



AS33C

Banco de Dados 2

Professor: Eduardo Cotrin Teixeira



cotrin@utfpr.edu.br



SQL/DDDL-Restrições de tabela

- Restrição de chave primária:

PRIMARY KEY (nome-coluna,[nome-coluna],...)

- Como *restrição de tabela*, define uma ou mais colunas que compõem a chave primária da tabela.

* Se a chave primária tiver uma única coluna, a cláusula pode aparecer como uma *restrição de coluna* na definição da tabela.



SQL/DDDL-Restrições de tabela

■ Exemplos:

```
CREATE TABLE FUNCIONARIO (  
    Nome VARCHAR(50) NOT NULL,  
    CPF CHAR(11) PRIMARY KEY, --restrição de coluna  
    Salario DECIMAL(10,2) ,  
    Idade INT);
```

```
CREATE TABLE ALUNO (  
    RA CHAR(7) ,  
    Campus VARCHAR(30) ,  
    Nome VARCHAR (50) NOT NULL,  
    PRIMARY KEY (RA,Campus) ); --restrição de tabela
```

** Considerando que alunos podem ter RA iguais em campi diferentes.*



SQL/DDDL-Restrições de tabela

Restrição de chave estrangeira:

- **FOREIGN KEY** (nome-coluna1) **REFERENCES** nome-tabela (nome-coluna2) [ação-integridade]

Exemplo:

```
FOREIGN KEY (CPF_ger) REFERENCES FUNCIONARIO (CPF) ON  
DELETE RESTRICT ON UPDATE CASCADE;
```

- (nome-coluna1) = coluna que contém chaves estrangeiras.
- REFERENCES nome-tabela (nome-coluna2) = tabela e nome da coluna onde o valor é chave primária.



SQL/DDDL-Restrições de tabela

- Pode ser usado **CONSTRAINT** para nomear as restrições de tabela (facilita localização de erros e alterações).
- Se não for dado nome a uma restrição (como chave primária ou estrangeira, por exemplo), o PostgreSQL define um nome padrão.

SQL/DDDL-Restrições de tabela

■ Exemplos:

```
CREATE TABLE Departamento(  
  Num INT PRIMARY KEY,  
  Nome VARCHAR(30),  
  CPF_ger CHAR(11) NOT NULL,  
  FOREIGN KEY (CPF_ger) REFERENCES FUNCIONARIO (CPF) );
```

▼ ▶◀ Constraints (2)

🔑 departamento_cpf_ger_fkey

🔑 departamento_pkey

SQL/DDDL-Restrições de tabela

■ Exemplos:

```
CREATE TABLE Departamento(  
    Num INT,  
    Nome VARCHAR(30),  
    CPF_ger CHAR(11) NOT NULL,  
    CONSTRAINT dep_pk PRIMARY KEY (Num),  
    CONSTRAINT dep_fk_cpfger FOREIGN KEY (CPF_ger)  
        REFERENCES FUNCIONARIO (CPF) );
```

▼ ▶◀ Constraints (2)

🔑 dep_fk_cpfger

🔑 dep_pk



SQL/DDDL-Restrições de tabela

- **FOREIGN KEY** (nome-coluna1) **REFERENCES** nome-tabela (nome-coluna2) [ação-integridade]

Exemplo:

```
FOREIGN KEY (CPF_ger) REFERENCES FUNCIONARIO (CPF) ON  
DELETE RESTRICT ON UPDATE CASCADE;
```

- [ação-integridade] = **ON DELETE** (para agir em ações de exclusão) e/ou **ON UPDATE** (para agir em ações de atualização) seguidos de:
 - **RESTRICT** (padrão) – rejeita a operação que viola a integridade.
 - **SET NULL** – atribui NULL à chave estrangeira, que fica sem "referência".
 - **CASCADE** – propaga a alteração feita na chave referenciada para as linhas que a referenciam.



SQL/DDDL - Exemplo

```
CREATE TABLE Departamento(  
    Num INT PRIMARY KEY,  
    Nome VARCHAR(30),  
    CPF_ger CHAR(11) NOT NULL,  
    FOREIGN KEY(CPF_ger) REFERENCES FUNCIONARIO(CPF)  
    ON DELETE RESTRICT ON UPDATE CASCADE );
```

Não permite apagar um
funcionário que seja gerente
de algum departamento.

```
CREATE TABLE Departamento(  
    Num INT PRIMARY KEY,  
    Nome VARCHAR(30),  
    CPF_ger CHAR(11) NOT NULL,  
    FOREIGN KEY(CPF_ger) REFERENCES FUNCIONARIO(CPF)  
    ON DELETE CASCADE ON UPDATE CASCADE );
```

Exclui o departamento de um
gerente excluído.

DROP TABLE

- Objetivo: Apagar a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão excluídos todos dados, estrutura e índices de acesso que estejam associados à tabela.

DROP TABLE <nome-tabela> [CASCADE | RESTRICT];

- Com CASCADE: São removidos todos os objetos em outras tabelas que dependem da tabela removida (*views*, restrições de chave estrangeira, etc.).
- Com RESTRICT (padrão): Tabela só é removida se não houver no BD nenhum objeto que dependa dela.
- **Exemplo:** DROP TABLE FUNCIONARIO CASCADE; --Remove todas **as restrições** de outras tabelas que referenciam funcionários.



Testando o exemplo

```
CREATE TABLE FUNCIONARIO (  
    Nome VARCHAR(50) NOT NULL,  
    CPF CHAR(11) PRIMARY KEY,  
    Salario DECIMAL(10,2) ,  
    Idade INT) ;
```

```
CREATE TABLE Departamento(  
    Num INT PRIMARY KEY,  
    Nome VARCHAR(30) ,  
    CPF_ger CHAR(11) NOT NULL,  
    FOREIGN KEY(CPF_ger) REFERENCES FUNCIONARIO(CPF)  
ON DELETE RESTRICT ON UPDATE CASCADE ) ;
```



Testando o exemplo

```
INSERT INTO Funcionario (Nome,CPF,Salario, Idade)
VALUES ('Paulo','11111111111',3500, 32);

INSERT INTO Departamento (Num,Nome,CPF_ger) VALUES
(1,'Computação','11111111111');

DELETE FROM Funcionario WHERE CPF = '11111111111';
--ON DELETE RESTRICT impede a exclusão

UPDATE Funcionario SET CPF='22222222222' WHERE Nome =
'Paulo';
--ON UPDATE CASCADE propaga a alteração

SELECT * FROM Departamento;
```



Testando o exemplo

```
DROP TABLE Funcionario; --não funciona
```

```
DROP TABLE Funcionario CASCADE --apaga constraint em  
Departamento
```

```
DROP TABLE Departamento;
```



Testando o exemplo

--criando novamente as tabelas com CASCADE no DELETE

```
CREATE TABLE FUNCIONARIO (  
    Nome VARCHAR(50) NOT NULL,  
    CPF CHAR(11) PRIMARY KEY,  
    Salario DECIMAL(10,2) ,  
    Idade INT) ;
```

```
CREATE TABLE Departamento(  
    Num INT PRIMARY KEY,  
    Nome VARCHAR(30) ,  
    CPF_ger CHAR(11) NOT NULL,  
    FOREIGN KEY(CPF_ger) REFERENCES FUNCIONARIO(CPF)  
    ON DELETE CASCADE ON UPDATE CASCADE ) ;
```



Testando o exemplo

```
INSERT INTO Funcionario (Nome,CPF,Salario, Idade)
VALUES ('Paulo','11111111111',3500, 32);
INSERT INTO Departamento (Num,Nome,CPF_ger) VALUES
(1,'Computação','11111111111');
SELECT * FROM Departamento;

DELETE FROM Funcionario WHERE CPF = '11111111111';
--ON DELETE CASCADE propaga a exclusão
SELECT * FROM Departamento;
```

ALTER TABLE

- Objetivo: Altera a estrutura de uma tabela. Pode ser usado de várias formas, as principais são descritas a seguir.

- Mudar o nome da tabela:

ALTER TABLE <nome-tabela> **RENAME TABLE** <novo-nome-tabela>;

ALTER TABLE Funcionario **RENAME TABLE** Fun_geral;

- Mudar o nome de uma coluna:

ALTER TABLE <nome-tabela> **RENAME** <nome-coluna> **TO** <novo-nome-coluna>;

ALTER TABLE Funcionario **RENAME** CPF_ger **TO** CPF_gerente;

- Adicionar uma nova coluna:

ALTER TABLE <nome-tabela> **ADD COLUMN** <nome-coluna> <tipo-dado> [restrições-coluna];

ALTER TABLE Funcionario **ADD COLUMN** Endereco
VARCHAR(50) NOT NULL;

- Remover uma coluna :

ALTER TABLE <nome-tabela> **DROP COLUMN** <nome-coluna>
[RESTRICT | CASCADE];

ALTER TABLE Funcionario **DROP COLUMN** CPF **CASCADE;**

** RESTRICT e CASCADE funcionam como na restrição de tabela.*

- Definir uma nova cláusula DEFAULT para uma coluna:

ALTER TABLE <nome-tabela> **ALTER COLUMN** <nome-coluna>
SET DEFAULT Valor;
ALTER TABLE Funcionario **ALTER COLUMN** Idade SET
DEFAULT 18;

- Remover a cláusula DEFAULT de uma coluna:

ALTER TABLE <nome-tabela> **ALTER COLUMN** <nome-coluna>
DROP DEFAULT;
ALTER TABLE Funcionario **ALTER COLUMN** Idade DROP
DEFAULT;

- Adicionar restrição de chave primária a uma coluna:

ALTER TABLE <nome-tabela> **ADD PRIMARY KEY** <nome-coluna>;

ALTER TABLE Funcionario **ADD PRIMARY KEY** CPF;

- Adicionar restrição de chave estrangeira a uma coluna:

ALTER TABLE <nome-tabela> **ADD FOREIGN KEY** (nome-coluna)
REFERENCES <nome-tabela> (nome-coluna) [ação-integridade];

ALTER TABLE Departamento **ADD FOREIGN KEY**
(CPF_gerente) **REFERENCES** FUNCIONARIO(CPF) **ON UPDATE**
CASCADE;

***DICA IMPORTANTE:** Crie todas as tabelas sem restrições de chave, depois altere-as para incluir as restrições. Dessa forma, a ordem da criação das tabelas não importa.

- Definir uma nova restrição nomeada:

ALTER TABLE <nome-tabela> **ADD CONSTRAINT** [nome-restrição]
restrição;

ALTER TABLE Departamento **ADD CONSTRAINT**
departamento_fk **FOREIGN KEY**(CPF_gerente) **REFERENCES**
FUNCIONARIO(CPF) **ON DELETE RESTRICT ON UPDATE CASCADE**;

*** Lembrando que se não for dado um nome a uma restrição, o PostgreSQL define um nome padrão.**

- Remover uma restrição nomeada:

ALTER TABLE <nome-tabela> **DROP CONSTRAINT** <nome-restrição>;
ALTER TABLE Departamento **DROP CONSTRAINT** depto_fk;

*** Para exclusão de uma restrição só é possível usar esse comando, com o nome da restrição.**



Testando os exemplos

```
CREATE TABLE DEPARTAMENTO (  
    DNumero INT,  
    DNome VARCHAR(30) NOT NULL,  
    CPF_gerente CHAR(11) NOT NULL,  
    CONSTRAINT departamento_pk PRIMARY KEY(DNumero) ,  
    CONSTRAINT departamento_fk FOREIGN  
KEY(CPF_gerente) REFERENCES FUNCIONARIO(CPF) ON DELETE  
RESTRICT ON UPDATE CASCADE);  
  
ALTER TABLE Departamento DROP CONSTRAINT departamento_fk;  
ALTER TABLE Departamento ADD CONSTRAINT departamento_fk  
FOREIGN KEY(CPF_gerente) REFERENCES FUNCIONARIO(CPF) ON  
DELETE RESTRICT ON UPDATE CASCADE;
```