

# Normalização de Banco de Dados



**Universidade Federal do Maranhão - UFMA**

**Departamento de Informática**

**Banco de Dados I**

Prof<sup>a</sup>.MSc Simara Rocha

[simararocha@gmail.com/simara@deinf.ufma.br](mailto:simararocha@gmail.com/simara@deinf.ufma.br)

<http://www.deinf.ufma.br/~simara>

Referências: Elmasri, R. and Navathe, S.B. Sistemas de Bancos de Dados. Editora Addison-Wesley, 6ª edição, 2001.  
Date, C.J. Introdução a Sistemas de Bancos de Dados. Editora Campus, 8ª edição, 2004.  
Korth, H.F. e Silberschatz, A. Sistemas de Bancos de Dados. Makron Books, 5ª edição, 2006.  
Notas de Aula do Prof. Msc. Tiago Eugenio de Melo



# Sumário

---

- Conceitos
- Dependência Funcional (DF)
- Regras para encontrar as DFs
- Normalização
- Formas Normais
- Conclusão



# Conceitos

---

- Normalização é um processo através do qual esquemas de relação, que não sejam satisfatórios às características do modelo relacional, são decompostos em esquemas menores que satisfaçam as propriedades desejáveis.
- A normalização inicialmente foi proposta como uma ferramenta de auxílio no projeto físico para a definição de relações



# Conceitos

---

- Porém na prática tornou-se uma ferramenta de verificação
  - Pois serve para verificar se os esquemas do projeto físico satisfazem algumas características básicas.
- Na normalização, são analisadas algumas medidas de qualidade para o projeto de um esquema de relação.
  - Estas medidas de qualidade visam, por exemplo, evitar um mau uso da memória.



# Conceitos

---

- As medidas são as seguintes:
  - Correta representação semântica:
    - os dados devem ser projetados de forma a terem seus significados bem definidos e coerentes com o que realmente querem representar.
  - Não geração de tuplas espúrias (sem sentido):
    - durante o processo de normalização deve-se atentar para evitar que sejam geradas tuplas que não façam sentido diante da realidade, isto pode ocorrer devido a alguma decomposição.



# Conceitos

---

- As medidas são as seguintes (cont.):
  - Redução de valores redundantes:
    - Sempre que possível deve-se reduzir ao máximo os valores redundantes desnecessários, ou seja, valores que muitas vezes aparecem repetidos quando isto não seria preciso.
    - Se os NULLs forem inevitáveis, garanta que eles se apliquem apenas em casos excepcionais e não na maioria das tuplas da relação.



# Conceitos

---

- Baseando-se nas medidas de qualidade já apresentadas, são definidas algumas diretrizes a serem seguidas durante o projeto, são elas:
  - D1: Projete um esquema de relação de tal forma que seja fácil explicar o seu *significado semântico*.
  - D2: Projete um esquema de relação de forma a evitar anomalias de atualização, para que possa ser evitada redundância de dados.



# Conceitos

---

- Diretrizes a serem seguidas durante o projeto, são elas (cont.)
  - D3: Sempre que possível, projete um esquema de relação de forma a evitar atributos que possam assumir valores nulos.
  - D4: Projete esquemas de relação de tal forma que as junções entre as relações correspondentes possam ser feitas através de condições de igualdade sobre atributos que são chaves primárias ou chaves estrangeiras de forma a garantir a não geração de tuplas espúrias.





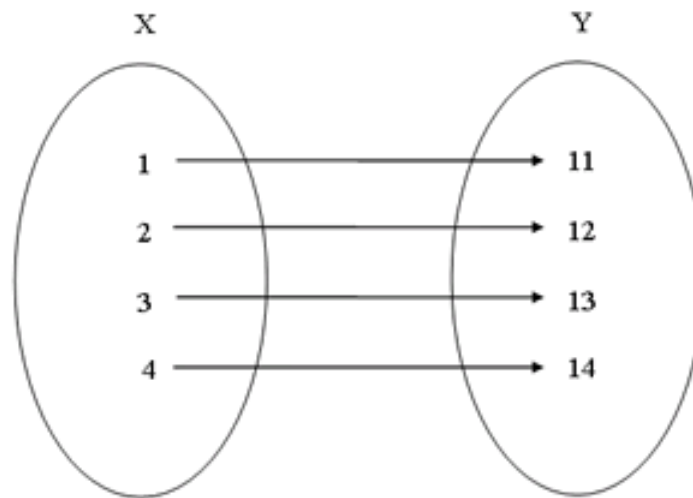
# Dependência Funcional

---

- O Modelo Relacional pegou emprestado da teoria de funções da matemática o conceito de dependência funcional
- Iremos utilizar então a teoria de funções para explicar a dependência funcional do Modelo Relacional.

# Dependência Funcional

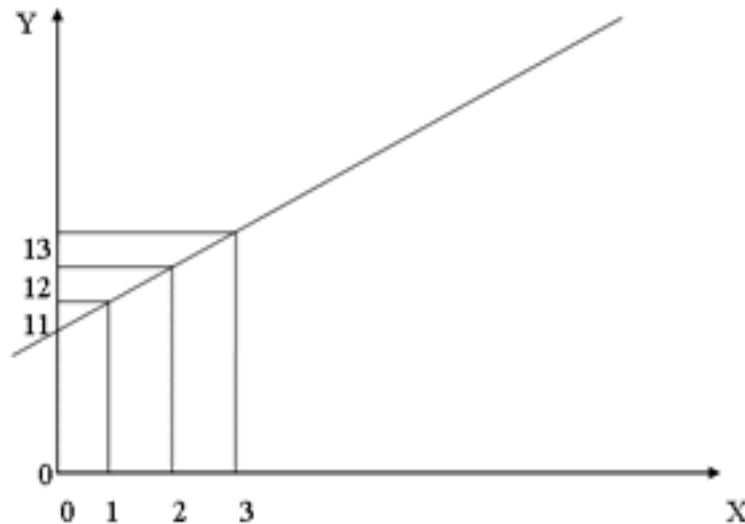
- Considerando os seguintes conjuntos:



- Observe que existe uma dependência entre os valores dos conjuntos, que pode ser expressa pela função  $f(x) = x + 10$ , ou seja,  $y$  é função de  $x$ , ou seja,  $y = f(x) = x + 10$ .

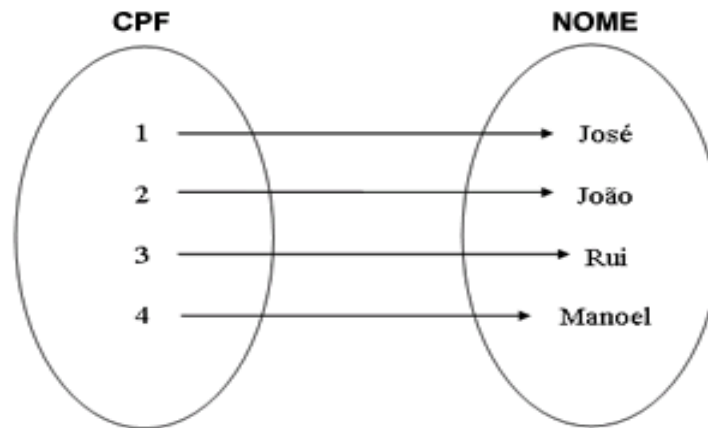
# Dependência Funcional

- Esta dependência (função) pode também ser expressa através do gráfico abaixo:



# Dependência Funcional

- Agora, observe os conjuntos abaixo:



- Observe que existe uma dependência entre os valores dos conjuntos, que pode ser expressa pela função  $f(\text{CPF})=\text{nome}$ .
  - ou seja, nome é função do CPF, se tivermos um número de CPF, poderemos encontrar o nome da pessoa correspondente.



# Dependência Funcional

---

- É claro que não existe uma figura gráfica que possa descrever esta função, mas ela existe
- Esta dependência é expressa no Modelo Relacional da seguinte maneira:
  - CPF  $\rightarrow$  NOME
- Leia-se a notação acima das seguintes maneiras:
  - Com um número de CPF posso encontrar o nome da pessoa
  - Nome depende funcionalmente do CPF.



# Dependência Funcional

---

- Restrição entre dois conjuntos de atributos do banco de dados:

**Definição.** Uma dependência funcional, indicada por  $X \rightarrow Y$ , entre dois conjuntos de atributos  $X$  e  $Y$  que são subconjuntos de  $R$ , especifica uma *restrição* sobre possíveis tuplas que podem formar um estado de relação  $r$  de  $R$ . A restrição é que, para quaisquer duas tuplas  $t_1$  e  $t_2$  em  $r$  que tenham  $t_1[X] = t_2[X]$ , elas também devem ter  $t_1[Y] = t_2[Y]$ .

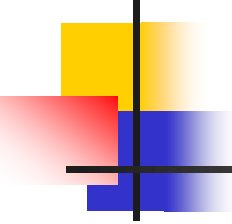
- Propriedade da semântica ou significado dos atributos



# Dependência Funcional

---

- Sejam os seguintes subconjuntos de atributos de um esquema T:
  - $A = (A_1, A_2, \dots, A_n)$  e  $B = (B_1, B_2, \dots, B_n)$
- Dizemos que B é dependente funcionalmente de um outro atributo A contido em T se a cada valor de A existir nas linhas da relação T, em que aparece, um único valor de B.
  - Notação:  $A \Rightarrow B$



# Regras para encontrar Dependências Funcionais

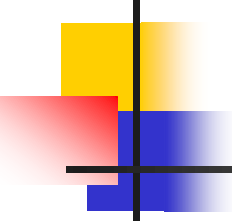
---

## 1. Separação

**$A \rightarrow BC$  então  $A \rightarrow B$  e  $A \rightarrow C$**

- Exemplo:
  - $CPF \rightarrow nome, endereço$  então  $CPF \rightarrow nome$  e  $CPF \rightarrow endereço$
- Leia o exemplo acima da seguinte maneira:
  - Se com um número de CPF encontramos o nome e o endereço de uma pessoa, então com este mesmo número podemos encontrar apenas o nome e com este mesmo número podemos encontrar apenas o endereço





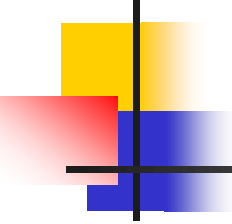
# Regras para encontrar Dependências Funcionais

---

## 2. Acumulação

**$A \rightarrow B$  então  $AC \rightarrow B$**

- Exemplo:
  - $CPF \rightarrow endereço$  então  $CPF, idade \rightarrow endereço$
- Leia o exemplo acima da seguinte maneira:
  - Se com um número de CPF encontramos o endereço de uma pessoa, então com este mesmo número mais a idade da pessoa podemos encontrar o endereço também



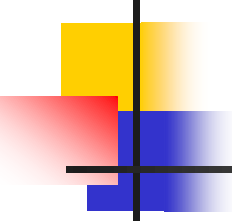
# Regras para encontrar Dependências Funcionais

---

## 3. Transitividade

**$A \rightarrow B$  e  $B \rightarrow C$  então  $A \rightarrow C$**

- Exemplo:
  - $\text{CPF} \rightarrow \text{código-cidade}$  e  $\text{código-cidade} \rightarrow \text{nome-cidade}$  então  $\text{CPF} \rightarrow \text{nome-cidade}$
- Leia o exemplo acima da seguinte maneira:
  - Se com um número de CPF encontramos o código da cidade de uma pessoa, e com o código da cidade encontramos o nome da cidade, então com o número do CPF podemos encontrar o nome da cidade.



# Regras para encontrar Dependências Funcionais

---

## 4. Pseudo-Transitividade

**$A \rightarrow B$  e  $BC \rightarrow D$  então  $AC \rightarrow D$**

- Exemplo:
  - $CPF \rightarrow \text{código-funcionário}$  e  $\text{código-funcionário, mês} \rightarrow \text{salário-funcionário}$  então  
 $CPF, \text{mês} \rightarrow \text{salário-funcionário}$
- Leia o exemplo acima da seguinte maneira:
  - Se com um número de CPF encontramos o código do funcionário, e com o código do funcionário mais um certo mês encontramos o salário que ele recebeu naquele mês, então com o número do CPF mais um certo mês podemos encontrar o salário que ele recebeu naquele mês.



# Normalização

---

- O processo de Normalização, proposto primeiramente por Codd, faz uma série de testes para certificar se um Esquema Relacional satisfaz a uma Forma Normal.
- Cada Relação é avaliada e decomposta em novas Relações, se necessário.
  - Projeto Relacional por Análise.
- Inicialmente, Codd propôs três formas normais.



# Normalização

---

- Propriedades que os esquemas relacionais devem ter:
  - Propriedade de junção não aditiva ou junção sem perdas
    - Extremamente crítica e deve ser alcançada a todo custo
  - Propriedade de preservação de dependência
    - Desejável, mas às vezes é sacrificada por outros fatores



# Normalização

---

- Uso prático das formas normais
  - A normalização é executada na prática:
    - De modo que os projetos resultantes sejam de alta qualidade e que atendam as propriedades desejáveis indicadas anteriormente
  - Embora várias formas normais mais altas tenham sido definidas
    - A utilidade prática destas torna-se questionável quando as restrições sobre as quais elas estão baseadas são raras e difíceis de entender ou detectar pelos projetistas e usuários de BD



# Normalização

---

- Uso prático das formas normais (cont.)
  - Outro ponto que merece ser observado é que os projetistas de BD não precisam normalizar para a forma normal mais alta possível
    - As relações podem ser deixadas em um estado de normalização inferior, por questões de desempenho
    - Porém, fazer isso gera as penalidades correspondentes a lidar com as anomalias

**Definição.** Desnormalização é o processo de armazenar a junção de relações na forma normal mais alta como uma relação da base, que está em uma forma normal mais baixa.



# Normalização

---

- Consequências:
  - Problemas de anomalias e inconsistências diminuem.
  - Relações simplificadas e estrutura regular.
  - Aumento da integridade dos dados.
  - Necessidade de realização de junções.
  - Eventual queda na performance.





# Formas Normais - 1FN

---

- Parte da definição formal de uma relação no modelo relacional básico (plano)
- Uma relação R está na 1FN se e somente se, em todo valor válido dessa relação, cada tupla contém exatamente um valor para cada atributo.
- Os únicos valores de atributo permitidos são os valores atômicos (ou indivisíveis)



# Formas Normais - 1FN

---

- Técnicas principais para conseguir a primeira forma normal
  - Remover o atributo e colocá-lo em uma relação separada
  - Expandir a chave
  - Usar vários atributos atômicos



# Formas Normais - 1FN

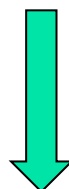
---

- Não permite relações aninhadas
  - Cada tupla pode ter uma relação dentro dela
- Para alterar a 1FN:
  - Remova os atributos da relação aninhada para uma nova relação
  - Propague a chave primária para ela
  - Desaninhar a relação para um conjunto de relações 1FN

# Formas Normais - 1FN

**Tabela: PEDIDO (não normalizada)**

NumPed	DataEmis	Fornecedor	CGC	End	CodProd	NomeProd	Qtde	Preço
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	111	Prod 1	10	R\$100.00
					222	Prod 2	44	R\$150.00
					333	Prod 3	50	R\$120.00
4	Feb 10	Computer	2222222-22	Itu 49	222	Prod 4	73	R\$150.00
					333	Prod 5	80	R\$120.00



**Tabela: PEDIDO – com Atributos Atômicos**

NumPed	DataEmis	Fornecedor	CGC	Bairro	Rua	Cod Prod	NomeProd	QtdeProd
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	111	Prod 1	10
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	222	Prod 2	44
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	333	Prod 3	50
4	Feb 10	Computer	2222222-22	Itu	Rua 49	222	Prod 2	73
4	Feb 10	Computer	2222222-22	Itu	Rua 49	333	Prod 3	80



# Formas Normais - 1FN

**Tabela: PEDIDO - com Atributos Atômicos**

NumPed	DataEmis	Fornecedor	CGC	Bairro	Rua	Cod Prod	NomeProd	QtdeProd
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	111	Prod 1	10
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	222	Prod 2	44
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	333	Prod 3	50
4	Feb 10	Computer	2222222-22	Itu	Rua 49	222	Prod 2	73
4	Feb 10	Computer	2222222-22	Itu	Rua 49	333	Prod 3	80



**Tabela: PEDIDO - com Chave Primária: NumPed+CodProd**

<u>NumPed</u>	<u>Cod Prod</u>	DataEmis	Fornecedor	CGC	Bairro	Rua	NomeProd	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 1	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 2	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 3	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 2	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 3	80



# Formas Normais - 2FN

---

- Baseada no conceito de dependência funcional total

- Versus dependência parcial

**Definição.** Um esquema de relação  $R$  está em 2FN se cada atributo não principal  $A$  em  $R$  for *total e funcionalmente dependente* da chave primária de  $R$ .

- Um esquema de relação está na 2FN se: estiver na 1FN e, além disso, todo atributo que não pertença a alguma de suas chaves for totalmente dependente da sua chave primária.



# Formas Normais - 2FN

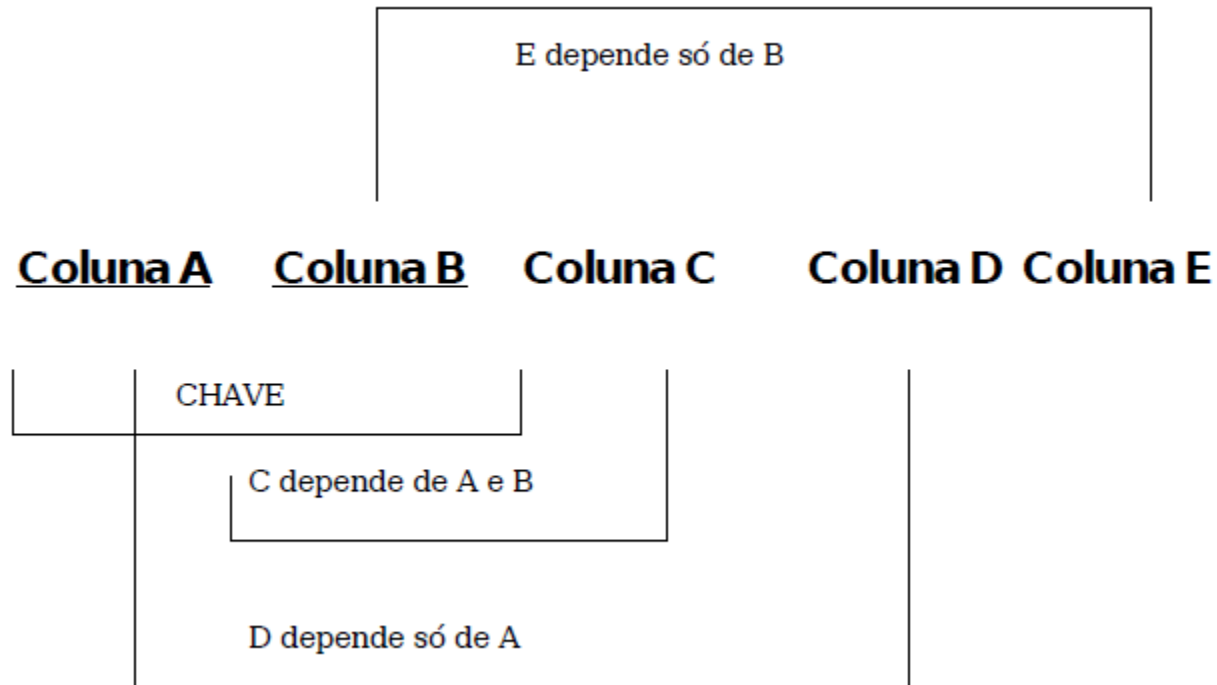
---

- Baseada no conceito de dependência funcional total (cont.)
  - Em outras palavras, para que uma relação esteja na 2FN é preciso que esteja na 1FN e que, havendo uma chave primária composta, todos os dados que não são chaves dependem de toda a chave primária (a chave primária completa).



# Formas Normais - 2FN

- Esquemáticamente falando...







# Formas Normais - 2FN

---

- Processo para obtenção da 2FN
  - identificar as colunas que não participam da chave primária da tabela
  - para cada uma das colunas identificadas, analisar se seu valor é determinado por parte, ou pela totalidade da chave
  - para as colunas dependentes parcialmente:
    - criar novas tabelas onde a chave primária será(ão) a(s) coluna(s) da chave primária original que determinou o valor da coluna analisada
    - excluir da tabela original as colunas dependentes parcialmente da chave



# Formas Normais - 2FN

**Tabela na 1FN: Pedido**

<u>NumPed</u>	<u>Cod Prod</u>	<u>DataEmis</u>	<u>Fornecedor</u>	<u>CGC</u>	<u>Bairro</u>	<u>Rua</u>	<u>NomeProd</u>	<u>QtdeProd</u>	<u>PreçoProd</u>
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 1	10	R\$100.00
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 2	44	R\$150.00
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 3	50	R\$120.00
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 2	73	R\$150.00
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 3	80	R\$120.00

**Tabela na 2FN: Pedido**

<u>NumPed</u>	<u>Cod Prod</u>	<u>DataEmis</u>	<u>Fornecedor</u>	<u>CGC</u>	<u>Bairro</u>	<u>Rua</u>	<u>QtdeProd</u>
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	80

**Tabela na 2FN: Produto**

<u>Cod Prod</u>	<u>NomeProd</u>	<u>PreçoProd</u>
111	Prod 1	R\$100.00
222	Prod 2	R\$150.00
333	Prod 3	R\$120.00



# Formas Normais - 3FN

---

- Baseada no conceito de dependência transitiva

**Definição.** De acordo com a definição original de Codd, um esquema de relação  $R$  está na 3FN se ele satisfizer a 2FN e nenhum atributo não principal de  $R$  for transitivamente dependente da chave primária.

- Ou seja, um esquema de relação está na 3FN se:
  - estiver na 2FN e, além disso, nenhum atributo que não pertença a alguma das suas chaves for transitivamente dependente da sua chave primária



# Formas Normais - 3FN

---

- Em outras palavras, para que uma relação esteja na 3FN é preciso que esteja na 2FN e todo atributo, que não pertença a alguma chave for não dependente de algum outro atributo, que também não pertença a alguma chave.



# Formas Normais - 3FN

---

- Esquemáticamente falando...

C determina D



<u>Coluna A</u>	Coluna B	Coluna C	Coluna D	Coluna E

A determina B

A determina C

A determina E



# Formas Normais - 3FN

**Tabela na 2FN: Pedido**

<u>NumPed</u>	<u>Cod Prod</u>	<u>DataEmis</u>	<u>Fornecedor</u>	<u>CGC</u>	<u>Bairro</u>	<u>Rua</u>	<u>QtdeProd</u>
3	111	20/jan	Casa Software	1111111-11	Lapa	Rua A	10
3	222	20/jan	Casa Software	1111111-11	Lapa	Rua A	44
3	333	20/jan	Casa Software	1111111-11	Lapa	Rua A	50
4	222	10/Fev	Computer	2222222-22	Itu	Rua 49	73
4	333	10/Fev	Computer	2222222-22	Itu	Rua 49	80

**Tabela na 3FN: Pedido**

<u>NumPed</u>	<u>Cod Prod</u>	<u>DataEmis</u>	<u>CGC</u>	<u>QtdeProd</u>
3	111	20/jan	1111111-11	10
3	222	20/jan	1111111-11	44
3	333	20/jan	1111111-11	50
4	222	10/Fev	2222222-22	73
4	333	10/Fev	2222222-22	80

**Tabela na 3FN: Produto**

<u>Cod Prod</u>	<u>NomeProd</u>	<u>PreçoProd</u>
111	Prod 1	R\$100,00
222	Prod 2	R\$150,00
333	Prod 3	R\$120,00

**Tabela na 3FN: Fornecedor**

<u>CGC</u>	<u>Fornecedor</u>	<u>Bairro</u>	<u>Rua</u>
1111111-11	Casa Software	Lapa	Rua A
2222222-22	Computer	Itu	Rua 49



# Anomalias - 1FN

- Considere a tabela Empregados, sendo chave primária os campos Matrícula e CodProj.

Matrícula	Nome	CodCargo	NomeCargo	CodProj	DataFim	Horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10
270	Gabriel	2	Analista	12	21/03/96	38
273	Silva	3	Projetista	01	17/07/95	22
274	Abraão	2	Analista	12	21/03/96	31



# Anomalias - 1FN

---

- Anomalias:

- Inserir: não é possível inserir um empregado sem que este esteja alocado num projeto, nem inserir um projeto sem que haja um empregado trabalhando nele.
- Remover: se for necessário remover um projeto, as informações de empregado que estiverem alocados naquele projeto serão perdidas.
- Atualizar: se um empregado for promovido de cargo, teremos que atualizar os atributos CodCargo e NomeCargo em todas as tuplas nas quais aquele empregado está presente.





# Anomalias - 2FN

---

## EMPREGADO

Matrícula	Nome	CodCargo	NomeCargo
120	João	1	Programador
121	Hélio	1	Programador
270	Gabriel	2	Analista
273	Silva	3	Projetista
274	Abraão	2	Analista

## ALOCAÇÃO

Matrícula	CodProj	Horas
120	01	37
120	08	12
121	01	45
121	08	21
121	12	107
270	08	10
270	12	78
273	01	22
274	12	31

## PROJETO

CodProj	DataFim
01	17/07/95
08	12/01/96
12	21/03/96



# Anomalias - 2FN

---

- Anomalias:

- Inserir: só é possível criar cargos se houver empregados designados para eles.
- Remover: se apagarmos um empregado que ocupa unicamente um cargo na empresa, perderemos a informação do cargo.
- Atualizar: se um cargo mudar de nome, será necessário mudar todas as tabelas em que este cargo aparece.



# Anomalias - 3FN

---

## EMPREGADO

<u>Matrícula</u>	Nome	CodCargo
120	João	1
121	Hélio	1
270	Gabriel	2
273	Silva	3
274	Abraão	2

## CARGO

<u>CodCargo</u>	Nome
1	Programador
2	Analista
3	Projetista



# Formas Normais – Boyce-Codd

---

- Foi proposta como uma forma mais simples de 3FN, mas é considerada mais rígida que a 3FN
  - isto é, toda relação na BCNF está na 3FN, porém uma relação na 3FN não está necessariamente na BCNF



# Formas Normais – Boyce-Codd

---

- A 3FN não tratou satisfatoriamente casos onde uma relação tem mais de uma chave candidata e estas chaves são compostas e possuem atributos em comum.
- Uma relação está em BCNF se todo determinante for uma chave candidata



# Formas Normais – Boyce-Codd

Estudante	Disciplina	Professor
100	Matemática	José
100	Física	Maria
200	Matemática	José
200	Física	Roberto

- Essa relação tem o seguinte significado:
  - Para cada disciplina, cada estudante recebe aula de apenas um Professor.
  - Cada professor ensina somente uma disciplina.
  - Uma disciplina pode ser ensinada por diversos professores.



# Formas Normais – Boyce-Codd

## Chaves Candidatas:

(estudante, disciplina)

(estudante, professor)

## Dependências Funcionais:

(estudante, disciplina)  $\rightarrow$  professor

Professor  $\rightarrow$  disciplina

Estudante	Disciplina	Professor
100	Matemática	José
100	Física	Maria
200	Matemática	José
200	Física	Roberto

- Se a PK for (estudante, disciplina) a relação está em 3FN a mesma coisa para (estudante, professor).
- Em ambos os casos, a relação não está em BCNF porque o determinante Professor não é uma chave candidata.



# Formas Normais – Boyce-Codd

---

- Processo para obtenção da BCFN:
  1. Identificar as dependências funcionais que violem a BCNF.
  2. Para cada dependência funcional achada em 1, criar uma relação com a PK igual ao determinante.
  3. As colunas que têm seu valor determinado em 1, são excluídas da relação original.





# Formas Normais – Boyce-Codd

---

ENSINO (estudante, professor)

Estudante	Professor
100	Jose
100	Maria
200	Jose
200	Roberto


LECIONA (professor, disciplina)

Professor	Disciplina
Jose	Matemática
Maria	Física
Roberto	Física



# Formas Normais – Boyce-Codd

<u>Id_Propried</u>	Nome_Região	Lote	Área
Pr1	Central	L23I	500
Pr2	Central	L14	500
Pr3	Sul	L2	1000
Pr4	Sul	L54	1000
Pr5	Leste	L400	800
Pr6	Norte	L43I	1500
Pr7	Central	L414	1500



Id\_propried → Nome\_região, Lote, Área  
Nome\_região, Lote → Id\_propried, Área  
Área → Nome\_região

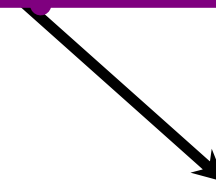


# Formas Normais – Boyce-Codd

<u>Id_Propried</u>	Nome_Região	Lote	Área
Pr1	Central	L23I	500
Pr2	Central	L14	500
Pr3	Sul	L2	1000
Pr4	Sul	L54	1000



<u>Id_Propried</u>	Lote	Área
Pr1	L23I	500
Pr2	L14	500
Pr3	L2	1000
Pr4	L54	1000



<u>Área</u>	Nome_Região
500	Central
1000	Sul



# Formas Normais - 4FN

---

- Uma tabela encontra-se na quarta forma normal, quando, além de estar na 3FN, não contém dependências multivaloradas.
  - Em outras palavras, uma tabela está na 4FN caso, além de estar na 3FN, não possua mais que uma dependência funcional multivalorada.
- Dependência Multivalorada
  - Uma coluna ou conjunto de colunas depende multivaloradamente de uma coluna (determinante) da mesma tabela quando um valor do atributo determinante identifica repetidas vezes um conjunto de valores na coluna dependente

# Formas Normais - 4FN

<u>Piloto</u>	<u>Avião</u>	<u>Trajeto</u>
0020	101	Rec-Rio
0020	105	Rio-Spa
0020	105	Spa-Rec
0020	101	Spa-Rec
0020	101	Rio-Spa
0020	105	Rec-Rio
0010	101	Rec-For
0010	104	Rec-For
0015	103	Rio-Spa

- A relação Vão não está na 4FN pois existem dependência multivaloradas

- Piloto  $\twoheadrightarrow$  Avião
- Piloto  $\twoheadrightarrow$  Trajeto

- Como corrigir?

- Separar a relação em relações, cada uma contendo um atributo (A) que multidetermina os outros (B,C), ou seja R1 (A, B) e R2 (A, C)



# Formas Normais - 4FN

---

Vôo1

<u>Piloto</u>	<u>Avião</u>
0020	101
0020	105
0010	101
0010	104
0015	103

Vôo2

<u>Piloto</u>	<u>Trajeto</u>
0020	Rec-Rio
0020	Rio-Spa
0020	Spa-Rec
0010	Rec-For
0015	Rio-Spa



## Formas Normais - 5FN

---

- Uma relação de 4FN estará em 5FN, quando seu conteúdo não puder ser reconstruído (existir perda de informação) a partir das diversas relações menores que não possuam a mesma chave primária.
- Esta forma normal trata especificamente dos casos de perda de informação, quando da decomposição de relacionamentos múltiplos.



## Formas Normais - 5FN

---

- A 5FN trata de casos bastante particulares, que ocorrem na modelagem de dados, que são os relacionamentos múltiplos (ternários, quaternários e n-ários).
- A 5NF trata de fatos multivalorados dependentes, enquanto que a 4NF trata de fatos independentes.





# Formas Normais - 5FN

---

- Um registro está na sua 5FN, quando o conteúdo deste mesmo registro não puder ser reconstruído (junção) a partir de outros registros menores, extraídos deste registro principal.
  - Ou seja, se ao particionar um registro, e sua junção posterior não conseguir recuperar as informações contidas no registro original, então este registro está na 5FN.



## Formas Normais - 5FN

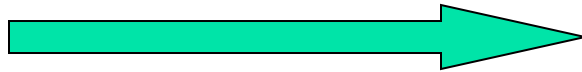
---

Vendedor	Fabricante	Veiculo
Smith	Ford	Automóvel
Smith	Ford	Caminhão
Smith	GM	Automóvel
Smith	GM	Caminhão
Jones	Ford	Automóvel

- Se um vendedor vende um certo tipo de veículo e ele representa o fabricante daquele tipo de veículo, então ele vende aquele tipo de veículo para aquele fabricante, a relação acima pode ser decomposta em três outras relações, e portanto não está na 5NF.

# Formas Normais - 5FN

Transformando em 5FN,  
teremos:



- As relações ao lado não podem ser decompostas, estando assim na 5NF. Para recompor a mesma informação serão necessários as três relações.

Vendedor	Fabricante
Smith	Ford
Smith	GM
Jones	Ford

Vendedor	Veiculo
Smith	Automóvel
Smith	Caminhão
Jones	Automóvel

Fabricante	Veiculo
Ford	Automóvel
Ford	Caminhão
GM	Automóvel
GM	Caminhão



# Conclusão

---

- Normalizar evita introduzir inconsistências quando se alteram relações; porém obriga a execução de custosas operações de junção para a consulta de informações.
- Mas, e aí? Normalizar ou não normalizar?
- A decisão deve ser tomada considerando-se o compromisso entre se garantir a eliminação de inconsistências na base, e eficiência de acesso.