

# [MAC0350] Introdução ao Desenvolvimento de Sistemas de Software

## Projeto Lógico de Bancos de Dados: o Modelo de Dados Relacional

Kelly Rosa Braghetto

DCC-IME-USP

Abril de 2024

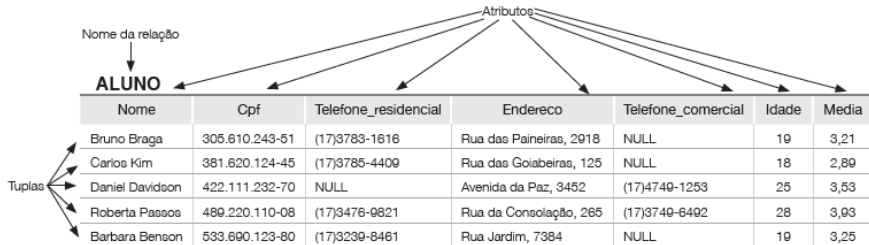
# Modelo relacional

- ▶ É um modelo de dados de implementação
- ▶ Foi introduzido por Ted Codd, da IBM Research, em 1970
- ▶ Primeiros **Sistemas Gerenciadores de Banco de Dados Relacionais** (SGBDRs) comerciais surgiram na década de 1980
- ▶ É um modelo simples, mas muito bem fundamentado matematicamente
  - ▶ Bloco de construção básico: conceito de *relação matemática*
  - ▶ Base teórica: *teoria dos conjuntos* e a *lógica de predicados de primeira ordem*

# Modelo Relacional

- ▶ No modelo relacional, um banco de dados é uma **coleção de relações**
- ▶ Cada **relação** pode ser vista como uma **tabela de valores**
- ▶ Cada **linha da tabela** representa uma **entidade ou relacionamento** do mundo real
- ▶ Os nomes de tabela e de coluna auxiliam a interpretação do significado dos valores de cada linha
- ▶ Todos os valores em uma mesma coluna possuem um mesmo **domínio** de valores possíveis
- ▶ No modelo relacional, os nomes correspondentes à linha, coluna e tabela são, respectivamente, **tupla**, **atributo** e **relação**

# Exemplo: relação ALUNO



# Domínio

- ▶ Um domínio  $D$  é um conjunto de **valores atômicos**
- ▶ Um domínio geralmente é especificado por:
  - ▶ um **nome** (para ajudar na interpretação de seus valores)
  - ▶ um **tipo de dado** (do qual são retirados os valores que formam o domínio)
  - ▶ um **formato**
- ▶ Ex.: o domínio `Numeros_telefone_nacionais`
  - ▶ Sequência de caracteres na forma *(dd)dddddd-dddd* onde cada *d* é um dígito numérico e os dois primeiros dígitos formam um código de área de telefone válido

## Esquema de relação

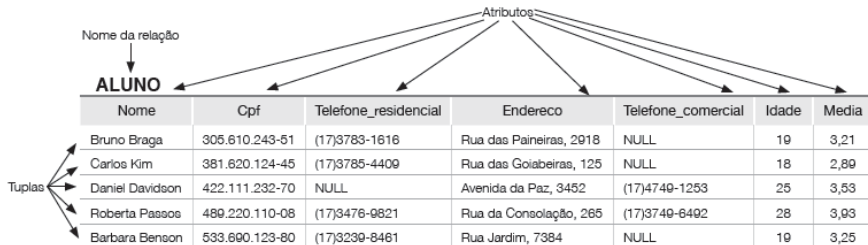
- ▶ **Esquema de relação** é usado para descrever uma relação
- ▶ Um esquema de relação  $R$ , indicado por  $R = (A_1, A_2, \dots, A_n)$ , é composto de um **nome de relação**  $R$  e uma **lista de atributos**  $A_1, A_2, \dots, A_n$ .
- ▶ Cada atributo  $A_i$  é o **nome de um papel** desempenhado por algum domínio  $D$  no esquema de relação  $R$ .  $D$  é chamado domínio de  $A_i$  e é denotado por  $dom(A_i)$
- ▶ O **grau de uma relação** é o número de atributos  $n$  no seu esquema de relação

## Relação (ou estado de relação)

- ▶ Uma **relação**  $r$  (ou **estado de relação**) do esquema de relação  $R(A_1, A_2, \dots, A_n)$ , também denotada por  $r(R)$ , é um conjunto de tuplas  $r = \{t_1, t_2, \dots, t_m\}$ .
- ▶ Cada tupla  $t_i$  é uma lista ordenada de  $n$  valores  $t = \langle v_1, v_2, \dots, v_n \rangle$ , onde cada valor  $v_i$ ,  $1 \leq i \leq n$ , é um elemento de  $dom(A_i)$  ou o valor especial NULL.
- ▶ Outros termos usados:
  - ▶ **intenção da relação**  $\Rightarrow$  esquema de relação  $R$
  - ▶ **extensão da relação**  $\Rightarrow$  estado de relação  $r(R)$

## Exemplo: relação ALUNO

- ▶ Relação de grau sete, que armazena informações sobre alunos universitários:  
ALUNO(Nome, Cpf, Telefone\_residencial, Endereco, Telefone\_comercial, Idade, Media)
- ▶ Usando o tipo de dado de cada atributo, a definição algumas vezes é descrita por:  
ALUNO(Nome: string, Cpf: string, Telefone\_residencial: string, Endereco: string, Telefone\_comercial: string, Idade: integer, Media: real)





## Relação – definição mais formal

- ▶ Uma relação  $r(R)$  é uma **relação matemática** de grau  $n$  sobre os domínios  $dom(A_1), dom(A_2), \dots, dom(A_n)$
- ▶  $r(R)$  é um **subconjunto do produto cartesiano** dos domínios que definem  $R$

$$r(R) \subseteq (dom(A_1) \times dom(A_2) \times \dots \times dom(A_n))$$

# Características das relações

## Ordenação das tuplas e dos atributos

- ▶ Uma relação é um conjunto de tuplas. Logo, **as tuplas em uma relação não possuem nenhuma ordem em particular.**
  - ▶ Quando os registros são armazenados fisicamente no disco, ou quando exibimos a relação como uma tabela, as linhas possuem uma ordem.
  - ▶ Essa ordenação das tuplas não faz parte da definição da relação. Uma relação tenta expressar fatos em um nível lógico ou abstrato.
- ▶ Entretanto, para simplificar a notação, é conveniente considerar que **os atributos nas relações e os seus respectivos valores nas tuplas são ordenados.**

# Características das relações

## Valores nas tuplas

- ▶ Cada valor em uma tupla é **atômico**
- ▶ Atributos compostos ou multivalorados não são permitidos
  - ▶ Atributos compostos são representados nas relações apenas em termos dos seus atributos componentes simples
  - ▶ Atributos multivalorados são representados em relações separadas

# Características das relações

## Valores NULL

- ▶ O valor especial NULL é usado para indicar:
  - ▶ que uma dada tupla não possui um valor para um atributo
  - ▶ que o valor para um atributo é desconhecido para uma dada tupla
  - ▶ que um atributo não se aplica a uma dada tupla

A comparação de valores NULL leva à ambiguidade; por essa razão, devemos sempre que possível evitar o NULL no projeto de BDs

## Interpretação de uma relação

- ▶ O modelo relacional representa fatos sobre entidades e relacionamentos uniformemente: tudo são **relações**
- ▶ Dificuldade de compreensão: descobrir se uma relação representa um tipo de entidade ou um tipo de relacionamento
- ▶ O projeto conceitual (modelo ER) lida com esse problema de forma apropriada

## Restrições do modelo relacional

- ▶ O estado de um BD como um todo corresponde aos estados de todas as suas relações em um determinado instante
- ▶ Geralmente, há muitas restrições para os valores reais em um estado do BD
- ▶ A maioria das restrições são derivadas de regras do minimundo que o BD representa
- ▶ Algumas restrições são inerentes ao modelo relacional

# Restrições do modelo relacional

## Categorias de restrições

- ▶ **Restrições implícitas** – as que são inerentes ao modelo de dados relacional
- ▶ **Restrições explícitas** – as que podem ser expressas diretamente nos esquemas do modelo de dados, em geral, por meio de uma linguagem de definição de dados (DDL – *data definition language*)
- ▶ **Restrições semânticas (ou regras de negócio)** – as que devem ser expressas e impostas pelos programas de aplicação

# Restrições implícitas

Já foram descritas em slides anteriores:

- ▶ Uma relação não possui tuplas repetidas (já que ela é um conjunto de tuplas)
- ▶ As tuplas em uma relação não possuem qualquer ordem em particular



## Restrições explícitas

São de 4 tipos:

1. Restrições de domínio
2. Restrições de chave
3. Restrições sobre valores NULL
4. Restrições de integridade (de entidade e referencial)

# Restrições explícitas

## 1. Restrições de domínio

- ▶ Especificam que, dentro de cada tupla, o valor de cada atributo  $A$  deve ser um valor atômico do domínio  $dom(A)$
- ▶ Veremos mais detalhes sobre como domínios podem ser definidos quando estudarmos a linguagem SQL

# Restrições explícitas

## 2. Restrições de chave

- ▶ Não existe em uma relação duas tuplas que possuam a mesma combinação de valores para **todos** os seus atributos
- ▶ Normalmente, existem subconjuntos de atributos de um esquema de relação  $R$  com a propriedade de que duas tuplas em qualquer estado de relação  $r$  de  $R$  não têm a mesma combinação de valores para esses atributos
- ▶ Qualquer um desses subconjuntos é chamado de **superchave** de  $R$
- ▶ Uma superchave especifica uma restrição de unicidade, na qual duas tuplas distintas em qualquer estado  $r$  de  $R$  não podem ter o mesmo valor para os atributos da superchave

# Restrições explícitas

## 2. Restrições de chave (continuação)

- ▶ Uma **chave**  $K$  de um esquema  $R$  é uma **superchave mínima** de  $R$ , ou seja, não podemos remover dela nenhum atributo e ainda manter a restrição de unicidade.
- ▶ Um esquema de relação pode possuir mais de uma chave; nesses casos, cada uma das chaves é chamada de **chave candidata**.
- ▶ Geralmente, indica-se uma das chaves candidatas como **chave primária** da relação; essa é a chave usada para identificar as tuplas da relação.  
As demais chaves candidatas são designadas como **chaves únicas** (*unique*).
- ▶ Num esquema de relação, os atributos da chave primária devem aparecer sublinhados.

## Exemplo: a relação CARRO com duas chaves candidatas

CARRO

<u>Placa</u>	Numero_chassi	Marca	Modelo	Ano
Itatiaia ABC-7039	A6935207586	Volkswagen	Gol	02
Itu TVP-3470	B4369668697	Chevrolet	Corsa	05
Santos MPO-2902	X8355447376	Fiat	Uno	01
Itanhaem TFY-6858	C4374268458	Chevrolet	Celta	99
Itatiba RSK-6279	Y8293586758	Renault	Clio	04
Atibaia RSK-6298	U0283657858	Volkswagen	Parati	04

**Figura 3.4**

A relação CARRO, com duas chaves candidatas: Placa e Numero\_chassi.

Nesse

exemplo, a chave candidata Placa foi escolhida como **chave primária**; por isso, a outra chave candidata Numero\_chassi é uma **chave única**. Não há notação gráfica para indicar a chave única no diagrama: é preciso usar uma anotação textual.

## Restrições explícitas

### 3. Restrições sobre valores NULL

- ▶ Esse tipo de restrição sobre os atributos especifica se valores NULL são ou não permitidos

## Restrições explícitas

Até aqui, vimos apenas restrições explícitas que se aplicam apenas a relações isoladas e seus atributos.

As demais restrições explícitas se referem à forma como as relações de um BD se relacionam entre si.

Por isso, antes de prosseguir na definição das restrições, definiremos o que é um BD relacional.

## Banco de dados relacional – definição

- ▶ Um **esquema de banco de dados relacional**  $S$  é um **conjunto de esquemas de relação**  $S = \{R_1, R_2, \dots, R_m\}$  e um **conjunto de restrições de integridade**  $RI$ .
- ▶ Um **estado de um banco de dados relacional**  $BD$  com esquema  $S$  é um conjunto de relações  $BD = \{r_1, r_2, \dots, r_m\}$  tal que  $r_i$  é uma relação de  $R_i$  e satisfaz as restrições de integridade especificadas em  $RI$ .
- ▶ O termo **banco de dados relacional** refere-se, implicitamente, ao seu esquema e ao seu estado atual.
- ▶ Um **estado válido** é um estado de um BD relacional que satisfaz a todas as restrições de integridade.



# Exemplo de um diagrama de esquema para um BD relacional

## FUNCIONARIO

Prnome	Minicial	Unome	<u>Cpf</u>	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
--------	----------	-------	------------	----------	----------	------	---------	----------------	-----

## DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

## LOCALIZACAO\_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

## PROJETO

Projnome	<u>Projnumero</u>	Projlocal	Dnum
----------	-------------------	-----------	------

## TRABALHA\_EM

<u>Fopf</u>	<u>Pnr</u>	Horas
-------------	------------	-------

## DEPENDENTE

<u>Fopf</u>	<u>Nome_dependente</u>	Sexo	Datanaso	Parentesco
-------------	------------------------	------	----------	------------

Figura 3.5

# Banco de dados relacional

## Algumas observações sobre o diagrama de exemplo

- ▶ Os atributos que representam um mesmo conceito no mundo real não necessariamente precisam ter o mesmo nome em relações diferentes  
Ex.: Dnumero em DEPARTAMENTO, Dnum em PROJETO e Dnr em FUNCIONARIO
- ▶ Atributos que representam conceitos diferentes não necessariamente precisam ter nomes diferentes em diferentes relações  
Ex.: Poderíamos ter um atributo Nome em PROJETO e outro atributo nome em DEPARTAMENTO (no lugar dos atributos Projnome e Dnome)

# De volta às restrições explícitas

## 4. Restrições de integridade

- ▶ **Restrições de integridade de entidade** – estabelecem que nenhum valor de chave primária pode ser NULL. Ter valores NULL para uma chave primária implica em não podermos identificar alguma(s) tupla(s).
- ▶ **Restrições de integridade referencial** – declaram que uma tupla em uma relação que faz referência a uma outra relação deve se referir a uma tupla existente nessa relação. Essa referência é feita por meio de **chaves estrangeiras**.

# Restrição de integridade referencial

## Chave estrangeira

- ▶ Uma chave estrangeira  $ChE$  de um esquema de relação  $R_1$  para um esquema de relação  $R_2$  é um subconjunto de atributos de  $R_1$  que possuem os mesmo domínios dos atributos da chave primária  $ChP$  de  $R_2$ .
- ▶ Além disso, toda tupla  $t_1$  no estado atual  $r_1$  de  $R_1$  ou possui um valor NULL para  $ChE$  ou  $t_1[ChE] = t_2[ChP]$ , onde  $t_2$  é uma tupla do estado atual  $r_2$  de  $R_2$ .

# Exemplo de chaves estrangeiras em um BD relacional

## FUNCIONARIO

Pnome	Minicial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
-------	----------	-------	-----	----------	----------	------	---------	----------------	-----

## DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

## LOCALIZACOES\_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

## PROJETO

Projnome	<u>Projnumero</u>	Projlocal	Dnum
----------	-------------------	-----------	------

## TRABALHA\_EM

<u>Fcpf</u>	<u>Pnr</u>	Horas
-------------	------------	-------

## DEPENDENTE

<u>Fcpf</u>	<u>Nome_dependente</u>	Sexo	Datanasc	Parentesco
-------------	------------------------	------	----------	------------

Figura 3.7

# Restrições semânticas

## Exemplo:

*o salário de um empregado não pode exceder o do seu supervisor*

## Mecanismos de implementação desse tipo de regras

- ▶ Programas de aplicação
- ▶ *Triggers* (gatilhos), no SGBDR
- ▶ *Assertions* (asserções), no SGBDR

# Operações do modelo relacional

As operações do modelo relacional podem ser divididas em duas categorias: *recuperações* e *modificações*.

## Recuperações

- ▶ As recuperações podem ser especificadas por meio de operações da **Álgebra Relacional** – que provê fundamentos formais para as operações de consulta do modelo relacional
- ▶ A Álgebra Relacional é usada como uma base para implementar e otimizar consultas nos SGBDRs
- ▶ Alguns dos seus conceitos estão implementados na linguagem de consulta padrão para os SGBDRs – a SQL (*Structured Query Language*)

# Operações do modelo relacional

As operações do modelo relacional podem ser divididas em duas categorias: *recuperações* e *atualizações*.

## Modificações

- ▶ Compreendem 3 tipos de operações sobre relações: **inserção** de tuplas, **remoção** de tuplas e **alteração de valores de atributos** de tuplas
- ▶ Essas operações só são aplicadas quando as restrições de integridade especificadas no esquema do banco de dados relacional não são violadas
- ▶ A linguagem SQL também provê comandos para a realização de modificações



## Operação de inserção

- ▶ Oferece uma lista de valores de atributo para que uma nova tupla  $t$  possa ser inserida em uma relação  $R$
- ▶ Pode violar qualquer um dos quatro tipos de restrições (de domínio, de integridade de entidade, de chave, de integridade referencial)
- ▶ Se uma inserção violar uma ou mais restrições, a ação padrão do SGBDR é rejeitar a inserção

## Operação de exclusão

- ▶ Pode violar apenas a integridade referencial (se a tupla que está sendo excluída for referenciada por chaves estrangeiras de outras tuplas)
- ▶ Opções disponíveis para o tratamento de uma violação:
  - ▶ **Restrict** – rejeita a exclusão (opção padrão nos SGBDRs)
  - ▶ **Cascade** – propaga a exclusão excluindo tuplas que referenciam aquela que está sendo excluída
  - ▶ **Set null** ou **set default** – modifica os valores de atributo que referenciam a causa da violação

## Operação de alteração

- ▶ Requer a especificação de uma condição sobre os atributos da relação, para selecionar a tupla (ou tuplas) a serem modificadas
- ▶ Se o atributo a ser alterado não faz parte de uma chave primária nem de uma chave estrangeira, em geral só pode causar problemas de domínio
- ▶ A alteração de uma chave primária/estrangeira gera problemas semelhantes aos da Inserção/Exclusão
  - ▶ as opções de tratamento de uma violação (restrict, cascade e set null ou set default) também fazem sentido quando o valor da chave primária é modificado

# Exemplo de Esquema Relacional com opções para tratamento de violações de chave estrangeira

## FUNCIONARIO

Pnome	Minicial	Unome	<u>Cpf</u>	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
-------	----------	-------	------------	----------	----------	------	---------	----------------	-----

Remoção: restrict  
Alteração: cascade

## DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

Remoção: set NULL  
Alteração: cascade

Remoção: restrict  
Alteração: cascade

## LOCALIZACAO\_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

Remoção: cascade  
Alteração: cascade

## PROJETO

Projnome	<u>Projnumero</u>	Projlocal	Dnum
----------	-------------------	-----------	------

Remoção: restrict  
Alteração: cascade

## TRABALHA\_EM

<u>Fcpf</u>	<u>Pnr</u>	Horas
-------------	------------	-------

Remoção: cascade  
Alteração: cascade

Remoção: cascade  
Alteração: cascade

## DEPENDENTE

<u>Fcpf</u>	<u>Nome_dependente</u>	Sexo	Datanasc	Parentesco
-------------	------------------------	------	----------	------------

Remoção: cascade  
Alteração: cascade

**Obs.:** Departamento. Dnumero e Projeto.ProjNome são chaves únicas.

# Referências Bibliográficas

## Sobre o Modelo Relacional:

- ▶ *Sistemas de Bancos de Dados* (7ª edição), Elmasri e Navathe. Pearson, 2016. – Capítulo 3
- ▶ *Database Systems – the complete book* (2ª edição), Garcia-Molina, Ullman e Widom. Prentice Hall, 2009. Capítulo 2
- ▶ *Sistemas de Gerenciamento de Banco de Dados* (3ª edição), Ramakrishnan e Gehrke, 2008. – Capítulo 3

## Cenas dos próximos capítulos...

- ▶ Mapeamento dos Modelos ER e EER para o Modelo Relacional