

# SQL

Vinicius da Silveira Segalin

[vinicius.segalin@posgrad.ufsc](mailto:vinicius.segalin@posgrad.ufsc)

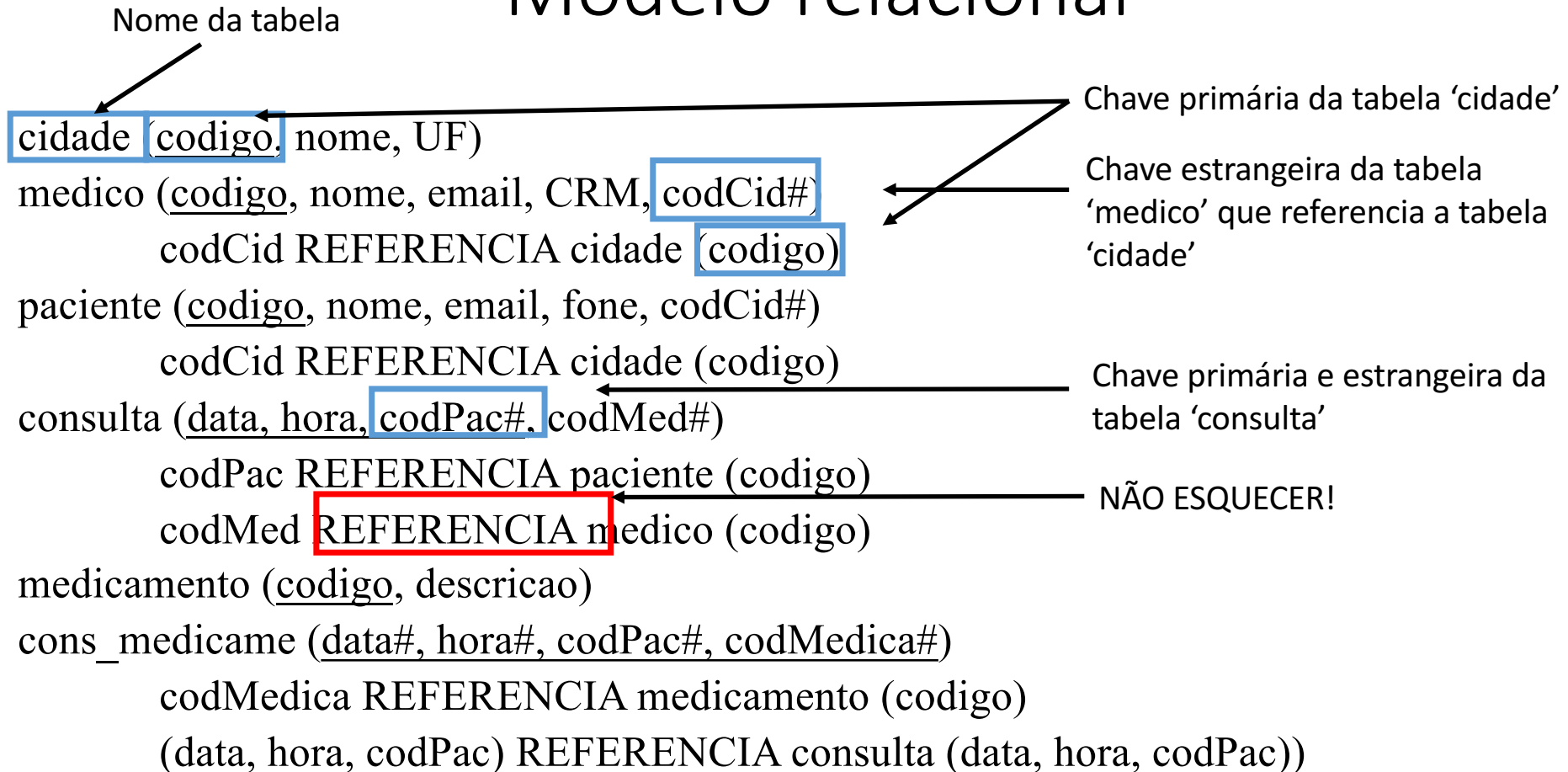
# Sumário

- Modelo relacional
- DDL
- DML
- Diferença entre SGBDs
- Dicas
- Exercícios

# Sumário

- **Modelo relacional**
- DDL
- DML
- Diferença entre SGBDs
- Dicas
- Exercícios

# Modelo relacional



# Sumário

- Modelo relacional
- **DDL**
  - **CREATE TABLE**
  - ALTER TABLE
  - DROP TABLE
- DML
- Diferença entre SGBDs
- Dicas
- Exercícios

# CREATE TABLE

- Modelo relacional:

cidade (codigo, nome UF)

- SQL

```
CREATE TABLE cidade  
(
```

```
    codigo integer NOT NULL, ← Restrição
```

```
    nome varchar (40), ← Tipo do atributo
```

```
    UF char (2)
```

```
); ← Necessário quando houver outro bloco de comandos
```

# CREATE TABLE

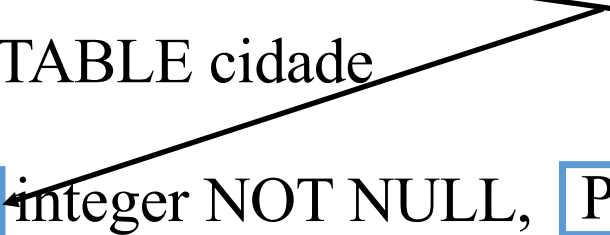
- Modelo relacional:

cidade (codigo nome UF)

- SQL

```
CREATE TABLE cidade  
(  
  codigo integer NOT NULL, PRIMARY KEY,  
  nome varchar (40),  
  UF char (2),  
  PRIMARY KEY (codigo)  
);
```

Chave primária



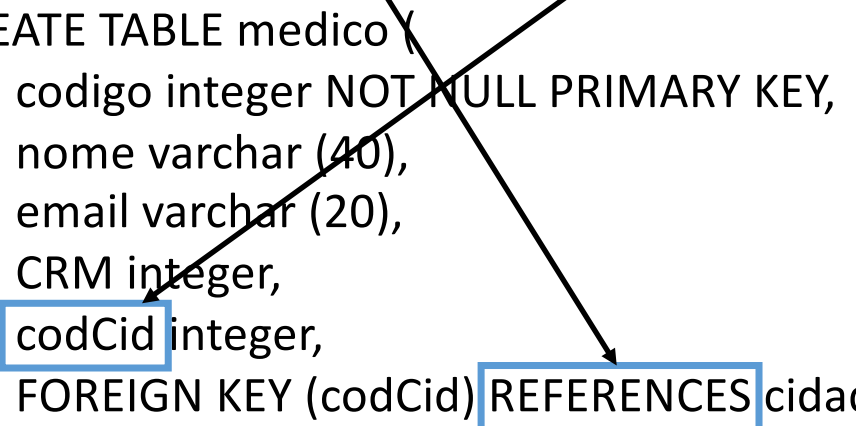
# CREATE TABLE

- Modelo relacional

medico (codigo, nome, email, CRM, **codCid#**)  
codCID **REFERENCIA** cidade (codigo)

- SQL

```
CREATE TABLE medico (  
  codigo integer NOT NULL PRIMARY KEY,  
  nome varchar (40),  
  email varchar (20),  
  CRM integer,  
  codCid integer,  
  FOREIGN KEY (codCid) REFERENCES cidade (codigo)  
);
```





# CREATE TABLE

- Modelo relacional

cons\_medicame (data#, hora#, codPac#, codMedica#)  
codMedica REFERENCIA medicamento (codigo)  
(data, hora, codPac) REFERENCIA consulta (data, hora, codPac)

- SQL

```
CREATE TABLE cons_medicame (  
  data DATE NOT NULL,  
  hora TIME NOT NULL,  
  codPac INTEGER NOT NULL, Chave primária composta  
  codMedica INTEGER NOT NULL, Chave estrangeira composta referenciando 'consulta'  
  PRIMARY KEY (data, hora, codPac, codMedica),  
  FOREIGN KEY (data, hora, codPac) REFERENCES consulta (data, hora, codPac),  
  FOREIGN KEY (codMedica) REFERENCES medicamento (codigo)  
);
```

Chave estrangeira simples referenciando 'medicamento'

# CREATE TABLE

- Constraints

- Nomes de PK e FK

```
CREATE TABLE medico (  
    codigo INTEGER NOT NULL,  
    ...  
    codCid INTEGER,  
    CONSTRAINT pk_medico PRIMARY KEY (codigo),  
    CONSTRAINT fk_cidade_medico FOREIGN KEY (codCid) REFERENCES cidade  
        (codigo) ON DELETE CASCADE  
);
```

CONSTRAINT + nome\_da\_constraint

Permite exclusão automática  
de registros filhos quando o  
registro pai for excluído

Obs: nome padrão do Postgres: {tablename}\_{columnname(s)}\_{suffix},  
ou seja, medico\_codCid\_pkey

# CREATE TABLE

- Constraints
  - Toda coluna pode ter um valor padrão

```
CREATE TABLE cidade  
(  
    codigo integer NOT NULL,  
    nome varchar (40),  
    UF char (2) DEFAULT 'SC'  
);
```

# CREATE TABLE

- Constraints
  - Colunas podem ter valores únicos sem serem PKs

```
CREATE TABLE medico
```

```
(
```

```
  codigo integer NOT NULL,
```

```
  nome varchar (40),
```

```
  email varchar (20),
```

```
  CRM integer UNIQUE,
```

```
  UNIQUE (CRM)
```

```
)
```

Obs: valores únicos, porém não referenciáveis.  
NÃO SÃO PK!

Duas formas possíveis no PostgreSQL

# CREATE TABLE

- Constraints
  - Podemos criar restrições que impedem dados incorretos de serem inseridos

```
CREATE TABLE medico
(
    codigo integer NOT NULL,
    nome varchar (40) CHECK (char_length(nome) > 10),
    email varchar (20),
    CRM integer,
    CONSTRAINT check_medico CHECK (email <> 'teste@teste.com' AND CRM > 100)
)
```

# Sumário

- Modelo relacional
- **DDL**
  - CREATE TABLE
  - **ALTER TABLE**
  - DROP TABLE
- DML
- Diferença entre SGBDs
- Dicas
- Exercícios

# ALTER TABLE

- Comando que permite alterar tabelas
  - Adicionar/remover coluna
  - Adicionar/remover constraints
  - Renomear tabela/coluna
  - Alterar tipo de coluna

# ALTER TABLE

- Adicionar/remover coluna

cidade

	<b>codigo</b> <b>integer</b>	<b>nome</b> <b>character varying(100)</b>	<b>uf</b> <b>character(2)</b>
<b>1</b>	1	Florianópolis	SC

ALTER TABLE cidade  
ADD prefeito varchar(40);

	<b>codigo</b> <b>integer</b>	<b>nome</b> <b>character varying(100)</b>	<b>uf</b> <b>character(2)</b>	<b>prefeito</b> <b>character varying(40)</b>
<b>1</b>	1	Florianópolis	SC	

ALTER TABLE cidade  
DROP prefeito;

	<b>codigo</b> <b>integer</b>	<b>nome</b> <b>character varying(100)</b>	<b>uf</b> <b>character(2)</b>
<b>1</b>	1	Florianópolis	SC

ALTER TABLE cidade  
ADD prefeito varchar (40) default 'sem prefeito'

	<b>codigo</b> <b>integer</b>	<b>nome</b> <b>character varying(100)</b>	<b>uf</b> <b>character(2)</b>	<b>prefeito</b> <b>character varying(40)</b>
<b>1</b>	1	Florianópolis	SC	sem prefeito



# ALTER TABLE

- Adicionar/remover constraint

```
ALTER TABLE paciente  
ADD CONSTRAINT pk_paciente PRIMARY KEY (codigo);
```

```
ALTER TABLE paciente  
DROP pk_paciente;
```

```
ALTER TABLE medico  
ADD CONSTRAINT pk_cidade_medico  
FOREIGN KEY (codCid) REFERENCES cidade(codigo);
```

```
ALTER TABLE medico  
DROP pk_cidade_medico;
```


```
ALTER TABLE cidade  
ALTER UF SET DEFAULT 'RS';
```

```
ALTER TABLE cidade  
ALTER UF DROP DEFAULT;
```

# ALTER TABLE

- Renomear tabela/coluna
  - ALTER TABLE cidade RENAME TO cidades;
  - ALTER TABLE cidade RENAME UF TO estado;
- Alterar tipo de coluna
  - ALTER TABLE cidade  
ALTER COLUMN nome TYPE varchar (200);

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC



	<b>codigo</b> integer	<b>nome</b> character varying(200)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC

Cuidado com compatibilidade de tipos!

# Sumário

- Modelo relacional
- **DDL**
  - CREATE TABLE
  - ALTER TABLE
  - **DROP TABLE**
- DML
- Diferença entre SGBDs
- Dicas
- Exercícios

# DROP TABLE

- Comando que exclui toda a tabela e seu conteúdo

```
DROP TABLE medico;  
DROP TABLE cidade;
```

← Primeiro tabelas filhas, depois a tabela pai

```
DROP TABLE cidade CASCADE;
```

← Com o comando CASCADE, todas as referências à tabela pai são removidas

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>email</b> character varying(100)	<b>crm</b> integer	<b>codcid</b> integer
<b>1</b>	1	João	joao@gmail.com	123	1

← Não referencia ninguém!

# Sumário

- Modelo relacional
- DDL
- **DML**
  - **Insert**
  - Update
  - Delete
- Diferença entre SGBDs
- Dicas
- Exercícios

# INSERT

- Inserção de dados nas tabelas
- Exemplo: cidade (codigo, nome, uf)
- Sem informar os atributos
  - **INSERT INTO** cidade **VALUES** (1, 'Florianópolis', 'SC');
  - **INSERT INTO** cidade **VALUES** (2, 'São José');
- Informando os atributos
  - **INSERT INTO** cidade (codigo, nome, uf) **VALUES** (3, 'Porto Alegre', 'RS');
  - **INSERT INTO** cidade (codigo, uf) **VALUES** (4, 'RS');

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC
<b>2</b>	2	São José	SC
<b>3</b>	3	Porto Alegre	RS
<b>4</b>	4		RS

Possui valor default

Não possui valor default e permite nulo

# INSERT

- Inserção em tabelas com chave estrangeira
- Exemplo: medico (codigo, nome, email, CRM, codCid#)  
codCid REFERENCIA cidade (codigo)

```
INSERT INTO medico (codigo, nome, email, codCid) VALUES (1, 'João', 'joao@gmail.com', 1) ;
```

```
INSERT INTO medico (codigo, nome, email) VALUES (2, 'Maria', 'maria@gmail.com');
```

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>email</b> character varying(100)	<b>crm</b> integer	<b>codcid</b> integer
<b>1</b>	1	João	joao@gmail.com		1
<b>2</b>	2	Maria	maria@gmail.com		

# INSERT

- Inserção em tabelas com chave estrangeira
- Valor da chave estrangeira já deve existir na tabela referenciada!

	codigo integer	nome character varying(100)	uf character(2)
1	1	Florianópolis	SC
2	2	São José	SC
3	3	Porto Alegre	RS
4	4		RS

INSERT INTO medico (codigo, nome, codCid) VALUES (3, 'José', 5);

```
ERROR: insert or update on table "medico" violates foreign key constraint "medico_codcid_fkey"  
DETAIL: Key (codcid)=(5) is not present in table "cidade".
```

```
***** Error *****
```

```
ERROR: insert or update on table "medico" violates foreign key constraint "medico_codcid_fkey"  
SQL state: 23503  
Detail: Key (codcid)=(5) is not present in table "cidade".
```



# Sumário

- Modelo relacional
- DDL
- **DML**
  - Insert
  - **Update**
  - Delete
- Diferença entre SGBDs
- Dicas
- Exercícios

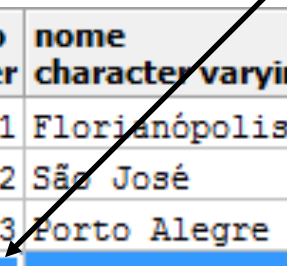
# UPDATE

- Alteração de dados já existentes nas tabelas

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
1	1	Florianópolis	SC
2	2	São José	SC
3	3	Porto Alegre	RS
4	4		RS

**UPDATE** cidade  
**SET** nome = 'Caxias do Sul'  
**WHERE** codigo = 4;

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
1	1	Florianópolis	SC
2	2	São José	SC
3	3	Porto Alegre	RS
4	4	Caxias do Sul	RS



# UPDATE

- Alteração de dados já existentes nas tabelas

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC
<b>2</b>	2	São José	SC
<b>3</b>	3	Porto Alegre	RS
<b>4</b>	4		RS

**UPDATE** cidade  
**SET** nome = 'Caxias do Sul';

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Caxias do Sul	SC
<b>2</b>	2	Caxias do Sul	SC
<b>3</b>	3	Caxias do Sul	RS
<b>4</b>	4	Caxias do Sul	RS

# Sumário

- Modelo relacional
- DDL
- **DML**
  - Insert
  - Update
  - **Delete**
- Diferença entre SGBDs
- Dicas
- Exercícios

# DELETE

- Remoção de dados nas tabelas

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC
<b>2</b>	2	São José	SC
<b>3</b>	3	Porto Alegre	RS
<b>4</b>	4	Caxias do Sul	RS

DELETE FROM cidade  
WHERE codigo = 4;

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC
<b>2</b>	2	São José	SC
<b>3</b>	3	Porto Alegre	RS

# DELETE

- Remoção de dados nas tabelas

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC
<b>2</b>	2	São José	SC
<b>3</b>	3	Porto Alegre	RS

DELETE FROM cidade;

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
--	--------------------------	---------------------------------------	---------------------------

# DELETE

- Remoção de dados referenciados

cidade

	codigo integer	nome character varying(100)	uf character(2)
1	1	Florianópolis	SC

	codigo integer	nome character varying(100)	email character varying(100)
1	1	João	joao@gmail.com

Primeiro excluir registros da  
tabela filha, para depois  
excluir registros da tabela pai

DELETE FROM medico WHERE codigo = 1;  
DELETE FROM cidade WHERE codigo = 1;

OU

DELETE FROM cidade WHERE codigo = 1;

Se a FK tiver sido criada com o comando  
'ON DELETE CASCADE'

```
CREATE TABLE medico (  
  codigo INTEGER NOT NULL,  
  ...  
  codCid INTEGER,  
  CONSTRAINT fk_cidade_medico  
    FOREIGN KEY (codCid) REFERENCES  
      cidade (codigo) ON DELETE CASCADE  
);
```

# Sumário

- Modelo relacional
- DDL
- DML
- **Diferença entre SGBDs**
- Dicas
- Exercícios



# Diferença entre SGBDs

- Tipos de dados
  - Boolean
    - PostgreSQL: Boolean
    - Microsoft SQL Server: Bit
  - Float
    - PostgreSQL: Numeric
    - Microsoft SQL Server: Float, Real
- Limitando consultas
  - PostgreSQL: `select * from tabela limit 10;`
  - Microsoft SQL Server: `select top 10 from tabela;`
- Diversas outras...

# Sumário

- Modelo relacional
- DDL
- DML
- Diferença entre SGBDs
- **Dicas**
- Exercícios

# Dicas

- Ordem na criação de tabelas
  - Tabelas referenciadas devem ser criadas antes
  - Exemplo

~~medico (codigo, nome, email, CRM, codCid#)  
          codCid REFERENCIA cidade (codigo)  
cidade (codigo, nome, UF)~~

cidade (codigo, nome, UF)  
medico (codigo, nome, email, CRM, codCid#)  
          codCid REFERENCIA cidade (codigo)



# Dicas

- Dica na criação de tabelas
  - Criar todas as tabelas sem chaves estrangeiras

```
CREATE TABLE tabela  
(  
    ...  
);
```

- Alterar as tabelas que possuem chaves estrangeiras, inserindo-as

```
ALTER TABLE tabela  
ADD CONSTRAINT fk_destino_fonte  
FOREIGN KEY (colunas_tabela_fonte)  
REFERENCES tabela_destino (colunas_tabela_destino);
```

# Dicas

- Comentar código
  - Comentário de uma linha: --

```
-- CREATE TABLE cidade...
```

- Comentário de um bloco: /\* ... \*/

```
/* CREATE TABLE cidade  
(  
    codigo integer,  
    ....  
);  
*/
```

# Dicas

- Utilize transações
  - Se tudo OK -> commit (salva as alterações)
  - Se ocorreu algum erro -> rollback (volta ao estado anterior)

```
BEGIN TRANSACTION;  
insert into cidade values (2, 'São José', 'SC');  
COMMIT TRANSACTION;
```

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC
<b>2</b>	2	São José	SC

```
BEGIN TRANSACTION;  
insert into cidade values (3, 'Porto Alegre', 'RS');  
ROLLBACK TRANSACTION;
```

	<b>codigo</b> integer	<b>nome</b> character varying(100)	<b>uf</b> character(2)
<b>1</b>	1	Florianópolis	SC
<b>2</b>	2	São José	SC

# Sumário

- Modelo relacional
- DDL
- DML
- Diferença entre SGBDs
- Dicas
- **Exercícios**