



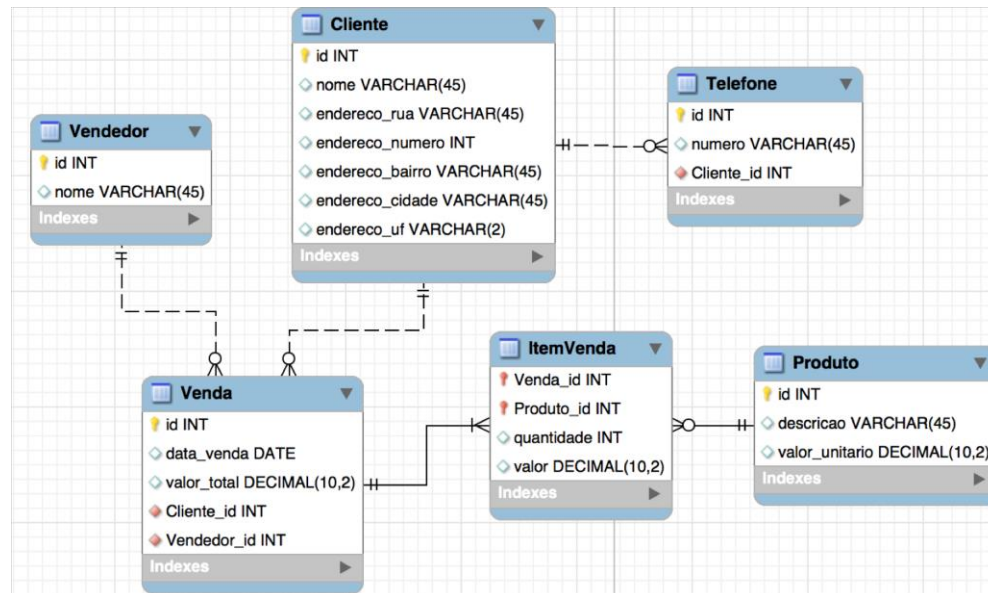
NORMALIZAÇÃO

Banco de dados

Prof. Me. Joice Wolfrann

NORMALIZAÇÃO

- A normalização de dados é uma série de passos que são realizados com a finalidade de propiciar um armazenamento mais consistente e um acesso mais eficiente aos dados de um BD relacional.
- A normalização é realizada antes de se implementar fisicamente o BD (*create table* etc)

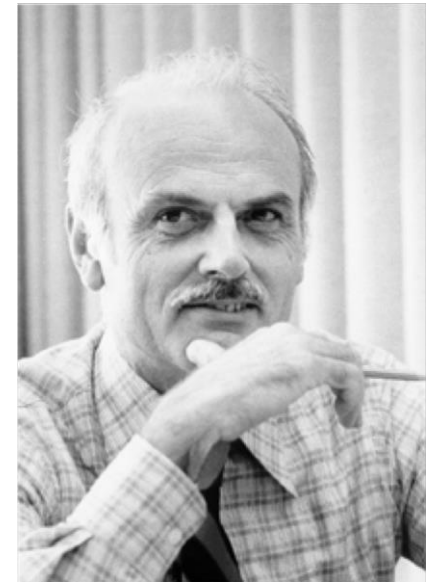


NORMALIZAÇÃO

- A normalização de tabelas tem por objetivo principal resolver problemas de atualização de bases de dados, minimizando redundâncias.
- Ele pode ser visto como o processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios, decompondo-os, através da separação de seus atributos em esquemas de relações menos complexas, mas que satisfaçam as propriedades desejadas.



- O processo de normalização como foi proposto inicialmente por **Codd** conduz um esquema de relação através de uma bateria de testes para certificar se o mesmo está na 1^a, 2^a e 3^a **Formas Normais**.
- Cada etapa ou teste corresponde uma determinada forma normal, que representa um progressivo refinamento na estrutura das tabelas.



- A normalização possui caráter organizativo e pode ocorrer durante a concepção do modelo conceitual, durante a derivação do modelo lógico para o relacional, ou após a derivação do modelo lógico.
- As principais características de uma base de dados normalizada são:
 - Geração de aplicações mais estáveis.
 - Aumento do número de tabelas utilizadas.
 - Diminuição dos tamanhos médios das tabelas.



BENEFÍCIOS DA NORMALIZAÇÃO:

- **Estabilidade do modelo lógico**: capacidade de um modelo manter-se inalterado face às mudanças que venham a ser percebida ou introduzidas no ambiente que tenha sido modelado.
- **Flexibilidade**: capacidade de adaptação a demandas diferenciadas, a expansão e redução, a omissão ou presença, etc.
- **Integridade**: diz respeito à qualidade do dado. Um dado mapeado em mais de um local de modo diferente, com valores instanciados de modo diferentes, pode ser indício de que não há integridade entre eles.



BENEFÍCIOS DA NORMALIZAÇÃO:

- **Economia**: no espaço de armazenamento em relação ao custo de manipulação de dados (que representa todo e qualquer esforço, tempo, ou valor agregado ao fato de manipularmos volumes de dados maiores do que os efetivamente necessários); custo causado pelo atraso do fornecimento da informação desejada.
- **Fidelidade ao ambiente observado**: ajuda a definir elementos que foram despercebidos durante o processo de modelagem.



- Os dados redundantes desperdiçam espaço em disco e criam problemas de manutenção.
 - Se os dados existentes em mais de um local precisarem ser alterados, os dados devem ser alterados exatamente da mesma maneira em todos os locais.
- Uma alteração de endereço do cliente é muito mais fácil de ser implementada se esses dados são armazenados apenas na tabela clientes e em outro lugar no banco de dados.



POR QUE NORMALIZAR ?

- 1º) Minimizar de redundâncias e inconsistências;
- 2º) Facilitar manipulações do Banco de Dados;
- 3º) Facilitar manutenção do Sistema de Informações



O QUE É UMA "DEPENDÊNCIA INCONSISTENTE"?

- Embora seja intuitivo que um usuário procure a tabela clientes para obter o endereço de um cliente específico, talvez não seja bom procurar o salário do funcionário que faz a chamada no cliente.
- O salário do funcionário está relacionado, **ou dependente**, do funcionário e, portanto, deve ser movido para a tabela funcionários.
- Dependências inconsistentes podem dificultar o acesso dos dados porque o caminho para localizar os dados pode estar ausente ou quebrado.



NORMALIZAÇÃO

Codd definiu 3 formas normais, conhecidas por:

- 1ª Forma Normal (1FN)
- 2ª Forma Normal (2FN)
- 3ª Forma Normal (3FN)



NORMALIZAÇÃO

1ª FORMA NORMAL (1FN)

- Uma relação está na primeira forma normal quando todos os atributos contêm apenas um valor correspondente, singular e não existem grupos de atributos repetidos — ou seja, não admite repetições ou campos que tenham mais que um valor.



NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Uma tabela está na 1FN se não houver grupo de dados repetidos, ou seja, se todos os valores forem únicos
- Visa eliminar grupos repetidos de dados (multivalorados) ou campos formados por mais de 1 valor (compostos)



NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Exemplo-1: campo composto

Id	Nome	Residência
1	João	Natal, RN
2	Maria	Caicó, RN
3	José	São Paulo, SP
4	Alex	Fortaleza, CE



Campo composto

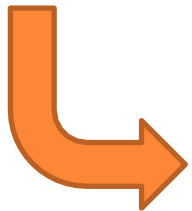
- Como saber quem mora em Natal? Ou no RN?



NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Exemplo-1: campo composto (NORMALIZAÇÃO)

Id	Nome	Residência
1	João	Natal, RN
2	Maria	Caicó, RN
3	José	São Paulo, SP
4	Alex	Fortaleza, CE

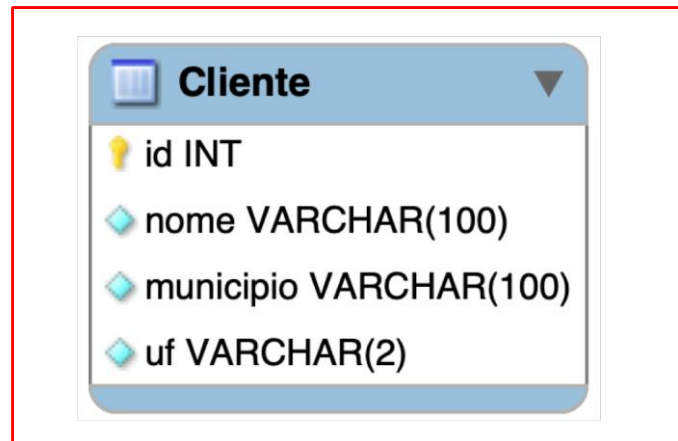
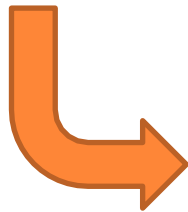
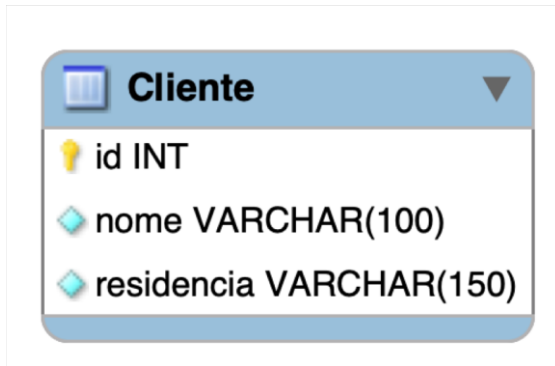


Id	Nome	Município	UF
1	João	Natal	RN
2	Maria	Caicó	RN
3	José	São Paulo	SP
4	Alex	Fortaleza	CE

- Cria-se um campo para cada valor do campo composto

NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Exemplo-1: campo composto (NORMALIZAÇÃO)



- Cria-se um campo para cada valor do campo composto

NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Exemplo-2: campo multivalorado

Id	Nome	Telefones
1	João	3322-1122
1	João	9999-8877
2	Maria	8888-4433
3	José	2211-5566
3	José	9765-1234

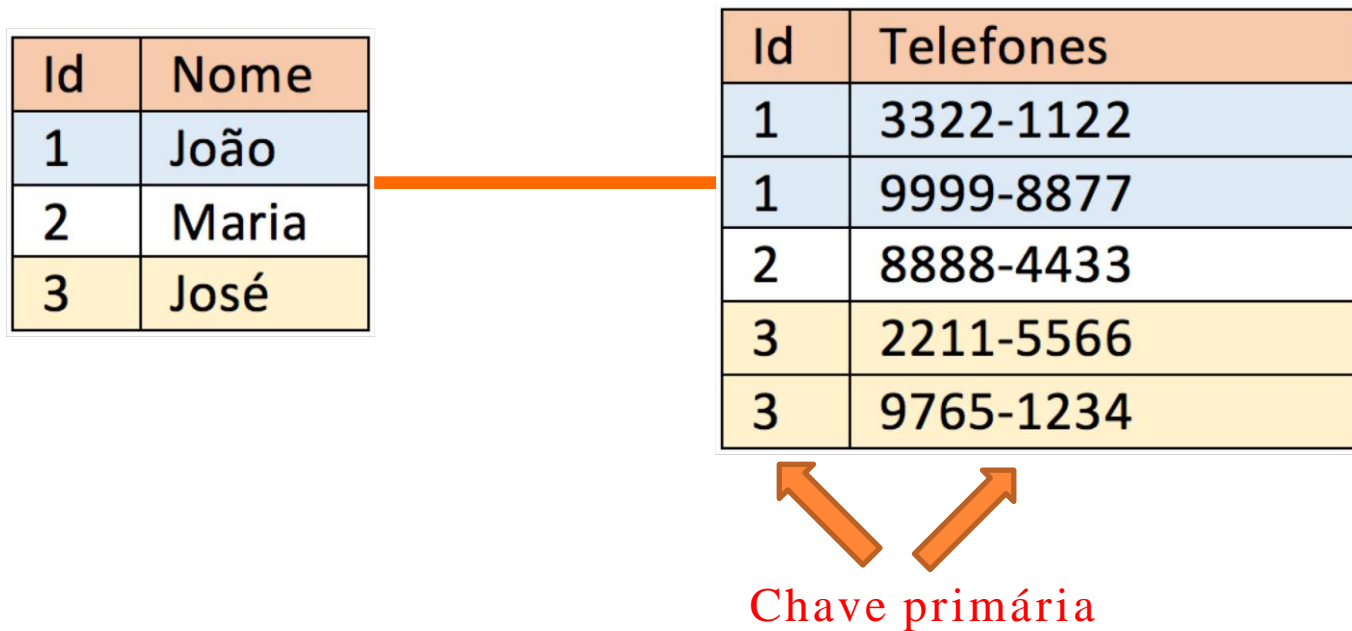


Campo multivalorado

- Campos ID e NOME duplicados na tabela
- Se precisarmos alterar o nome de João, teremos que atualizar mais de 1 registro

NORMALIZAÇÃO

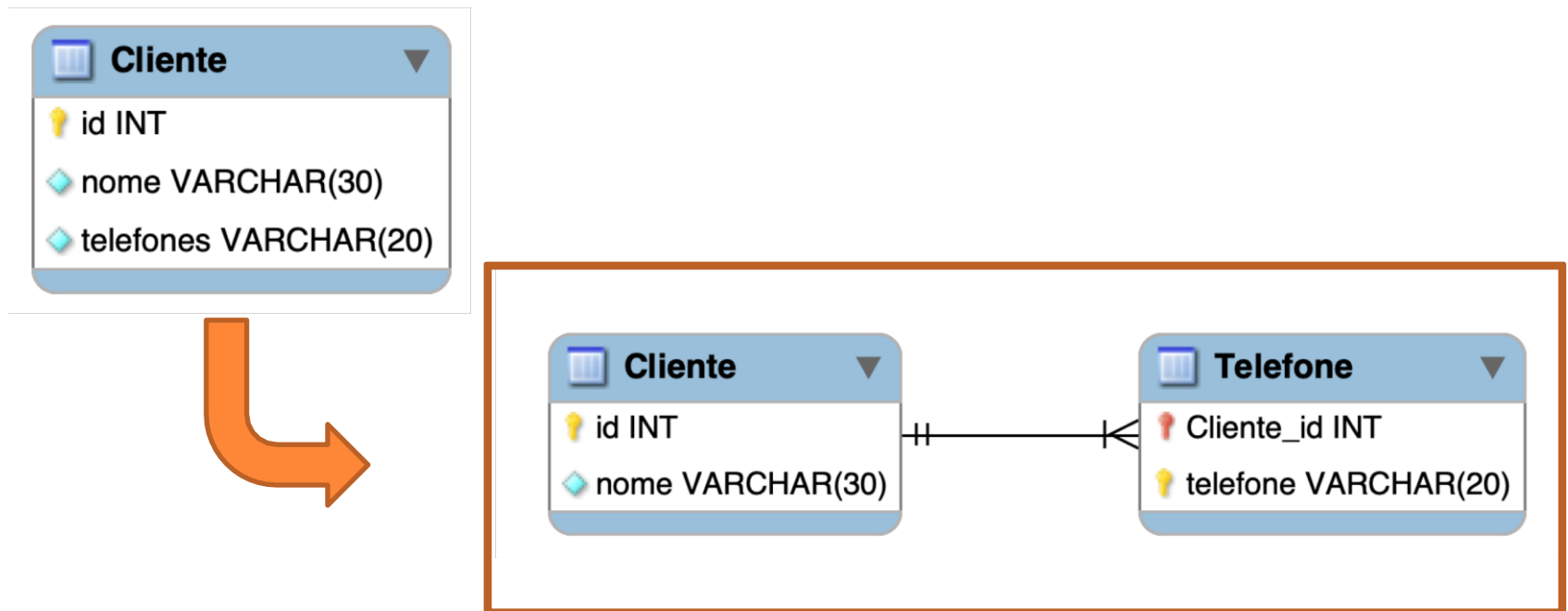
- 1ª Forma Normal (1FN)
- Exemplo-2: campo multivalorado (NORMALIZAÇÃO)



- Cria-se uma nova tabela contendo os valores do campo multivalorado

NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Exemplo-2: campo multivalorado (NORMALIZAÇÃO)



- Cria-se uma nova tabela contendo os valores do campo multivalorado

NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Exemplo-3:

Projetos	
🔑	CodProjetos: INTEGER
◆	TipoProjeto: VARCHAR(20)
◆	Descrição: VARCHAR(45)
◆	Matricula: VARCHAR(20)
◆	Nome: VARCHAR(20)
◆	Categoria: VARCHAR
◆	TipoSalario: FLOAT
◆	Data_2: INTEGER
◆	Tempo: INTEGER

Arquivo de Projetos (CodProjetos, TipoProjeto, Descrição, Matricula, Nome, Categoria, TipoSalario, data, Tempo)



NORMALIZAÇÃO

- 1ª Forma Normal (1FN)
- Exemplo-3:

O resultado após a aplicação da primeira forma normal (1FN) será:

Projetos	
🔑	idProjetos: INTEGER
◆	TipoProjeto: VARCHAR(20)
◆	Descrição: VARCHAR(45)

Equipes	
🔑	CodProj: INTEGER
🔑	Matricula: VARCHAR(20)
◆	Nome: VARCHAR(20)
◆	Categoria: VARCHAR
◆	TipoSalario: FLOAT
◆	Data_2: DATE
◆	Tempo: INTEGER

Considerando-se, agora, as entidades:

Arquivo de Projetos (CodProjetos, TipoProjeto, Descrição, Matricula, Nome, Categoria, TipoSalario, data, Tempo)

Arquivo de Equipe (CodProjetos, Matricula, Nome, Categoria, Tiposalario, Data, Tempo)



NORMALIZAÇÃO

○ 2ª Forma Normal (2FN)

- A 2FN só é aplicável para tabelas que possuem uma **chave primária composta** e que, além disso, tenham outros atributos que não façam parte da chave primária.
- Uma tabela está na 2FN se estiver na 1FN e todo atributo que não compõe a chave primária deve ter dependência funcional total em relação à chave primária.



Código pedido	Valor Pedido	Data pedido	Telefone de contato	Produto				
				Código produto	Nome produto	Valor unitário do produto	Quantidade	Valor pago por produto
100	3300,00	30/11/2009	2222-2222 9999-9999	1	Computador	1500,00	1	1500,00
				5	Impressora	600,00	2	1200,00
				6	Papel A4	12,00	50	600,00
101	3800,00	15/12/2009	2121-2121 9191-9191	2	Mouse	30,00	10	300,00
				5	Impressora	600,00	5	3000,00
				7	Teclado	50,00	10	500,00

No modelo relacional, a **dependência funcional** entre dois atributos, A e B, ocorre quando, em todas as linhas da tabela, para cada valor de A irá aparecer sempre o mesmo valor de B.

Por exemplo, na tabela sempre que aparecer o código de produto “5” teremos como nome do produto “Impressora”.

Assim, o nome do produto depende funcionalmente do código do produto.

Para denotar a dependência funcional, usa-se uma expressão na forma “codigo_produto → nome_produto”. Isso significa que o nome do produto depende funcionalmente do código do produto.

NORMALIZAÇÃO

- 2ª Forma Normal (2FN)
- Portanto: Uma tabela está na 2FN quando está na 1FN e seus campos dependem funcionalmente da totalidade da chave primária (quando composta)
- Não pode haver atributos que dependam apenas de uma parte da chave primária composta



NORMALIZAÇÃO

o 2ª Forma Normal (2FN)

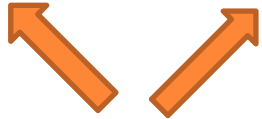
Para cada tabela que tiver chave primária composta e pelo menos um atributo que não faz parte da chave, deve-se verificar se cada um dos atributos não chave têm **dependência funcional total**.

Caso a dependência não seja total, deve-se criar uma tabela com o atributo que depende parcialmente, mais o atributo do qual ele depende (que será chave primária na nova tabela e será chave estrangeira na tabela inicial).

NORMALIZAÇÃO

- 2ª Forma Normal (2FN)
- Exemplo-1:

<u>CPF</u>	<u>Projeto Id</u>	Horas Trab.	Funcionário	<u>Projeto Nome</u>	<u>Projeto UF</u>
1	100	150	João	SUAP	RN
2	100	120	Maria	SUAP	RN
3	200	90	José	SIS-VENDAS	SP
1	300	45	João	SIGA-A	CE




Chave primária

- Cada campo não-chave depende totalmente dos campos da chave-primária?

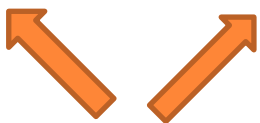
NORMALIZAÇÃO

- 2ª Forma Normal (2FN)

- Exemplo-1:



<u>CPF</u>	<u>Projeto_Id</u>	Horas Trab.	Funcionário	<u>Projeto_Nome</u>	<u>Projeto_UF</u>
1	100	150	João	SUAP	RN
2	100	120	Maria	SUAP	RN
3	200	90	José	SIS-VENDAS	SP
1	300	45	João	SIGA-A	CE




Chave primária

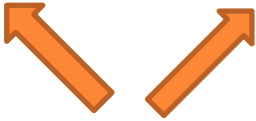
- Campo **Horas Trab** depende do CPF e do Projeto_Id, obedecendo assim à 2FN

NORMALIZAÇÃO

- 2ª Forma Normal (2FN)
- Exemplo-1:



<u>CPF</u>	<u>Projeto_Id</u>	Horas Trab.	Funcionário	<u>Projeto_Nome</u>	<u>Projeto_UF</u>
1	100	150	João	SUAP	RN
2	100	120	Maria	SUAP	RN
3	200	90	José	SIS-VENDAS	SP
1	300	45	João	SIGA-A	CE




Chave primária

- Campo **Funcionário** depende apenas do CPF e não obedece à 2FN

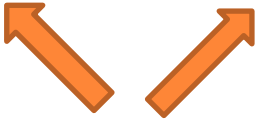


NORMALIZAÇÃO

- 2ª Forma Normal (2FN)
- Exemplo-1:



<u>CPF</u>	<u>Projeto_Id</u>	Horas Trab.	Funcionário	<u>Projeto_Nome</u>	<u>Projeto_UF</u>
1	100	150	João	SUAP	RN
2	100	120	Maria	SUAP	RN
3	200	90	José	SIS-VENDAS	SP
1	300	45	João	SIGA-A	CE




Chave primária

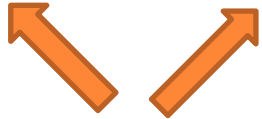
- Campo **Projeto_Nome** depende apenas do **Projeto_Id** e não obedece à 2FN

NORMALIZAÇÃO

- 2ª Forma Normal (2FN)
- Exemplo-1:



<u>CPF</u>	<u>Projeto_Id</u>	Horas Trab.	Funcionário	<u>Projeto_Nome</u>	<u>Projeto_UF</u>
1	100	150	João	SUAP	RN
2	100	120	Maria	SUAP	RN
3	200	90	José	SIS-VENDAS	SP
1	300	45	João	SIGA-A	CE



Chave primária

- Campo **Projeto_UF** depende apenas do Projeto_Id e não obedece à 2FN

NORMALIZAÇÃO

- 2ª Forma Normal (2FN)
- Exemplo-1: (NORMALIZAÇÃO)

<u>CPF</u>	<u>Projeto_Id</u>	Horas Trab.	Funcionário	<u>Projeto_Nome</u>	<u>Projeto_UF</u>
1	100	150	João	SUAP	RN
2	100	120	Maria	SUAP	RN
3	200	90	José	SIS-VENDAS	SP
1	300	45	João	SIGA-A	CE



<u>CPF</u>	Funcionário
1	João
2	Maria
3	José

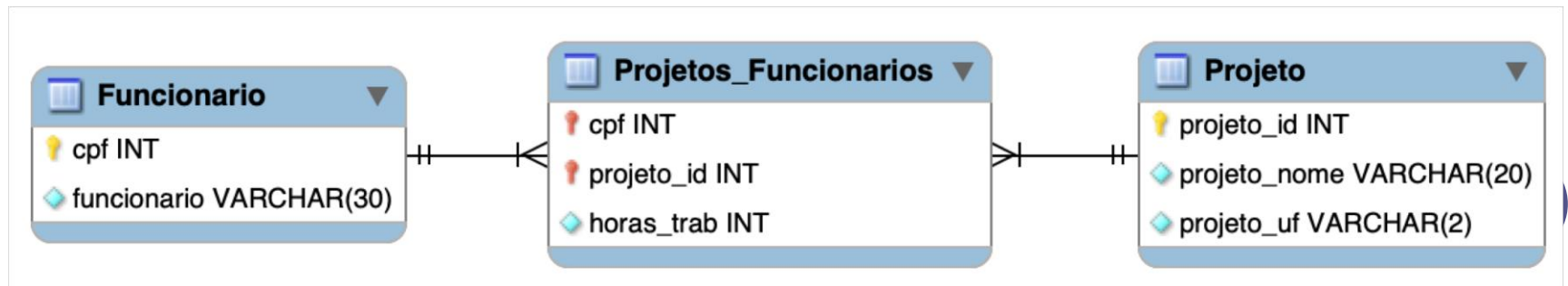
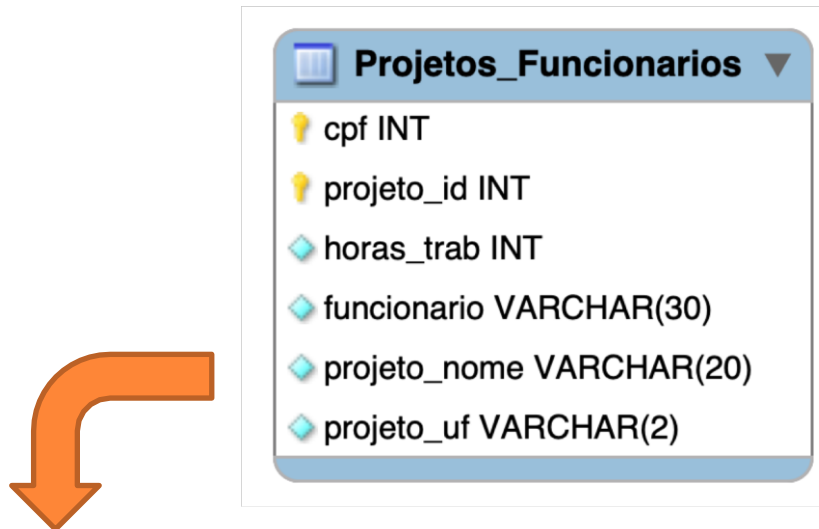
<u>CPF</u>	<u>Projeto_Id</u>	Horas Trab.
1	100	150
2	100	120
3	200	90
1	300	45

<u>Projeto_Id</u>	<u>Projeto_Nome</u>	<u>Projeto_UF</u>
100	SUAP	RN
200	SIS-VENDAS	SP
300	SIGA-A	CE

- Criam-se tabelas para armazenar os campos que dependam parcialmente da chave-primária, restando na tabela original apenas os campos que dependam totalmente da chave-primária

NORMALIZAÇÃO

- **2ª Forma Normal (2FN)**
- **Exemplo-1: (NORMALIZAÇÃO)**

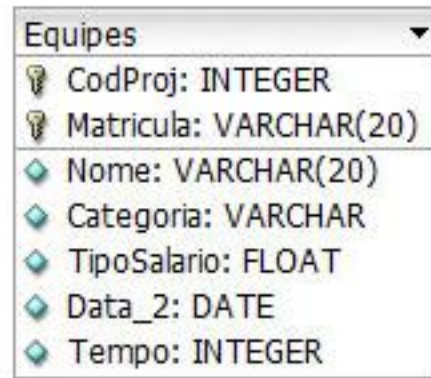


NORMALIZAÇÃO

2ª FORMA NORMAL (2FN)

EXEMPLO-2

- **Arquivo de Equipes** (CodProjetos, Matricula, Nome, Categoria, Tiposalario, Data, Tempo)



Equipes	
🔑	CodProj: INTEGER
🔑	Matricula: VARCHAR(20)
💎	Nome: VARCHAR(20)
💎	Categoria: VARCHAR
💎	TipoSalario: FLOAT
💎	Data_2: DATE
💎	Tempo: INTEGER

- O resultado após a aplicação da segunda forma normal (2FN) será:
- **Arquivo de Equipe** (CodProjetos, Matricula, Data, Tempo)
- **Arquivo de Funcionario** (Matricula, Nome, Categoria, Tiposalario)



NORMALIZAÇÃO

- 3ª Forma Normal (3FN)
- Uma tabela está na 3FN quando está na 2FN e não existir dependência funcional transitiva entre seus atributos
- Não pode existir dependência funcional transitiva entre atributos

Dependência Transitiva ocorre quando existe um atributo que não é chave e nem faz parte da chave, mas que identifica outros atributos.

Ou seja, existe um atributo não chave que depende de outro atributo não chave.

NORMALIZAÇÃO

- 3ª Forma Normal (3FN)
- Uma tabela está na 3FN quando está na 2FN e não existir dependência funcional transitiva entre seus atributos
- Não pode haver um campo que seja determinado por outro campo não-chave



NORMALIZAÇÃO

- 3ª Forma Normal (3FN)
- Exemplo-1:

<u>CPF</u>	Funcionário	Sexo	<u>Depart Id</u>	<u>Depart Nome</u>
1	João	<u>Masc</u>	11	RH
2	Maria	<u>Fem</u>	22	Financeiro
3	José	<u>Masc</u>	11	RH


Estes 3 campos dependem da chave-primária e obedecem à 3FN



NORMALIZAÇÃO

- 3ª Forma Normal (3FN)
- Exemplo-1:

<u>CPF</u>	Funcionário	Sexo	<u>Depart_Id</u>	<u>Depart_Nome</u>
1	João	<u>Masc</u>	11	RH
2	Maria	<u>Fem</u>	22	Financeiro
3	José	<u>Masc</u>	11	RH



Já este campo não depende da chave-primária, mas sim do campo Depart_Id e não obedece à 3FN



NORMALIZAÇÃO

- 3ª Forma Normal (3FN)
- Exemplo-1: (NORMALIZAÇÃO)

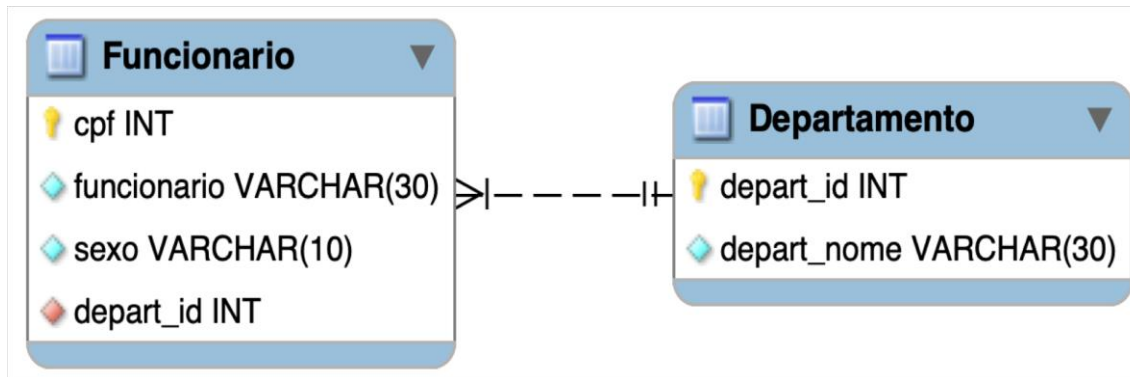
<u>CPF</u>	Funcionário	Sexo	<u>Depart_Id</u>
1	João	<u>Masc</u>	11
2	Maria	<u>Fem</u>	22
3	José	<u>Masc</u>	11

Departamento	
Depart_id	Depart_nome
11	RH

- Cria-se uma tabela para armazenar o campo que não depende da chave-primária, tendo esta nova tabela como chave-primária o campo da qual é dependente na tabela original

NORMALIZAÇÃO


- 3ª Forma Normal (3FN)
- Exemplo-1: (NORMALIZAÇÃO)







NORMALIZAÇÃO

3ª FORMA NORMAL (3FN)

- Exemplo 2:
- ❖ **Arquivo de Equipes** (CodProjetos, Matricula, Data, Tempo)
- ❖ **Arquivo de Funcionario** (Matricula, Nome, Categoria, Tiposalario)

Equipes	
	CodProj: INTEGER
	Matricula: VARCHAR(20)
	Data_2: DATE
	Tempo: INTEGER



Funcionario	
	Matricula: INTEGER
	Nome: VARCHAR(20)
	Categoria: VARCHAR
	TipoSalario: FLOAT





NORMALIZAÇÃO

3ª FORMA NORMAL (3FN)

- *O resultado após a aplicação da Terceira forma normal (3FN) será:*

Categoria	
	CodCategoria: INTEGER
	Descricao: VARCHAR(20)

TipoSalario	
	CodTipoSalario: INTEGER
	Descricao: VARCHAR(20)
	Valor: FLOAT
	DataInicio: DATE

TipoProjeto	
	CodTipoProjeto: INTEGER
	Descricao: VARCHAR

- ❖ **Arquivo de Categoria** (CodCategoria, Descricao)
- ❖ **Arquivo de TipoSalario** (CodTipoSalario, Descricao, Valor, DataInicio)
- ❖ **Arquivo de TipoProjeto** (CodTipoProjeto, Descricao)



NORMALIZAÇÃO

4ª FORMA NORMAL (4FN)

Uma tabela está na 4FN quando:

- Está na 3FN;
- Na tabela não pode conter múltiplos fatos multivalorados (diferente de campos multivalorados).
- Identificar as relações entre os atributos para determinação dos fatos multivalorados;
- Estabelecer subconjuntos para cada fato multivalorado.



NORMALIZAÇÃO

4ª FORMA NORMAL (4FN)

A Quarta Forma Normal (4FN) fala sobre fatos multivalorados, ou seja, sobre relacionamentos muitos-para-muitos e muitos-para-um.

De uma forma mais específica, para que um esquema enquadre-se nesta diretriz, este deve estar na 3FN e não pode conter dois ou mais fatos multivalorados independentes.

O objetivo desta restrição é tentar minimizar o número de campos da chave composta de uma entidade.



NORMALIZAÇÃO

4ª FORMA NORMAL (4FN)

- Considere uma tabela que armazena informações sobre planos de saúde e exames de um paciente:
 - ❖ Paciente;
 - ❖ Plano;
 - ❖ Exame.

Paciente	Plano	Exame
Murilo	São Camilo	Endoscopia
Murilo	Unimed	Endoscopia
Murilo	São Camilo	Hemograma
Murilo	Unimed	Hemograma



NORMALIZAÇÃO

4ª FORMA NORMAL (4FN)

<u>Paciente</u>	<u>Plano</u>
Murilo	São Camilo
Murilo	Unimed

<u>Paciente</u>	<u>Exame</u>
Murilo	Endoscopia
Murilo	Hemograma



NORMALIZAÇÃO

4ª FORMA NORMAL (4FN)

○ Exemplo 2:

Professor	Disciplina	Titulação
Murilo	Arquitetura e Organização de Computadores	Especialista
Murilo	Inteligência Artificial	Especialista
João	Lógica de Programação	Doutor
João	Sistemas Microcontrolados	Doutor

<u>Professor</u>	<u>Disciplina</u>	<u>Professor</u>	<u>Titulacao</u>
Murilo	Arquitetura e Organização de Computadores	Murilo	Especialista
Murilo	Inteligência Artificial	João	Doutor
João	Lógica de Programação		
João	Sistemas Microcontrolados		

NORMALIZAÇÃO

5ª FORMA NORMAL (5FN)

- Uma tabela está na 5FN quando:
- Está na 4FN;
- Quando um campo (**atributo**) está em outra tabela sem a necessidade de estar na tabela pesquisada.
 - Não há a perda de nenhuma informação.



NORMALIZAÇÃO

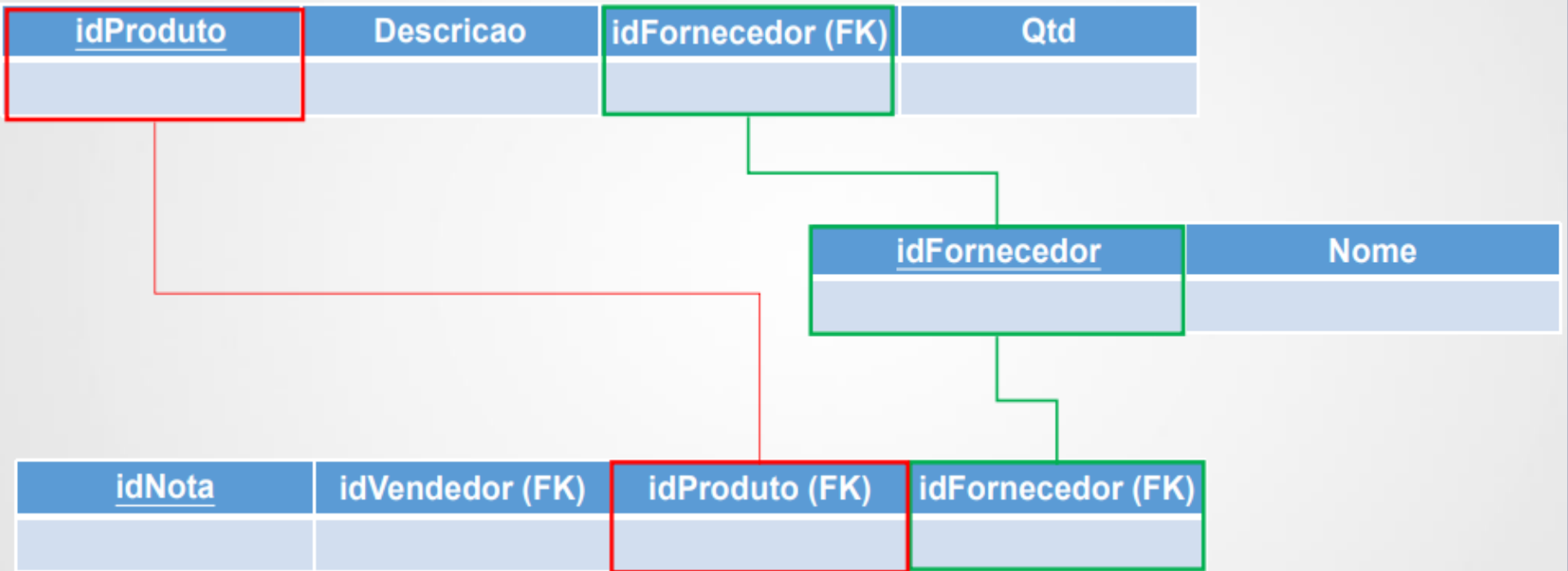
5ª FORMA NORMAL (5FN)

- A Quinta Forma Normal (5FN) trata os casos onde uma determinada informação pode ser reconstruída de informações menores combinadas.
- Ela em nada difere da 4FN se não houver uma constante simétrica que atue como uma regra de mundo entre as tabelas em questão.
- Na ausência desta contante, se o esquema estiver na 4FN, automaticamente estará, também, na 5FN.



NORMALIZAÇÃO

5ª FORMA NORMAL (5FN)



NORMALIZAÇÃO

5ª FORMA NORMAL (5FN)

