

FGA0137

# Sistemas de Banco de Dados 1

Prof. Maurício Serrano

Material original: Profa. Elaine Parros Machado de Sousa

Prof. Jose Fernando Rodrigues Junior

**2021/2**

# Modelo Relacional

## Módulo 2

# Modelo Relacional

- Criado por E. F. Codd (IBM)
  - *"A relational model of data for large shared data banks"*. Communications of the ACM, Volume 13 , Issue 6, June 1970.
- Modelo de Implementação
  - projeto lógico

# Sistemas de Banco de Dados

## Desenvolvimento de Sistemas de Banco de Dados

## Ciclo de Vida



# Definição do Modelo

- *“O modelo relacional representa uma base de dados como uma **coleção de relações**”*  
[Elmasri&Navathe]
- Modelo Relacional – base teórica em **Teoria de Conjuntos**

# Definição do Modelo

- **Valores**

- dados do mundo real

- **Tabelas**

- dados mantidos em tabelas  $\Rightarrow$  representam coleções de objetos, entidades, associações, etc, do mundo real
- tabelas são uma noção intuitiva para as **RELAÇÕES**

# Terminologia

- Relação
  - Tabela
- Tupla
  - Registro, linha
- Atributo
  - Campo
- Valor
- *Relation Intension*
  - Esquema
- *Relation Extension*
  - Instância

# Modelo Intuitivo

| <b>Nome</b> | <b>Nro</b> | <b>Curso</b> | <b>....</b> |
|-------------|------------|--------------|-------------|
|-------------|------------|--------------|-------------|

Esquema

|          |      |      |  |
|----------|------|------|--|
| Paulo    | 9999 | Elet |  |
| Izabella | 8888 | Comp |  |
| João     | 1111 | Comp |  |
|          |      |      |  |
|          |      |      |  |

Instância



# Valores

- Modelo relacional  $\Rightarrow$  valores são **atômicos**
- **Valor Atômico**
  - **indivisível**  $\Rightarrow$  não pode ser recuperado em partes
    - ex: sobrenome, idade
  - **monovalorado**  $\Rightarrow$  pode ter apenas um valor
    - ex:
      - Idade de aluno é monovalorado
      - Irmãos de aluno é multivalorado

# Modelo Intuitivo

| Nome | Nro | Curso |  |
|------|-----|-------|--|
|------|-----|-------|--|

**Esquema  
de Relação**

**Tupla**

**Atributo**

|          |      |      |  |
|----------|------|------|--|
| Paulo    | 9999 | Elet |  |
| Izabella | 8888 | Comp |  |
| João     | 1111 | Comp |  |
|          |      |      |  |
|          |      |      |  |

**Valor**

**Relação**

# Domínios

- **Domínio de aplicação**
- Exemplos:
  - Escola
  - Universidade
  - Cidade
- **Domínio de atributo**
- Exemplos:
  - Nomes de Alunos
  - Códigos de Disciplinas
  - Idade

# Domínios

- Especificação do **Domínio de atributo:**
  - **Nome**
  - **Definição lógica**
  - **Tipo de dado e formato de dado**

# Especificação do Domínio

- **Nome e Definição lógica.** Ex:
  - **Nomes de Alunos:** conjunto de todos os nomes possíveis para pessoas
  - **Códigos de Disciplinas:** conjunto dos códigos das disciplinas oferecidas na FGA
  - **Idade:** conjunto de idades possíveis para alunos

# Especificação do Domínio

- **Tipo de dado e/ou formato.** Ex:
  - **Nomes de Alunos** – *string* de 60 caracteres
  - **Códigos de Disciplinas** – *string* com três letras seguidas de um traço e de quatro dígitos: FGA-0137
  - **Idade** – inteiro entre 15 e 100

# Esquema de Relações

- **Esquema de relação:** descreve a relação
  - $\mathcal{R}(A_1, A_2, \dots, A_n)$ 
    - ou  $\mathcal{R} = \{A_1, A_2, \dots, A_n\}$
  - $\mathcal{R}$  - nome da relação
  - $(A_1, A_2, \dots, A_n)$  - conjunto de atributos que formam a relação

# Esquema de Relações

- N - **grau** da relação descrita por  $\mathcal{R}$ 
  - número de atributos em  $\mathcal{R}$
- **Dom( $A_i$ )** - Domínio do Atributo  $A_i$
- Ex:
  - uma relação de Alunos que tenha os atributos Nome, RG e Idade, tem o seguinte esquema:

**Aluno(Nome, RG, Idade)**



# Exemplo

- **Especificação dos domínios:**
  - **Nomes de Alunos:** conjunto de todos os nomes possíveis para pessoas – *strings* de 60 caracteres
  - **RG:** conjunto dos RGs válidos no Brasil – números de 9 dígitos \_
  - **Idade:** conjunto de idades possíveis para alunos – inteiro entre 0 e 100

## Exemplo (cont.)

- Esquema da relação Aluno:
  - $\text{Aluno} = \{\text{Nome}, \text{RG}, \text{Idade}\}$
- Domínios dos atributos de Aluno:
  - $\text{Dom}(\text{Nome}) = \text{Nomes de Alunos}$
  - $\text{Dom}(\text{RG}) = \text{RG}$
  - $\text{Dom}(\text{Idade}) = \text{Idade}$

# Relações

- **Relação R** – instância do Esquema de Relação  $\mathcal{R}(A_1, A_2, \dots, A_n)$ 
  - $R(\mathcal{R})$
  - $R \subseteq \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n)$
  - **R** é um conjunto de tuplas
$$R = \{t_1, t_2, \dots, t_k\}$$
$$t = \{v_1, v_2, \dots, v_n\}, v_i \in \text{Dom}(A_i)$$

# Relações

- Número total de tuplas possíveis:
  - $| \text{Dom}(A_1) | \times | \text{Dom}(A_2) | \times \dots \times | \text{Dom}(A_n) |$
- $R(\mathcal{R})$  contém apenas as **tuplas válidas** que representam a situação de um determinado instante do mundo real
- **Esquema de Relação**  $\mathcal{R}$  (*relation intension*)  $\Rightarrow$  mudanças pouco freqüentes
- **Relação R** (*relation extension*)  $\Rightarrow$  dinâmica

# Relações

- Exemplo:
  - Esquema de Relação Aluno:
    - $\text{Aluno} = \{\text{Nome}, \text{RG}, \text{Idade}\}$
  - Possível relação:
    - $R(\text{Aluno}) = \{ \langle \text{José}, 12345, 21 \rangle, \langle \text{Pedro}, 54321, 18 \rangle, \langle \text{Paulo}, 321321, 22 \rangle \}$

# Relações

- **Ordem das tuplas de uma relação**

- **relação  $\Rightarrow$  conjunto** de tuplas
- matematicamente não existe a idéia de ordem em conjuntos  $\Rightarrow$  **não existe uma ordem em particular para as tuplas de uma relação**

**OBS:** na implementação de um SGBDR existe uma ordem física de armazenamento das tuplas, determinando uma ordem na recuperação das informações  $\rightarrow$  esta ordem é eventual e pode ser alterada pelo SGBDR por questões de desempenho ou otimização

# Relações

- **Ordem dos valores de uma tupla**
  - **tupla**  $\Rightarrow$  lista de **n** valores dispostos em uma ordem determinada de acordo com a disposição dos atributos no esquema da relação
- Valores nas tuplas
  - os **valores** de uma tupla são **atômicos**
  - valor nulo (*null*)
    - valor desconhecido
    - valor não se aplica
    - valor indisponível

# Restrições das Relações

- **Restrição de domínio**
  - o valor de cada atributo **A** deve ser um valor atômico pertencente a **Dom(A)**
- **Restrição de *null* para atributo**
  - determina quando o valor especial *null* é ou não permitido para um atributo
- **Restrição de unicidade (CHAVE)**
  - deve ser possível identificar univocamente cada tupla da relação



# Restrição de Unicidade

- **Relação** é um **conjunto** de tuplas
  - pela teoria de conjuntos  $\Rightarrow$  **todas as tuplas devem ser distintas**
  - para garantir esta propriedade de maneira eficiente
    - especifica-se uma **Restrição de Unicidade**  $\Rightarrow$  definição de **chave**

# Restrição de Unicidade

- **Superchave**

- conjunto de atributos de uma relação R que identifique univocamente cada tupla
- $\mathbf{SC}_{Hk}(\mathcal{R}) = \{\mathbf{A}_j, \dots, \mathbf{A}_i\} \mid \{\mathbf{A}_j, \dots, \mathbf{A}_i\} \subseteq \mathcal{R}$
- Combinação de valores não se repete
- Exemplo:
  - **Aluno** = {Nome, Idade, Curso, Nro}
  - $SC_{H1}(\text{Aluno}) = \{\text{Nome}, \text{Curso}, \text{Idade}\}$
  - $SC_{H2}(\text{Aluno}) = \{\text{Nro}, \text{Nome}\}$

# Restrição de Unicidade

- **Chave**

- é uma superchave da qual não se pode retirar nenhum atributo e ainda preservar a propriedade de identificação unívoca  $\Rightarrow$  **superchave mínima**

# CHAVE

- $C_{Hk}(\mathcal{R}) = \{A_i, \dots, A_j\} \mid \{A_i, \dots, A_j\} \subseteq \mathcal{R}$

$$t_g[C_{Hk}] \neq t_h[C_{Hk}] \quad \forall g, h \in R, g \neq h$$

- Exemplo:
  - **Aluno = {Nome, Idade, Curso, Nro}**
  - $SC_{H1}(\text{Aluno}) = \{\text{Nome}, \text{Nro}\}$
  - $C_{H1}(\text{Aluno}) = \{\text{Nome}\}$
  - $C_{H2}(\text{Aluno}) = \{\text{Nro}\}$

# Chave

- **Chave Candidata:**
  - pode existir mais de uma chave para uma mesma relação
  - cada uma das chaves é chamada de **Chave Candidata**
    - $C_{H1}(\text{Aluno}) = \{\text{Nome}\}$
    - $C_{H2}(\text{Aluno}) = \{\text{Nro}\}$

# Chave

- **Chave Primária**
  - escolhida entre as chaves candidatas
  - a chave primária é frequentemente a mais utilizada para acessos à relação
  - Exemplo:
    - **$C_{H0}(\text{Aluno}) = \{\text{Nro}\}$**

# Chave

- Notação no Esquema da Relação
  - $C_{H0}(\text{Aluno}) = \{\text{Nro}\}$
  - $C_{H1}(\text{Aluno}) = \{\text{Nome}\}$

**Aluno = { Nome, Idade, Curso, Nro }**

**Chave  
secundária**

**Chave primária**

# Base de Dados Relacional

- O **esquema  $\mathcal{S}$  de uma base de dados relacional** é composto por:
  - 1) um conjunto de esquemas de relações

$$\mathcal{S} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_n\}$$

- 2) um conjunto de **Restrições de Integridade  $I$**



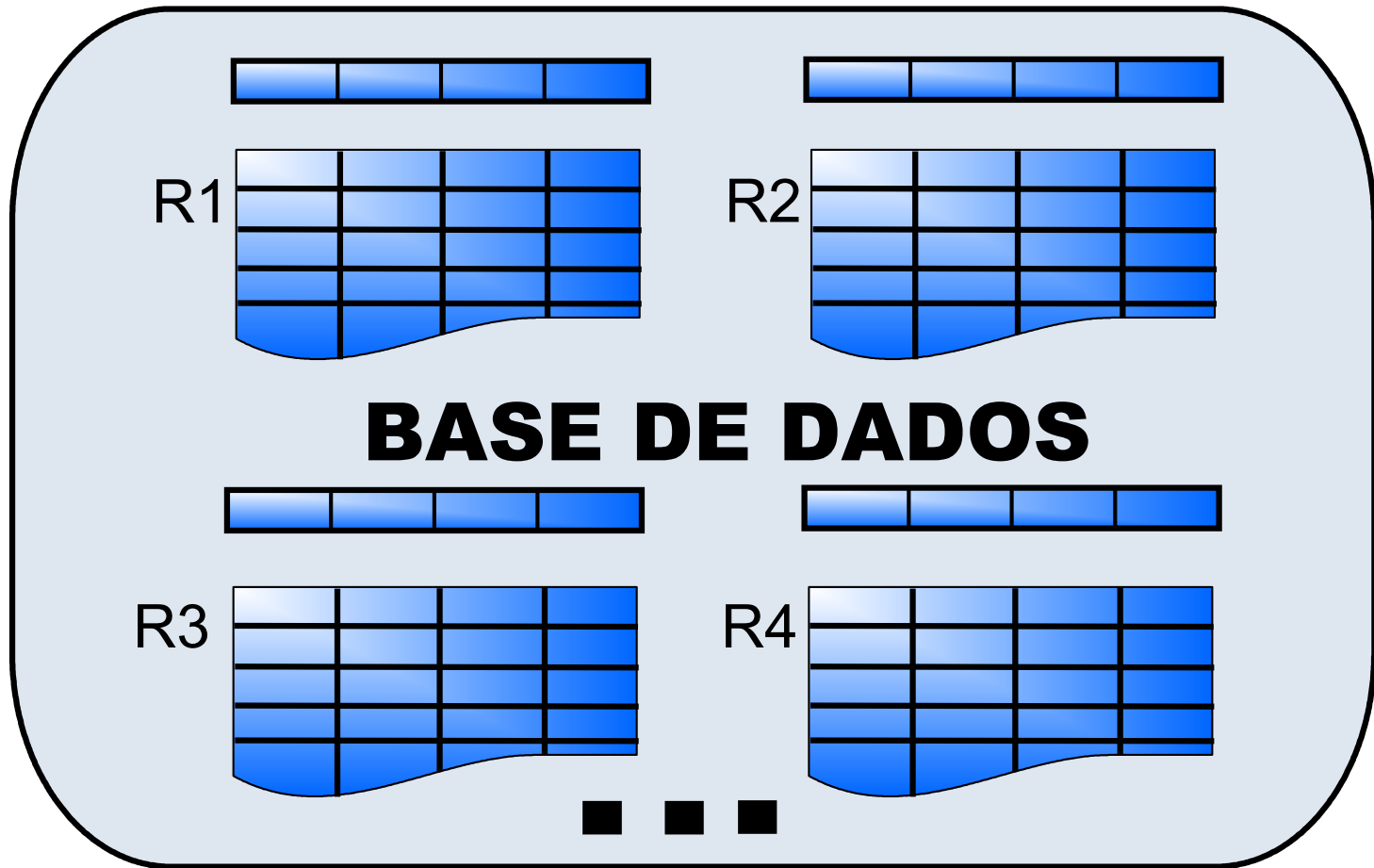
# Base de Dados Relacional

- Uma **base de dados** relacional (uma instância) é composta por:
  - um conjunto de relações

$$\mathbf{BD} = \{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_n\}$$

tal que cada  $R_i$  é uma instância de  $\mathcal{R}_i$  e cada  $R_i$  satisfaz todas as restrições indicadas em  $I$

# Base de Datos Relacional



# Exemplo

- Base de Dados para armazenar informações sobre as diversas turmas de disciplinas oferecidas num semestre
- Esquemas de Relações:
  - Aluno = {Nome, Nro, Idade, Curso}
  - Disciplina = {Sigla, Nome, NCreditos}
  - Matricula = {Aluno, Disciplina, Semestre, Ano, Nota}

# Restrições de Integridade

- **Restrições de integridade**
  - regras a respeito dos valores que podem ser armazenados nas relações
    - objetivo: garantir consistência
  - quando definidas no domínio do problema, devem ser sempre satisfeitas na base de dados

# Restrições de Integridade

- Principais restrições de integridade para um BD relacional:
  - **Restrições de Integridade da Entidade**
  - **Restrições de Integridade Referencial**

# Restrições de Integridade

- **Restrição de Integridade da Entidade**
  - a **chave primária não pode ser nula** em nenhuma tupla de qualquer relação
  - se a chave primária for **composta** por mais de um atributo, **nenhum deles** pode ser nulo

# Restrições de Integridade

- **Restrição de Integridade Referencial**
  - definida entre duas relações
  - usada para manter consistência entre tuplas de duas relações
  - define que: **se** uma tupla  $t_1$  em uma relação  $R_1$  **faz referência** a uma relação  $R_2$ , **então**  $t_1$  deve fazer referência a uma **tupla existente em  $R_2$**

# Restrições de Integridade Referencial

- Restrição de Integridade Referencial está vinculada ao conceito de **chave estrangeira**
  - conceito fundamental: **compatibilidade de domínio**



# Restrições de Integridade Referencial

- **Compatibilidade de Domínio:**
  - dados dois conjuntos de atributos quaisquer C e D, ambos são **compatíveis** quando o primeiro atributo de C tem o mesmo domínio do primeiro atributo de D, o segundo atributo de C tem o mesmo domínio do segundo atributo de D, e assim por diante

# Restrições de Integridade Referencial

- **FK** é uma **Chave estrangeira** em  $R_1$  que referencia  $R_2$  se:
  - 1) FK é compatível em domínio com **toda a chave primária** PK de  $R_2$
  - 2) o **valor** dos atributos FK numa tupla  $t_i$  qualquer da relação  $R_1$ :
    - ou é igual ao **valor** dos atributos PK de alguma tupla  $t_k$  da relação  $R_2 \Rightarrow t_i[\mathbf{FK}] = t_k[\mathbf{PK}]$ ,  $t_i \in R_1$ ,  $t_k \in R_2$
    - ou é nulo  $\Rightarrow t_i[\mathbf{FK}] = \text{null}$

# Restrições de Integridade Referencial

- As duas condições para a ocorrência da chave estrangeira determinam a **Restrição de Integridade Referencial** entre duas relações  $R_1$  e  $R_2$

$$\mathcal{R}_1[\text{FK}] \xrightarrow{\text{CF}} \mathcal{R}_2[\text{PK}]$$

# Restrições de Integridade Referencial

- Chave Estrangeira:

$X = \{\underline{A}, \underline{B}, C\}$

$Y = \{F, G, \underline{H}\}$



$\text{Dom}(F, G) = \text{Dom}(A, B)$

$\{A, B\}$  é chave primária em X

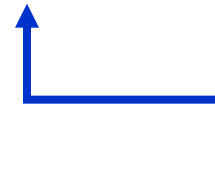
$\{F, G\}$  é chave estrangeira em Y

# Restrições de Integridade Referencial

- Exemplo:

Pergunta: a chave estrangeira {Departamento} pode ser nula? Por que?

Departamento = {Cod, NomeD}



Empregado = {NomeE, Departamento}

# Exemplo

**Alunos = {Nome, Nro, Idade}**

R1(Alunos) = {<Mario, 1234, 20>, <Paulo, 4321, *null*>, <*null*, 1234, 22>, <Thais, *null*, 24>, <Mario, 1235, 22>}

**Disciplina = {Sigla, Monitor}**

R2(Disciplina) = {<FGA-0104, 1234>, <FGA\_123, 2222>, <FGA-0149, Mario>, <FGA-532, *null*>}

Quais restrições de relação e de integridade não são satisfeitas nas tuplas do exemplo? Por quê?