



# Banco de Dados

## Restrições de Integridade (RI's)

**Prof. Fernando Rodrigues de Almeida Júnior**

**UFC – Universidade Federal do Ceará**

Curso: Eng. da Computação

e-mail: [fernandorodrigues@sobral.ufc.br](mailto:fernandorodrigues@sobral.ufc.br)

# 7. Restrições de Integridade

## - Conceitos Básicos -

- ❑ Restrições de Integridade em um BD Relacional
  - ↳ Restrição de domínio
  - ↳ Restrição de entidade
  - ↳ Restrição de chave
  - ↳ Restrição de integridade referencial
- ❑ Especificação de restrições de integridade
  - ↳ Não procedimental
  - ↳ Procedimental
- ❑ Especificação não procedimental de restrições
  - ↳ Através de expressões da DDL SQL
  - ↳ Impedem qualquer alteração no BD que violem as restrições

# 7. Restrições de Integridade

## - Restrição de Domínio -

### ❑ Especificação não procedimental de restrições (cont.)

#### ↳ Restrição de domínio

⇒ O valor de cada atributo A

⇒ Tem que ser um valor atômico de  $\text{dom}(A)$

⇒ Exemplo

```
create table Empregado  
(matr integer not null,  
  salário dec(9,2) not null
```

⋮

```
constraint c1  
  check (matr > 0 and matr <= 99999)
```

```
constraint c2  
  check (salário > 998,00)
```

**Observação:** O MySQL NÃO implementa as cláusulas “CREATE ASSERTION” <Constraint name> e “CHECK”.

O manual do MySQL diz:

“The CHECK clause is parsed but **ignored** by all storage engines”

# 7. Restrições de Integridade

## - Restrição de Domínio -

- ❑ O MySQL implementa o tipo “ENUM”:
  - ➡ ENUM é um objeto string com um valor escolhido de uma lista de valores permitidos;
  - ➡ Esta lista de valores permitidos é enumerada explicitamente na especificação do atributo no momento da criação da tabela.

### ❑ Exemplo:

```
CREATE TABLE shirts (  
    name VARCHAR(40),  
    size ENUM('x-small', 'small', 'medium', 'large', 'x-large')  
);  
INSERT INTO shirts (name, size) VALUES ('dress shirt','large'), ('t-shirt','medium'),  
    ('polo shirt','small');  
SELECT name, size FROM shirts WHERE size = 'medium';  
+-----+-----+  
| name   | size   |  
+-----+-----+  
| t-shirt | medium |  
+-----+-----+  
UPDATE shirts SET size = 'small' WHERE size = 'large';  
COMMIT;
```

# 7. Restrições de Integridade

## - Restrição de Chave -

### ❑ Especificação não procedimental de restrições (cont.)

#### ↳ Restrição de chave

⇒ **Tuplas de uma relação têm que ser distintas entre si**

⇒ Duas tuplas em uma relação não podem ter a mesma combinação de valores para seus atributos

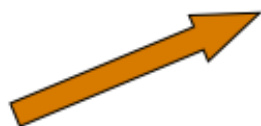
⇒ Formas de especificação não procedimental

⇒ **Primary key**

⇒ Unique key

⇒ Exemplo

```
Create table Departamento  
(cod_depart integer not null,  
  nome        varchar(30) not null,  
  ender       varchar(30),  
  primary key (cod_depart))
```



# 7. Restrições de Integridade

## - Restrição de Chave -

- ❑ Especificação não procedimental de restrições (cont.)
  - ↳ Restrição de integridade referencial
    - ⇒ Um conjunto de atributos FK de uma tabela R é **chave estrangeira (foreign key)** se
      - ⇒ Os atributos em FK têm o mesmo domínio que a chave primária PK de uma tabela S, e
      - ⇒ Um valor de FK em uma tupla  $t_1$  de R
        - ♦ Ou ocorre em como valor de PK para uma tupla  $t_2$  em S
$$t_1[\text{FK}] = t_2[\text{PK}]$$
        - ♦ ou é nulo
    - ⇒ Forma de especificação  
[CONSTRAINT nome-restrição]  
FOREIGN KEY (nome-coluna {, nome-coluna})  
REFERENCES nome-tabela [ON DELETE CASCADE |  
SET NULL | NO ACTION ],  
[ON UPDATE CASCADE],



# Banco de Dados

## Criando e Gerenciando Usuários no MySQL

**Prof. Fernando Rodrigues de Almeida Júnior**

**UFC – Universidade Federal do Ceará**

Curso: Eng. da Computação

e-mail: [fernandorodrigues@sobral.ufc.br](mailto:fernandorodrigues@sobral.ufc.br)

# 7. Restrições de Integridade

## - Gerenciamento de Acesso -

### ❑ Criando um usuário [Forma Geral]:

> mysql --user=nome\_usuario --password=senha\_usuario nome\_bd

- ou no modo resumido:

> mysql -u nome\_usuario -psenha\_usuario nome\_bd

- Se você omitir o valor da senha após a opção --password ou -p, o cliente irá solicitar uma senha.
- Se for usada a opção -p, não deve haver nenhum espaço em branco entre -p e o valor da senha.

Exemplos:

> mysql --user=fernando --password=minha@senha meuBanco  
ou

> mysql -u fernando -pminha@senha meuBanco



## 7. Restrições de Integridade

### - Gerenciamento de Acesso -

#### ❑ Criando novos usuários:

- Conectando como usuário *root*:

```
>mysql --user=root mysql
```

- Se houver uma senha para o root, a mesma deve ser informada na opção '--password' ou '-p'
- Depois de conectar como root, você poderá adicionar novas contas;
- Deve-se usar as sentenças CREATE USER e GRANT para criar e configurar novas contas.

## 7. Restrições de Integridade

### - Gerenciamento de Acesso -

#### ❑ Criando novos usuários:

Os exemplos a seguir usam as sentenças CREATE USER e GRANT para criar e configurar quatro novas contas:

```
mysql> CREATE USER 'monty'@'localhost' IDENTIFIED BY 'some_pass';  
mysql> GRANT ALL PRIVILEGES ON *.* TO 'monty'@'localhost'  
      -> WITH GRANT OPTION;  
mysql> CREATE USER 'monty'@'%' IDENTIFIED BY 'some_pass';  
mysql> GRANT ALL PRIVILEGES ON *.* TO 'monty'@'%'  
      -> WITH GRANT OPTION;  
mysql> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin_pass';  
mysql> GRANT RELOAD,PROCESS ON *.* TO 'admin'@'localhost';  
mysql> CREATE USER 'dummy'@'localhost';
```

PS: WITH GRANT OPTION cria um usuário que pode editar as permissões de outros usuários.

## 7. Restrições de Integridade

- *Gerenciamento de Acesso* -

### ■ Exemplos de criação de novos usuários:

```
mysql> CREATE USER 'custom'@'localhost' IDENTIFIED BY 'obscure';
```

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
```

```
-> ON bankaccount.*
```

```
-> TO 'custom'@'localhost';
```

```
mysql> CREATE USER 'custom'@'host47.example.com' IDENTIFIED BY  
      'obscure';
```

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
```

```
-> ON expenses.*
```

```
-> TO 'custom'@'host47.example.com';
```

```
mysql> CREATE USER 'custom'@'%.example.com' IDENTIFIED BY  
      'obscure';
```

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
```

```
-> ON customer.*
```

```
-> TO 'custom'@'%.example.com';
```



# Banco de Dados

## Asserções (Assertions)

**Prof. Fernando Rodrigues de Almeida Júnior**

**UFC – Universidade Federal do Ceará**

Curso: Eng. da Computação

e-mail: [fernandorodrigues@sobral.ufc.br](mailto:fernandorodrigues@sobral.ufc.br)

# 7. Restrições de Integridade

## - *Assertion* -

- ❑ **Asserção [Significado]:**
  - Proposição que se assume como verdadeira, independentemente de seu conteúdo. Afirmação: proposição que se faz com muita certeza; assertiva, alegação, afirmativa.
- ❑ **Pode-se especificar restrições gerais em SQL por meio de asserções declarativas (não procedimentais):**
  - Cada asserção recebe um nome de restrição e é especificada por uma condição semelhante à cláusula WHERE de uma consulta SQL.
- ❑ **Overhead sobre operações no BD => poucos SGBDs implementam. Ex: PostgreSQL.**

# 7. Restrições de Integridade

## - *Assertion* -

- ❑ Criando uma Asserção [Forma Geral - Padrão SQL-1999]:

```
CREATE ASSERTION <constraint name>  
CHECK ( <search condition> );
```

- ❑ Removendo uma Asserção [Forma Geral]:

```
DROP ASSERTION <constraint name> ;
```

# 7. Restrições de Integridade

- *Assertion* -

## ■ Asserção [Exemplo]:

- Especificar a restrição de que o salário de um funcionário não pode ser maior que o salário do gerente do departamento para o qual o funcionário trabalha:

```
CREATE ASSERTION RESTRICAO_SALARIAL
CHECK ( NOT EXISTS
    ( SELECT *
      FROM   FUNCIONARIO F,
            FUNCIONARIO G,
            DEPARTAMENTO D
      WHERE F.Salario > G.Salario
            AND F.Lotacao = D.Cod_Depart
            AND D.Matr_Gerente = G.Matr ) );
```