

# SQL DDL

## Esquemas Relacionais



Restrições de Integridade  
Chave Primária  
Chaves Estrangeiras

Prof. Dr. Luis Mariano del Val Cura

# Esquema do Banco de Dados em SQL

- Um *esquema de banco de dados* está formado pelas declarações das relações (“tabelas”) do Banco de Dados.
- Restrições de integridade incluindo gatilhos(*triggers*) e asserções
- Outros elementos também aparecem no esquema do banco de dados como visões (*views*) e índices.

# Banco de dados para os exemplos



*Empregados* (RG, CPF, Nome, Idade, Salario,  
CodDpto)

*Departamentos* (Codigo, Nome, RGChefe)

*Projetos* (Num, Nome, RGCoord, Orçamento, CodDpto)

*TrabXProjeto* (RG, Num, NumeroHoras)

# Criação de uma Relação

- Forma mais simples é:

```
CREATE TABLE <nome> (  
    <lista de elementos>  
);
```

- Para remover uma relação:

```
DROP TABLE <nome>;
```

# Elementos da declaração de uma tabela

- Elemento básico: um atributo e seu tipo.
- Tipos mais comuns:
  - INT or INTEGER (sinônimos).
  - REAL or FLOAT (sinônimos).
  - CHAR( $n$ ) = strings de tamanho fixo de  $n$  caracteres.
  - VARCHAR( $n$ ) = strings de tamanho variável de  $n$  caracteres.

**BASE PARA RESTRIÇÕES DE DOMÍNIO**

# Exemplo

```
create table Empregado (  
    RG      char(10),  
    CPF     char (12)  ,  
    Nome    varchar(30)  ,  
    Idade   int    ,  
    Salario real,  
    Coddpto char(3)  
)
```

# Definição de chaves

- Um atributo ou lista de atributos podem ser declarados como chaves primárias:

## **PRIMARY KEY**

- Um atributo que não pode ter valores repetidos nas tuplas

## **UNIQUE**

- Usado para definir outras chaves Candidatas.

- **BASE PARA RESTRIÇÕES DE:**

- **INTEGRIDADE DE CHAVE**

- **INTEGRIDADE DE ENTIDADE**

# Declaração de chaves de Atributo simples

```
create table Empregado (  
    RG      char(10) primary key,  
    CPF     char (12) unique ,  
    Nome    varchar(30)      ,  
    Idade   int               ,  
    Salario real,  
    coddpto char(3)  
)
```



# Declaração de chaves multi-atributo.

---

- A declaração das chaves multi-atributo pode ser um outro elemento na lista de elementos do comando `CREATE TABLE`.
- Esta forma é imprescindível se a chave está definida por mais de um atributo.

# Exemplo : Chave multi-atributo

- RG e Numero do Projeto formam uma chave.

```
CREATE TABLE TrabXProjeto (  
    RG                char(10),  
    Num               int,  
    NumeroHoras      int,  
    PRIMARY KEY (RG, Num)  
);
```

# PRIMARY KEY vs UNIQUE

---

- O padrão SQL permite aos implementadores de DBMS fazer sua própria distinção entre PRIMARY KEY e UNIQUE.
- Exemplo: alguns DBMS podem criar automaticamente um *índice* (estrutura para realizar acesso mais rápido) em resposta a um PRIMARY KEY, mas não para UNIQUE.

# PRIMARY KEY vs UNIQUE



- SQL padrão precisa distinguir PRIMARY KEY e UNIQUE:
  1. Pode existir uma única PRIMARY KEY para a relação mas muitos atributos UNIQUE.
  2. Atributo PRIMARY KEY nunca pode ser NULL para nenhuma tupla. Mas atributos declarados UNIQUE podem ter NULL's.
  3. Chaves candidatas não selecionadas como chaves primárias são definidas como UNIQUE

# Algumas outras declarações para atributos

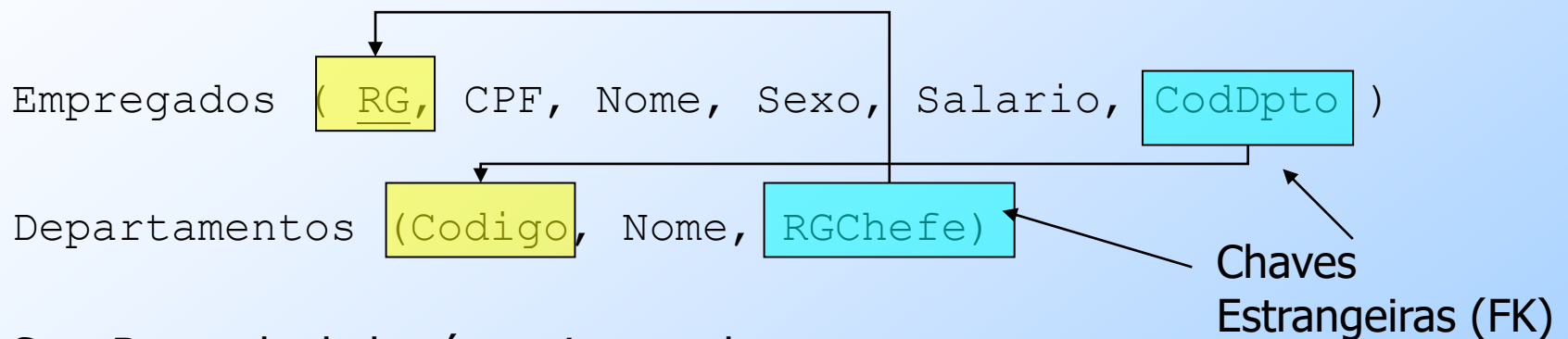
---

1. NOT NULL significa que o valor do atributo nunca pode ser NULL.
2. DEFAULT <valor> indica que se nenhum valor específico é associado ao atributo em alguma tupla, esta se inicializa com um valor pre-definido.

# Exemplo: Valores padrão

```
CREATE TABLE Alunos (  
    nome CHAR(30) PRIMARY KEY,  
    endereço CHAR(50)  
        DEFAULT `Guatemala 123`  
    fone CHAR(16)  
);
```

# Chaves Estrangeiras. Relembrando



Se o Banco de dados é consistente devemos esperar que:

- Cada valor de **CodDpto** nas tuplas da relação **Empregado** seja um valor existente no atributo **Dpto** de alguma tupla de **Departamento** ou NULL
- Cada valor de **RGChefe** nas tuplas da relação **Departamento** seja um valor existente no atributo **RG** em alguma tupla de **Empregado** ou NULL

# Definição de chaves estrangeiras.

- Uso da palavra reservada REFERENCES:
  1. Como parte de definição de um atributo (somente para chaves estrangeiras de um atributo).
  2. Como elemento do esquema relacional:  
FOREIGN KEY ( <lista de atributos> )  
REFERENCES <relação> ( <atributos> )
- Atributos referenciados devem estar declarados como PRIMARY KEY ou UNIQUE.



# Exemplo: No atributo

```
create table Empregados (  
    RG      char(10) primary key,  
    CPF     char (12) unique,  
    Nome    varchar(30) not null,  
    Idade   int,  
    coddpto char(3) references Departamentos(codigo)  
)
```

```
create table Departamentos(  
    codigo   char(3) primary key,  
    Nome     varchar(30) not null,  
    RgChefe  char(10) references Empregados(RG)  
)
```

# Exemplo: Como elemento da declaração

```
create table Empregados (  
    RG      char(10),  
    CPF     char (12) unique,  
    Nome    varchar(30) not null,  
    Idade   int  
    coddpto char(3),  
    foreign key (coddpto) references Departamentos(codigo),  
    primary key (RG)  
)
```

```
create table Departamentos(  
    codigo   char(3),  
    Nome     varchar(30) not null,  
    RgChefe  char(10),  
    primary key (codigo),  
    foreign key (c RgChefe) references Empregados(RG)  
)
```

# Controle de restrições de chave estrangeira

- Se existe uma restrição de chave estrangeira do atributo da relação  $R$  para a chave da relação  $S$ , duas violações são possíveis:
  1. Uma inserção ou atualização em  $R$  inclui valores não encontrados em  $S$ .
  2. Uma remoção ou atualização em  $S$  provoca que varias tuplas em  $R$  fiquem sem referência

# Ações do SGBD (1)

- Suponha  $R = \text{Empregados}$ ,  
 $S = \text{Departamentos}$ .
- Uma inserção ou atualização que introduza um Empregado de um Departamento não existente deve ser rejeitada.
- Uma remoção ou atualização que elimine um Departamento que aparece em varias tuplas de Empregados pode ser manipulada de três formas:

# Ações do SGBD (2)

1. *Default* : Rejeita a modificação .
2. *Cascade* : Aplica a mesma modificação em Empregados.
  - Delete Departamento: remoção da tupla em Empregados.
  - Update Departamento: modifica o valor em Emregados.
3. *Set NULL* : Modifica o valor do Empregado para NULL.

# Exemplo: Cascade

- Remoção da tupla *Pesquisa* em Departamentos:
  - Remover todas as tuplas em Empregados que tem como Departamento a *Pesquisa*. (código 2)
- Modificação do código de *Pesquisa* de 2 para 4.
  - Modificar todas as chaves estrangeiras em Empregados do departamento *Pesquisa* de 2 para 4

# Operação de remoção


**Empregado (**

<u>RG</u>	Nome	Sexo	Salario	Dpto
2232	João	M	700.00	1
2245	Ana	F	1100.00	2
8960	Roberto	M	1800.00	3
7865	Claudia	F	1200.00	2
0983	Helena	F	600.00	NULL

**Departamento (**

<u>Dpto</u>	NomeDpto	Local	RGChefe
1	RH	Andar1	2232
2	Pesquisa	Andar2	7865
3	Manutenção	Andar2	8960

**Remoção**



# Operação de remoção. Cascade

**Empregado (**

<b><u>RG</u></b>	<b>Nome,</b>	<b>Sexo,</b>	<b>Salario,</b>	<b>Dpto )</b>
2232	João	M	700.00	1
8960	Roberto	M	1800.00	3
0983	Helena	F	600.00	NULL

**Departamento (**

<b><u>Dpto</u></b>	<b>NomeDpto,</b>	<b>Local,</b>	<b>RGChefe )</b>
1	RH	Andar1	2232
3	Manutenção	Andar2	8960



# Operação de atualização.

Empregado (	<u>RG</u> ,	Nome,	Sexo,	Salario,	Dpto )
	2232	João	M	700.00	1
	2245	Ana	F	1100.00	2
	8960	Roberto	M	1800.00	3
	7865	Claudia	F	1200.00	2
	0983	Helena	F	600.00	NULL

Departamento (	<u>Dpto</u> ,	NomeDpto,	Local,	RGChefe )
	1	RH	Andar1	2232
4 →	2	Pesquisa	Andar2	7865
Atualização	3	Manutenção	Andar2	8960

# Operação de atualização. Cascade

Empregado (	<u>RG</u> ,	Nome,	Sexo,	Salario,	Dpto )
	2232	João	M	700.00	1
	2245	Ana	F	1100.00	4
	8960	Roberto	M	1800.00	3
	7865	Claudia	F	1200.00	4
	0983	Helena	F	600.00	NULL

Departamento (	<u>Dpto</u> ,	NomeDpto,	Local,	RGChefe )
	1	RH	Andar1	NULL
	4	Pesquisa	Andar2	7865
	3	Manutenção	Andar2	8960

# Exemplo: Set NULL

- Remoção da tupla *Pesquisa* em Departamentos:
  - Modifica todas as tuplas de Empregados do departamento *Pesquisa* para o valor NULL.
- Modificação do código de *Pesquisa* de 2 para 4.
  - Modifica todas as tuplas de Empregados do departamento *Pesquisa* para o valor NULL.

# Operação de remoção

**Empregado (**


<u>RG</u>	Nome	Sexo	Salario	Dpto
2232	João	M	700.00	1
2245	Ana	F	1100.00	2
8960	Roberto	M	1800.00	3
7865	Claudia	F	1200.00	2
0983	Helena	F	600.00	NULL

**Departamento (**

<u>Dpto</u>	NomeDpto	Local	RGChefe
1	RH	Andar1	2232
2	Pesquisa	Andar2	7865
3	Manutenção	Andar2	8960

**)**

**Remoção**



# Operação de remoção. Set NULL

Empregado (	<u>RG</u> ,	Nome,	Sexo,	Salario,	Dpto )
	2232	João	M	700.00	1
	2245	Ana	F	1100.00	NULL
	8960	Roberto	M	1800.00	3
	7865	Claudia	F	1200.00	NULL
	0983	Helena	F	600.00	NULL

Departamento (	<u>Dpto</u> ,	NomeDpto,	Local,	RGChefe )
	1	RH	Andar1	2232
	3	Manutenção	Andar2	8960

# Operação de atualização. Set NULL

**Empregado ( RG , Nome, Sexo, Salario, Dpto )**

2232	João	M	700.00	1
2245	Ana	F	1100.00	2
8960	Roberto	M	1800.00	3
7865	Claudia	F	1200.00	2
0983	Helena	F	600.00	NULL

**Departamento ( Dpto , NomeDpto, Local, RGChefe )**

1	RH	Andar1	2232
2	Pesquisa	Andar2	7865
3	Manutenção	Andar2	8960

4 →

**Atualização**

# Operação de remoção. Set NULL

Empregado (	<u>RG</u> ,	Nome,	Sexo,	Salario,	Dpto )
	2232	João	M	700.00	1
	2245	Ana	F	1100.00	NULL
	8960	Roberto	M	1800.00	3
	7865	Claudia	F	1200.00	NULL
	0983	Helena	F	600.00	NULL

Departamento (	<u>Dpto</u> ,	NomeDpto,	Local,	RGChefe )
	1	RH	Andar1	NULL
	4	Pesquisa	Andar2	7865
	3	Manutenção	Andar2	8960

# Definição de uma política

- Quando uma chave estrangeira é declarada devemos definir uma política (SET NULL ou CASCADE) para cada caso independente de remoção e de atualização.
- A seguir da declaração de chave estrangeira:  
ON [UPDATE, DELETE][SET NULL CASCADE]
- Duas destas cláusulas podem ser usadas para cada chave estrangeira.
- Caso não seja definido, o padrão (rejeitar a mudança) será aplicado.



# Exemplo

```
create table Empregados (  
    RG    char(10) primary key ,  
    CPF   char (12) unique,  
    Nome  varchar(30) not null,  
    Idade int,  
    Salario real,  
    coddpto char(3),  
    foreign key(coddpto) references Departamentos(codigo),  
    On Delete Set NULL,  
    On Update Cascade  
)
```

# Restrições baseadas em atributos

---

- Restrições dos valores para um atributo em particular.
- Adicionar: CHECK( <condição> ) na declaração do atributo.
- A condição deve usar explicitamente o nome do atributo mas outras relações ou atributos podem aparecer como subconsulta.

# Exemplo

```
create table Empregados (  
    RG      char(10) primary key ,  
    CPF     char (12) unique,  
    Nome    varchar(30) not null,  
    Idade   int,  
    Salario real  check (Salario > 100 and Salario < 50000 ),  
    coddpto char(3),  
    foreign key(coddpto) references Departamentos(codigo),  
        On Delete Set NULL,  
        On Update Cascade  
)
```

# Momento da verificação

- Restrições baseadas em atributos são verificadas unicamente quando o valor do atributo é inserido ou atualizado.
- **Exemplo:** CHECK (Salario .... ) é verificado para permitir a inserção de um novo Empregado e rejeitada a modificação (para essa tupla) se o novo salário é maior do que 50000 ou menor que 100.

□

# Restrições baseadas em tuplas

- CHECK ( <condição> ) pode ser adicionado como um elemento do esquema relacional.
- A <condição> pode se referir a qualquer atributo da relação.
  - Quaisquer outros atributos e relações precisam de uma sub-consulta.
- A verificação é realizada na inserção e atualização.

# Exemplo: Restrição baseada em tuplas

- Somente os Empregados de mais de 25 anos podem ganhar mais de 6000.00

```
create table Empregados (  
  RG    char(10) primary key ,  
  CPF   char (12) unique,  
  Nome  varchar(30) not null,  
  Idade  int,  
  Salario real,  
  check (Idade > 25 or Salario < 6000 ),  
)
```

# Adição de Atributos

- Podemos adicionar um novo atributo no esquema relacional:

```
ALTER TABLE <nome> ADD  
    <declaração de atributo>;
```

- Exemplo:**

```
ALTER TABLE Empregados ADD  
fone CHAR(16) DEFAULT  
    'naodefinido';
```

# Remoção de atributos

- Eliminar um atributo do esquema relacional como:

```
ALTER TABLE <nome>  
    DROP <atributo>;
```

- Exemplo: Queremos eliminar o atributo Sexo de Empregados:

```
ALTER TABLE Empregados DROP  
Sexo;
```