**Anexo 4**

**Script DDL e DML de Preparação das Tabelas Envolvidas no Exemplo do Capitulo 5**

CREATE TABLE IF NOT EXISTS public.editora

(

id integer,

nome character varying NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public.autor

(

id integer,

nome character varying NOT NULL,

dt\_nasc date NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public.livro

(

id integer,

titulo character varying NOT NULL,

ano\_pub integer NOT NULL,

preco numeric NOT NULL,

editora\_id integer NOT NULL,

PRIMARY KEY (id)

);

-- Criar a tabela LivroAutor sem a restrição UNIQUE inicialmente

CREATE TABLE IF NOT EXISTS public.livro\_autor

(

id integer,

livro\_id integer,

autor\_id integer,

papel "char" NOT NULL,

PRIMARY KEY (id)

);

-- Adicionar as chaves estrangeiras

ALTER TABLE IF EXISTS public.livro\_autor

ADD CONSTRAINT "livro\_autor\_livroFk" FOREIGN KEY (livro\_id)

REFERENCES public.livro (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.livro\_autor

ADD CONSTRAINT "livro\_autor\_autorFk" FOREIGN KEY (autor\_id)

REFERENCES public.autor (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

-- Adicionar a restrição UNIQUE após as chaves estrangeiras

ALTER TABLE IF EXISTS public.livro\_autor

ADD CONSTRAINT "livro\_autorUk" UNIQUE (livro\_id, autor\_id);

ALTER TABLE public.livro

ADD CONSTRAINT "livro\_editoraFk" FOREIGN KEY (editora\_id)

REFERENCES public.editora (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

-- Gerar 1.000 editoras

INSERT INTO public.editora (id, nome)

SELECT

i,

'Editora ' || i

FROM generate\_series(1, 1000) AS i;

-- Gerar 1.000 autores

INSERT INTO public.autor (id, nome, dt\_nasc)

SELECT

i,

'Autor ' || i,

date('1950-01-01') + (random() \* (date('2000-01-01') - date('1950-01-01'))) \* interval '1 day'

FROM generate\_series(1, 1000) AS i;

-- Gerar 1.000.000 de livros

INSERT INTO public.livro (id, titulo, ano\_pub, preco, editora\_id)

SELECT

i,

'Livro ' || i,

floor(random() \* (2024 - 2020 + 1)) + 2020,

round((random() \* 100)::numeric, 2),

floor(random() \* 1000 + 1) -- "editora\_id" aleatório entre 1 e 1000

FROM generate\_series(1, 1000000) AS i;

-- Gerar registros na tabela livro\_autor

DO $$

DECLARE

livro\_id integer;

autor\_id integer;

papel char;

autores\_adicionados integer[]; -- Array para armazenar os autores já adicionados ao livro

BEGIN

FOR livro\_id IN 1..1000000 LOOP

autores\_adicionados := array[]::integer[]; -- Inicializar o array para cada livro

-- Autor principal

autor\_id := floor(random() \* 1000 + 1);

papel := 'P';

INSERT INTO public.livro\_autor (id, livro\_id, autor\_id, papel)

VALUES (nextval('livro\_autor\_id\_seq'), livro\_id, autor\_id, papel);

autores\_adicionados := array\_append(autores\_adicionados, autor\_id); -- Adicionar o autor ao array

-- Coautores (0 a 3 coautores por livro)

FOR i IN 1..floor(random() \* 4) LOOP

autor\_id := floor(random() \* 1000 + 1);

-- Verificar se o autor já foi adicionado ao livro

WHILE autor\_id = ANY(autores\_adicionados) LOOP

autor\_id := floor(random() \* 1000 + 1);

END LOOP;

papel := 'C';

INSERT INTO public.livro\_autor (id, livro\_id, autor\_id, papel)

VALUES (nextval('livro\_autor\_id\_seq'), livro\_id, autor\_id, papel);

autores\_adicionados := array\_append(autores\_adicionados, autor\_id); -- Adicionar o autor ao array

END LOOP;

END LOOP;

END $$;