

Exercício 3

Comandos de criação das tabelas

Tipos de dados pré-definidos

Utilizamos de alguns tipos definidos por nós para atributos que possuem valores possíveis pré-definidos pelo enunciado

```
CREATE TYPE tipo_prop_enum AS ENUM ('casa', 'quarto individual', 'quarto compartilhado');  
CREATE TYPE tipo_cama_enum AS ENUM ('solteiro', 'casal', 'beliche');  
CREATE TYPE condicao_enum AS ENUM ('confirmado', 'cancelado');  
CREATE TYPE genero_enum AS ENUM ('masculino', 'feminino', 'outro');  
CREATE TYPE rate_range AS ENUM ('bom', 'medio', 'ruim');  
CREATE TYPE tipo_conta_enum AS ENUM ('poupanca', 'conta corrente');
```

Tipos de dados utilizados

Utilizamos dos seguintes tipos de dados do PostgreSQL

- **VARCHAR(50)**: Utilizado para armazenar qualquer tipo de nome.
- **VARCHAR(15)**: Utilizado para armazenar números de telefone.
- **DATE**: Utilizado para datas.
- **TIME(0)**: Utilizado para armazenar horas (sem minutos e segundos), quando somente essa é relevante.
- **TIMESTAMP**: Utilizado para armazenar data e hora.
- **MONEY**: Utilizado para armazenar valores monetários.
- **SMALLINT**: Utilizado para armazenar números inteiros pequenos, para atributos que números muito grandes não faria sentido, exemplo: número de camas.
- **INTEGER**: Utilizado para armazenar números inteiros.
- **SERIAL**: Utilizado para criar uma sequência única de inteiros (auto-incremento) para os atributos de id.
- **TEXT**: Utilizado para armazenar grandes quantidades de texto, como nos atributos de mensagem.
- **TEXT []**: Utilizado para armazenar um array de grandes quantidades de texto, como no atributo fotos, da relação Avaliacao.
- **DATE[]**: Utilizado para armazenar um array de datas no atributo datas disponíveis da relação Propriedade.

Print da execução dos comandos

```
%%sql
COMMIT;
DROP DATABASE IF EXISTS trabalho1 WITH (FORCE);
COMMIT;
CREATE DATABASE trabalho1
  WITH OWNER = postgres
  ENCODING = 'UTF8'
  LOCALE = 'pt_BR.UTF-8';
COMMIT;
```

✓ 0.1s

Python

```
* postgresql://ziel:***@localhost:5432/template1
postgresql://ziel:***@localhost:5432/trabalho1
Done.
Done.
Done.
Done.
Done.
```

```
%%sql
CREATE TYPE tipo_prop_enum AS ENUM ('casa', 'quarto individual', 'quarto compartilhado');
CREATE TYPE tipo_cama_enum AS ENUM ('solteiro', 'casal', 'beliche');
CREATE TYPE condicao_enum AS ENUM ('confirmado', 'cancelado');
CREATE TYPE genero_enum AS ENUM ('masculino', 'feminino', 'outro');
CREATE TYPE rate_range AS ENUM ('bom', 'medio', 'ruim');
CREATE TYPE tipo_conta_enum AS ENUM ('poupanca', 'conta corrente');
```

✓ 0.0s

Python

```
postgresql://ziel:***@localhost:5432/template1
* postgresql://ziel:***@localhost:5432/trabalho1
Done.
Done.
Done.
Done.
Done.
Done.
```

```
%%sql
DROP TABLE IF EXISTS Localizacao CASCADE;
CREATE TABLE Localizacao (
  id_local SERIAL PRIMARY KEY,
  pais VARCHAR(50),
  estado VARCHAR(50),
  cidade VARCHAR(50),
  bairro VARCHAR(50)
);
```

✓ 0.0s

Python

```
postgresql://ziel:***@localhost:5432/template1
* postgresql://ziel:***@localhost:5432/trabalho1
Done.
Done.
```

```
%%sql
DROP TABLE IF EXISTS Anfitriao CASCADE;
CREATE TABLE Anfitriao (
    nome VARCHAR(50),
    sobrenome VARCHAR(50),
    data_nasc DATE,
    endereco VARCHAR(50),
    genero genero_enum,
    num_tel VARCHAR(15),
    email VARCHAR(50),
    senha VARCHAR(50),
    id_local INTEGER,
    PRIMARY KEY (nome, sobrenome, num_tel),
    FOREIGN KEY (id_local) REFERENCES Localizacao
);
```

✓ 0.0s

Python

postgresql://ziel:***@localhost:5432/template1

* postgresql://ziel:***@localhost:5432/trabalho1

Done.

Done.

```
%%sql
DROP TABLE IF EXISTS Locatario CASCADE;
CREATE TABLE Locatario (
    nome VARCHAR(50),
    sobrenome VARCHAR(50),
    data_nasc DATE,
    endereco VARCHAR(50),
    genero genero_enum,
    num_tel VARCHAR(15),
    email VARCHAR(50),
    senha VARCHAR(50),
    id_local INTEGER,
    PRIMARY KEY (nome, sobrenome, num_tel),
    FOREIGN KEY (id_local) REFERENCES Localizacao
);
```

✓ 0.0s

Python

postgresql://ziel:***@localhost:5432/template1

* postgresql://ziel:***@localhost:5432/trabalho1

Done.

Done.

```
%%sql
DROP TABLE IF EXISTS Propriedade CASCADE;
CREATE TABLE Propriedade (
    nome VARCHAR(50),
    endereco VARCHAR(50),
    tipo_propriedade tipo_prop_enum,
    qtd_quartos SMALLINT,
    qtd_banheiros SMALLINT,
    max_hospedes SMALLINT,
    min_noites SMALLINT,
    max_noites SMALLINT,
    preco_noite MONEY,
    taxa_limpeza MONEY,
    datas_disponivel DATE[],
    horario_checkIn TIME(0),
    horario_checkOut TIME(0),
    nome_anf VARCHAR(50),
    sobrenome_anf VARCHAR(50),
    num_tel_anf VARCHAR(15),
    id_local INTEGER,
    PRIMARY KEY (nome, endereco),
    FOREIGN KEY (nome_anf, sobrenome_anf, num_tel_anf) REFERENCES Anfitriao,
    FOREIGN KEY (id_local) REFERENCES Localizacao
);
```

✓ 0.0s

Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

```
%%sql
DROP TABLE IF EXISTS Mensagem CASCADE;
CREATE TABLE Mensagem (
    nome_envia VARCHAR(50),
    sobrenome_envia VARCHAR(50),
    num_tel_envia VARCHAR(50),
    nome_recebe VARCHAR(50),
    sobrenome_recebe VARCHAR(50),
    num_tel_recebe VARCHAR(50),
    horario_envio TIMESTAMP,
    texto_mensagem TEXT,
    PRIMARY KEY(nome_envia, num_tel_envia, sobrenome_envia, horario_envio)
);
```

✓ 0.0s

Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

```
%%sql
```

```
DROP TABLE IF EXISTS Locacao CASCADE;
CREATE TABLE Locacao (
    id_locacao SERIAL,
    data_checkIn DATE,
    data_checkOut DATE,
    data_reserva DATE,
    qtd_hospedes SMALLINT,
    imposto MONEY,
    preco_estadia MONEY,
    cod_promocional VARCHAR(30),
    desconto MONEY,
    preco_total MONEY,
    condicao condicao_enum,
    nome_hosp VARCHAR(50),
    sobrenome_hosp VARCHAR(50),
    num_tel_hosp VARCHAR(15),
    nome_prop VARCHAR(50),
    endereco_prop VARCHAR(50),
    FOREIGN KEY (nome_hosp, sobrenome_hosp, num_tel_hosp) REFERENCES Locatario,
    FOREIGN KEY (nome_prop, endereco_prop) REFERENCES Propriedade,
    PRIMARY KEY(id_locacao)
);
```

✓ 0.0s

```
* postgresql://ziel:***@localhost:5432/template1
postgresql://ziel:***@localhost:5432/trabalho1
```

Done.

Done.

```
%%sql
```

```
DROP TABLE IF EXISTS Avaliacao CASCADE;
CREATE TABLE Avaliacao (
    id_avaliacao SERIAL PRIMARY KEY,
    fotos TEXT[],
    rate_limpeza rate_range,
    rate_comunicacao rate_range,
    rate_localizacao rate_range,
    rate_valor rate_range,
    nome_envia VARCHAR(50),
    sobrenome_envia VARCHAR(50),
    num_tel_envia VARCHAR(50),
    horario_envio TIMESTAMP,
    FOREIGN KEY (nome_envia, num_tel_envia, sobrenome_envia, horario_envio) REFERENCES Mensagem
);
```

✓ 0.0s

Python

```
postgresql://ziel:***@localhost:5432/template1
```

```
* postgresql://ziel:***@localhost:5432/trabalho1
```

Done.

Done.

```
%%sql
DROP TABLE IF EXISTS Quarto CASCADE;
CREATE TABLE Quarto (
    id_quarto SERIAL,
    qtd_camas SMALLINT,
    tipo_cama tipo_cama_enum,
    nome_prop VARCHAR(50),
    endereco_prop VARCHAR(50),
    FOREIGN KEY (nome_prop, endereco_prop) REFERENCES Propriedade,
    PRIMARY KEY(id_quarto,nome_prop, endereco_prop)
);
```

✓ 0.0s Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

[]

```
%%sql
DROP TABLE IF EXISTS Ponto_interesse CASCADE;
CREATE TABLE Ponto_interesse (
    id_ptnInteresse SERIAL,
    id_local INTEGER,
    nome_ponto VARCHAR(50),
    FOREIGN KEY (id_local) REFERENCES Localizacao,
    PRIMARY KEY(id_ptnInteresse,id_local)
);
```

✓ 0.0s Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

```
%%sql
DROP TABLE IF EXISTS Regras CASCADE;
CREATE TABLE Regras (
    id_regra SERIAL,
    nome_regra VARCHAR(50),
    nome_prop VARCHAR(50),
    endereco_prop VARCHAR(50),
    FOREIGN KEY (nome_prop, endereco_prop) REFERENCES Propriedade,
    PRIMARY KEY(id_regra,nome_prop, endereco_prop)
);
```

✓ 0.0s Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

```
%%sql
DROP TABLE IF EXISTS Comodidades CASCADE;
CREATE TABLE Comodidades (
    id_comodidade SERIAL,
    nome_comodidade VARCHAR(50),
    nome_prop VARCHAR(50),
    endereco_prop VARCHAR(50),
    FOREIGN KEY (nome_prop, endereco_prop) REFERENCES Propriedade,
    PRIMARY KEY(id_comodidade,nome_prop, endereco_prop)
);
```

✓ 0.0s Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

```

%%sql
DROP TABLE IF EXISTS Conta CASCADE;
CREATE TABLE Conta (
    num_conta INTEGER,
    num_roteamento INTEGER,
    tipo_conta tipo_conta_enum,
    nome_anf VARCHAR(50),
    sobrenome_anf VARCHAR(50),
    num_tel_anf VARCHAR(15),
    FOREIGN KEY (nome_anf, sobrenome_anf, num_tel_anf) REFERENCES Anfitriao,
    PRIMARY KEY(num_conta,nome_anf, sobrenome_anf, num_tel_anf)
);

```

✓ 0.0s

Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

```

%%sql
DROP TABLE IF EXISTS EnviaRecebeMensagem CASCADE;
CREATE TABLE EnviaRecebeMensagem (
    nome_envia VARCHAR(50),
    sobrenome_envia VARCHAR(50),
    num_tel_envia VARCHAR(50),
    horario_envio TIMESTAMP,
    nome_hosp VARCHAR(50),
    sobrenome_hosp VARCHAR(50),
    num_tel_hosp VARCHAR(50),
    nome_anf VARCHAR(50),
    sobrenome_anf VARCHAR(50),
    num_tel_anf VARCHAR(50),
    PRIMARY KEY(nome_hosp, sobrenome_hosp, num_tel_hosp,
               nome_anf, sobrenome_anf, num_tel_anf,
               nome_envia, num_tel_envia, sobrenome_envia, horario_envio) ,
    FOREIGN KEY (nome_hosp, sobrenome_hosp, num_tel_hosp) REFERENCES Locatario,
    FOREIGN KEY (nome_anf, sobrenome_anf, num_tel_anf) REFERENCES Anfitriao,
    FOREIGN KEY (nome_envia, num_tel_envia, sobrenome_envia, horario_envio) REFERENCES Mensagem
);

```

✓ 0.0s

* [postgresql://ziel:***@localhost:5432/template1](#)

[postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.

```

%%sql
DROP TABLE IF EXISTS FazAvaliacaoPara CASCADE;
CREATE TABLE FazAvaliacaoPara (
    id_avaliacao SERIAL,
    nome_hosp VARCHAR(50),
    sobrenome_hosp VARCHAR(50),
    num_tel_hosp VARCHAR(50),
    nome_prop VARCHAR(50),
    endereco_prop VARCHAR(50),
    FOREIGN KEY (nome_hosp, sobrenome_hosp, num_tel_hosp) REFERENCES Locatario,
    FOREIGN KEY (id_avaliacao) REFERENCES Avaliacao,
    FOREIGN KEY (nome_prop, endereco_prop) REFERENCES Propriedade,
    PRIMARY KEY(nome_hosp, sobrenome_hosp, num_tel_hosp,nome_prop, endereco_prop, id_avaliacao)
);

```

✓ 0.0s

Python

[postgresql://ziel:***@localhost:5432/template1](#)

* [postgresql://ziel:***@localhost:5432/trabalho1](#)

Done.

Done.