

SQL

Restrições de integridade

INE5613 – Banco de Dados I

Carina F. Dorneles
dorneles@inf.ufsc.br

Restrições de Integridade

- ▶ Manutenção da **consistência da base de dados** através da validação de restrições definidas sobre os dados
- ▶ **Controle de valores** que um campo pode ter
 - ▶ Por ex.: O número de assentos de um voo só pode variar entre 100 e 700
- ▶ **Controle de relacionamento** entre tabelas
 - ▶ Por ex.: Um voo só pode pertencer a uma Cia já cadastrada



Restrições de Integridade

▶ Restrições

- ▶ De **domínio**: são os tipos associados às colunas (integer, char, varchar ...)
- ▶ De **chave primária**: estabelece que um dado atributo de uma tabela não pode ter valor duplicado
- ▶ Valores **NULL**: indica se um atributo pode ou não ter valor NULL
- ▶ De **integridade**
 - ▶ Integridade de **entidade**: valor da PK não pode ser NULL
 - ▶ Integridade **referencial**: o valor de uma FK deve ser o valor de uma PK
- ▶ De **valores**
 - ▶ Delimita os valores permitidos para o valor de uma coluna
- ▶ **Regras de negócio**
 - ▶ São restrições aplicadas aos **valores** dos atributos e que normalmente estão relacionados ao **domínio de problema**



Restrições de domínio, PK, NULL, integridade e de valores

Restrições já vistas...

- ▶ Restrição de domínio
 - ▶ Determina o conjunto de valores permitidos (o tipo da coluna)
- ▶ Restrição de “Chave primária” (PK) – PRIMARY KEY
 - ▶ Determina que um atributo de uma tabela não pode ter valor duplicado, nem ser nulo e que pode ser referenciado por outra tabela
- ▶ Restrição de valores NULL
 - ▶ Permite, ou não, que um atributo de uma tabela possua valor nulo (ausência de valor)
- ▶ Integridade de entidade: valor da PK não pode ser NULL

▶ Exemplo

```
CREATE TABLE pessoa
(
  codigo INT NOT NULL,
  nome VARCHAR(40) NOT NULL,
  PRIMARY KEY (codigo)
)
```

Check

- ▶ Regra “Check”

- ▶ Define uma condição que deve ser satisfeita para que a coluna possa ser incluída ou alterada numa linha da tabela.

- ▶ Exemplo:

```
CREATE TABLE pessoa
(  
  codigo integer NOT NULL,  
  nome varchar(40) NOT NULL,  
  codCidade integer,  
  dtaIngresso date,  
  PRIMARY KEY (codigo),  
  FOREIGN KEY (codCidade) REFERENCES Cidade (codigo),  
  CHECK (dtaIngresso <= now())  
)
```

*now(): Pega data corrente no Postgresq.
-- Depende do SGBD*



Restrições de integridade referencial

- ▶ Considerada uma das mais importantes
 - ▶ A referência de uma FK para uma PK deve ser **íntegra**



Restrições de integridade referencial

- ▶ Considerada uma das mais importantes
 - ▶ A referência de uma FK para uma PK deve ser **íntegra**
 - ▶ Determina que:

**valores de atributos da “chave estrangeira”
possuem obrigatoriamente:**

1. correspondentes em
chaves primárias da tabela referenciada

2. valor **NULL**


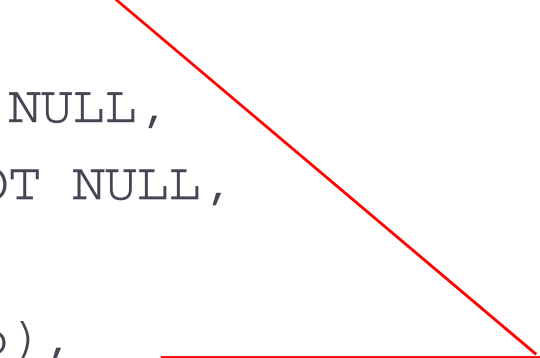


Restrições de integridade referencial

► Exemplo:

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

```
CREATE TABLE pessoa  
(codigo integer NOT NULL,  
  nome varchar(40) NOT NULL,  
  codCidade integer,  
PRIMARY KEY (codigo),  
FOREIGN KEY (codCidade) REFERENCES Cidade (codigo))
```



Restrições de integridade referencial

► Exemplo:

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

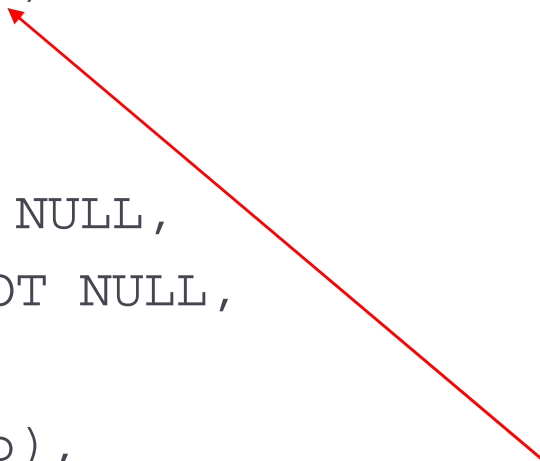
```
CREATE TABLE pessoa  
(codigo integer NOT NULL,  
  nome varchar(40) NOT NULL,  
  codCidade integer,  
PRIMARY KEY (codigo),  
FOREIGN KEY (codCidade) REFERENCES Cidade (codigo))
```

Pessoa

codigo	nome	codCidade
1	Beto	1
2	Ana	2

Cidade

codigo	nome
1	São Paulo
3	Rio de Janeiro
4	Londrina



Restrições de integridade referencial

► Exemplo:

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

```
CREATE TABLE pessoa  
(codigo integer NOT NULL,  
  nome varchar(40) NOT NULL,  
  codCidade integer,  
  PRIMARY KEY (codigo),  
  FOREIGN KEY (codCidade) REFERENCES Cidade (codigo))
```

Não obedece a regra de
Integridade Referencial

Pessoa

<i>codigo</i>	<i>nome</i>	<i>codCidade</i>
1	Beto	1
2	Ana	2

Cidade

<i>codigo</i>	<i>nome</i>
1	São Paulo
3	Rio de Janeiro
4	Londrina



Como garantir ??

Pessoa

<i>codigo</i>	<i>nome</i>	<i>codCidade</i>
1	Beto	10
2	Ana	12

Cidade

<i>codigo</i>	<i>nome</i>
10	São Paulo
12	Rio de Janeiro



Como garantir ??

Pessoa

codigo	nome	codCidade
1	Beto	10
2	Ana	12

Cidade

codigo	nome
10	São Paulo
12	Rio de Janeiro

- E se a cidade “Rio de Janeiro” for deletada do BD?



Como garantir ??

<i>Pessoa</i>		
<i>codigo</i>	<i>nome</i>	<i>codCidade</i>
1	Beto	10
2	Ana	12

<i>Cidade</i>	
<i>codigo</i>	<i>nome</i>
10	São Paulo
12	Rio de Janeiro

- E se a cidade “Rio de Janeiro” for deletada do BD?

Lembrem:

valores de atributos da “chave estrangeira” possuem obrigatoriamente:

1. correspondentes em **chaves primárias da tabela referenciada**
2. valor **NULL**

Como garantir ??

<i>Pessoa</i>			<i>Cidade</i>	
<i>codigo</i>	<i>nome</i>	<i>codCidade</i>	<i>codigo</i>	<i>nome</i>
1	Beto	10	10	São Paulo
2	Ana	12	12	Rio de Janeiro

- ▶ E se a cidade “Rio de Janeiro” for deletada do BD?
 - ▶ Valor de `codCidade` deve receber NULL

Ou

- ▶ Não podemos permitir a deleção de “Rio de Janeiro”



Como garantir ??

- ▶ SQL permite o gerenciamento deste comportamento em dois casos:

- ▶ Ao **apagar** a PK referenciada

- ▶ ON DELETE

- ▶ Ao **alterar** a PK referenciada

- ▶ ON UPDATE



Deleção e Alteração de PK

- ▶ ON DELETE
- ▶ ON UPDATE
- ▶ Ações
 - ▶ Default:
 - ▶ não permite as operações de DELETE ou UPDATE em PK
 - ▶ SET NULL
 - ▶ Quando uma PK é deletada ou alterada, seta as FK para NULL
 - ▶ CASCADE
 - ▶ Quando uma PK é deletada ou alterada, apaga ou altera as FK



Exemplo

Pessoa

codigo	nome	codCidade
1	Beto	10
2	Ana	12

Cidade

codigo	nome
10	São Paulo
12	Rio de Janeiro

▶ ON DELETE CASCADE

- ▶ Deletou cidade, deleta a linha que a referencia



Exemplo

Pessoa

codigo	nome	codCidade
1	Beto	10
2	Ana	NULL

Cidade

codigo	nome
10	São Paulo
12	Rio de Janeiro

► ON DELETE SET NULL

- Deletou cidade, seta NULL para a FK que a referencia



Exemplo

Pessoa

codigo	nome	codCidade
1	Beto	10
2	Ana	121

Cidade

codigo	nome
10	São Paulo
121	Rio de Janeiro

► ON UPDATE CASCADE

- Alterou PK de cidade, altera a FK que a referencia



Combinações possíveis

- ▶ ON DELETE SET NULL
- ▶ ON DELETE CASCADE
- ▶ ON DELETE RESTRICT (*default* , não precisa especificar)

- ▶ ON UPDATE SET NULL
- ▶ ON UPDATE CASCADE
- ▶ ON UPDATE RESTRICT



Exemplo em SQL

- Supondo que a tabela cidade já tenha sido criada

```
CREATE TABLE pessoa
(codigo integer NOT NULL,
 nome varchar(40) NOT NULL,
 codCidade integer,
 Codprof integer,
 PRIMARY KEY (codigo),
 FOREIGN KEY (codCidade) REFERENCES Cidade (codigo)
   ON DELETE SET NULL /* seta null para a coluna codCidade */
   ON UPDATE CASCADE, /* altera o valor da coluna codCidade */
 FOREIGN KEY (codProf) REFERENCES profissao (codigo)
   ON DELETE SET NULL
   ON UPDATE SET NULL /* seta null para a coluna codCidade */)
```

```
CREATE TABLE empresa
(codigo integer NOT NULL,
 nome varchar(40) NOT NULL,
 codCidade integer,
 PRIMARY KEY (codigo),
 FOREIGN KEY (codCidade) REFERENCES Cidade (codigo)
   ON DELETE CASCADE /* deleta a linha que possui a referência */
   ON UPDATE CASCADE)
```



Exercícios

