

# Banco de Dados

Prof. Anthony Ferreira La Marca

[anthony@computacao.cua.ufmt.br](mailto:anthony@computacao.cua.ufmt.br)

# Projeto de BD

- Objetivo
  - Gerar um conjunto de esquemas de relações
    - Sem redundância desnecessária
    - Que permita recuperar informações de forma fácil
- Projetar esquemas na forma normal apropriada

Para atingir esses objetivos



Conceitos de normalização

# Projeto de BD

- Para atingir um bom projeto de BD
  - Decompor uma relação em relações menores
- Decomposição
  - Realizada através de dependências funcionais

# Dependência Funcional

- Baseia-se no reconhecimento que os valores de alguns atributos podem ser **determinados** a partir de outros
- Esse conhecimento não pode ser inferido pelo SGBD
- Deve ser identificado durante a fase de Projeto do BD

# Dependência Funcional

✓ Exemplo:

•  $RA \rightarrow \text{nome, idade}$

**Aluno (RA, nome, disciplina, idade)**

{2, Rodolfo, CC-302, 20  
2, Rodolfo, CC-304, 20  
10, Eduardo, CC-304, 21  
10, Eduardo, CC-308, 21  
25, Vanessa, CC-304, 20  
32, César, CC-304, 22  
38, Mariana, CC-303, 21  
54, Érica, CC-302, 20 }

Sempre que o **RA** se repete,  
**nome** e **idade** se repetem?

**SIM !!!**

**nome** e **idade** SÃO  
FUNCIONALMENTE DEPENDENTES  
de **RA**

# Dependência Funcional

## ✓ Exemplo:

▪ **idade** → **disciplina**

**Aluno** (**RA**, **nome**, **disciplina**, **idade**)

{2, Rodolfo, CC-302, 20  
2, Rodolfo, CC-304, 20  
10, Eduardo, CC-304, 21  
10, Eduardo, CC-308, 21  
25, Vanessa, CC-304, 20  
32, César, CC-304, 22  
38, Mariana, CC-303, 21  
54, Érica, CC-302, 20 }

Sempre que o **idade** se repete,  
**disciplina** se repete?

**NÃO !!!**

**disciplina NÃO É**  
FUNCIONALMENTE DEPENDENTE  
de **idade**

# Dependência Funcional

- São informações **semânticas** fornecidas pelo projetista
- São usadas para minimizar redundância
- Minimizar anomalias em atualizações
- Contruir um BD mais eficiente

# Processo de Normalização

- Proposto por Codd e 1972
- Processo para simplificar as relações através de formais normais
  - Regras que devem ser obedecidas
- Inicialmente codd propôs 3 formas normais
  - 1FN
  - 2FN
  - 3FN
- Posteriormente Codd e Boyce proporam
  - FNBC



# Processo de Normalização

- Todas essas formas normais são baseadas no conceito de dependência funcional entre atributos de uma relação
- Posteriormente surgiram a 4FN e a 5FN
  - Dependências Multivaloradas
  - Dependências de junção

# Normalização

➤ Como deixar as relações na 1FN:

Se existirem atributos compostos:

- Substituí-los por atributos atômicos

Exemplo:

Aluno (RA, nome, endereço)

Aluno (RA, nome, rua , número , bairro , cidade , estado)

# Normalização

➤ Como deixar as relações na 1FN:

Se existirem atributos multi-valorados:

**Duas  
situações**

Quantidade de valores é  
pequena e conhecida  
previamente

Quantidade de valores é  
desconhecida, grande ou  
variável.

# Normalização

## ➤ Como deixar as relações na 1FN:

- Quantidade de valores é **pequena e conhecida previamente**
  - ➔ Substitui-se o atributo multi-valorado por um **conjunto de atributos de mesmo domínio**, cada um monovalorado representando uma ocorrência do valor.

### Exemplo:

Aluno (RA, nome, {notas-bimestrais})

Aluno (RA, nome, nota1, nota2, nota3, nota4)



# Normalização

## ➤ Como deixar as relações na 1FN:

- Quantidade de valores é desconhecida, grande ou variável.
  - ➡ retira-se da relação o atributo multi-valorado
  - ➡ cria-se uma nova relação que tem o mesmo conjunto de atributos chave, mais o atributo multivalorado. Verificar a formação da chave primária.

Exemplo:

Aluno (RA, nome, {disciplinas-matriculadas})

# Normalização

## ➤ Como deixar as relações na 1FN:

- Quantidade de valores é desconhecida, grande ou variável.

### Exemplo:

Aluno (RA, nome, {disciplinas-matriculadas})



Aluno (RA, nome)

Matricula (RA, disciplina-matriculada)

# Normalização

- Existe cassos ainda que temos atributos multivalorados compostos
- Chamados de relações aninhadas
- Pois cada tupla pode ter uma relação dentro dela
- Ex: Func:Proj
  - Cada tupla representa uma entidade funcionário
  - E a relação projeto (projnumero, Horas) detro de cada tupla

```
FUNC_PROJ(Cpf, Fnome, {PROJS(Projnumero, Horas)})
```

# (a) FUNC\_PROJ

		Projs	
Cpf	Fnome	Projnumero	Horas

(b)

## FUNC\_PROJ

Cpf	Fnome	Projnumero	Horas
12345678966	Silva, João B.	1	32,5
		2	7,5
66688444476	Lima, Ronaldo K.	3	40,0
45345345376	Leite, Joice A.	1	20,0
		2	20,0
33344555587	Wong, Fernando T.	2	10,0
		3	10,0
		10	10,0
		20	10,0
99988777767	Zelaya, Alice J.	30	30,0
		10	10,0
98798798733	Pereira, André V.	10	35,0
		30	5,0
98765432168	Souza, Jennifer S.	30	20,0
		20	15,0
88866555576	Brito, Jorge E.	20	NULL

## FUNC\_PROJ1

Cpf	Fnome
-----	-------

## FUNC\_PROJ2

Cpf	Projnumero	Horas
-----	------------	-------

(c)

Normalizando relações aninhadas para a 1FN. (a) Esquema da relação FUNC\_PROJ com um atributo de relação aninhada PROJS. (b) Exemplo de extensão da relação FUNC\_PROJ mostrando relações aninhadas dentro de cada tupla. (c) Decomposição de FUNC\_PROJ nas relações FUNC\_PROJ1 e FUNC\_PROJ2 pela propagação da chave primária.



# 2FN

- Deve estar na 1FN
- Se a chave primária foi única não aplica-se esta forma normal, caso contrário
- Há atributos que não participam da chave primária e são **funcionalmente dependentes** de toda a chave primária?
- Caso haja atributos dependentes de somente parte da chave primária, a relação deve ser normalizada

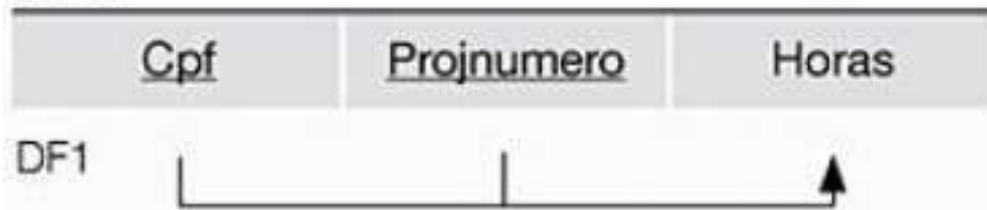
# 2FN

## FUNC\_PROJ



## Normalizacao 2FN

### FP1



### FP2



### FP3



# 2FN

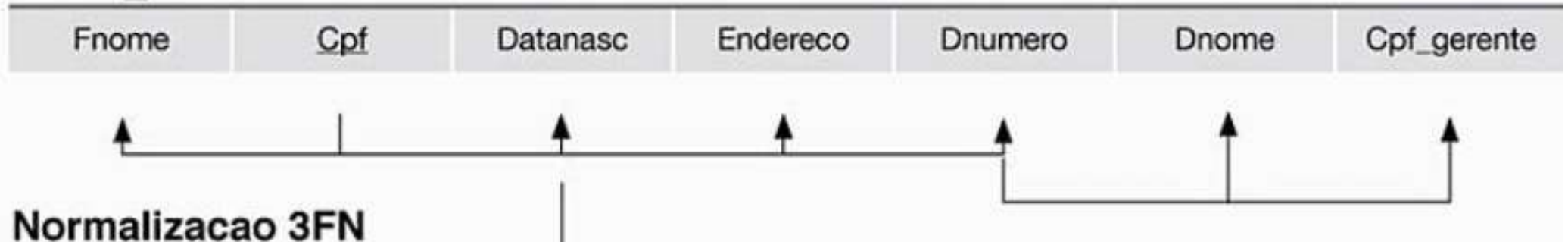
- Outro Exemplo
- Matricula (RA, cod-disc, nome-aluno, nome-disc, quant-aula-disc, frequencia)
- Como ficaria?

# 3FN

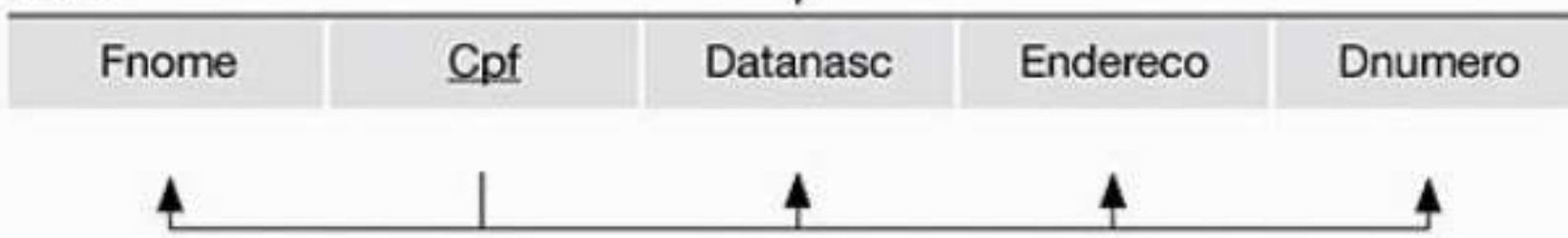
- Deve estar na 2FN
- 1 - Elimina-se atributos calculados
- Exemplo
  - Aluno (RA, nome-aluno, data-nasc, idade)
- 2 - Há atributos que não participam da chave primária e são **funcionamento dependentes** de outros atributos que não são chaves?

# 3FN

## FUNC\_DEP



## DF1



## DF2



# Resumindo

- A seguinte relação não está na 3FN (a 3FN não pode ter dependência transitiva).
  - $A \rightarrow B \rightarrow C$
- A solução seria dividir esse relacionamento em duas, o que resulta em:
  - $A \rightarrow B$
  - $B \rightarrow C$

# 3FN

- Outro Exemplo
- Aluno (RA, nome-aluno, cod-curso, nome-curso, titulo-curso)
- Como ficaria?

# Exercícios

- 1- Lotes(Propriedade-num, nome-cidade, num-lote, area, preço, imposto)



# FNBC

- Problemas que a 3FN não resolve
  - Relação com duas ou mais chaves candidatas
  - Essas candidatas fossem compostas
  - Elas tivessem sobreposição (atributo em comum)
- Caso essas condições não ocorra em uma tabela, basta aplicar a 3FN

# FNBC

- Proposta para ser mais simples que a 3FN
- No entanto, se tornou mais rigorosa
- Deve estar na 3FN
- Mas o contrário não é obrigatoriamente verdadeiro.
- Uma relação R está na Forma Normal de Boyce-Codd
  - Se para toda dependência funcional,  $X \rightarrow A$ , X é uma chave candidata (primária ou secundária) de R
  - Ou seja, nenhum atributo não chave de R pode determinar outro atributo (principal ou não)
  - Trata apenas das dependências envolvendo atributos não principais

# FNBC

- Para normalizar a tabela devemos decompor a tabela da seguinte maneira
- Encontrar uma dependência funcional  $X \rightarrow Y$  que viole a FNBC.  $X$  não deve ser uma superchave
- Dividir a tabela em duas
  - Uma com os atributos  $X Y$
  - Outra com os atributos  $X$  juntamente com os atributos restantes da tabela original

# FNBC

- Temos quantas possíveis chaves candidatas?
- Quais?
- Temos sobreposição de atributo?
- Qual?

F#	<u>F_Nome</u>	P#	Qtde
F1	Acme	P1	600
F1	Acme	P2	300
F1	Acme	P3	250
F1	Acme	P4	280
F2	Umbrella	P1	350

# FNBC

Forn	
F#	<u>F_Nome</u>
F1	Acme
F2	Umbrella
...	...

+

Forn_Prod		
F#	P#	Qtde
F1	P1	600
F1	P2	300
F1	P3	250
F1	P4	280
F2	P1	350
...	...	...

ou

Forn_Nome_Prod		
<u>F_Nome</u>	P#	Qtde
Acme	P1	600
Acme	P2	300
Acme	P3	250
Acme	P4	280
Umbrella	P1	350

# FNBC

- Outro Exemplo
  - Ensina (aluno, disciplina, professor)
- Restrições
- Cada estudante aprende uma disciplina lecionado por um professor
- Cada professor leciona apenas 1 disciplina
- Uma disciplina pode ser lecionada por vários professores

tbl_ADP		
Aluno	Disciplina	Professor
500	Matemática	Fábio
501	Física	Jorge
501	História	Ana Maria
503	Matemática	Sandra
503	História	Nunes

# FNBC

- Problemas
  - Se um aluno for excluído, as informações do orientador somem também
  - Se um novo aluno ou orientador for adicionado, obrigatoriamente devemos adicionar as informações relacionadas, as quais podem não existir

tbl_ADp		
Aluno	Disciplina	Professor
500	Matemática	Fábio
501	Física	Jorge
501	História	Ana Maria
503	Matemática	Sandra
503	História	Nunes

# FNBC

- Temos quantas possíveis chaves candidatas?
- Quais?
- Temos sobreposição de atributo?
- Qual?

tbl_ADP		
Aluno	Disciplina	Professor
500	Matemática	Fábio
501	Física	Jorge
501	História	Ana Maria
503	Matemática	Sandra
503	História	Nunes



# FNBC

- Temos duas DFs na relação
  - DF1: {aluno, disciplina} -> professor
  - DF2: professor -> disciplina
- {aluno, disciplina} é uma chave candidata para a relação
- Ao decompor uma relação para atender FNBC, deve-se tentar preservar as DFs nas relações decompostas

# FNBC - Decomposição

- Três possíveis decomposição para a relação
  - D1: {aluno, professor} e {aluno, disciplina}
  - D2: {disciplina, professor} e {disciplina, aluno}
  - D3: {professor, disciplina} e {professor, aluno}
- Todas as 3 decomposições vão perder a DF1
  - Temos que nos conformar em sacrificar as DFs
  - Porém não podemos sacrificar a propriedade de **junção aditiva**

# FNBC - Decomposição

- Das três possíveis, apenas a D3 não irá gerar tuplas espúrias após a junção
  - Espúrias: resultados inválidos após junções
- Teste para checar decomposições binárias não aditivas ->>

# Propriedade JBN (Junção Binária Não-aditiva)

- Uma decomposição  $D=\{R1, R2\}$  de  $R$  tem a JBN, no que diz respeito um conjunto de dependências funcionais  $F$  de  $R$ , **se e somente se**, alguma das DFs a seguir for válida
  - $DF((R1 \cap R2) \rightarrow (R1 - R2))$
  - $DF((R1 \cap R2) \rightarrow (R2 - R1))$

# Propriedade JBN (Junção Binária Não-aditiva)

- D1: {aluno, professor} e {aluno, disciplina}
- D2: {disciplina, professor} e {disciplina, aluno}
- D3: {professor, disciplina} e {professor, aluno}
- Ao aplicar o teste da propriedade JBN às 3 decomposições da relação Ensina, temos:
- D1 (ambas inválidas)
  - Aluno -> professor      ou      aluno -> disciplina
- D2 (ambas inválidas)
  - Disciplina -> professor      ou      disciplina -> aluno
- D3 (boa decomposição)
  - Professor -> disciplina      ou      professor -> aluno

# Decomposição ideal

- Seja  $R$  uma relação que não está na FNBC
- Seja  $X$  um subconjunto de  $R$
- E seja  $X \rightarrow A$  a DF que causa violação da FNBC
- Então  $R$  pode ser decomposta em 2 relações:
  - $R - A$
  - $X \cup A$
- Se  $R - A$  ou  $X \cup A$  não estiver na FNBC, repita o processo

# Decomposição ideal

- Note que a DF que violou a FNBC, em nosso exemplo Ensina, foi professor -> disciplina
- Portanto, sua decomposição seria:
  - Ensina – Disciplina
  - Professor U Disciplina
- Resultando em:
  - R1 (professor, aluno)
  - R2 ( professor, disciplina)

# FNBC

- Solução
  - Tabela1 (aluno, professor)
  - Tabela2 (professor, disciplina)



# FNBC

- E no nosso exemplo do lote?
  - Suponhamos lotes de duas cidades, Barra e aragarças
  - Em Barra os lotes têm tamanhos de 300 ou 360  $m^2$
  - Em Aragarças tamanhos de 420 a 450  $m^2$
  - Sendo assim, dizemos que há dependência funcional
    - Area -> Nome-cidade
  - No entanto, nome-cidade é chave primaria e área não!

# FNBC

- Termos que quebrar a relação Lotes1A do exercício anterior
- Lotes1AX (prop-num, area, num-lote)
- Lotes1AY (area, nome-cidade)

# 4FN

- Estar na 3FN
- Há relações que possuem restrições que não podem ser especificadas a partir da DF
- Utilizaremos dependência multivalorada
- São consequência da 1FN
- Se por acaso tivermos 2 atributos multivalorados em uma mesma relação
  - Obtemos o problema de ter que repetir cada valor de um dos atributos com cada valor do outro atributo
  - Afim de manter o estado da relação coerente e
  - Independência dos atributos envolvidos

# 4FN

- Para manter a relação coerente
- Devemos ter uma tupla separada para representar cada combinação
  - Equipe
  - Projeto
  - Funcionário
- Essa restrição é especificada pela DM
- Informalmente, sempre que dois relacionamentos 1:N independentes são misturados na mesma relação uma MVD pode surgir

# 4FN

COD_PROJETO	COD_FUNCIONARIO	COD_EQUIPE
11	1001	A10
11	1002	A10
11	1001	A20
11	1002	A20
12	1001	A10
12	1001	A20

- Colocado na 4FN

COD_PROJETO	COD_FUNCIONÁRIO	COD_PROJETO	COD_EQUIPE
11	1001	11	A10
11	1002	11	A20
12	1001	12	A10
		12	A20

# 4FN – Outro Exemplo

<u>Sname</u>	<u>Part_name</u>	<u>Proj_name</u>
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

- Está na 4ªFN, pois não há combinação de todos os valores dos atributos
- E Por que algumas situações eu repito e outras não? Quem decide isso?
  - A regra de negócio!!!!

# Exercícios

- Passe para a 4ªFN

MUSICA	ARTISTA	ALBUM
MUSICA 1	ARTISTA 1	ALBUM 1
MUSICA 1	ARTISTA 2	ALBUM 2
MUSICA 1	ARTISTA 1	ALBUM 2
MUSICA 2	ARTISTA 3	ALBUM 1
MUSICA 2	ARTISTA 2	ALBUM 1
MUSICA 2	ARTISTA 3	ALBUM 2

# Solução

MUSICA	ARTISTA
MUSICA 1	ARTISTA 1
MUSICA 1	ARTISTA 2
MUSICA 2	ARTISTA 3
MUSICA 2	ARTISTA 2

MUSICA	ALBUM
MUSICA 1	ALBUM 1
MUSICA 1	ALBUM 2
MUSICA 2	ALBUM 1
MUSICA 2	ALBUM 2



# 5FN

- Estar na 4FN
- Não pode ter dependência funcional de junção
  - Ou seja, uma dependência multivalorada incompleta
- Muito difícil de perceber-la em um BD com centenas de tabelas
- Quando aplicamos a 4FN em apenas 2 tabelas (em algumas vezes)
  - Acontece perda de semântica
  - Assim precisaremos que a decomposição seja feita em 3 ou mais tabelas

# Exemplo

## FORNECE

<u>Nome_fornece</u>	<u>Nome_peca</u>	<u>Nome_proj</u>
Silva	Peneira	ProjX
Silva	Porca	ProjY
Adam	Peneira	ProjY
Walter	Porca	ProjZ
Adam	Prego	ProjX
Adam	Peneira	ProjX
Silva	Peneira	ProjY

- Dependência de Junção
  - Não iremos gerar todas as combinações
  - Geraremos algumas dependendo da regra de negócio
  - Resultando em decomposição não binária

# Exemplo

**R1**

<u>Nome_fornece</u>	<u>Nome_peca</u>
Silva	Peneira
Silva	Porca
Adam	Peneira
Walter	Porca
Adam	Prego

**R2**

<u>Nome_fornece</u>	<u>OR_proj</u>
Silva	ProjX
Silva	ProjY
Adam	ProjY
Walter	ProjZ
Adam	ProjX

**R3**

<u>Nome_peca</u>	<u>OR_proj</u>
Peneira	ProjX
Porca	ProjY
Peneira	ProjY
Porca	ProjZ
Prego	ProjX

- E se tivéssemos a seguinte regra de negócio:
  - Se um fornecedor fornece uma peça x ele é obrigado a fornecer a todos os projetos

# 5FN

## FORNECE

<u>Nome_fornece</u>	<u>Nome_peca</u>	<u>Nome_proj</u>
Silva	Peneira	ProjX
Silva	Porca	ProjY
Adam	Peneira	ProjY
Walter	Porca	ProjZ
Adam	Prego	ProjX

<u>Nome_fornece</u>	<u>Nome_peca</u>	<u>Nome_proj</u>
Silva	Peneira	ProjX
Silva	Porca	ProjY
Adam	Peneira	ProjY
Walter	Porca	ProjZ
Adam	Prego	ProjX
Adam	Peneira	ProjX
Silva	Peneira	ProjY

tuplas falsas

Decomposição

Junção Natural

R1

<u>Nome_fornece</u>	<u>Nome_peca</u>
Silva	Peneira
Silva	Porca
Adam	Peneira
Walter	Porca
Adam	Prego

R2

<u>Nome_fornece</u>	<u>OR_proj</u>
Silva	ProjX
Silva	ProjY
Adam	ProjY
Walter	ProjZ
Adam	ProjX

R3

<u>Nome_peca</u>	<u>OR_proj</u>
Peneira	ProjX
Porca	ProjY
Peneira	ProjY
Porca	ProjZ
Prego	ProjX

# Exercício 4FN e 5FN

Revendas

	Representante	Cliente	Produto
1	Martins	Sá e filhos	Louça
2	Martins	Sá e filhos	Talheres
3	Martins	DMI, Lda	Louça
4	Martins	DMI, Lda	Talheres
5	Castro	Sá e filhos	Cintos
6	Castro	Sá e filhos	Meias
7	Castro	Sá e filhos	Lenços
8	Castro	YSL	Cintos
9	Castro	YSL	Meias
10	Castro	YSL	Lenços

Revendas

	Representante	Cliente	Produto
1	Martins	Sá e filhos	Louça
3	Martins	DMI, Lda	Louça
4	Martins	DMI, Lda	Talheres
5	Castro	Sá e filhos	Cintos
7	Castro	Sá e filhos	Lenços
9	Castro	YSL	Meias
10	Castro	YSL	Lenços

# Exercícios

- 1) Considere o formulário abaixo:

## RELATÓRIO DE AVALIAÇÃO DE CURSOS

Cód. Curso: INF001		Nome Curso: Projeto BD	Cód. Área: INF		Descrição Área: Informática	
Matrícula Funcionário	Data de Admissão	Nome do Funcionário	Ano Concl.	Cód. Cargo	Nome do Cargo	Avaliação
00129	01/03/1999	Alberto dos Santos	2000	001	Analista Junior	Regular
93821	05/03/1976	José da Silva	2002	002	Analista Sênior	Muito Bom
29841	09/09/2000	Maria José da Silva	2001	001	Analista Junior	Excelente
93820	08/07/1998	Rosa Maria	2000	003	Analista Pleno	Bom
00129	01/03/1999	Alberto dos Santos	2002	002	Analista Sênior	Muito Bom

# Exerícios

- a) Encontre as dependências Funcionais:
- Ex:  $A \rightarrow B, C$ ;  $C \rightarrow E$ ;
- B) Aplique a 1, 2 e 3FN. Caso julgue necessário aplique a FNBC.
- 2) Considere a relação R (A, B, C, D, E, F) onde a chave primária é A, B e que apresenta as seguintes dependências funcionais:
- $A \rightarrow C$ ;  $B \rightarrow D$ ;  $(A, B) \rightarrow E$ ;  $E \rightarrow F$ ;
- Normalize esta relação até a 3FN

# Exercícios

- 3) Considere a seguinte relação para automóveis alugados
- Aluguel (cod-cliente, nome-cliente, telefone, cod-carro, marca, data-aluguel, data-devolução, valor-devido, cod-fornecedor, nome-fornecedor, valor-diaria-carro)
- A) mostre as dependências funcionais
- B) Deixe a relação na 3FN