

cin.ufpe.br



# Centro de **Informática**

U • F • P • E



UNIVERSIDADE FEDERAL DE PERNAMBUCO

## Modelo Relacional

Apresentado Por:

Robson do Nascimento Fidalgo

[rdnf@cin.ufpe.br](mailto:rdnf@cin.ufpe.br)

# Modelo Relacional

- Foi definido em 1970 por E.F. Cood
- Principais razões para sua grande aceitação:
  - Simplicidade (teoria dos conjuntos) e Formalismo (álgebra relacional)
- É um modelo lógico que representa dados como relações
  - Neste modelo, o BD é visto como um conjunto de relações
  - Os dados em uma relação representam fatos reais a respeito de uma **entidade** ou de um **relacionamento** do mundo real

- Formalização Básica

- 1) Sejam  $D_1, D_2, \dots, D_n$  conjuntos, não necessariamente disjuntos,  $R$  é uma relação sobre estes  $n$  conjuntos se ela for um conjunto de  $n$ -uplas (tuplas) ordenadas  $\langle a_1, a_2, \dots, a_n \rangle$  tal que  $a_1, a_2, \dots, a_n$  pertence a  $D_1, D_2, \dots, D_n$  ou a nulo
- Onde:
  - $D_1, D_2, \dots, D_n =$  Domínios (Conjunto de valores indivisíveis)
  - $a_1, a_2, \dots, a_n =$  Atributos (Papel desempenhado por algum domínio)
  - $n =$  índice máximo da relação (grau da relação ou qtd de atributos)
    - **dupla**  $\langle d_1, d_2 \rangle$ , **tripla**  $\langle d_1, d_2, d_3 \rangle$ , ... ,  **$n$ -upla**  $\langle d_1, d_2, \dots, d_n \rangle$

- Propriedades de uma Relação:
  - Não há tuplas duplicadas em uma mesma relação
  - As tuplas de uma relação não são ordenadas
  - Os atributos em uma tupla não são ordenados
  - Cada relação possui um número fixo de atributos, todos com nomes distintos

- Composição básica de um BD Relacional
  - Tabelas (ou Relações)
    - Compostas de
      - Linhas
      - Colunas
      - Chaves primárias
    - Relacionadas através de
      - Chaves estrangeiras

# Modelo Relacional

- Terminologias

profissional	acadêmica
tabela	relação
linha	tupla
coluna	atributo
valor de coluna	valor de atributo

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

- Mais sinônimos:

- Tabela = Relação = Arquivo
- Linha = Tupla = Registro
- Coluna = Atributo = Campo
- Valor de Coluna = Valor de Atributo = Valor de Campo

- Chaves
  - Conceito usado para especificar restrições de integridade básicas de um SGBD relacional
- Quatro tipos:
  - chave candidata
  - chave primária
  - chave estrangeira
  - chave alternativa



- Chave Candidata

- É uma coluna (chave simples) ou uma combinação de colunas (chave composta) cujos valores distinguem uma linha das demais dentro de uma tabela.

- Esta deve ser mínima.

- EX:

## Chaves Candidatas

**simples**

Cliente		
Matrícula	CPF	Nome
1111	101.010.101-01	Rita
2222	202.020.202-02	Ana
3333	303.030.303-03	Pedro
4444	404.040.404-04	José

**composta**

Dependente		
Matrícula	Num	Nome
1111	1	Pedro
1111	2	Ruth
2222	1	Rosa
3333	1	João

# Modelo Relacional - Chaves

- Chave Primária
  - É a chave candidata que foi escolhida para identificar preferencialmente uma tupla
  - EX:

## Chaves Primárias

**simples**

Cliente		
<u>Matrícula</u>	CPF	Nome
1111	101.010.101-01	Rita
2222	202.020.202-02	Ana
3333	303.030.303-03	Pedro
4444	404.040.404-04	José

**composta**

Dependente		
<u>Matrícula</u>	<u>Num</u>	Nome
1111	1	Pedro
1111	2	Ruth
2222	1	Rosa
3333	1	João

- Chave Estrangeira
  - Uma coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na ch. primária de uma tabela
  - Mecanismo que permite a implementação de relacionamentos em um banco de dados relacional
  - EX:

## Chave Estrangeira

**simples**

Cliente		
Matrícula	CPF	Nome
1111	101.010.101-01	Rita
2222	202.020.202-02	Ana
3333	303.030.303-03	Pedro
4444	404.040.404-04	José

Dependente		
Matrícula	Num	Nome
1111	1	Pedro
1111	2	Ruth
2222	1	Rosa
3333	1	João

# Modelo Relacional - Chaves

- Chave Estrangeira na mesma tabela
  - EX:

Chave Primária

**simples**

Chave Estrangeira

**simples**

Empregado		
<u>CodEmpregado</u>	Nome	<u>CodChefe</u>
1010	Ana	
2020	Paulo	1010
3030	Fred	1010
4040	Silvia	2020

Auto-Relacionamento

- Chave Estrangeira - Validação pelo SGBD
  - Quando da **inclusão** de uma linha na tabela que contém a chave estrangeira
    - O valor da chave estrangeira deve aparecer na coluna da chave primária referenciada
  - Quando da **alteração** do valor da chave estrangeira
    - O novo valor de uma chave estrangeira deve aparecer na coluna da chave primária referenciada
  - Quando da **exclusão** de uma linha da tabela que contém a chave primária referenciada pela chave estrangeira
    - Na coluna chave estrangeira não deve aparecer o valor da chave primária que está sendo excluída

- Chave Alternativa
  - É a chave candidata que não foi escolhida como chave primária
  - EX:

## Chave Alternativa

**simples**

Cliente		
Matrícula	CPF	Nome
1111	101.010.101-01	Rita
2222	202.020.202-02	Ana
3333	303.030.303-03	Pedro
4444	404.040.404-04	José

Dependente		
Matrícula	Num	Nome
1111	1	Pedro
1111	2	Ruth
2222	1	Rosa
3333	1	João

**ATENÇÃO:**

**Chave Alternativa não faz  
relacionamento com  
Chave Estrangeira !**

# Modelo Relacional - Valor Vazio

- Um valor de campo pode assumir o valor especial vazio (“null” em inglês)
- Colunas nas quais não são admitidos valores vazios são chamadas de colunas obrigatórias
- Abordagem relacional
  - Todas colunas que compõem a chave primária devem ser obrigatórias
  - As colunas que compõem as demais chaves podem ser opcionais



- Objetivo primordial de um SGBD
  - Garantir a integridade de dados. Isto é, evitar que o BD entre em um estado inconsistente.
- Para garantir a integridade de um BD, os SGBD oferecem o mecanismo de restrições de integridade
- Uma restrição de integridade é uma regra de consistência de dados que é garantida pelo próprio SGBD
- Restrições de integridade são checadas pelo SGBD quando o BD sofre qualquer modificação.

- Restrições de integridade básicas
  - Restrições de Domínio
  - Restrições de Chaves
  - Integridade da Entidade
  - Restrições em Nulo
  - Integridade referencial
- As restrições acima são:
  - Garantidas automaticamente por um SGBD relacional
  - Não sendo exigido que o programador escreva procedimentos para garanti-las explicitamente

- Restrições de integridade básicas
  - **Restrições de Domínio**
    - Especifica que para uma coluna A de uma tabela, todo valor associado a A deve ser atômico e pertencer ao domínio desta coluna.
  - **Restrições de Chaves**
    - Especifica que todas as tuplas em uma relação devem ser distintas.
  - **Integridade da Entidade**
    - Especifica que os valores das chaves primárias não pode ser vazio.
  - **Restrições em Nulo**
    - Controla quais colunas de uma tabela podem receber valores nulos.
  - **Integridade referencial**
    - Especifica que os valores de uma chave estrangeira devem aparecer na chave primária da tabela referenciada

- Há muitas outras restrições de integridade que não se encaixam nas categorias básicas
- Essas restrições são chamadas de restrições semânticas (ou regras de negócio)
- Exemplos de restrições semânticas:
  - Um empregado não pode ter um salário maior que seu superior imediato.
- Diferente das restrições de integridade básicas, estas devem ser implementadas pelos programadores.

# Esquema Relacional

- Esquema Relacional = definição das tabelas
  - Representação básica (incompleta mas compacta)
  - Exemplo:

Agência (código, tipo, endereço, CEP, ...);

*Chave Primária*

Conta(número, saldo, dtAbertura, codAgência, ...)

codAgência referencia Agência

*Chave Estrangeira*

## Álgebra Relacional

“Linguagem de Consulta Relacional”

Por:

Robson do Nascimento Fidalgo

[rdnf@cin.ufpe.br](mailto:rdnf@cin.ufpe.br)

- Linguagens de consulta (*query languages*)
  - Permitem recuperar e manipular dados
- Modelo relacional suporta linguagens de consultas simples e poderosas
  - Forte embasamento formal → permiti desenvolvimento de algoritmos otimizados
- Não são linguagens de programação
  - Não são “computacionalmente completas”
    - Falta de operações de decisão e repetição.
    - Inadequadas para cálculos complexos
  - Oferecem acesso e manipulação fácil e eficiente de BDs

# Álgebra Relacional

- Álgebra = um conjunto de objetos + um conjunto de operações sobre estes.
  - EX: Aritmética → conj. de números + operações sobre os números
- Álgebra Relacional
  - Desenvolvida para descrever operações sobre uma BDR
    - Conhecimentos de álgebra relacional ajudam a entender SQL
  - Objetos: são as tabelas (relações)
  - Operações: são as definidas sobre conjuntos + as relacionais
- Principais operações:

<ul style="list-style-type: none"><li>▪ União</li><li>▪ Interseção</li><li>▪ Diferença</li><li>▪ Produto Cartesiano</li><li>▪ Complemento</li></ul>	} Operações sobre Conjuntos	<ul style="list-style-type: none"><li>– Seleção</li><li>– Projeção</li></ul>	} Operações Relacionais Unárias
		<ul style="list-style-type: none"><li>– Junção</li><li>– Divisão</li></ul>	} Operações Relacionais Binárias



# Operações sobre conjuntos

- Duas relações  $A(a_1, a_2, \dots, a_n)$  e  $B(b_1, b_2, \dots, b_n)$  são ditas compatíveis em domínio se ambas têm o mesmo grau  $n$  e se  $\text{Dom}(a_i) = \text{Dom}(b_i)$ ,  $1 \leq i \leq n$ .
  - Exemplo:
    - Aluno (nome, idade, curso)
    - Professor (nome, idade, curso)
    - Funcionario (nome, curso, idade)

$\text{Dom}(\text{nome}) = \text{char}(30)$   
 $\text{Dom}(\text{idade}) = \text{int}$   
 $\text{Dom}(\text{curso}) = \text{char}(5)$   
 $\text{Dom}(\text{depto.}) = \text{char}(5)$

**Aluno é compatível com Professor, mas não é com Funcionario.**

- Note que :
  - A estrutura de uma relação (tabela) é mais importante do que sua semântica.
  - A ordem dos atributos prevalece

# Operações sobre conjuntos

- As operações binárias sobre conjuntos da álgebra relacional podem operar sobre pares de relações compatíveis em domínio
- Tais operações são as usuais da teoria dos conjuntos:
  - União:  $(A \cup B) \Rightarrow$  Une as tuplas das relações A e B.
  - Interseção:  $(A \cap B) \Rightarrow$  Retorna as tuplas cujos valores sejam comuns às A e B
  - Diferença:  $(A - B) \Rightarrow$  Retorna as tuplas de A cujos valores não estão em B
  - Produto cartesiano:  $(A \times B) \Rightarrow$  Combinação de todas as tuplas das relações A e B
  - Complemento ou união exclusiva:  $(A \cup | B) \Rightarrow$  Retorna todas as tuplas de A ou a B, que não estão em ambas, ou seja,  $A \cup | B = A \cup B - A \cap B$ .
- OB: As operações de União, Interseção, Diferença e Complemento só aplicam-se a relações compatíveis em domínio

## Aluno (nome, idade, curso)

{José, 25, Computação; (José como aluno de Doutorado)  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação}

## Professor (nome, idade, curso)

{Ruth, 35, Computação;  
Rosa, 32, Química;  
José, 25, Computação}

- EX: Retornar todos os alunos e professores da Universidade

## Aluno $\cup$ Professor = (nome, idade, curso)

{José, 25, Computação;  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação;  
Ruth, 35, Computação;  
Rosa, 32, Química}

*Convenciona-se usar os nomes dos atributos da relação a esquerda, quando não especificado.*

Note que José só aparece uma vez !

# Interseção

## **Aluno (nome,idade,curso)**

{José, 25, Computação;  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação}

## **Professor (nome,idade,curso)**

{Ruth, 35, Computação;  
Rosa, 32, Química;  
José, 25, Computação}

- EX: Retornar todos que ao mesmo tempo sejam alunos e professores da Universidade

**Aluno  $\cap$  Professor = (nome,idade,curso)**

{José, 25, Computação}

# Diferença

## Aluno (nome,idade,curso)

{José, 25, Computação;  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação}

## Professor (nome,idade,curso)

{Ruth, 35, Computação;  
Rosa, 32, Química;  
José, 25, Computação}

- EX1: Retornar todos os alunos que não são professores
- EX2: Retornar todos os professores que não são alunos

## EX1: Aluno $\ominus$ Professor = (nome,idade,curso)

{ Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação}

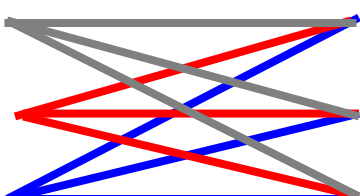
## EX2: Professor $\ominus$ Aluno = (nome,idade,curso)

{Ruth, 35, Computação;  
Rosa, 32, Química}

Note que a Diferença não é comutativa, ou seja,  $A-B \neq B-A$

# Produto Cartesiano

**Curso (curso, departamento)      Professor (nome, idade, curso)**

{ Computação, EC; Computação, CC; Matemática, MA}		{Ruth, 35, Computação; Rosa, 32, Química; José, 25, Computação}
---	---	---

- EX: Retornar todas as combinações entre os cursos e os professores da Universidade

**Curso X Professor = (curso, departamento, nome, idade, P.curso)**

{Computação, EC, Ruth, 35, Computação;  
 Computação, EC, Rosa, 32, Química;  
 Computação, EC, José, 25, Computação;  
 Computação, CC, Ruth, 35, Computação;  
 Computação, CC, Rosa, 32, Química;  
 Computação, CC, José, 25, Computação;  
 Matemática, MA, Ruth, 35, Computação;  
 Matemática, MA, Rosa, 32, Química;  
 Matemática, MA, José, 25, Computação}

# Complemento ou União Exclusiva

## Aluno (nome,idade,curso)

{José, 25, Computação;  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação}

## Professor (nome,idade,curso)

{Ruth, 35, Computação;  
Rosa, 32, Química;  
José, 25, Computação}

- EX: Retornar todos que ao mesmo tempo não são aluno e professor da Universidade

## $\text{Aluno} \cup \text{Professor} = (\text{nome}, \text{idade}, \text{curso})$

{ Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação;  
Ruth, 35, Computação;  
Rosa, 32, Química}

# Operações relacionais unárias

- Produzem como resultado uma nova relação que é um subconjunto (horizontal ou vertical) da relação origem
- São elas:
  - **Seleção:**  $(\sigma_{\langle \text{condição} \rangle}(\text{Relação}))$   $\Rightarrow$  seleciona tuplas de uma relação que satisfazem um dada condição.
    - Onde (Relação) é uma tabela ou uma expressão de álgebra relacional, e  $\langle \text{condição} \rangle$ , uma expressão booleana (and, or, not, =,  $\neq$ , <,  $\leq$ , >,  $\geq$ ) envolvendo atributos da tabela
    - Produz um subconjunto horizontal de uma relação
  - **Projeção:**  $(\pi_{\langle \text{atributos} \rangle}(\text{Relação}))$   $\Rightarrow$  seleciona de uma relação os atributos de interesse
    - Onde (Relação) é uma tabela ou uma expressão de álgebra relacional, e  $\langle \text{atributos} \rangle$ , uma lista de colunas da tabela operando
    - Produz um subconjunto vertical de uma relação



# Seleção

## Aluno (nome,idade,curso)

{José, 25, Computação;  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação}

## Professor (nome,idade,curso)

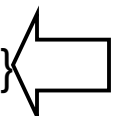
{Ruth, 35, Computação;  
Rosa, 32, Química;  
José, 25, Computação}

- EX: Retornar todos os alunos maiores de 20 anos do curso de computação da Universidade

$\sigma_{\langle \text{idade} \rangle > 20 \text{ and curso} = \text{“Computação”}}(\text{Aluno}) = (\text{nome,idade,curso})$   
 {José, 25, Computação}

- Outra solução:

$\sigma_{\langle \text{idade} \rangle > 20}(\sigma_{\langle \text{curso} = \text{“Computação”} \rangle}(\text{Aluno})) = (\text{nome,idade,curso})$

{José, 25, Computação}  {José, 25, Computação;  
Ana, 19, Computação}

# Projeção

## Aluno (nome,idade,curso)

{José, 25, Computação;  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação}

## Professor (nome,idade,curso)

{Ruth, 35, Computação;  
Rosa, 32, Química;  
José, 25, Computação}

- EX: Retornar todos os alunos e seus cursos

$\pi_{\langle \text{nome, curso} \rangle}(\text{Aluno}) = (\text{nome, curso})$

{José, Computação;  
Pedro, Química;  
Paulo, Física;  
Ana, Computação}

- OB: Na Projeção pode haver eliminação de linhas
- EX: Retornar todos os cursos que têm alunos matriculados na Universidade

$\pi_{\langle \text{curso} \rangle}(\text{Aluno}) = (\text{curso})$

{ Computação;  
Química;  
Física}

# Operações relacionais binárias

- Produz como resultado uma nova relação que é um subconjunto (seleção) do produto cartesiano das relações envolvidas
  - Em geral, após o produto cartesiano, é necessário comparar um ou um grupo de atributos de uma relação com um ou um grupo atributos (compatíveis em domínio) da outra relação.
- São elas:
  - **Junção:**
  - **Divisão:**

- Retorna apenas as tuplas do produto cartesiano de seus argumentos que satisfaçam uma dada condição
- **Sintaxe:**  $((R1) \bowtie_{\langle \text{condição} \rangle} (R2)) = (\sigma_{\langle \text{condição} \rangle} (R1 \times R2))$ 
  - Onde (R1) e (R2) são relações ou expressões de álgebra relacional, e  $\langle \text{condição} \rangle$ , uma expressão booleana envolvendo os atributos das duas relações.

# Junção

## Aluno (nome,idade,curso)

{José, 25, Computação;  
Pedro, 21, Química;  
Paulo, 19, Física;  
Ana, 19, Computação;  
**João, 34, Computação}**

## Professor (nome,idade,curso)

{Ruth, 35, Computação;  
Rosa, 32, Química;  
José, 25, Computação}

- EX: Retornar todos os alunos mais velhos do que qualquer professor da Universidade

$$((\text{Aluno}) \bowtie_{\langle \text{Aluno.idade} > \text{Professor.idade} \rangle} (\text{Professor}))$$

=

$$(\text{nome, idade, curso, nome, idade, curso})$$

{João, 34, Computação, Rosa, 32, Química  
João, 34, Computação, José, 25, Computação}

- Utilizada quando se deseja extrair de uma relação R1 um conjunto de valores que também existem na relação R2
- Produz uma relação R(X) com as tuplas de R1(A) que estão combinadas com todas as tuplas de R2(B), onde:  
 $B \subseteq A$  e  $X = A - B$
- Normalmente utilizada em operações que envolve “todos”
- **Sintaxe: (R1) ÷ (R2)**
  - Onde R1 e Relação2 são tabelas ou expressões de álgebra relacional

## Piloto (nome, avião)

{Pedro, 101;  
Pedro, 105;  
Bruno, 101;  
Bruno, 104;  
Bruno, 105;  
Bruno, 103;  
Paulo, 103;  
Paulo, 104}

## Avião (identificação)

{101;  
104;  
105;  
103}

- EX:Retornar os pilotos que estão habilitados para conduzir todos os aviões da companhia.

**(Piloto) ÷ (Avião) = (nome)**

**{ Bruno }**

## Matricula (nome-a, discipl, nota)

{José, IF111, 9,0;  
Pedro, IF333, 3,5;  
Paulo, IF111, 7,5;  
Paulo, IF333, 6,5;  
José, IF333, 10,0;  
José, IF222, 6,5;  
Ana, IF222, 7,0 }

## Aulas (nome-p, discipl)

{Lopes, IF111;  
Joana, IF222;  
Lopes, IF333}

- EX:Retornar os alunos que cursam todas as disciplinas ministradas pelo Prof. Lopes?

$\pi_{\langle \text{nome-a, discipl} \rangle}(\text{Matricula}) \div \pi_{\langle \text{discipl} \rangle}(\sigma_{\text{nome-p}=\text{"Lopes"}}(\text{Aulas}))$

**{ José , Paulo }**

- **Note:** Todas as operações podem ser combinadas entre si



## Normalização

Por:

Robson do Nascimento Fidalgo

[rdnf@cin.ufpe.br](mailto:rdnf@cin.ufpe.br)

# Normalização - Introdução

---

- Introduzida por E.F.Codd em 1970 (1FN, 2FN e 3FN)
- Processo matemático fundamentado na teoria dos conjuntos
- Aplica uma série de regras sobre as tabelas de um BD para verificar se estas foram bem projetadas

- Objetivo
  - Gerar um conjunto de esquemas de relações que permita:
    - Armazenar informações sem redundância desnecessária, e
    - Recuperar informações eficientemente.
  - Em resumo...
    - Evita anomalias de atualização e redundâncias no projeto do BDR
    - Permite representar eficientemente os dados do mundo real, tornando o modelo mais estável e de fácil manutenção

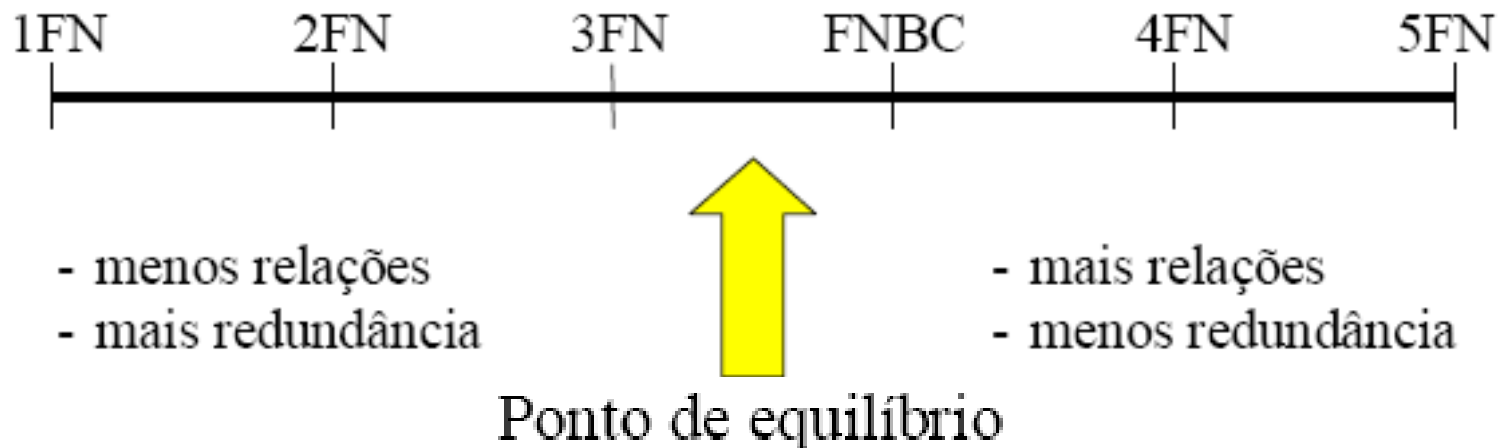
- Visão geral:
  - Converter progressivamente uma tabela em tabelas menores, até que pouca ou nenhuma redundância de dados exista
  - Processo sistemático de geração de tabelas

- Usada para validação do modelo relacional
- Seu benefício é melhor percebido quando aplicado em modelos relacionais obtidos por engenharia reversa de sistemas de arquivos
- Também pode ser usada para geração do modelo relacional a partir de documentos da organização
  - Gerar tabelas a partir de notas fiscais, históricos escolar, relatórios, etc.

- É baseada no conceito de formas normais (regras)
  - Existem 5 formas normais
    - 1FN, 2FN, 3FN, FNBC, 4FN e 5FN
- Do ponto de vista de desempenho, sua aplicação nem sempre é ideal.
  - Proliferação de tabelas
  - Usar o bom senso!

# Normalização - Introdução


- Aplicar normalização até qual forma normal?
  - Na prática o processo de normalização encerra-se na 3FN ou FNBC



- Inicialmente, usa-se o conceito de dependência funcional para expressar fatos acerca dos dados.

- Dependência Funcional (DF)
  - Sempre que um conjunto de colunas  $C_1$  identifica uma coluna ou um conjunto de colunas  $C_2$ , diz-se que há uma dependência funcional entre  $C_1$  e  $C_2$ , onde  $C_1$  é o determinante e  $C_2$  é o dependente.
  - Representação:
    - $C_1 \rightarrow C_2$  (lê-se:  $C_1$  determina  $C_2$  ou  $C_2$  é dependente de  $C_1$ )
  - EX: Empregado (CPF, Nome, DtNasc, Cargo, Gratificacao)
    - CPF  $\rightarrow$  Nome
    - CPF  $\rightarrow$  DtNasc
    - CPF  $\rightarrow$  Categoria
    - CPF  $\rightarrow$  Salario
    - CPF  $\rightarrow$  (Nome, Cargo, Gratificacao)

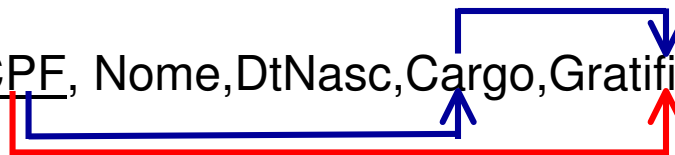


- Dependência Funcional Parcial (DFP)
  - Ocorre quando um conjunto de colunas é depende apenas de parte das colunas da ch. primária **composta**
    - EX:  
EmpProj(CPF,CodP,DtInicio,Nome,DtNas,Cargo,Gratificacao)  

- Dependência Funcional Total (DFT)
  - É o oposto da DFP
    - EX: (CPF,CodP) → (Cargo,Gratificacao)

- Dependência Funcional Transitiva (DFT)
  - Dada uma tabela qualquer, diz-se que existe DFT quando um conjunto de colunas  $C_3$  depende de uma coluna  $C_2$ , que não é ch. primária, mas que  $C_2$  depende funcionalmente da chave primária  $C_1$
  - Ou seja, se numa relação tivermos  $C_1 \rightarrow C_2$  e  $C_2 \rightarrow C_3$  então diz-se que  $C_3$  depende transitivamente de  $C_1$ , através de  $C_2$ .

- EX:

- Empregado (CPF, Nome, DtNasc, Cargo, Gratificacao)



- Diz-se que **Gratificacao** é dependente transitivo de **Cargo**. Então, se **Cargo** é determinado por **CPF**, este, indiretamente (transitivamente), também determina **Gratificacao**.

- 1ª Forma Normal (1FN)
  - Uma tabela esta na 1FN quando:
    - Os domínios de todas as suas colunas são atômicos. Ou seja, a tabela não pode ter atributo composto ou multivalorado
    - Em resumo: Uma tabela está na 1FN se nesta não houver uma tabela aninhada

# Normalização – Formas Normais

- 1ª Forma Normal (1FN)
  - Contra-exemplo (Tabelas aninhadas)

Projeto (CodP, Descricao, {Localizacao}, {Empregado(CPF, Nome, Cargo, Gratificacao)})

Projeto						
<u>CodP</u>	Descricao	Localizacao	Empregado ( <u>CPF</u> , Nome, Cargo, Gratificacao)			
1010	Tapa Buraco	Av. Sul	10101010	Paulo	Operador N1	2000
		Av. Norte	20202020	Rui	Operador N1	2000
		Av. Caxangá	30303030	Rita	Engenheiro	3000
		...	...	...	...	
2020	Asfaltamento	Av. Norte	30303030	Rita	Gerente	4000
		Av. Recife	40404040	João	Operador N2	2500
			50505050	Pedro	Operador N2	2500
		...	...	...	...	
...	...	...	...	...	...	...

**Atributo Multivalorado**

**Atributo Composto e Multivalorado**

# Normalização – Formas Normais

- 1ª Forma Normal (1FN)
  - Como transformar uma tabela para a 1FN
  - Atributo Composto:
    - Decompor o atributo composto em atributos simples e colocá-los:
      - Na mesma tabela (indicado quando o atributo composto é monovalorado). EX:
        - » Paciente (CPF, Nome, Endereco(Logradouro, CEP)) **× 1FN**
        - » Paciente (CPF, Nome, Logradouro, CEP) **✓ 1FN**
      - Em uma tabela relacionada (indicado quando o atributo composto é multivalorado). EX:
        - » Paciente (CPF, Nome, {Telefone(DDD, Prefixo, Sufixo)}) **× 1FN**
        - » Paciente (CPF, Nome) **✓ 1FN**
        - » PacienteTelefone (CPF, DDD, Prefixo, Sufixo) **✓ 1FN**  
CPF referencia Paciente

# Normalização – Formas Normais

- 1ª Forma Normal (1FN) - Transformação
  - Como transformar uma tabela para a 1FN
  - Atributo Multivalorado:
    - Decompor o atributo multivalorado em atributos simples e colocá-los:
      - Na mesma tabela (indicado quando a quantidade de valores é pequena e conhecida a priori). EX:
        - » Paciente (CPF, Nome, {GrausDeLente}) **× 1FN**
        - » Paciente (CPF, Nome, GrauLenteE, GrauLenteD) **✓ 1FN**
      - Em uma tabela relacionada (indicado quando a multivaloração é desconhecida ou grande). EX:
        - » Paciente (CPF, Nome, {ImagemRX}) **× 1FN**
        - » Paciente (CPF, Nome) **✓ 1FN**
        - » PacienteRX (CPF, ImagemRX) **✓ 1FN**  
CPF referencia Paciente

# Normalização – Formas Normais

- 1ª Forma Normal (1FN) - Transformação

Projeto						
CodP	Descricao	Localizacao	Empregado (CPF, Nome, Cargo, Gratificacao)			
1010	Tapa Buraco	Av. Sul	10101010	Paulo	Operador N1	2000
		Av. Norte	20202020	Rui	Operador N1	2000
		Av. Caxangá	30303030	Ana	Engenheiro	3000
		...	...	...	...	
2020	Asfaltamento	Av. Norte	30303030	Ana	Gerente	4000
		Av. Recife	40404040	João	Operador N2	2500
		...	50505050	Pedro	Operador N2	2500
		...	...	...	...	
...	...	...	...	...	...	...

× 1FN

Projeto		ProjetoLocalizacao		ProjetoEmpregado				
CodP	Descricao	CodP	Localizacao	CodP	CPF	Nome	Cargo	Gratificacao
1010	Tapa Buraco	1010	Av. Sul	1010	10101010	Paulo	Operador N1	2000
2020	Asfaltamento	1010	Av. Norte	1010	20202020	Rui	Operador N1	2000
...	...	1010	Av. Caxangá	1010	30303030	Ana	Engenheiro	3000
		2020	Av. Norte	2020	30303030	Ana	Gerente	4000
		2020	Av. Recife	2020	40404040	João	Operador N2	2500
		...		2020	50505050	Pedro	Operador N2	2500
					...	...	...	...

✓ 1FN


- 2ª Forma Normal (2FN)
  - Uma tabela esta na 2FN quando:
    - Está na 1FN,
    - A chave primária é **composta** e
    - Todas as colunas que não participam da chave primária são dependentes de todas as colunas que compõem a chave primária. Isto é, não existe DFP.



# Normalização – Formas Normais

- 2ª Forma Normal (2FN)
  - Contra-Exemplo (Dependência Funcional Parcial)

EmpregadoProjeto (CodP, CPF, Nome, Cargo, Gratificacao)



ProjetoEmpregado				
<u>CodP</u>	<u>CPF</u>	Nome	Cargo	Gratificacao
1010	10101010	Paulo	Operador N1	2000
1010	20202020	Rui	Operador N1	2000
1010	30303030	Rita	Engenheiro	3000
2020	30303030	Rita	Gerente	4000
2020	40404040	João	Operador N2	2500
2020	50505050	Pedro	Operador N2	2500
	...	...	...	...

- 2ª Forma Normal (2FN)
  - Como transformar uma tabela para a 2FN
    - Retira-se a(s) coluna(s) com DFP da tabela original
    - A partir dessa(s) coluna(s) retirada(s), cria-se uma ou mais tabelas compostas pela parte da chave primária e suas colunas dependentes
    - A parte da chave primária que gerou a dependência será a nova chave primária da tabela criada. EX:
      - Consulta(CPF, CRM, NomeP, NomeM, Especialidade, Tipo, Valor) ✗ **2FN**
      - Paciente (CPF, NomeP) ✓ **2FN**
      - Medico (CRM, NomeM, Especialidade) ✓ **2FN**
      - Consulta (CPF, CRM, Tipo, Valor) ✓ **2FN**  
CPF referencia Paciente  
CRM referencia Medico

# Normalização – Formas Normais

- 2ª Forma Normal (2FN)
  - Como transformar uma tabela para a 2FN

ProjetoEmpregado				
<u>CodP</u>	<u>CPF</u>	Nome	Cargo	Gratificacao
1010	10101010	Paulo	Operador N1	2000
1010	20202020	Rui	Operador N1	2000
1010	30303030	Rita	Engenheiro	3000
2020	30303030	Rita	Gerente	4000
2020	40404040	João	Operador N2	2500
2020	50505050	Pedro	Operador N2	2500
	...	...	...	...

× 2FN

Empregado		ProjetoEmpregado			
<u>CPF</u>	Nome	<u>CodP</u>	<u>CPF</u>	Cargo	Gratificacao
10101010	Paulo	1010	10101010	Operador N1	2000
20202020	Rui	1010	20202020	Operador N1	2000
30303030	Rita	1010	30303030	Engenheiro	3000
40404040	João	2020	30303030	Gerente	4000
50505050	Pedro	2020	40404040	Operador N2	2500
...	...	2020	50505050	Operador N2	2500
			...	...	...

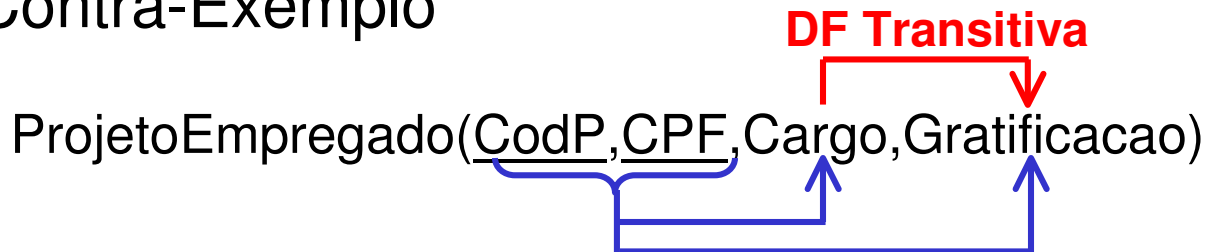
✓ 2FN

- 3ª Forma Normal (3FN)
  - Uma tabela esta na 3FN quando:
    - Está na 2FN e
    - Não contém DF Transitiva, sendo que todas as colunas que **não participam** da chave primária devem ser **exclusivamente** dependentes desta

# Normalização – Formas Normais

- 3ª Forma Normal (3FN)

- Contra-Exemplo



ProjetoEmpregado			
<u>CodP</u>	<u>CPF</u>	Cargo	Gratificacao
1010	10101010	Operador N1	2000
1010	20202020	Operador N1	2000
1010	30303030	Engenheiro	3000
2020	30303030	Gerente	4000
2020	40404040	Operador N2	2500
2020	50505050	Operador N2	2500
	...	...	...

- 3ª Forma Normal (3FN)
  - Como transformar uma tabela para a 3FN
    - Retira-se a(s) coluna(s) com DF Transitiva da tabela original
    - A partir dessa(s) coluna(s) retirada(s), cria-se uma ou mais tabelas compostas pela coluna determinante (como chave primária) + suas colunas dependentes
    - Verifica-se a 2FN para cada nova tabela. EX:
      - Consulta (CPF, CRM, Tipo, Valor) ✗ **3FN**
      - Consulta (CPF, CRM, Tipo) ✓ **3FN**  
Tipo referencia ConsultaTipo
      - ConsultaTipo (Tipo, Valor) ✓ **3FN**
    - OBS: Além de não conter DFT, as tabelas na 3FN não devem possuir colunas com valores calculados (derivados)

# Normalização – Formas Normais

- 3ª Forma Normal (3FN)
  - Como transformar uma tabela para a 3FN

ProjetoEmpregado			
<u>CodP</u>	<u>CPF</u>	Cargo	Gratificacao
1010	10101010	Operador N1	2000
1010	20202020	Operador N1	2000
1010	30303030	Engenheiro	3000
2020	30303030	Gerente	4000
2020	40404040	Operador N2	2500
2020	50505050	Operador N2	2500
	...	...	...

× 3FN

ProjetoEmpregado			CargoGratificacao	
<u>CodP</u>	<u>CPF</u>	Cargo	<u>Cargo</u>	Gratificacao
1010	10101010	Operador N1	Operador N1	2000
1010	20202020	Operador N1	Engenheiro	3000
1010	30303030	Engenheiro	Gerente	4000
2020	30303030	Gerente	Operador N2	2500
2020	40404040	Operador N2	...	...
2020	50505050	Operador N2	...	...
	...	...		

✓ 3FN

- Forma Normal BOYCE/CODD (FNBC)
  - É um refinamento da 3FN (usada em casos particulares)
    - Uma relação em FNBC é uma forma mais rigorosa do que a 3FN
    - Uma relação em FNBC está de acordo com a 3FN, mas o contrário não é verdade
  - Uma tabela esta na FNBC quando:
    - Está na 3FN e
    - **Todos** os determinante da tabela são chaves candidatas



# Normalização – Formas Normais

- Forma Normal BOYCE/CODD (FNBC)

- Contra-Exemplo

Localizacao (Cidade, Endereco, CEP) ✓ **3FN** ✗ **FNBC**



Localizacao		
<u>Cidade</u>	<u>Endereço</u>	CEP
Bom Jesus	Rod. da Paz, 2000	20.020-020
Bom Jesus	Rod. da Paz, 1000	20.020-020
Esperança	Rod. da Paz, 1000	30.030-030
Vitória	Av. Luz, 1000	40.040-040
Vitória	R. Glória, 1000	50.050-050
...	...	...

Não está na FNBC, pois CEP é determinante e não é chave candidata!

# Normalização – Formas Normais

- Forma Normal BOYCE/CODD (FNBC)
  - Como transformar uma tabela para a FNBC
    - Decompor a tabela (em duas ou mais), separando da tabela original as colunas que são determinantes e não são chaves candidatas, e as colunas dependentes destes determinantes
    - O determinante que não é chave candidata na tabela original deve fazer parte das chaves primárias das novas tabelas
    - Verifica-se a 3FN para cada nova tabela.

# Normalização – Formas Normais

- Forma Normal BOYCE/CODD (FNBC)
  - Como transformar uma tabela para a FNBC
    - Exemplo: **Avaliação** (Aluno, Disciplina, Professor, Media) ✓ **3FN** ✗ **FNBC**
      - Supondo que cada aluno tem um único professor por disciplina,
      - Cada professor só ensina uma única disciplina e
      - Uma disciplina pode ser ministrada por vários professores
    - Dependências Funcionais:
      - Aluno, Disciplina → Professor, Media
      - Aluno, Professor → Disciplina, Media
      - Professor → Disciplina (**Professor é determinante e não é chave candidata**)
    - Então:
      - ProfDisciplina (Professor, Disciplina) ✓ **FNBC**
      - ProfAluno (Professor, Aluno, Media) ✓ **FNBC**

# Normalização – Formas Normais

- Forma Normal BOYCE/CODD (FNBC)
  - Como transformar uma tabela para a FNBC

Localizacao		
Cidade	Endereço	CEP
Bom Jesus	Rod. da Paz, 2000	20.020-020
Bom Jesus	Rod. da Paz, 1000	20.020-020
Esperança	Rod. da Paz,1000	30.030-030
Vitória	Av. Luz, 1000	40.040-040
Vitória	R. Glória, 1000	50.050-050
...	...	...

✓ 3FN  
✗ FNBC

CEPCidade		EndereçoCEP	
CEP	Cidade	Endereço	CEP
20.020-020	Bom Jesus	Rod. da Paz, 2000	20.020-020
30.030-030	Esperança	Rod. da Paz, 1000	20.020-020
40.040-040	Vitória	Rod. da Paz,1000	30.030-030
50.050-050	Vitória	Av. Luz, 1000	40.040-040
...	...	R. Glória, 1000	50.050-050
...	...	...	...

✓ FNBC

- A construção de modelos ERs muitas vezes já encontram-se na 3FN, ficando as demais (FNBC, 4FN e 5FN) para serem avaliadas e aplicadas para casos particulares
- Desnormalização: Técnica usada para converter uma ou mais tabelas relacionadas em uma única tabela com informações possivelmente redundantes
  - Usada em casos particulares para evitar junções
  - Preocupar-se com a integridade dos dados redundantes
  - Muito usada no modelo estrela de um Data Warehouse

[cin.ufpe.br](http://cin.ufpe.br)



# Centro de **Informática**

U • F • P • E



UNIVERSIDADE FEDERAL DE PERNAMBUCO