

---

# Banco de Dados

## **SQL: CRIANDO TABELAS**

Prof. Dr. Edimar Manica

[edimar.manica@ibiruba.ifrs.edu.br](mailto:edimar.manica@ibiruba.ifrs.edu.br)

# PRINCIPAIS COMANDOS

---

## DDL

- **CREATE TABLE**
- ALTER TABLE
- DROP TABLE

## DML

- INSERT INTO
- UPDATE
- DELETE FROM

## DQL

- SELECT

## DCL

- GRANT TO
- REVOKE FROM

## DTL

- BEGIN
- COMMIT
- ROLLBACK

# ESQUEMA DE UM CONSULTÓRIO

---

cidade (codigo, nome, UF)

paciente (codigo, nome, email, fone, cod\_cid#)  
(cod\_cid) REFERENCIA cidade (codigo)

medico (codigo, nome, email, CRM, cod\_cid#)  
(cod\_cid) REFERENCIA cidade (codigo)

consulta (data, hora, cod\_med#, cod\_pac#)  
(cod\_pac) REFERENCIA paciente (codigo)  
(cod\_med) REFERENCIA medico (codigo)

medicamento (codigo, descricao)

cons\_medicame (data, hora, cod\_med#, cod\_medica#)  
(cod\_medica) REFERENCIA medicamento (codigo)  
(data, hora, cod\_med) REFERENCIA consulta (data, hora, cod\_med)

# criação de tabela

---

```
CREATE TABLE nometabela
(
    nomeatributo1 tipoatributo1 listadeopcoes1,
    nomeatributo2 tipoatributo2 listadeopcoes2,
    ...
    nomeatributon tipoatributon listadeopcoesn,

    CONSTRAINT PK_nometabela PRIMARY KEY (nomeatributochave)
);
```

**OBS:** poderia ser tudo em uma única linha.

# CRIAÇÃO DE TABELA

---

## Modelo Lógico

cidade(codigo, nome, uf)

## Script SQL

```
CREATE TABLE cidade
(
    codigo    SERIAL          NOT NULL,
    nome      VARCHAR(80)     NOT NULL,
    uf        CHAR(2)         DEFAULT 'RS',
    CONSTRAINT pk_cidade PRIMARY KEY (codigo)
);
```

# CRIAÇÃO DE TABELA

---

## Modelo Lógico

cidade(codigo, nome, uf)

## Script SQL

```
CREATE TABLE cidade  
(  
    codigo SERIAL NOT NULL,  
    nome VARCHAR(80) NOT NULL,  
    uf CHAR(2) DEFAULT 'RS',  
    CONSTRAINT pk_cidade PRIMARY KEY (codigo)  
);
```

Nome da  
Tabela

Parênteses

# CRIAÇÃO DE TABELA

## Modelo Lógico

cidade(codigo, nome, uf)

## Script SQL

```
CREATE TABLE cidade
```

```
(
```

codigo	SERIAL	NOT NULL	,
nome	VARCHAR(80)	NOT NULL	,
uf	CHAR(2)	DEFAULT 'RS'	,

Nome do  
atributo

Tipo do  
atributo

Opções

Vírgula

```
    CONSTRAINT pk_cidade PRIMARY KEY (codigo)  
);
```

# CRIAÇÃO DE TABELA

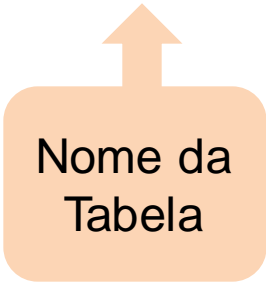
---

## Modelo Lógico

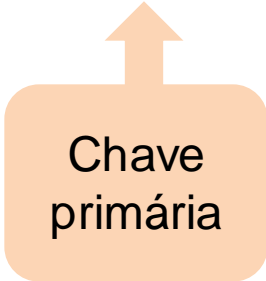
cidade(codigo, nome, uf)

## Script SQL

```
CREATE TABLE cidade
(
    codigo    SERIAL          NOT NULL,
    nome      VARCHAR(80)     NOT NULL,
    uf        CHAR(2)         DEFAULT 'RS',
    CONSTRAINT pk_cidade PRIMARY KEY (codigo)
);
```



Nome da  
Tabela



Chave  
primária



# CRIAÇÃO DE TABELA

---

## Modelo Lógico

cidade(codigo, nome, uf)

## Script SQL

```
CREATE TABLE cidade
(
    codigo SERIAL NOT NULL,
    nome VARCHAR(80) NOT NULL,
    uf CHAR(2) DEFAULT 'RS',

    CONSTRAINT pk_cidade PRIMARY KEY (codigo)
);
```

## OBS:

- não esqueça de declarar a chave primária
- colunas são especificadas antes das chaves
- chaves primárias obrigatoriamente são únicas

# POSTGRESQL 13: TIPOS NUMÉRICOS

Name	Storage Size	Description	Range
smallint	2 bytes	small-range integer	-32768 to +32767
integer	4 bytes	typical choice for integer	-2147483648 to +2147483647
bigint	8 bytes	large-range integer	-9223372036854775808 to +9223372036854775807
decimal	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
numeric	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
real	4 bytes	variable-precision, inexact	6 decimal digits precision
double precision	8 bytes	variable-precision, inexact	15 decimal digits precision
smallserial	2 bytes	small autoincrementing integer	1 to 32767
serial	4 bytes	autoincrementing integer	1 to 2147483647
bigserial	8 bytes	large autoincrementing integer	1 to 9223372036854775807

**Fonte:** <https://www.postgresql.org/docs/13/datatype-numeric.html>

# TIPOS MAIS COMUNS

---

- Inteiro
  - int ou integer
- Sequencial
  - serial
    - É um **integer** com uma sequência, fazendo com que o atributo receba automaticamente o valor 1 para o primeiro registro, 2 para o segundo e assim sucessivamente
- Fracionário
  - numeric(t, nr. de dígitos depois da vírgula)
  - Exemplo
    - 23.5141 → numeric(6, 4)

# TIPOS MAIS COMUNS

---

- Manipulação de strings
  - char(t) – string de tamanho t
    - Indicado para atributos de tamanho fixo, como sexo e UF
  - varchar(t) – string de tamanho **máximo** t
    - Indicado para atributos de tamanho variável, como nome e descrição

# TIPOS MAIS COMUNS

---

- Data
  - date
- Hora
  - time
- Data hora
  - timestamp

# OPÇÕES MAIS COMUNS

---

- NOT NULL → obrigatório
  - Exemplo

```
CREATE TABLE cidade
(
    codigo    SERIAL          NOT NULL,
    nome      VARCHAR(80) NOT NULL,
    uf        CHAR(2)         DEFAULT 'RS',
    CONSTRAINT PK_cidade PRIMARY KEY (codigo),
);
```

# OPÇÕES MAIS COMUNS

---

- DEFAULT default\_exp → valor padrão
  - Exemplo

```
CREATE TABLE cidade
(
    codigo    SERIAL          NOT NULL,
    nome      VARCHAR(80)     NOT NULL,
    uf        CHAR(2)         DEFAULT 'RS',
    CONSTRAINT PK_cidade PRIMARY KEY (codigo),
);
```

# OPÇÕES MAIS COMUNS

---

- **UNIQUE** → valores únicos (chave alternativa)
  - Exemplo

```
CREATE TABLE medico
(
    codigo SERIAL NOT NULL,
    nome VARCHAR(80) NOT NULL,
    email VARCHAR(80),
    CRM INT NOT NULL UNIQUE,
    cod_cid INT NOT NULL,
    CONSTRAINT pk_medico PRIMARY KEY (codigo),
    ...
);
```



# OPÇÕES MAIS COMUNS

---

- Opções podem ser combinadas
  - Exemplo

```
CREATE TABLE cidade
(
    codigo    SERIAL          NOT NULL,
    nome      VARCHAR(80)     NOT NULL,
    uf        CHAR(2)         NOT NULL DEFAULT 'RS',
    CONSTRAINT PK_cidade PRIMARY KEY (codigo),
);
```

# CHAVE PRIMÁRIA COMPOSTA

---

```
CREATE TABLE consulta
(
    data DATE NOT NULL,
    hora TIME NOT NULL,
    cod_med INT NOT NULL,
    cod_pac INT NOT NULL,
    CONSTRAINT pk_consulta
        PRIMARY KEY (data, hora, cod_med),
    CONSTRAINT fk_consulta_paciente FOREIGN KEY
        (cod_pac) REFERENCES paciente (codigo),
    CONSTRAINT fk_consulta_medico FOREIGN KEY
        (cod_med) REFERENCES medico (codigo)
);
```

# RESTRIÇÕES DE VALORES

---

```
CREATE TABLE cidade
(
    codigo SERIAL NOT NULL,
    nome VARCHAR(80) NOT NULL,
    uf CHAR(2) DEFAULT 'RS',
    CONSTRAINT PK_cidade PRIMARY KEY (codigo),
    CONSTRAINT ckc_uf CHECK (uf = 'RS' or uf = 'SC')
);
```

