



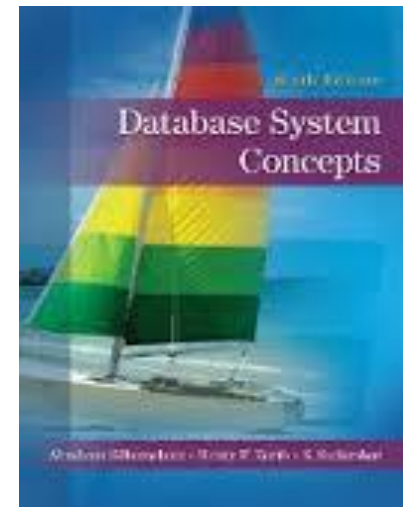
Unidade 8 – Projeto de Banco de Dados Boas Práticas e Normalização



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP



Bibliografia





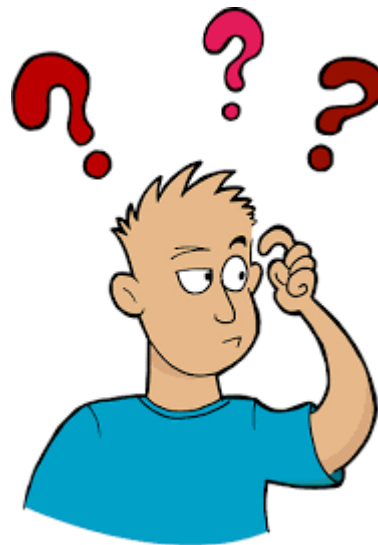
Projeto de Banco de Dados

- ⊕ Cada esquema de Relação consiste em uma série de atributos, e o esquema de banco de dados consiste em uma série de esquemas de relação.
- ⊕ O projeto Conceitual não apresenta detalhes de implementação e, por isso, é de fácil compreensão e pode ser usada para comunicação com usuários;
- ⊕ Na modelagem conceitual emprega-se usualmente o MER – Modelo Entidade Relacionamento, notação criada por Peter Chen em 1976.
- ⊕ No projeto lógico, aplica-se um algoritmo de mapeamento do modelo conceitual para o modelo relacional.
- ⊕ Assume-se que atributos são agrupados para formar relações e emprega-se o **BOM SENSO** do projetista de banco de dados.





Como avaliar se o agrupamento de atributos em um esquema é melhor do que outro ?





Projeto de Banco de Dados

- ✦ Deve-se utilizar **BOAS PRÁTICAS** para se medir a qualidade de um projeto, além da intuição do projetista;
- ✦ Deve-se empregar um modo **FORMAL** de análise.
- ✦ Os objetivos implícitos da atividade de projeto são **preservação da informação** e **redundância mínima**.





Boas Práticas – Diretriz 1

■ Semântica clara dos atributos

- ⊕ Atributos agrupados em uma relação têm certo **significado** no mundo real;
- ⊕ A semântica de uma relação refere-se a seu **SIGNIFICADO** resultante da interpretação dos valores de atributo em uma tupla.



Boas Práticas – Diretriz 1

■ Semântica clara dos atributos – Exemplo 1

Funcionario

					FK
Fnome	<u>CPF</u>	DataNasc	Endereco	idDepto	

- ✓ O significado desta relação é simples de ser compreendido;
- ✓ Cada tupla representa um FUNCIONARIO, com valores para nome (**Fnome**), **CPF**, endereço (**Endereco**) e o departamento para o qual trabalha (**idDepto**);
- ✓ O atributo idDepto é uma chave estrangeira que representa um relacionamento implícito entre **FUNCIONARIO** e **DEPARTAMENTO**.



Boas Práticas – Diretriz 1

■ Semântica clara dos atributos – Exemplo 2

Departamento

			FK
Dnome	<u>idDepto</u>	CPF_Gerente	

- ✓ O significado desta relação é simples de ser compreendido;
- ✓ Cada tupla representa um **DEPARTAMENTO**, com valores para nome (**Dnome**), **idDepto** e (**CPF_Gerente**);
- ✓ O atributo **CPF_Gerente** relaciona um **DEPARTAMENTO** ao **FUNCIONARIO** que é seu gerente.



Boas Práticas – Diretriz 1

■ Semântica clara dos atributos – Exemplo 3

Projeto

FK

ProjNome	<u>ProjNumero</u>	ProjLocal	idDepto
----------	-------------------	-----------	---------

- ✓ O significado desta relação é simples de ser compreendido;
- ✓ Cada tupla representa um **PROJETO**, com valores para nome do projeto (**ProjNome**), número do projeto (**ProjNumero**), localização (**ProjLocal**) e departamento responsável pelo projeto (**idDepto**).
- ✓ O atributo **idDepto** relaciona um **PROJETO** a seu **DEPARTAMENTO**.



Boas Práticas – Diretriz 1



- ✓ Projete um esquema de relação de modo que seja **fácil** explicar o seu **SIGNIFICADO**;
- ✓ **Não combine** atributos de vários tipos de entidade e de relacionamento em uma única relação;
- ✓ Caso contrário, se a relação corresponder a uma mistura de várias entidades e relacionamentos, haverá **AMBIGUIDADES SEMÂNTICAS** e a relação não poderá ser explicada com facilidade.

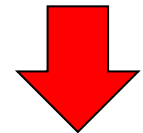


Violação da Diretriz 1

■ Violação da Diretriz 1 – Exemplo 1

Funcionario'

Fnome	<u>CPF</u>	DataNasc	Endereco	idDepto	NomeDepto	EnderecoDepto
-------	------------	----------	----------	---------	-----------	---------------



- ✓ A semântica também é clara. A relação representa dados de um funcionário;
- ✓ Porém, inclui **informações adicionais** do departamento em que o funcionário trabalha, tais como o nome do departamento e o endereço do departamento.
- ✓ Embora não haja logicamente nada de errado com essa relação, ela viola a Diretriz 1 ao **misturar** atributos de entidades distintas do mundo real: **FUNCIONARIO** e **DEPARTAMENTO**.

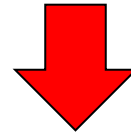
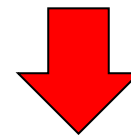


Violação da Diretriz 1

■ Violação da Diretriz 1 – Exemplo 2

FUNC_PROJ

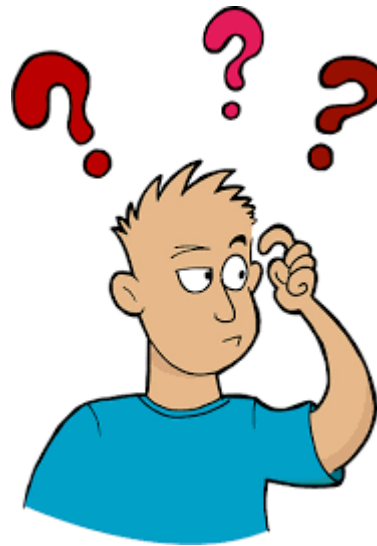
<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
------------	-------------------	-------	-------	----------	-----------



- ✓ Cada tupla da relação FUNC_PROJ' relaciona um FUNCIONARIO a um PROJETO;
- ✓ Mas, também inclui o nome do funcionário (Fnome), o nome do projeto (ProjNome) e local do projeto (ProjLocal).
- ✓ Embora não haja logicamente nada de errado com essa relação, ela viola a Diretriz 1 ao misturar atributos de entidades distintas do mundo real: **FUNCIONARIO** e **PROJETO**.



Qual o impacto da violação da Diretriz 1 ?





Redundância de Informações

- ✓ O agrupamento de atributos em esquemas de relação têm um efeito significativo no **espaço de armazenamento**.
- ✓ O objetivo do projeto de esquema é **MINIMIZAR** o espaço de armazenamento usado pelas relações e, portanto, pelos arquivos correspondentes.



Violação da Diretriz 1

■ Violação da Diretriz 1 – Exemplo 1

Funcionario'

Redundância

Redundância



Fnome	<u>CPF</u>	DataNasc	Endereco	idDepto	NomeDepto	EnderecoDepto
Antonio	123987098-11	19/10/1976	Rua A, 309	5	Compras	ALA 10 – SBC
Ana	657987541-12	31/10/1987	Rua Brasil, 87	5	Compras	ALA 10 - SBC
Paula	980652098-99	07/03/1982	Av. Souza, 45	5	Compras	ALA 10 - SBC
Pedro	721098098-23	15/06/1987	Rua X, 76	2	Vendas	CENTRO - SP
Carlos	876234099-45	12/06/1991	Av. Rui F. 23	2	Vendas	CENTRO - SP





Violação da Diretriz 1

■ Violação da Diretriz 1 – Exemplo 2

Redundância Redundância

FUNC_PROJ

<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
123987098-11	1	45	Antonio	ABX	SP
123987098-11	2	30	Antonio	TRW	Santo André
123987098-11	3	56	Antonio	RSA	Rio de Janeiro
765987098-34	3	65	Pedro	RSA	Rio de Janeiro
567987123-98	3	41	Ana	RSA	Rio de Janeiro
221897008-56	3	98	Carlos	RSA	Rio de Janeiro





Anomalias de Inserção

Funcionario'

Fnome	<u>CPF</u>	DataNasc	Endereco	idDepto	NomeDepto	EnderecoDepto
Antonio	123987098-11	19/10/1976	Rua A, 309	5	Compras	ALA 10 – SBC
Ana	657987541-12	31/10/1987	Rua Brasil, 87	5	Compras	ALA 10 - SBC
Paula	980652098-99	07/03/1982	Av. Souza, 45	5	Compras	ALA 10 - SBC
Pedro	721098098-23	15/06/1987	Rua X, 76	5	Compras	CAMPINAS

Funcionário NOVO



- ✓ Para inserirmos um novo funcionário, é necessário incluir-se os valores dos atributos do Departamento ou NULL se o funcionário ainda não trabalha em nenhum Departamento.
- ✓ Se inserirmos valores de forma incorreta, o banco de dados gera-se uma INCONSISTÊNCIA nos dados (incoerência).





Anomalias de Inserção

Funcionario'

Fnome	<u>CPF</u>	DataNasc	Endereco	idDepto	NomeDepto	EnderecoDepto
Antonio	123987098-11	19/10/1976	Rua A, 309	5	Compras	ALA 10 – SBC
Ana	657987541-12	31/10/1987	Rua Brasil, 87	5	Compras	ALA 10 - SBC
Paula	980652098-99	07/03/1982	Av. Souza, 45	5	Compras	ALA 10 - SBC
NULL	NULL	NULL	NULL	999	Escritório Regional	SÃO PAULO



Departamento NOVO em funcionário



- ✓ Para inserirmos um novo DEPARTAMENTO que ainda não tenha funcionários, deve-se colocar valores NULL nos atributos de funcionário.
- ✓ Isso viola regra de integridade de entidade, uma vez que CPF é chave primária.



Anomalia de Exclusão

Funcionario'

Fnome	<u>CPE</u>	DataNasc	Endereco	idDepto	NomeDepto	EnderecoDepto
Antonio	123987098-11	19/10/1976	Rua A, 309	5	Compras	ALA 10 – SBC
Ana	657987541-12	31/10/1987	Rua Brasil, 87	5	Compras	ALA 10 - SBC
Paula	980652098-99	07/03/1982	Av. Souza, 45	5	Compras	ALA 10 - SBC
Pedro	721098098-23	15/06/1987	Rua X, 76	15	Engenharia	SANTOS

Eliminação do último funcionário
do Departamento 15



- ✓ Ao se eliminar o último funcionário do Departamento 15, perde-se no banco de dados informações referentes a esse departamento.



Anomalia de Modificação

Funcionario'

Fnome	<u>CPE</u>	DataNasc	Endereco	idDepto	NomeDepto	EnderecoDepto
Antonio	123987098-11	19/10/1976	Rua A, 309	5	Compras	ALA 10 – SBC
Ana	657987541-12	31/10/1987	Rua Brasil, 87	5	Compras	ALA 10 - SBC
Paula	980652098-99	07/03/1982	Av. Souza, 45	5	Compras	ALA 10 - SBC
Pedro	721098098-23	15/06/1987	Rua X, 76	15	Engenharia	SANTOS

Modificação da localização
do Departamento 5



- ✓ Ao se modificar o endereço do Departamento = 5, deve-se proceder a alteração em todas as tuplas referentes aos funcionários que trabalham nesse departamento.



Boas Práticas – Diretriz 2

■ Anomalias de manutenção de dados

- ⊕ Projete os esquemas de relação da base de modo que nenhuma anomalia de inserção, exclusão ou modificação esteja presente nas relações.
- ⊕ Se houver alguma anomalia, anote-as claramente e cuide para que os programas que atualizam o banco de dados operem corretamente.





Tuplas com valores NULL

- ⊕ Relações 'gordas' (com muitos atributos) podem ter ocorrências de valores NULL;
- ⊕ Valores NULL podem ter diversas semânticas: atributo não se aplica para a tupla, ou valor é desconhecido do atributo ou ainda o valor é conhecido mas ainda não registrado.
- ⊕ Deve-se considerá-los em operações de contagem de registros (COUNT);
- ⊕ Valores NULL implicam em lógica de três valores: TRUE, FALSE e UNKNOWN.



Tuplas com valores NULL

Conectivos lógicos na lógica de três valores.

(a) **AND**

	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

(b) **OR**

	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

(c) **NOT**

TRUE	FALSE		
FALSE	TRUE		
UNKNOWN	UNKNOWN		

Exemplo: FALSE AND UNKNOWN -> **FALSE**

FALSE OR UNKNOWN -> **UNKNOWN**

NULL



Tuplas com valores NULL

Result Grid					
		Filter Rows:			
		Edit:			
		Export			
	idCliente	nome	email	limiteCredito	status
▶	1	Paulo de Souza	paulo@gmail.com	10000	5
	2	Ana Maria	anamaria@gmail.com	5000	4
	3	Paulo de Souza	paulodesouza@uol.com	8000	4
	4	Saulo Silva	saulosilva@gmail.com	NULL	3
	5	Pedro Couto	NULL	3000	NULL
	6	Rogério Aguiar	rogaguair@hotmail.com	4000	NULL

SELECT COUNT(*) FROM cliente ;

Result Grid	
Filter Rows:	
count(*)	
▶	6



Boas Práticas – Diretriz 3

■ Tuplas com valores NULL

- ⊕ Ao máximo possível, evite colocar atributos em uma relação de base cujos valores podem ser **NULL** com frequência;
- ⊕ Se **NULLs** forem inevitáveis, garanta que eles se apliquem apenas em casos excepcionais, e não à maioria das tuplas na relação.





Geração de Tuplas Falsas

- ✓ Considere o esquemas da relação **FUNC_PROJ**.

FUNC_PROJ

<u>Cpf</u>	<u>Projnumero</u>	Horas	Fnome	Projnome	Projlocal
------------	-------------------	-------	-------	----------	-----------

FUNC_PROJ			Redundância		Redundância	
			Fnome	Projnome	Projlocal	
123456789066	1	32,5	Silva, João B.	ProdutoX	Santo André	
123456789066	2	7,5	Silva, João B.	ProdutoY	Itu	
66688444478	3	40,0	Lima, Ronaldo K.	ProdutoZ	São Paulo	
45345345378	1	20,0	Leite, Joice A.	ProdutoX	Santo André	
45345345378	2	20,0	Leite, Joice A.	ProdutoY	Itu	
33344555587	2	10,0	Wong, Fernando T.	ProdutoY	Itu	
33344555587	3	10,0	Wong, Fernando T.	ProdutoZ	São Paulo	
33344555587	10	10,0	Wong, Fernando T.	Informatização	Mauá	
33344555587	20	10,0	Wong, Fernando T.	Reorganização	São Paulo	
00088777767	30	30,0	Zelaya, Alice J.	Novosbenefícios	Mauá	
00088777767	10	10,0	Zelaya, Alice J.	Informatização	Mauá	
08798798733	10	35,0	Pereira, André V.	Informatização	Mauá	
08798798733	30	5,0	Pereira, André V.	Novosbenefícios	Mauá	
08765432168	30	20,0	Souza, Jennifer S.	Novosbenefícios	Mauá	
08765432168	20	15,0	Souza, Jennifer S.	Reorganização	São Paulo	
88866555578	20	Null	Brito, Jorge E.	Reorganização	São Paulo	



Geração de Tuplas Falsas

- ✓ Separando-se a relação **FUNC_PROJ** em **FUNC_LOCAL** e **FUNC_PROJ1**.

(a)

FUNC_LOCAL

Fnome	Projlocal
-------	-----------

Chave Primária



Um funcionário cujo nome é Fnome trabalha em algum projeto cujo local é ProjLocal.

(b)

FUNC_PROJ1

Cpf	Projnumero	Horas	Projnome	Projlocal
-----	------------	-------	----------	-----------

Chave Primária



Um funcionário com um determinado CPF trabalha uma certa quantidade de horas em um projeto cujo nome, número e localização são Projnome, Projnumero e Projlocal.



Geração de Tuplas Falsas

- ✓ Estado das relações **FUNC_LOCAL** e **FUNC_PROJ1** equivalentes a **FUNC_PROJ**.

(c)

FUNC_LOCAL

Fnome	Projlocal
Silva, João B.	Santo André
Silva, João B.	Itu
Lima, Ronaldo K.	São Paulo
Leite, Joice A.	Santo André
Leite, Joice A.	Itu
Wong, Fernando T.	Itu
Wong, Fernando T.	São Paulo
Wong, Fernando T.	Mauá
Zelaya, Alice J.	Mauá
Pereira, André V.	Mauá
Souza, Jennifer S.	Mauá
Souza, Jennifer S.	São Paulo
Brito, Jorge E.	São Paulo

FUNC_PROJ1

Cpf	Projnumero	Horas	Projnome	Projlocalizacao
12345678966	1	32,5	ProdutoX	Santo André
12345678966	2	7,5	ProdutoY	Itu
66688444476	3	40,0	ProdutoZ	São Paulo
45345345376	1	20,0	ProdutoX	Santo André
45345345376	2	20,0	ProdutoY	Itu
33344555587	2	10,0	ProdutoY	Itu
33344555587	3	10,0	ProdutoZ	São Paulo
33344555587	10	10,0	Computadorização	Mauá
33344555587	20	10,0	Reorganização	São Paulo
99988777767	30	30,0	Novosbenefícios	Mauá
99988777767	10	10,0	Computadorização	Mauá
98765432168	10	35,0	Computadorização	Mauá
98765432168	30	5,0	Novosbenefícios	Mauá
98765432168	30	20,0	Novosbenefícios	Mauá
98798798733	20	15,0	Reorganização	São Paulo
88866555576	20	NULL	Reorganização	São Paulo



Geração de Tuplas Falsas

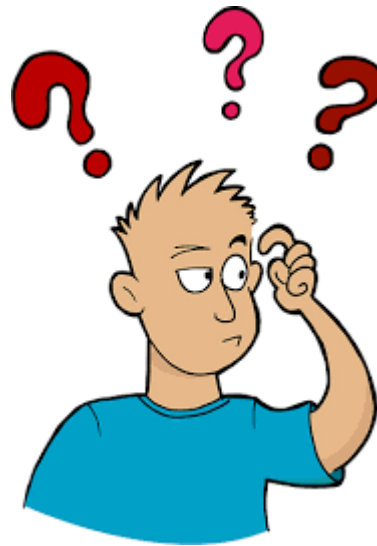
- ✓ Ao se executar uma operação JOIN NATURAL sobre **FUNC_LOCAL** e **FUNC_PROJ1**, o resultado produz mais tuplas do que o conjunto original de tuplas em **FUNC_PROJ**.

Cpf	Projnumero	Horas	Projnome	Projlocal	Fnome
12345678966	1	32,5	ProdutoX	Santo André	Silva, João B.
*12345678966	1	32,5	ProdutoX	Santo André	Leite, Joice A.
12345678966	2	7,5	ProdutoY	Itu	Silva, João B.
*12345678966	2	7,5	ProdutoY	Itu	Leite, Joice A.
*12345678966	2	7,5	ProdutoY	Itu	Wong, Fernando T.
66688444476	3	40,0	ProdutoZ	São Paulo	Lima, Ronaldo K.
*66688444476	3	40,0	ProdutoZ	São Paulo	Wong, Fernando T.
*45345345376	1	20,0	ProdutoX	Santo André	Silva, João B.
45345345376	1	20,0	ProdutoX	Santo André	Leite, Joice A.
*45345345376	2	20,0	ProdutoY	Itu	Silva, João B.
45345345376	2	20,0	ProdutoY	Itu	Leite, Joice A.
*45345345376	2	20,0	ProdutoY	Itu	Wong, Fernando T.
*33344555587	2	10,0	ProdutoY	Itu	Silva, João B.
*33344555587	2	10,0	ProdutoY	Itu	Leite, Joice A.
33344555587	2	10,0	ProdutoY	Itu	Wong, Fernando T.
*33344555587	3	10,0	ProdutoZ	São Paulo	Lima, Ronaldo K.
33344555587	3	10,0	ProdutoZ	São Paulo	Wong, Fernando T.
33344555587	10	10,0	Computadorização	Mauá	Wong, Fernando T.
*33344555587	20	10,0	Reorganização	São Paulo	Lima, Ronaldo K.
33344555587	20	10,0	Reorganização	São Paulo	Wong, Fernando T.

Tuplas Falsas



Qual a causa da geração de Tuplas Falsas ?





Geração de Tuplas Falsas

- ✚ A decomposição de **FUNC_PROJ** em **FUNC_LOCAL** e **FUNC_PROJ1** é indesejável porque, quando as juntamos (JUNÇÃO) de volta usando JOIN NATURAL, não obtemos a informação original correta.
- ✚ Isso porque, nesse caso, ProjLocal é o atributo que representa **FUNC_LOCAL** e **FUNC_PROJ1**, e **ProjLocal** não é a chave primária nem a chave estrangeira em **FUNC_LOCAL** ou **FUNC_PROJ1**.





Boas Práticas – Diretriz 4

■ Tuplas Falsas

- ⊕ Projete esquemas de relação de modo que possam ser unidos com condições de igualdade sobre os atributos que são pares relacionados corretamente (chave primária, chave estrangeira) de modo que se garanta que nenhuma tupla FALSA será gerada.
- ⊕ Evite relações com atributos correspondentes que sejam combinações (chave estrangeira, chave primária) pois a JUNÇÃO sobre tais atributos pode produzir TUPLAS FALSAS





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 semântica ou significado dos atributos.

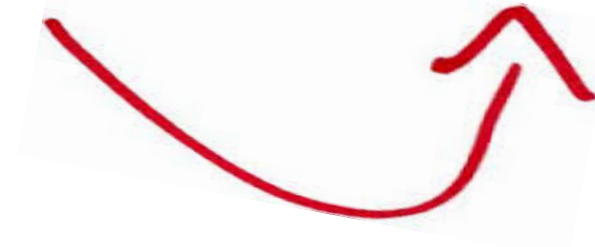


Dependência Funcional

- ⊕ Considere o esquema de relação FUNC_PROJ;

FUNC_PROJ

<u>Cpf</u>	<u>Projnumero</u>	Horas	Fnome	Projnome	Projlocal
------------	-------------------	-------	-------	----------	-----------



- ⊕ Pela semântica dos atributos e da relação, sabe-se que a seguinte dependência funcional deve ser mantida:

CPF → **Fnome** (CPF determina exclusivamente o nome do Funcionário)





Dependência Funcional

FUNC_PROJ

<u>Cpf</u>	<u>Projnumero</u>	Horas	Fnome	Projnome	Projlocal
------------	-------------------	-------	-------	----------	-----------

	<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
Tupla T1 →	123987098-11	1	45	Antonio	ABX	SP
	123987098-11	2	30	Antonio	TRW	Santo André
Tupla T2 →	123987098-11	3	56	Antonio	RSA	Rio de Janeiro
	765987098-34	3	65	Pedro	RSA	Rio de Janeiro
	567987123-98	3	41	Ana	RSA	Rio de Janeiro
	221897008-56	3	98	Carlos	RSA	Rio de Janeiro

$$T1(CPF) = T2(CPF) \rightarrow T1(Fnome) = T2(Fnome)$$



Dependência Funcional

- ⊕ Considere o esquema de relação FUNC_PROJ;



- ⊕ Pela semântica dos atributos e da relação, sabe-se que a seguinte dependência funcional deve ser mantida:

Projnumero → { Projnome, Projlocal }

O número do projeto (Projnumero) determina exclusivamente o nome do Projeto (Projnome) e seu local (Projlocal)





Dependência Funcional

FUNC_PROJ

<u>Cpf</u>	<u>Projnumero</u>	Horas	Fnome	Projnome	Projlocal
------------	-------------------	-------	-------	----------	-----------

<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
123987098-11	1	45	Antonio	ABX	SP
123987098-11	2	30	Antonio	TRW	Santo André
123987098-11	3	56	Antonio	RSA	Rio de Janeiro
765987098-34	3	65	Pedro	RSA	Rio de Janeiro
567987123-98	3	41	Ana	RSA	Rio de Janeiro
221897008-56	3	98	Carlos	RSA	Rio de Janeiro

Tupla T1



Tupla T2



$$T1(\text{ProjNumero}) = T2(\text{ProjNumero}) \rightarrow T1(\text{ProjNome}) = T2(\text{ProjNome}) \text{ e } T1(\text{ProjLocal}) = T2(\text{ProjLocal})$$



Dependência Funcional

- ⊕ Considere o esquema de relação FUNC_PROJ;

FUNC_PROJ

<u>Cpf</u>	<u>Projnumero</u>	Horas	Fnome	Projnome	Projlocal
------------	-------------------	-------	-------	----------	-----------



- ⊕ Pela semântica dos atributos e da relação, sabe-se que a seguinte dependência funcional deve ser mantida:

{ CPF, ProjNumero } → Horas

Uma combinação de valores de CPF e ProjNumero determina exclusivamente o número de horas/semana que o funcionário trabalha no projeto.





Dependência Funcional

FUNC_PROJ

<u>Cpf</u>	<u>Projnumero</u>	Horas	Fnome	Projnome	Projlocal
------------	-------------------	-------	-------	----------	-----------



<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
123987098-11	1	45	Antonio	ABX	SP
123987098-11	2	30	Antonio	TRW	Santo André
123987098-11	3	56	Antonio	RSA	Rio de Janeiro
765987098-34	3	65	Pedro	RSA	Rio de Janeiro
567987123-98	3	41	Ana	RSA	Rio de Janeiro
221897008-56	3	98	Carlos	RSA	Rio de Janeiro



Dependência Funcional

⊕ Uma dependência funcional é uma propriedade do esquema de relação **R**, e **não** um estado de relação válido e específico **r** de **R**.

⊕ Portanto, deve ser definida de maneira explícita por alguém que conhece a semântica dos atributos de **R**.



⊕ Exemplo:

ENSINA

Professor	Disciplina	Texto
Silva	Estruturas de Dados	Bartram
Silva	Gerenciamento de Dados	Martin
Neto	Compiladores	Hoffman
Braga	Estruturas de Dados	Horowitz

⊕ **A restrição deve valer para todos os estados da relação R.** Basta um contra-exemplo para refutar a dependência funcional. No exemplo, Professor não determina funcionalmente Disciplina (tuplas com o valor 'Silva' para o atributo Professor)



Dependência Funcional

	A	B	C	D
t_1	a1	b1	o1	d1
t_2	a1	b2	o2	d2
t_3	a2	b2	o2	d3
t_4	a3	b3	o4	d3



- ⊕ **A** não determina funcionalmente **B**, pois há violação da restrição de DF nas tuplas t_1 e t_2 .
- ⊕ **B** não determina funcionalmente **A**, pois há violação da restrição de DF nas tuplas t_2 e t_3 .
- ⊕ **D** não determina funcionalmente **C**, pois há violação da restrição de DF nas tuplas t_3 e t_4 .



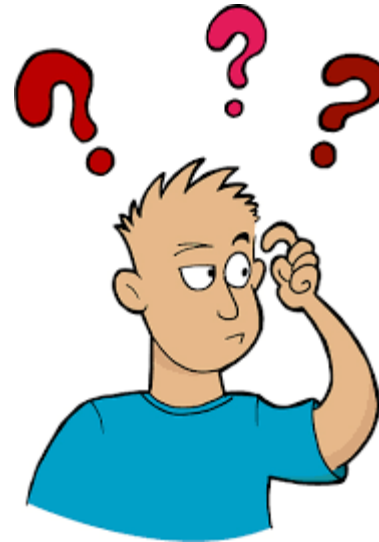
Normalização de Relações

- ✦ O processo de normalização, proposto inicialmente, por Codd (1972) permite que, por meio de uma série de testes, certifique-se que uma relação satisfaz certa **FORMA NORMAL**.
- ✦ Inicialmente, Codd propôs 3 formas normais, que ele chamou de **primeira**, **segunda** e **terceira** formas normais.
- ✦ Todas essas formas normais estão baseadas em uma única ferramenta analítica: as **dependências funcionais** entre os atributos de uma relação.





Qual a finalidade da Normalização de Relações ?





Normalização de Relações

⊕ Pode ser considerada um processo de se analisar os esquemas de relação com base em suas DEPENDÊNCIAS FUNCIONAIS e chaves primárias para conseguir as propriedades desejadas de:

✓ *Minimização da Redundância;*

✓ *Minimização das Anomalias de Inserção, Exclusão e Atualização dos dados.*





Normalização – Definição

- ⊕ A forma normal de uma relação refere-se à condição de forma normal mais alta a que ela atende e, portanto, indica o grau ao qual ela foi normalizada.
- ⊕ Se uma relação está na terceira forma normal, então também está na segunda e primeira formas normais.
- ⊕ Se uma relação está na segunda forma normal, então também está na primeira forma normal.
- ⊕ Para finalidades práticas, em geral, normaliza-se até a terceira forma normal.
- ⊕ 2FN e 3FN atacam diferentes problemas, mas por motivos históricos, é comum seguir a ordem de 1FN, 2FN e 3FN no processo de normalização das relações.





Lembrando . . .

- ⊕ Superchave
- ⊕ Chave
- ⊕ Chave Candidata
- ⊕ Chave Primária
- ⊕ Chave Única





Superchave

- ⊕ Duas tuplas **não** podem ter a mesma combinação de valores para todos os seus atributos.
- ⊕ Ou seja, todos os elementos de um conjunto devem ser distintos.
- ⊕ No entanto, normalmente existem subconjuntos de atributos de um esquema de relação **R**, com a propriedade de que duas tuplas em qualquer estado da relação **r** de **R** não deverão ter a mesma combinação de valores para esses atributos. Qualquer subconjunto de atributos desse tipo é chamado **SUPERCHAVE** do esquema de relação **R**.

ALUNO

Atributos →

CODIGO	NOME	Curso	CPF
3456	Paulo de Souza Alves	EC	109.098.340-12
9872	Francisco Silva	CC	239-345-986-12
1890	Francisco Silva	CC	234.876.456-45



Superchaves

- ◆ Considerando o estado da relação abaixo, quais as superchaves da Relação ?

Atributos →

ALUNO			
CODIGO	NOME	CURSO	CPF
3456	Paulo de Souza Alves	EC	109.098.340-12
9872	Francisco Silva	CC	239-345-986-12
1890	Francisco Silva	CC	234.876.456-45





Chave

- ◆ Uma superchave pode ter atributos redundantes. Assim, o conceito mais útil é o de uma superchave que não tenha redundância.
- ◆ Uma chave de um esquema de relação **R** é uma superchave de **R** com a propriedade adicional de que a remoção de qualquer atributo **A** não mais caracteriza uma superchave.
- ◆ Portanto, a chave é uma superchave mínima.

ALUNO

Atributos →

CODIGO	NOME	CURSO	CPF
3456	Paulo de Souza Alves	EC	109.098.340-12
9872	Francisco Silva	CC	239-345-986-12
1890	Francisco Silva	CC	234.876.456-45



Chave Candidata

- ◆ Toda chave também é uma superchave, mas não o contrário.
- ◆ Em geral, qualquer superchave formada por apenas um atributo também é chave.
- ◆ Em geral, um esquema de relação pode ter mais de uma chave.
- ◆ Nesse caso, cada uma das chaves é chamada **CHAVE CANDIDATA**.



Chave Primária e Chave Única

- ◆ Na modelagem relacional, designa-se uma das chaves candidatas como chave primária (**PK** = primary key).
- ◆ A escolha da chave primária é arbitrária, porém prefere-se escolher uma chave primária com um único atributo.
- ◆ Valores **NULL** não são permitidos para chaves primárias.
- ◆ No esquema de relação, indica-se a **PK** com sublinhado.
- ◆ As outras chaves candidatas são chamadas Chaves Únicas (**UK** = Unique Key)



Chave primária e Chave única

- ◆ Considerando o estado da relação abaixo, quais as chaves poderiam ser chaves primárias e chaves únicas ?

Atributos →

ALUNO			
CODIGO	NOME	CURSO	CPF
3456	Paulo de Souza Alves	EC	109.098.340-12
9872	Francisco Silva	CC	239-345-986-12
1890	Francisco Silva	CC	234.876.456-45





Atributo Principal – Definição

- ◆ Um atributo do esquema de relação **R** é chamada **ATRIBUTO PRINCIPAL** de R se ele for um membro de alguma chave candidata de **R**.
- ◆ Um atributo é chamado **NÃO PRINCIPAL** se não for um atributo principal – ou seja, se não for um membro de qualquer chave candidata.





Primeira Forma Normal

- ◆ Parte da definição formal de uma relação no modelo relacional;
- ◆ Os únicos valores permitidos de atributo são os valores atômicos (indivisíveis);
- ◆ Afirma que o domínio de um atributo deve incluir apenas valores atômicos e que o valor de qualquer atributo em uma tupla deve ser um único valor do domínio desse atributo.
- ◆ Logo, 1FN **reprova** atributos multivalorados, atributos compostos e suas combinações.





Primeira Forma Normal

- ◆ Seja a Relação R que representa um DEPARTAMENTO com os atributos **Dnome** (nome do departamento), **Dnumero** (número do departamento), CPF_Gerente (cpf do gerente do departamento) e Dlocal (localização do departamento).
- ◆ Supomos que o Departamento possa ter um certo número de locais.

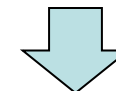
Departamento

Dnome	<u>Dnumero</u>	CPF_Gerente	Dlocal
-------	----------------	-------------	--------





Primeira Forma Normal



Departamento

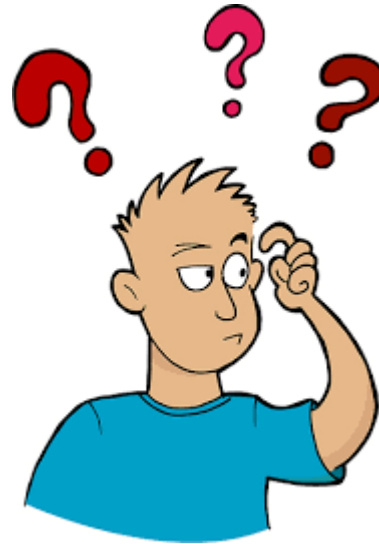
Dnome	<u>Dnumero</u>	CPF_Gerente	Dlocal
Vendas	5	333.876.334-15	Santo André, Itu, Santos
Administração	3	278.987.120-99	Rio de Janeiro
Engenharia	4	876.999.761-85	São Paulo, Sorocaba

- ◆ O domínio de **Dlocal** contém valores atômicos, mas algumas tuplas podem ter um conjunto desses valores.
- ◆ Nesse caso, **Dlocal** não é funcionalmente dependente da chave primária **Dnumero**.
- ◆ **Isso viola a 1FN**. Portanto a relação não está na 1FN. (Na verdade, R nem se qualifica como relação pois **não é aderente** ao modelo relacional.





Qual o procedimento para se obter a 1FN ?





1FN – Primeira Solução

- Remove-se o atributo Dlocal que viola a 1FN e cria-se uma nova relação LOCALIZACAO_DEP, compondo-se a chave primária do Departamento (Dnumero) com o atributo Dlocal.

DEPARTAMENTO

Dnome	<u>Dnumero</u>	CPF_Gerente
Vendas	5	333.876.334-15
Administração	3	278.987.120-99
Engenharia	4	876.999.761-85

LOCALIZACAO_DEP

<u>Dnumero</u>	<u>Dlocal</u>
5	Santo André
5	Itu
5	Santos
3	Rio de Janeiro
4	São Paulo
4	Sorocaba



1FN – Segunda Solução

- ◆ Expandir a chave de modo que haverá uma tupla separada na relação original DEPARTAMENTO. Nesse caso, a chave primária torna-se uma chave composta { Dnumero, Dlocal} .

DEPARTAMENTO

Dnome	<u>Dnumero</u>	CPF_Gerente	<u>Dlocal</u>
Vendas	5	333.876.334-15	Santo André
Vendas	5	333.876.334-15	Itu
Vendas	5	333.876.334-15	Santos
Administração	3	278.987.120-99	Rio de Janeiro
Engenharia	4	876.999.761-85	Sorocaba
Engenharia	4	876.999.761-85	São Paulo



1FN – Terceira Solução

- ◆ Se o número máximo de valores for conhecido para o atributo, por exemplo: se for conhecido no máximo três locais para um departamento, substituir o atributo Dlocal pelos atributos Dlocal1, Dlocal2 e Dlocal3.

DEPARTAMENTO

Dnome	<u>Dnumero</u>	CPF_Gerente	Dlocal1	Dlocal2	Dlocal3
Vendas	5	333.876.334-15	Santo André	Itu	Santos
Administração	3	278.987.120-99	Rio de Janeiro		
Engenharia	4	876.999.761-85	São Paulo	Sorocaba	



Dependência Funcional Total

- Uma dependência funcional $X \rightarrow Y$ é uma dependência funcional **total** se a remoção de qualquer atributo **A** de **X** significar que a dependência não se mantém mais.

FUNC_PROJ

<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
123987098-11	1	45	Antonio	ABX	SP
123987098-11	2	30	Antonio	TRW	Santo André
123987098-11	3	56	Antonio	RSA	Rio de Janeiro
765987098-34	3	65	Pedro	RSA	Rio de Janeiro
567987123-98	3	41	Ana	RSA	Rio de Janeiro
221897008-56	3	98	Carlos	RSA	Rio de Janeiro

- Exemplo: $\{ \text{CPF}, \text{ProjNumero} \} \rightarrow \text{Horas}$ é Dependência Funcional Total, pois a remoção de **CPF** ou de **ProjNumero** não determinará o atributo Horas.



Dependência Funcional Parcial

- Uma dependência funcional $X \rightarrow Y$ é uma dependência funcional **parcial** se algum atributo $A \in X$ puder ser removido de X e a dependência ainda se mantiver.

FUNC_PROJ

<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
123987098-11	1	45	Antonio	ABX	SP
123987098-11	2	30	Antonio	TRW	Santo André
123987098-11	3	56	Antonio	RSA	Rio de Janeiro
765987098-34	3	65	Pedro	RSA	Rio de Janeiro
567987123-98	3	41	Ana	RSA	Rio de Janeiro
221897008-56	3	98	Carlos	RSA	Rio de Janeiro

- Exemplo: $\{ \text{CPF}, \text{ProjNumero} \} \rightarrow \text{Fnome}$ é Dependência Funcional Parcial, pois a remoção de **ProjNumero** ainda configura uma Dependência Funcional.



Segunda Forma Normal

- ◆ Um esquema de relação **R** está na **2FN** se cada atributo não principal A em **R** for **TOTAL e FUNCIONALMENTE DEPENDENTE** da chave primária.
- ◆ O teste para **2FN** envolve testar as dependências funcionais cujos atributos do lado esquerdo fazem parte da chave primária.
- ◆ Se a chave primária tiver um único atributo, o teste não precisa aplicado.

Lembrando...



- Um atributo do esquema de relação **R** é chamada **ATRIBUTO PRINCIPAL** de **R** se ele for um membro de alguma chave candidata de **R**.
- Um atributo é chamado **NÃO PRINCIPAL** se **não** for um atributo principal – ou seja, se **não** for um membro de qualquer chave candidata.



Segunda Forma Normal – Contra Exemplo

- ◆ FUNC_PROJ NÃO está na 2FN (Primeiro motivo)

FUNC_PROJ

<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
123987098-11	1	45	Antonio	ABX	SP
123987098-11	2	30	Antonio	TRW	Santo André
123987098-11	3	56	Antonio	RSA	Rio de Janeiro
765987098-34	3	65	Pedro	RSA	Rio de Janeiro
567987123-98	3	41	Ana	RSA	Rio de Janeiro
221897008-56	3	98	Carlos	RSA	Rio de Janeiro

- ◆ O atributo não principal **Fnome** viola a 2FN.



- ◆ { **CPF** , **ProjNumero** } → **Fnome** não é DEPENDÊNCIA FUNCIONAL TOTAL (é parcial) da chave primária (CPF + ProjNumero)



Segunda Forma Normal – Contra Exemplo

◆ FUNC_PROJ NÃO está na 2FN

(Segundo motivo)

FUNC_PROJ

<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
123987098-11	1	45	Antonio	ABX	SP
123987098-11	2	30	Antonio	TRW	Santo André
123987098-11	3	56	Antonio	RSA	Rio de Janeiro
765987098-34	3	65	Pedro	RSA	Rio de Janeiro
567987123-98	3	41	Ana	RSA	Rio de Janeiro
221897008-56	3	98	Carlos	RSA	Rio de Janeiro

◆ Os atributos não principais **ProjNome** e **ProjLocal** violam a 2FN.



◆ { **CPF** , **ProjNumero** } → **ProjNome** **não** é DEPENDÊNCIA FUNCIONAL TOTAL (é parcial) da chave primária (CPF + ProjNumero) .

◆ { **CPF** , **ProjNumero** } → **ProjLocal** **não** é DEPENDÊNCIA FUNCIONAL TOTAL (é parcial) da chave primária (CPF + ProjNumero) .



Segunda Forma Normal – Exemplo

<u>CPF</u>	<u>ProjNumero</u>	Horas	Fnome	ProjNome	ProjLocal
------------	-------------------	-------	-------	----------	-----------



Normalizando para 2FN

FP_1

<u>CPF</u>	<u>ProjNumero</u>	Horas
------------	-------------------	-------

FP_2

<u>CPF</u>	Fnome
------------	-------

FP_3

<u>ProjNumero</u>	ProjNome	ProjLocal
-------------------	----------	-----------





Dependência Funcional Transitiva

- ◆ Uma dependência funcional $X \rightarrow Y$ em um esquema de relação R é uma dependência transitiva se houver um conjunto Z de atributos em R que nem sejam uma chave candidata nem um subconjunto de qualquer chave de R e tanto $X \rightarrow Z$ quanto $Z \rightarrow Y$ se mantiverem.



Dependência Funcional Transitiva – Exemplo

- ◆ A dependência **CPF** → **CPF_Gerente** por meio de **Dnumero** em FUNC_DEP, pois ambas as dependências **CPF** → **Dnumero** e **Dnumero** → **CPF_Gerente** se mantêm e **Dnumero** não é chave por si só nem um subconjunto da chave de FUNC_DEP.

FUNC_DEP

Fnome	<u>CPF</u>	DataNasc	Endereco	Dnumero	NomeDepto	CPF_Gerente
Antonio	123987098-11	19/10/1976	Rua A, 309	5	Compras	298.876.121-52
Ana	657987541-12	31/10/1987	Rua Brasil, 87	10	Vendas	876.098.776-81
Paula	980652098-99	07/03/1982	Av. Souza, 45	10	Vendas	876.098.776-81

- ◆ **CPF** → **Dnumero** é dependência funcional;
- ◆ **Dnumero** → **CPF_Gerente** é dependência funcional;
- ◆ **Dnumero** não é chave por si só;
- ◆ **Dnumero** não faz parte da chave primária de FUNC_DEP.



Terceira Forma Normal

- ◆ 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.

Lembrando...



- Um atributo do esquema de relação **R** é chamada **ATRIBUTO PRINCIPAL** de **R** se ele for um membro de alguma chave candidata de **R**.
- Um atributo é chamado **NÃO PRINCIPAL** se não for um atributo principal – ou seja, se não for um membro de qualquer chave candidata.



Terceira Forma Normal – Contra Exemplo

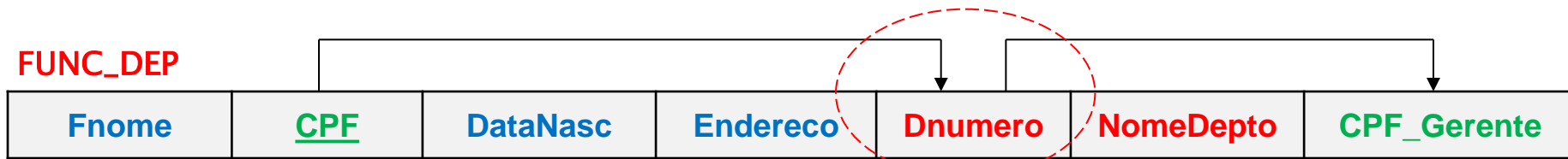
FUNC_DEP

Fnome	<u>CPF</u>	DataNasc	Endereco	Dnumero	NomeDepto	CPF_Gerente
Antonio	123987098-11	19/10/1976	Rua A, 309	5	Compras	298.876.121-52
Ana	657987541-12	31/10/1987	Rua Brasil, 87	10	Vendas	876.098.776-81
Paula	980652098-99	07/03/1982	Av. Souza, 45	10	Vendas	876.098.776-81

- ◆ A relação **FUNC_DEP** está na **2FN**, pois não existe dependência parcial sobre a chave;
- ◆ Mas, **FUNC_DEP** não está na **3FN**, devido à dependência funcional transitiva de CPF_Gerente (e também de NomeDepto) em CPF por meio de **Dnumero**.
- ◆ Havendo dependência funcional transitiva, a relação não está na 3FN.



Terceira Forma Normal – Exemplo



Normalizando para 3FN

DF1

Fnome	<u>CPF</u>	DataNasc	Endereco	Dnumero
-------	------------	----------	----------	---------

DF2

<u>Dnumero</u>	NomeDepto	CPF_Gerente
----------------	-----------	-------------

- Intuitivamente, vemos que DF1 e DF2 representam fatos de entidades independentes sobre funcionários e departamentos.



Mais exemplos ...



A relação a seguir NÃO está na 1FN:

Projeto						
<u>CodProj</u>	DescProj	<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc	DtInicio
11	Alfa	1001	Antonio	Analista Sr	1800	02/01/2005
		1004	Daniela	Analista Pl	1200	05/01/2005
12	Beta	1003	Claudio	Analista Sr	1800	10/02/2005



Passando para a 1FN:

ProjFunc					
<u>CodProj</u>	<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc	DtInicio
11	1001	Antonio	Analista Sr	1800	02/01/2005
11	1004	Daniela	Analista Pl	1200	05/01/2005
12	1003	Claudio	Analista Sr	1800	10/02/2005

Projeto	
<u>CodProj</u>	DescProj
11	Alfa
12	Beta



A relação a seguir NÃO está na 2FN:

ProjFunc					
<u>CodProj</u>	<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc	DtInicio
11	1001	Antonio	Analista Sr	1800	02/01/2005
11	1004	Daniela	Analista PI	1200	05/01/2005
12	1003	Claudio	Analista Sr	1800	10/02/2005



Passando para a 2FN:

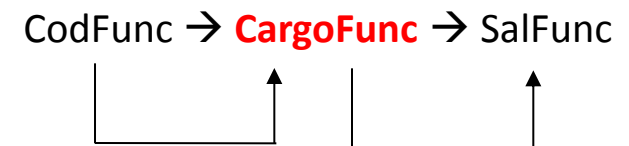
Funcionario			
<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc
1001	Antonio	Analista Sr	1800
1004	Daniela	Analista PI	1200
1003	Claudio	Analista Sr	1800

ProjFunc		
<u>CodProj</u>	<u>CodFunc</u>	DtInicio
11	1001	02/01/2005
11	1004	05/01/2005
12	1003	10/02/2005



A relação a seguir NÃO está na 3FN:

Funcionario			
<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc
1001	Antonio	Analista Sr	1800
1004	Daniela	Analista PI	1200
1003	Claudio	Analista Sr	1800



Dependência Funcional Transitiva



Passando para a 3FN:

Funcionario		
<u>CodFunc</u>	NomeFunc	CargoFunc
1001	Antonio	Analista Sr
1004	Daniela	Analista PI
1003	Claudio	Analista Sr

CargoSalario	
<u>CargoFunc</u>	SalFunc
Analista Sr	1800
Analista PI	1200