**Comandos do PostgreSQL - Cidades**

**CREATE**

-- Uma cidade tem somente um estado (1 para N)

-- Uma cidade tem somente um prefeito (1 para 1)

-- Uma cidade tem muitas empresas (N para N)

-- A FK que vem de estados precisa ser não nula, para garantir a integridade referencial entre estado e cidade, deve existir uma PK em estado para que seja feita a inserção de uma cidade

CREATE TABLE IF NOT EXISTS cidades(

  id SERIAL PRIMARY KEY,

  nome VARCHAR(255) NOT NULL,

  estado\_id INT NOT NULL,

  area DECIMAL(10, 2) NOT NULL,

  FOREIGN KEY (estado\_id) REFERENCES estados(id)

);

CREATE TABLE IF NOT EXISTS teste(

  id SERIAL PRIMARY KEY

);

DROP TABLE IF EXISTS teste;

**INSERT**

-- São Paulo tem id = 25

INSERT INTO cidades(nome, area, estado\_id)

VALUES('Campinas', 795, 25);

-- Rio de Janeiro tem id = 19

INSERT INTO cidades(nome, area, estado\_id)

VALUES('Niteroi', 133.9, 19);

-- Pernambuco tem id = 17, usando o SELECT com WHERE dentro do INSERT

INSERT INTO cidades(nome, area, estado\_id)

VALUES('Caruaru', 920.6, (SELECT id FROM estados WHERE sigla = 'PE'));

-- Ceará tem id = 6

INSERT INTO cidades(nome, area, estado\_id)

VALUES('Juazeiro do Norte', 248.2, (SELECT id FROM estados WHERE sigla = 'CE'));

**SELECT**

-- Active: 1704649501576@@127.0.0.1@5432@exercicios@public

-- Use de um Alias(Apelido) para fazer consultas com várias tabelas

-- Isso gera um produto cartesiano ou um CROSS JOIN

SELECT \* FROM estados est, cidades cid;

-- Retorna os estados e suas cidades associadas, estamos usando o ALIAS

SELECT

  est.nome AS "Estado",

  cid.nome AS "Cidade",

  regiao AS "Região"

FROM estados est, cidades cid

WHERE est.id = cid.estado\_id;

-- Retorna os estados e suas cidades associadas, estamos usando o INNER JOIN

-- INNER JOIN: Intersecção entre as tabelas, ou os dados que possuem relacionamento

SELECT

  est.nome AS "Estado",

  cid.nome AS "Cidade",

  regiao AS "Região"

FROM estados est

INNER JOIN cidades cid ON est.id = cid.estado\_id;