Fernandes','Bianca
Freitas',
        'Manuela Prado','Camila Fernandes','Nicole Carvalho','Clarice
Pacheco','Catarina Aguiar',
        'Helena Goes','Bruna Esteves','Rafaela Nunes','Fernanda Lima','Julia Castro',
        'Carolina Figueiredo','Patrícia Nunes','Larissa Gomes','Priscila
Melo','Natália Teixeira',
        'Luciana Batista','Vanessa Prado','Gabriela Barros','Amanda Dias','Daniela
Albuquerque',
        'Carla Ramos','Elis Regina Ferreira','Sabrina Rocha','Lorena Silveira','Tainá
Azevedo',
        'Ana Beatriz Costa','Ana Júlia Fernandes','Maria Clara Viana','Maria Cecília
Paes','Maria Alice Duarte',
        'João Miguel Silveira','João Lucas Santiago','Luiz Felipe Torres','Luiz Otávio
Mendes','Pedro Lucas Nogueira',
        'Pedro Miguel Freire','Carlos Eduardo Pires','José Augusto Melo','José
Henrique Fialho','Paulo Roberto Antunes',
        'Marco Antônio Pacheco','Antônio Carlos Bastos','João Victor Cardoso','João
Guilherme Rezende','Maria Vitória Elias',
        'Ana Carolina Lopes','Maria Gabriela Campos','Maria Luiza Barbosa','Arthur
Benjamin Braga','Rodrigo Rafael Silva',
        'João Marcos Ribeiro','Maria Eduarda Gomes','Ana Clara Ferreira','Lucas
Emanuel Costa','Matheus Gabriel Duarte',
        'Bianca Vitória Moreira','Camila Daniela Alves','Luiza Helena Pires','Isadora
Cristina Ramos','Marina Beatriz Lemos',
        'João Pedro Albuquerque','Gabriel Henrique Castro','Lara Vitória
Cardoso','Elisa Camargo','Paola Andrade'
    ])[floor(random() * 120) + 1] AS nome_leitor,

    -- Telefone fictício
    '(11) 9' || LPAD((floor(random()*900000) + 100000)::text, 6, '0') AS telefone,

    -- Datas de nascimento entre 1960 e 2010
    DATE '1960-01-01' + (floor(random() * 18250)) * INTERVAL '1 day' AS
data_nascimento,

    -- CPF fictício com 11 dígitos aleatórios (não corresponde a CPF real)
    LPAD((floor(random() * 99999999999))::text, 11, '0') AS cpf

FROM generate_series(1, 500) g;
```

```
--
*****************************************************************************************
***
Tabela Email_Leitor

INSERT INTO Email_Leitor (
    id_email_leitor,
    email_leitor,
    id_leitor
)
SELECT
    g AS id_email_leitor,

    -- Geração do e-mail
    LOWER(
        translate(
            replace(l.nome_leitor, ' ', '.'),
            'ÁÀÂÃÄáàâãäÉÈÊËéèêëÍÌÎÏíìîïÓÒÔÕÖóòôõöÚÙÛÜúùûüÇç',
            'AAAAAaaaaaEEEEeeeeIIIIiiiiOOOOOoooooUUUUuuuuCc'
        )
    )
    || floor(random() * 100)
    || '@' ||
    (ARRAY[
        'gmail.com',
        'outlook.com',
        'yahoo.com',
        'hotmail.com',
        'bol.com.br',
        'uol.com.br',
        'terra.com.br',
        'icloud.com',
        'protonmail.com'
    ])[floor(random() * 9) + 1]  -- domínio aleatório

    AS email_leitor,

    g AS id_leitor

FROM generate_series(1, 500) g
JOIN Leitor l ON l.id_leitor = g;

--*************************************************************************
--Carteira da Biblioteca

INSERT INTO Carteira_da_Biblioteca (id_carteira, validade, status, id_leitor)
SELECT
    g AS id_carteira,

    -- Gera a data de validade uma única vez
    validade_aleatoria,

    -- Status consistente com a validade
```

```
    CASE
        WHEN validade_aleatoria < CURRENT_DATE THEN 'Expirada'           -- validade
passada
        WHEN random() < 0.1 THEN 'Suspensa'                              -- 10%
suspensas aleatórias
        ELSE 'Ativada'                                                    -- restante
das válidas
    END AS status,

    g AS id_leitor

FROM (
    SELECT
        g,
        CURRENT_DATE + (floor(random() * 2555) - 730) * INTERVAL '1 day' AS
validade_aleatoria
    FROM generate_series(1, 500) g
) t;

--
**********************************************************************************
--Tabela Livro

INSERT INTO Livro (id_livro, titulo_livro, nome_autor, ano_publicacao, id_editora)
SELECT
    g AS id_livro,

    -- Título do livro (não nulo)
    COALESCE(
        (ARRAY[
            'Aventuras em ','Mistério de ','Segredos de ','O Mundo de ','Histórias de
',
            'O Diário de ','Lendas de ','O Enigma de ','Memórias de ','A Jornada de ',
            'O Retorno de ','O Caminho de ','A Busca por ','A Cidade de ','O Labirinto
de '
        ])[floor(random()*15)+1] ||
        (ARRAY[

'Mariana','Pedro','Lucas','Miguel','Ana','Sofia','Gabriel','Beatriz','Rafael','Carolin
a',
            'Floresta','Cidade
Perdida','Tesouro','Ilha','Castelo','Tempo','Universo','Mistério','Segredo','Aventura'
,

'Sombras','Luz','Montanha','Rio','Espelho','Portal','Estrela','Vento','Chuva','Mar',
            'Segredo
Antigo','Fantasma','Príncipe','Princesa','Dragão','Rei','Rainha','Caverna','Floresta
Negra','Caminho Proibido'
        ])[floor(random()*40)+1],
        'Livro Sem Título'
    ) AS titulo_livro,

    -- Nome do autor (não nulo)
```

```
    COALESCE(
        (ARRAY[
            'Miguel Souza','Arthur Almeida','Heitor Freitas','Theo Cardoso','Davi
Farias','Bernardo Siqueira',
            'Samuel Ribeiro','Gabriel Castro','Isaac Fernandes','Lucas
Pereira','Rafael Santos','Matheus Costa',
            'Vinícius Moura','Enzo Gabriel','Pedro Henrique','João Pedro','Caio
Vasconcelos','Gustavo Martins',
            'Eduardo Teixeira','Felipe Carvalho','Daniel Rodrigues','Henrique
Monteiro','Alexandre Pires','Rodrigo Antunes',
            'Leonardo Rezende','Diego Tavares','Renato Silveira','André
Moura','Marcelo Campos','Bruno Araújo',
            'Igor Batista','Murilo Nascimento','Tiago Lemos','Paulo Henrique','Álvaro
Assis','César Andrade',
            'Otávio Neves','Fábio Couto','Roberto Mendes','Jorge Lima','Ana
Clara','Maria Eduarda','Mariana Rocha',
            'Júlia Melo','Laura Ribeiro','Alice Barbosa','Valentina Duarte','Giovanna
Fonseca','Sofia Almeida',
            'Isabella Martins','Beatriz Azevedo','Yasmin Barros','Heloísa
Torres','Lívia Fernandes','Bianca Freitas',
            'Manuela Prado','Camila Fernandes','Nicole Carvalho','Clarice
Pacheco','Catarina Aguiar','Helena Goes',
            'Bruna Esteves','Rafaela Nunes','Fernanda Lima','Julia Castro','Carolina
Figueiredo','Patrícia Nunes',
            'Larissa Gomes','Priscila Melo','Natália Teixeira','Luciana
Batista','Vanessa Prado','Gabriela Barros',
            'Amanda Dias','Daniela Albuquerque','Carla Ramos','Elis Regina
Ferreira','Sabrina Rocha','Lorena Silveira',
            'Tainá Azevedo','Ana Beatriz Costa','Ana Júlia Fernandes','Maria Clara
Viana','Maria Cecília Paes','Maria Alice Duarte',
            'João Miguel Silveira','João Lucas Santiago','Luiz Felipe Torres','Luiz
Otávio Mendes','Pedro Lucas Nogueira',
            'Pedro Miguel Freire','Carlos Eduardo Pires','José Augusto Melo','José
Henrique Fialho','Paulo Roberto Antunes',
            'Marco Antônio Pacheco','Antônio Carlos Bastos','João Victor
Cardoso','João Guilherme Rezende','Maria Vitória Elias',
            'Ana Carolina Lopes','Maria Gabriela Campos','Maria Luiza Barbosa','Arthur
Benjamin Braga','Rodrigo Rafael Silva',
            'João Marcos Ribeiro','Maria Eduarda Gomes','Ana Clara Ferreira','Lucas
Emanuel Costa','Matheus Gabriel Duarte',
            'Bianca Vitória Moreira','Camila Daniela Alves','Luiza Helena
Pires','Isadora Cristina Ramos','Marina Beatriz Lemos'
        ])[floor(random()*120)+1],
        'Autor Desconhecido'
    ) AS nome_autor,

    -- Ano de publicação (não nulo)
    1950 + floor(random() * 76)::int AS ano_publicacao,

    -- id_editora (não nulo)
    1 + floor(random() * 500)::int AS id_editora

FROM generate_series(1, 500) g;
--*************************************************************
```

```sql
--Tabela Livro Leitor

-- Inserir registros
WITH leitor_aleatorio AS (
    -- Cada leitor terá entre 1 e 5 livros
    SELECT
        l AS id_leitor,
        (1 + floor(random() * 5)::int) AS qtd_livros
    FROM generate_series(1, 500) l
),
livros_por_leitor AS (
    -- Distribuir livros aleatórios para cada leitor
    SELECT
        ROW_NUMBER() OVER () AS id_livro_leitor,
        id_leitor,
        (1 + floor(random() * 500)::int) AS id_livro
    FROM leitor_aleatorio, generate_series(1, qtd_livros)
)
-- Inserção final, removendo possíveis duplicatas (mesmo livro para mesmo leitor)
INSERT INTO Livro_Leitor (id_livro_leitor, id_livro, id_leitor)
SELECT DISTINCT id_livro_leitor, id_livro, id_leitor
FROM livros_por_leitor
LIMIT 500;  -- garante exatamente 500 registros

--*************************************************************************************
--Tabela Gênero Livro

-- Gerar registros criativos para Genero_Livro
WITH livros AS (
    SELECT id_livro FROM Livro
),
palavras1 AS (
    SELECT unnest(ARRAY[
        'Ficção','Romance','Mistério','Aventura','Drama',
        'História','Suspense','Fantasia','Biografia','Épico'
    ]) AS p
),
palavras2 AS (
    SELECT unnest(ARRAY[
        'Moderna','Antiga','Épica','Contemporânea','Clássica',
        'Mágica','Sombria','Encantada','Proibida','Perdida'
    ]) AS p
),
generos_criativos AS (
    SELECT CONCAT(p1.p, ' ', p2.p) AS genero_livro
    FROM palavras1 p1 CROSS JOIN palavras2 p2
),
livro_genero AS (
    SELECT
        row_number() OVER () AS id_genero_livro,
        l.id_livro,
        g.genero_livro
    FROM livros l
    CROSS JOIN LATERAL (
```

```
        SELECT genero_livro
        FROM generos_criativos
        ORDER BY random()
        LIMIT (1 + floor(random()*2)::int)  -- 1 a 3 gêneros por livro
    ) g
)
INSERT INTO Genero_Livro (id_genero_livro, genero_livro, id_livro)
SELECT id_genero_livro, genero_livro, id_livro
FROM livro_genero;


--
**********************************************************************************
--Tabela Exemplar

-- Gerar exemplares com códigos de barras únicos
WITH livros AS (
    SELECT id_livro FROM Livro
),
exemplares AS (
    SELECT
        row_number() OVER () AS id_exemplar,
        l.id_livro,
        CASE
            WHEN random() < 0.4 THEN 'Novo'
            WHEN random() < 0.8 THEN 'Bom'
            WHEN random() < 0.95 THEN 'Regular'
            ELSE 'Danificado'
        END AS estado_de_conservacao,
        -- Código de barras único de 15 dígitos usando id_exemplar como base
        LPAD((row_number() OVER () + 100000000000000)::text, 15, '0') AS
codigo_de_barras,
        1 + floor(random()*500)::int AS id_localizacao
    FROM livros l
    CROSS JOIN generate_series(1, (1 + floor(random()*5)::int))  -- 1 a 5 exemplares
por livro
)
INSERT INTO Exemplar (id_exemplar, estado_de_conservacao, codigo_de_barras, id_livro,
id_localizacao)
SELECT id_exemplar, estado_de_conservacao, codigo_de_barras, id_livro, id_localizacao
FROM exemplares;


--
**********************************************************************************
**************************
--Tabela Empréstimo

-- Inserir 500 empréstimos aleatórios
INSERT INTO Emprestimo (data_emprestimo, data_devolucao, id_exemplar, id_carteira)
SELECT
    -- Data de empréstimo aleatória nos últimos 3 anos
    (CURRENT_DATE - (floor(random() * 1095)::int))::date AS data_emprestimo,

    -- Data de devolução: 70% já devolvidos, 30% ainda não
    CASE
```

```
        WHEN random() < 0.7 THEN (CURRENT_DATE - (floor(random() * 1095)::int) + (7 +
floor(random() * 24)::int))::date
        ELSE NULL
    END AS data_devolucao,

    -- id_exemplar aleatório entre os existentes
    (SELECT id_exemplar FROM Exemplar ORDER BY random() LIMIT 1) AS id_exemplar,

    -- id_carteira aleatório entre os existentes
    (SELECT id_carteira FROM Carteira_da_Biblioteca ORDER BY random() LIMIT 1) AS
id_carteira
FROM generate_series(1, 500);
```