



 **ime**
INSTITUTO MATEMÁTICA E ESTATÍSTICA

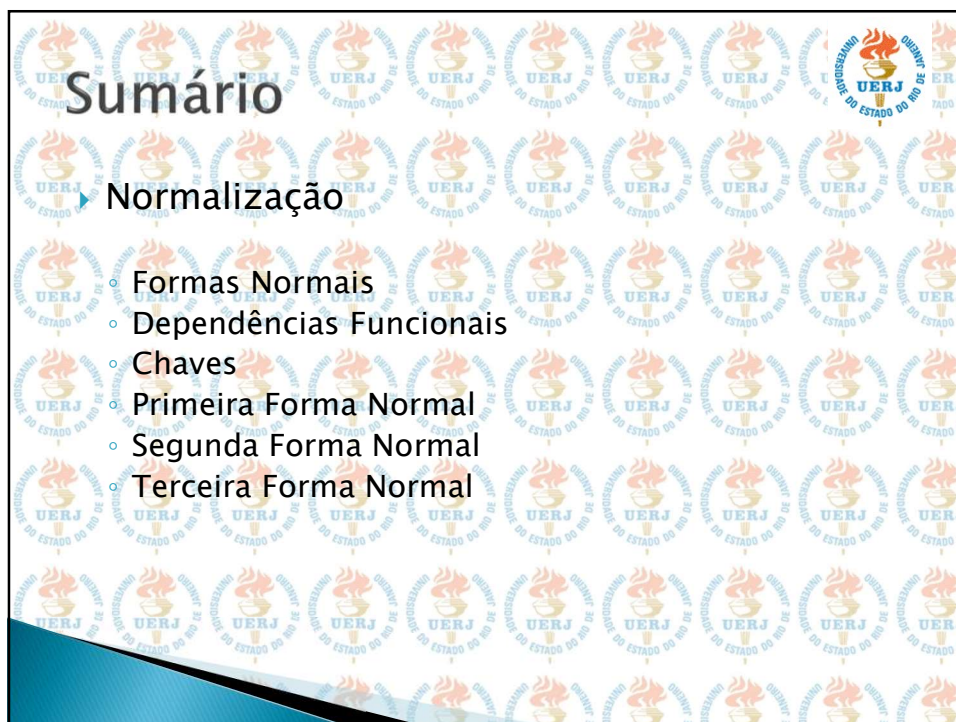
Normalização de Relações

Ana Carolina Almeida

Heuser, 2009 - Capítulo 6
Elmasri & Navathe, 2011 - Capítulo 15
Date, 2004 - Capítulos 11 e 12
Silberschätz et al, 2012 - Capítulo 8

Material adaptado das notas de aula do prof. Sérgio Côrtes (Puc-Rio)

1



Sumário

- ▶ Normalização
 - Formas Normais
 - Dependências Funcionais
 - Chaves
 - Primeira Forma Normal
 - Segunda Forma Normal
 - Terceira Forma Normal

2

Normalização – Formas Normais

► Normalização é a formalização de uma metodologia que, dada uma coleção de dados para serem representados em um banco de dados, defina uma estrutura lógica adequada para aqueles dados.

► Teoria da normalização baseia-se no conceito de forma normal de uma relação.

► Uma relação está em uma determinada forma normal se ela satisfaz a um determinado conjunto de restrições.



3

As Formas Normais

► Universo de Relações (Normalizadas e Não Normalizadas)

Relações 1ª FN (Relações Normalizadas)

Relações 2ª FN

Relações 3ª FN

Relações BCNF (Boyce-Codd)

Relações 4ª FN

Relações PJNF, (5ª FN)

PJNF (Project-Join Normal Form):
Forma Normal Projeção/Junção

4

Dependências Funcionais

► Dada uma relação “R”, um atributo “A1” de “R” é funcionalmente dependente do atributo “A2”, se e somente se, cada valor de “A2” em “R” tem associado a ele precisamente um valor de “A1”.

$R.A2 \rightarrow R.A1$

Os atributos “A1” e “A2” podem ser Compostos

5

Dependências Funcionais

► Fornecedor F

Código	Nome	Cidade	População
01	José	Fortaleza	30
02	Sandro	João Pessoa	20
03	Gabriel	Rio de Janeiro	40
04	José	Rio de Janeiro	40

Nome é funcionalmente dependente de Código, pois cada valor de Código possui associado somente um Nome.

$F.Código \rightarrow F.Nome$ $\rightarrow 01 \rightarrow José$

Os atributos “A1” e “A2” podem ser Compostos

$F.Código \rightarrow F.Nome$

$F.Código \rightarrow F.Cidade$

$F.Código \rightarrow F.População$

OU

$F.Código \rightarrow F.(Nome, cidade, população)$

$F.01 \rightarrow José$

$F.01 \rightarrow Fortaleza$

$F.01 \rightarrow 30$

Por exemplo, o mesmo não ocorreria para $F.Nome \rightarrow F.Código$ porque teríamos n valores de códigos associados à José (Códigos 01 e 04).

6

Chaves Candidatas

Seja "R" uma relação com atributos A_1, A_2, \dots, A_n . O conjunto de atributos (A_1, A_2, \dots, A_k) de "R" é o conjunto de chaves candidatas de "R" se satisfaz as seguintes condições:

Em nenhum momento, duas tuplas de "R" tem o mesmo valor para A_1, A_2, \dots, A_k
(Unicidade)

Nenhum dos A_1, A_2, \dots, A_k podem ser descartados do conjunto sem destruir a propriedade de unicidade acima. Nenhum subconjunto tem a propriedade da unicidade
(Minimalidade/Irreducibilidade)

Exemplos para a relação:

Casamento (NomeMarido, NomeEsposa, DtCasamento)

- Chaves Candidatas 1 = {NomeMarido, DtCasamento}
- Chaves Candidatas 2 = {NomeEsposa, DtCasamento}
- Chaves Candidatas 3 = {NomeMarido, NomeEsposa}

7

Chave Primária

Chave Primária é um caso particular de chave candidata.

Para uma determinada relação, uma chave candidata é determinada, teoricamente, de modo arbitrário, como **chave primária**.

As demais chaves candidatas, se existirem, são chamadas **chaves alternadas/alternativas**.

NOTAS: Toda relação tem pelo menos uma chave candidata. Em último caso a combinação de todos os seus atributos tem a propriedade da Unicidade (A_1, A_2, \dots, A_n).

Não faz sentido duas tuplas iguais.

8

Dependências Funcionais – Conclusão

- Se um atributo “A1” é uma chave candidata da relação “R” e, em particular ele é chave primária, então todos os demais atributos de “R” dependem funcionalmente de “A1”.
- ATENÇÃO:** Não existe nenhuma exigência na definição de dependência funcional que “A1” seja uma chave candidata de “R”.
 - Se na relação Fornecedor (código, nome, cidade, população), população representa o número de pessoas que vivem na cidade do fornecedor estamos modelando então:

Cidade → População

Existe dependência funcional, mas **Cidade não** é chave candidata de Fornecedor.

9

Por que normalizar?

Imagine um arquivo com as seguintes informações:

• Identificador do pedido	• CNPJ do cliente	◦ Preço unitário
◦ Identificador da filial	• Nome do cliente	◦ Preço total
◦ Identificador do pedido	• Endereço do cliente	• Valor total do pedido
• Data do pedido	• Itens solicitados, no limite de 10 itens por pedido	• Imposto a recolher
• Nome da filial	◦ Identificador do item	• Sinal
• Endereço da filial	◦ Descrição do item	• Restante a pagar
• Matrícula do vendedor	◦ Quantidade	
• Nome do vendedor		

Questões:

- Se o proprietário solicitar o aumento do número de itens por pedido?
- Se uma filial mudar de endereço?
- Se houver uma alteração na quantidade de um dos itens?
- Se for necessário enviar uma correspondência a todos os clientes?
- Se o espaço em disco for limitado e o volume de pedidos for muito grande?

10

Por que normalizar?

► Algumas observações:

Rever quantidade de itens

Duplicidade ou ausência de dado

Mudanças no arquivo e programa

Tempo + Esforço

Desperdício de espaço

Filial1
123456
29/04/2017
Filial Centro
Av. Presidente Vargas, 591
Vendedor1
Ana Carolina
123456789/0001-12
João da Silva
Rua São Francisco, 100
Item1
Pilha duracell
5
R\$ 10,00
R\$ 50,00
Item 2
...
R\$150,00
R\$ 15,00
R\$ 50,00
R\$ 115,00
...
10 itens por pedido

11

Por que normalizar?

► Dividir o arquivo em diversos arquivos?

Pedido

Identificador da filial
Identificador do pedido
Data do pedido
Matrícula do vendedor
CNPJ do cliente
Sinal
Preço total
Valor total do pedido
Imposto a recolher
Restante a pagar

Item

Identificador do item
Descrição do item
Preço unitário

Vendedor

Matrícula do vendedor
Nome do vendedor

Filial

Identificador da filial
Nome da filial
Endereço da filial

Item do Pedido

Identificador da filial
Identificador do pedido
Identificador do item
Quantidade

Cliente

CNPJ do cliente
Nome do cliente
Endereço do cliente

12

Anomalias de Atualização

Seja a relação/tabela de Fornecedor F

Fornecedor FN	Cidade	População	Peça PN	Nome da Peça	Quantidade
F01	Fortaleza	30	P01	Parafuso	200
F01	Fortaleza	30	P02	Porca	100
F01	Fortaleza	30	P03	Lixa	150
F01	Fortaleza	30	P04	Martelo	50
F01	Fortaleza	30	P05	Prego	250
F01	Fortaleza	30	P06	Bucha	200
F02	João Pessoa	20	P01	Parafuso	300
F02	João Pessoa	20	P06	Bucha	300
F03	Rio de Janeiro	40	P06	Bucha	50
F04	Recife	25	P06	Bucha	350
F04	Recife	25	P05	Prego	350
F04	Recife	25	P03	Lixa	150

13

Anomalias de Atualização

As redundâncias encontradas na relação do exemplo anterior levam a uma variedade de dificuldades ou exigência de maior atenção e controle para sua atualização.

Caracterizando:

- Anomalia de Inserção (Insert)
 - Só podemos incluir um Fornecedor caso haja algum suprimento/peça oriundo do mesmo.
- Anomalia de Exclusão (Delete)
 - Se o Fornecedor só possuir *um* suprimento/peça e este for excluído, perdemos os dados do Fornecedor. Ex.: F03.
- Anomalia de Atualização (Update)
 - Caso haja alteração em qualquer atributo de Fornecedor, esta deverá ser feita em todas as tuplas do mesmo. Ex.: população.

14

Primeira Forma Normal

Uma relação "R" está na PRIMEIRA FORMA NORMAL - 1ªFN se e somente se todos os domínios considerados contêm somente valores atômicos, isto é, nenhum dos seus atributos tem domínio multivalorado.

Fornecedor FN	Cidade	População	Peça PN	Nome da Peça	Quantidade
F01	Fortaleza	30	P01, P02, P03, P04, P05, P06	Parafuso, Porca, Lixa, Martelo, Prego, Bucha	200, 100, 150, 50, 250, 200
F02	João Pessoa	20	P01, P06	Parafuso, Bucha	300, 300
F03	Rio de Janeiro	40	P06	Bucha	50
F04	Recife	25	P06, P05, P03	Bucha, Prego, Lixa	350, 350, 150

Atributos multivalorados
- Não está na 1FN!

15

Primeira Forma Normal

Fornecedor FN	Cidade	População	Peça PN	Nome da Peça	Quantidade
F01	Fortaleza	30	P01	Parafuso	200
F01	Fortaleza	30	P02	Porca	100
F01	Fortaleza	30	P03	Lixa	150
F01	Fortaleza	30	P04	Martelo	50
F01	Fortaleza	30	P05	Prego	250
F01	Fortaleza	30	P06	Bucha	200
F02	João Pessoa	20	P01	Parafuso	300
F02	João Pessoa	20	P06	Bucha	300
F03	Rio de Janeiro	40	P06	Bucha	50
F04	Recife	25	P06	Bucha	350
F04	Recife	25	P05	Prego	350
F04	Recife	25	P03	Lixa	150

1FN → Atributos
monovalorados

16

Segunda Forma Normal

Uma relação "R" está na SEGUNDA FORMA NORMAL - 2ª FN se e somente se estiver na 1ª FN e cada atributo **NÃO CHAVE** for **totalmente** dependente da chave primária.

Chave

Fornecedor FN	Peça PN	Quantidade
F01	P01	200
F01	P02	100
F01	P03	150
F01	P04	50
F01	P05	250
F01	P06	200
F02	P01	300
F02	P06	300
F03	P06	50
F04	P06	350
F04	P05	350
F04	P03	150

Fornecedor FN	Cidade	População
F01	Fortaleza	30
F02	João Pessoa	20
F03	Rio de Janeiro	40
F04	Recife	25

Peça PN	Nome da Peça
P01	Parafuso
P02	Porca
P03	Lixa
P04	Martelo
P05	Prego
P06	Bucha
P01	Parafuso
P06	Bucha
P06	Bucha
P06	Bucha
P05	Prego
P03	Lixa

Relação anterior está na 2FN?

17

Terceira Forma Normal

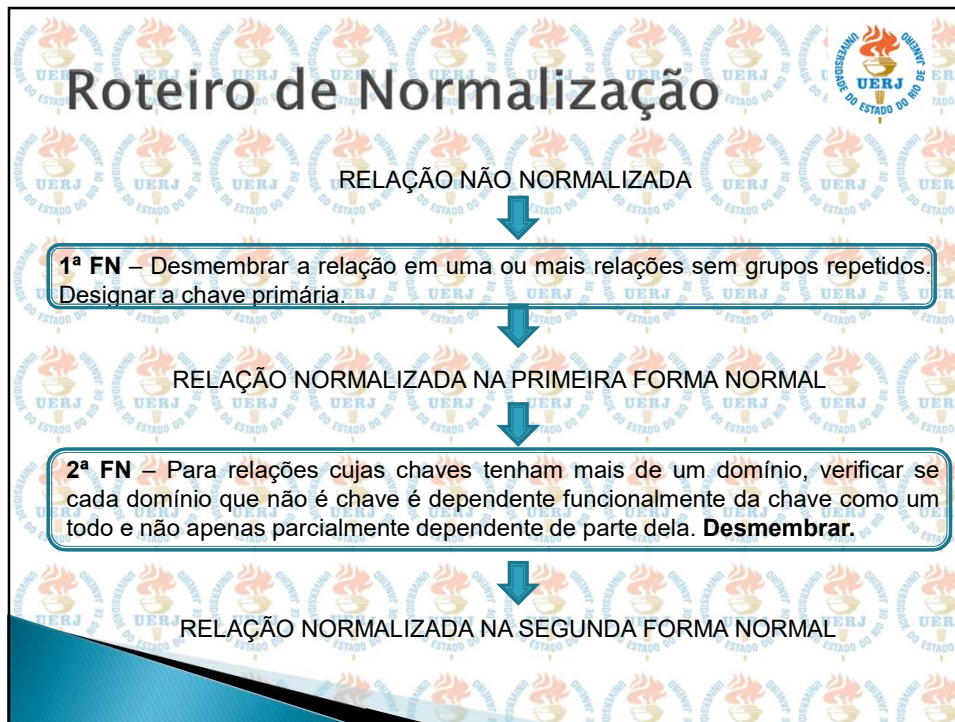
Uma relação "R" está na TERCEIRA FORMA NORMAL - 3ª FN se e somente se estiver na 2ª FN e cada atributo **NÃO CHAVE** **NÃO DEPENDE** transitivamente da chave primária.

Fornecedor FN	Cidade	População
F01	Fortaleza	30
F02	João Pessoa	20
F03	Rio de Janeiro	40
F04	Recife	25

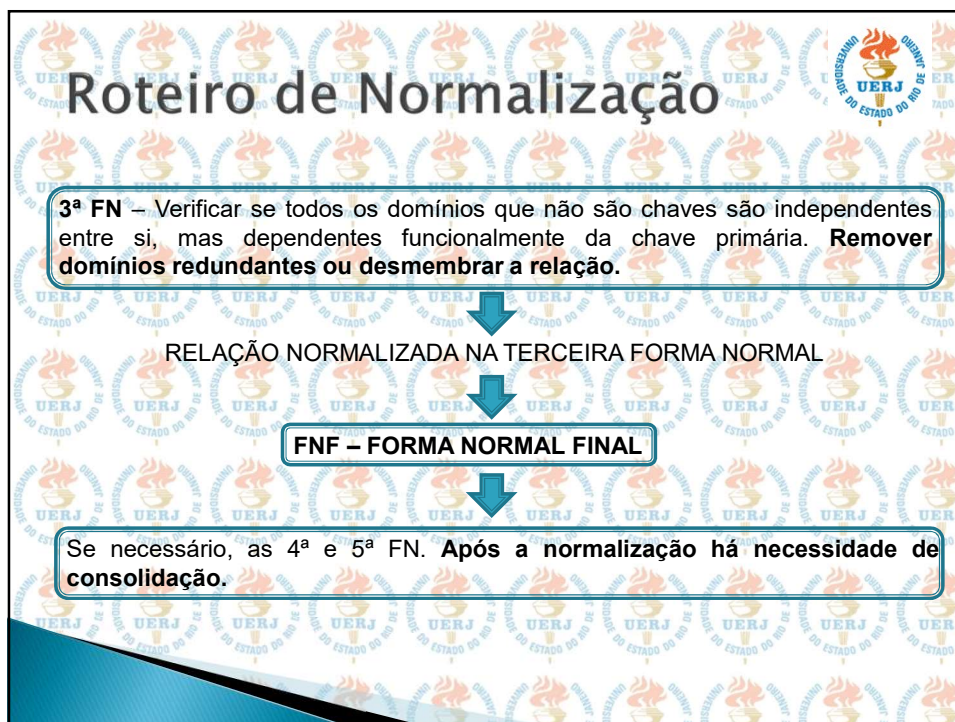
Fornecedor FN	Cidade
F01	Fortaleza
F02	João Pessoa
F03	Rio de Janeiro
F04	Recife

Cidade	População
Fortaleza	30
João Pessoa	20
Rio de Janeiro	40
Recife	25

18



19



20

Consolidação de Entidades

- ▶ Alinham-se, por coluna, todas as entidades com chaves candidatas ou com chaves primárias idênticas;
- ▶ Escolhe-se a chave candidata mais comum ou mais adequada;
- ▶ Comparam-se os atributos das entidades a serem consolidadas com os atributos da entidade relacional escolhida;
- ▶ Desprezam-se os atributos duplicados e inserem-se os atributos das demais entidades na entidade relacional escolhida;
- ▶ Verifica-se a necessidade de incluir outro atributo identificador de estado (um indicador, por exemplo);
- ▶ Se necessário, esse atributo deverá ser inserido na entidade relacional escolhida;
- ▶ Desprezam-se as demais entidades;
- ▶ O resultado será uma **única** entidade relacional.

21

Consolidação de Entidades – Exemplo

- ▶ Sejam as seguintes entidades após a normalização:
 - **Funcionário-ativo** (matrícula_funcionário, nome, endereço, código_departamento, salário, data_admissão)
 - **Funcionário-aposentado** (matrícula_aposentado, nome, endereço, salário, data_admissão, número_inss, data_aposentadoria)
 - **Departamento** (código_departamento, nome, matrícula_chefe)
- ▶ Aplicando a consolidação temos:
 - Alinham-se, por coluna, todas as entidades com chaves candidatas ou com chaves primárias idênticas
 - **Funcionário-ativo** (matrícula_funcionário, nome, endereço, código_departamento, salário, data_admissão)
 - **Funcionário-aposentado** (matrícula_aposentado, nome, endereço, salário, data_admissão, número_inss, data_aposentadoria)

22

Consolidação de Entidades – Exemplo

- ▶ Escolhe-se a chave candidata mais comum ou mais adequada;
 - **Funcionário** (matrícula, ...)
- ▶ Comparam-se os atributos das entidades a serem consolidadas com os atributos da entidade relacional escolhida;
 - **Funcionário-ativo** (matrícula, funcionário, nome, endereço, código_departamento, salário, data_admissão)
 - **Funcionário-aposentado** (matrícula, aposentado, nome, endereço, salário, data_admissão, número_inss, data_aposentadoria)
- ▶ Desprezam-se os atributos duplicados e inserem-se os atributos das demais entidades na entidade relacional escolhida;
 - **Funcionário** (matrícula, nome, endereço, código_departamento, salário, data_admissão, número_inss, data_aposentadoria)

23

Consolidação de Entidades – Exemplo

- ▶ Verifica-se a necessidade de incluir outro atributo identificador de estado (um indicador, por exemplo);
 - *Sem necessidade*
- ▶ Se necessário, esse atributo deverá ser inserido na entidade relacional escolhida;
 - *Sem necessidade*
- ▶ Desprezam-se as demais entidades;
 - **Funcionário-ativo** (matrícula, funcionário, nome, endereço, código_departamento, salário, data_admissão)
 - **Funcionário-aposentado** (matrícula, aposentado, nome, endereço, salário, data_admissão, número_inss, data_aposentadoria)
- ▶ O resultado será uma **única** entidade relacional.
 - **Funcionário** (matrícula, nome, endereço, código_departamento, salário, data_admissão, número_inss, data_aposentadoria)

24

Conclusão

Armazenar todas as informações em uma única relação/entidade (ex.: banco) é possível. Porém, isso resultaria em repetição de informações e a necessidade por valores nulos.



25

Exercícios

▶ Coloque as seguintes estruturas na 1FN, 2FN e 3FN:

- a) FuncionarioProjeto (MatrFunc, NomeFunc, CodDepto, NomeDepto, {CodProjeto, NomeProj, LocalProjeto, NumHorasFuncProj})
 - MatrFunc → NomeFunc
 - MatrFunc → CodDepto, NomeDepto
 - CodProjeto → NomeProj, LocalProjeto
 - MatrFunc, CodProjeto → NumHorasFuncProj
 - CodDepto → NomeDepto
- b) Encomendas (CodEncomenda, DtEncomenda, CodCliente, NomeCliente, EndCliente, {CodProduto, NomeProduto, ValorUnitario, QtdProd}). Cada encomenda pode ter mais do que um produto.
 - CodEncomenda → DtEncomenda
 - CodEncomenda → CodCliente, NomeCliente, EndCliente
 - CodCliente → NomeCliente, EndCliente
 - CodEncomenda, CodProduto → QtdProd
 - CodProduto → NomeProduto, ValorUnitario

OBS.: { } indicam atributos multivalorados

26

Exercício (a)

- ▶ FuncionarioProjeto (MatrFunc, NomeFunc, CodDepto, NomeDepto, {CodProjeto, NomeProj, LocalProjeto, NumHorasFuncProj})
- ▶ 1 FN:
 - Funcionario(MatFunc, NomeFunc, CodDepto, NomeDepto)
 - FuncionarioProjeto(MatFunc, CodProjeto, NomeProj, LocalProjeto, NumHorasFuncProj)
- ▶ 2 FN:
 - Funcionario(MatFunc, NomeFunc, CodDepto, NomeDepto)
 - FuncionarioProjeto(MatFunc, CodProjeto, NumHorasFuncProj)
 - Projeto(CodProjeto, NomeProj, LocalProjeto)
- ▶ 3 FN:
 - Funcionario(MatFunc, NomeFunc, CodDepto)
 - Departamento(CodDepto, NomeDepto)
 - FuncionarioProjeto(MatFunc, CodProjeto, NumHorasFuncProj)
 - Projeto(CodProjeto, NomeProj, LocalProjeto)

27

Exercício (b)

- ▶ Encomendas (CodEncomenda, DtEncomenda, CodCliente, NomeCliente, EndCliente, {CodProduto, NomeProduto, ValorUnitario, QtdProd}). Cada encomenda pode ter mais do que um produto.
- ▶ 1 FN:
 - ▶ Encomendas(CodEncomenda, DtEncomenda, CodCliente, NomeCliente, EndCliente)
 - ▶ Produtos(CodEncomenda, CodProduto, NomeProduto, ValorUnitario, QtdProd)
- ▶ 2 FN:
 - ▶ Encomendas(CodEncomenda, DtEncomenda, CodCliente, NomeCliente, EndCliente)
 - ▶ Produtos(CodEncomenda, CodProduto, QtdProd)
 - ▶ DetalheProduto(CodProduto, NomeProduto, ValorUnitario)
- ▶ 3 FN:
 - ▶ Encomendas(CodEncomenda, DtEncomenda, CodCliente)
 - ▶ Clientes(CodCliente, NomeCliente, EndCliente)
 - ▶ Produtos(CodEncomenda, CodProduto, QtdProd)
 - ▶ DetalheProduto(CodProduto, NomeProduto, ValorUnitario)

28

Exercícios

c. Paciente (CodPaciente, NomePaciente, NumQuarto, DescQuarto, NumCômodosQuarto, {CodMedico, NomeMedico, TelMedico}).

- CodPaciente -> NomePaciente
- CodPaciente -> NumQuarto, DescQuarto, NumCômodosQuarto
- CodMedico -> NomeMedico, TelMedico
- NumQuarto -> DescQuarto, NumCômodosQuarto

d. OrdemCompra (CodOrdem, DtEmissão, CodFornecedor, NomeFornecedor, EndFornecedor, {CodMaterial, DescMaterial, QtdComprada, ValorUnit, ValorTotalItem}, ValorTotalOrdem).

- CodOrdem -> DtEmissao
- CodOrdem -> CodFornecedor, NomeFornecedor, EndFornecedor
- CodOrdem -> ValorTotalOrdem
- CodFornecedor -> NomeFornecedor, EndFornecedor
- CodMaterial -> DescMaterial, ValorUnit

29

Exercício (c)

▶ Paciente (CodPaciente, NomePaciente, NumQuarto, DescQuarto, NumCômodosQuarto, {CodMedico, NomeMedico, TelMedico})

▶ 1 FN:

- Paciente(CodPaciente, NomePaciente, NumQuarto, DescQuarto, NumCômodosQuarto)
- PacienteMedico(CodPaciente, CodMedico, NomeMedico, TelMedico)

▶ 2 FN:

- Paciente(CodPaciente, NomePaciente, NumQuarto, DescQuarto, NumCômodosQuarto)
- PacienteMedico(CodPaciente, CodMedico)
- Medico(CodMedico, NomeMedico, TelMedico)

▶ 3 FN:

- Paciente(CodPaciente, NomePaciente, NumQuarto)
- Quarto(NumQuarto, DescQuarto, NumCômodosQuarto)
- PacienteMedico(CodPaciente, CodMedico)
- Medico(CodMedico, NomeMedico, TelMedico)

30

Exercício (d)

OrdemCompra (CodOrdem, DtEmissão, CodFornecedor, NomeFornecedor, EndFornecedor, {CodMaterial, DescMaterial, QtdComprada, ValorUnit, ValorTotalItem}, ValorTotalOrdem)

1FN:

OrdemCompra (CodOrdem, DtEmissão, CodFornecedor, NomeFornecedor, EndFornecedor, ValorTotalOrdem)
• MaterialCompra(CodOrdem, CodMaterial, DescMaterial, QtdComprada, ValorUnit, ValorTotalItem)

2FN:

OrdemCompra (CodOrdem, DtEmissão, CodFornecedor, NomeFornecedor, EndFornecedor, ValorTotalOrdem)
• MaterialCompra(CodOrdem, CodMaterial, QtdComprada, ValorTotalItem)
• Material(CodMaterial, DescMaterial, ValorUnit)

3FN:

OrdemCompra (CodOrdem, DtEmissão, CodFornecedor, ValorTotalOrdem)
• Fornecedor(CodFornecedor, NomeFornecedor, EndFornecedor)
• MaterialCompra(CodOrdem, CodMaterial, QtdComprada, ValorTotalItem)
• Material(CodMaterial, DescMaterial, ValorUnit)