

PROJETO DE BANCO DE DADOS: NOTA DE NORMALIZAÇÃO E EXERCÍCIOS (Até 3FN)

As tabelas que contêm dados redundantes podem sofrer de anomalias de atualização, o que pode introduzir inconsistências em um banco de dados.

As regras associadas às formas normais mais comumente usadas, ou seja, primeira (1NF), segunda (2NF) e terceira (3NF).

A identificação de vários tipos de anomalias de atualização, como anomalias de inserção, exclusão e modificação, pode ser encontrada quando tabelas que quebram as regras de 1NF, 2NF e 3NF e são passíveis de conter dados redundantes e sofrer anomalias de atualização.

A normalização é uma técnica para produzir um conjunto de tabelas com propriedades desejáveis que suportam os requisitos de um usuário ou empresa.

O principal objetivo do design de banco de dados relacional é agrupar colunas em tabelas para minimizar a redundância de dados e reduzir o espaço de armazenamento de arquivos exigido pelas tabelas base.

Dê uma olhada no exemplo a seguir:

StdSSN	StdCity	StdClass	OfferNo	OffTerm	OffYear	EnrGrade	CourseNenhum	CrsDesc	
S1	SEATTLE	JUN	O1	OUTONO	DE 2006	3,5	C1		BD
S1	SEATTLE	JUN	O2	OUTONO	DE 2006	3,3	C2		VB
S2	BOTHELL	JUN	O3	PRIMAVERA	2007	3,1	C3		OO
S2	BOTHELL	JUN	O2	OUTONO	DE 2006	3,4	C2		VB

A **anomalia de inserção**: Ocorre quando dados extras além dos dados desejados devem ser adicionados ao banco de dados. Por exemplo, para inserir um curso (CourseNo), é necessário conhecer um aluno (StdSSN) e uma oferta (OfferNo), pois a combinação de StdSSN e OfferNo é a chave primária. Lembre-se de que uma linha não pode existir com valores NULL para parte de sua chave primária.

A **anomalia de atualização**: Ocorre quando é necessário alterar várias linhas para modificar **APENAS** um único fato. Por exemplo, se alterarmos o StdClass do aluno S1 (JUN), duas linhas, linha 1 e 2 devem ser alteradas. Se S1 foi matriculado em 10 aulas, 10 linhas devem ser alteradas.

A **anomalia de exclusão**: ocorre sempre que excluir uma linha inadvertidamente faz com que outros dados sejam excluídos. Por exemplo, se excluirmos a matrícula (EnrGrade) de S2 em O3 (terceira linha), perdemos as informações sobre oferta de O3 e curso C3, pois esses valores são exclusivos da tabela (célula). Além disso, O3 é uma chave primária.

RECAPITULAR

Os problemas associados à redundância de dados são ilustrados comparando a equipe e a filial tabelas com a tabela StaffBranch. Tabelas que possuem dados redundantes podem ter problemas chamados de anomalias de atualização, que são classificadas como anomalias de inserção, exclusão ou modificação. Consulte a Figura a seguir para obter um exemplo de uma tabela com dados redundantes chamada StaffBranch. Existem dois tipos principais de anomalias de inserção, que ilustramos usando esta tabela.

Anomalias de inserção

1. Para inserir os dados de um novo funcionário (staffNo, nome, cargo e salário) localizado em uma determinada ramificação na tabela StaffBranch, também devemos inserir os detalhes corretos para essa ramificação (branchNo, branchAddress e telNo). Por exemplo, para inserir os detalhes de um novo funcionário na filial B002, devemos inserir os detalhes corretos da filial B002 para que os detalhes da filial sejam consistentes com os valores da filial B002 em outros registros do Tabela StaffBranch. Os dados mostrados na tabela StaffBranch também são mostrados na e tabelas de filiais. Essas tabelas possuem dados redundantes e não sofrem com essa potencial inconsistência, pois para cada membro da equipe apenas inserimos o número da agência apropriada na tabela Staff. Além disso, os detalhes da filial B002 são registrados apenas uma vez no banco de dados como um único registro na tabela Filial.
2. Para inserir detalhes de uma nova filial que atualmente não possui membros da equipe no StaffBranch tabela, é necessário inserir NULLs nas colunas relacionadas à equipe, como staffNo. No entanto, como staffNo é a chave primária para a tabela StaffBranch, tentar inserir nulos para staffNo viola a integridade da entidade e não é permitido. O design das tabelas mostradas em Staff e a Filial evita esse problema porque os novos detalhes da filial são inseridos na tabela Filial separadamente dos detalhes da equipe. Os detalhes dos funcionários localizados em uma nova filial podem ser inseridos na tabela Funcionários posteriormente.

Anomalias de exclusão

Se excluirmos um registro da tabela StaffBranch que representa o último membro da equipe localizado em uma filial, os detalhes sobre essa filial também serão perdidos do banco de dados. Por exemplo, se excluirmos o registro da equipe Art Peters (S0415) da tabela StaffBranch, os detalhes relacionados à filial B003 serão perdidos do banco de dados. O desenho das tabelas que separam a tabela Staff e Branch evita este problema, pois os registros da filial são armazenados separadamente dos registros da equipe e apenas a coluna branchNo relaciona as duas tabelas. Se excluirmos o registro da equipe Art Peters (S0415) da tabela Equipe, os detalhes da filial B003 na tabela Filial permanecerão inalterados.

Anomalias de modificação

Se quisermos alterar o valor de uma das colunas de uma determinada ramificação no StaffBranch tabela, por exemplo, o número de telefone da agência B001, devemos atualizar os registros de todos os funcionários localizados nessa agência (linha 1 e 2). Se esta modificação não for realizada em todos os registros apropriados da tabela StaffBranch, o banco de dados ficará inconsistente. Neste exemplo, a filial B001 teria números de telefone diferentes em registros de funcionários diferentes. Os exemplos acima ilustram que as tabelas Staff e Branch têm propriedades mais desejáveis do que a tabela StaffBranch.

Tabela de Filial da Equipe

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000

Tabelas de funcionários e filiais

Staff

staffNo	name	position	salary	branchNo
S1500	Tom Daniels	Manager	46000	B001
S0003	Sally Adams	Assistant	30000	B001
S0010	Mary Martinez	Manager	50000	B002
S3250	Robert Chin	Supervisor	32000	B002
S2250	Sally Stern	Manager	48000	B004
S0415	Art Peters	Manager	41000	B003

Branch

branchNo	branchAddress	telNo
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
B002	City Center Plaza, Seattle, WA 98122	206-555-6756
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131

A tabela StaffBranch tem dados redundantes; **os detalhes de uma filial** são repetidos para cada membro da equipe, por exemplo, linha 1 e 2, linha 3 e 4 em branchNo, branchAddress e telNo.

Em contraste, as informações da agência aparecem apenas uma vez para cada agência na tabela Branch e apenas o número da agência (branchNo) é repetido na tabela Staff, para representar onde cada membro da equipe está localizado.

Vamos dar uma olhada em outro exemplo.

Tabela AIRCRAFT_1

	AC_NUM	AC_MODEL	AC_RENT_CHG	AC_SEATS	AC_TOTAL_TIME
►	1178R	Cessna C-172 Shyhawk	\$58.50	4	4512.6
	205IY	Cessna C-172 Shyhawk	\$58.50	4	5325.5
	2087V	Cessna C-152 Commuter	\$51.75	2	4889.8
	226BR	Cessna C-172 Shyhawk	\$58.50	4	4299.6
	2867W	Piper PA28-181 Archer II	\$64.00	4	3267.4
	3213R	Piper PA28-181 Archer II	\$64.00	4	2517.9
	4112E	Piper PA28-181 Archer II	\$64.00	4	5211.3
	45ZU	Cessna C-152 Commuter	\$51.75	2	7003.1
	5725Y	Cessna C-172 Shyhawk	\$58.50	4	3968.2

Se usarmos a tabela AIRCRAFT_1 conforme mostrado na Figura acima, uma mudança nas taxas de aluguel por hora (AC_RENT_CHG) para o Cessna 172 Skyhawk deve ser feita quatro vezes; se esquecermos de alterar apenas uma dessas taxas, teremos um problema de integridade de dados.

Como seria melhor ter dados críticos em apenas um lugar. Então, se uma mudança deve ser feita, ela precisa ser feita apenas uma vez.

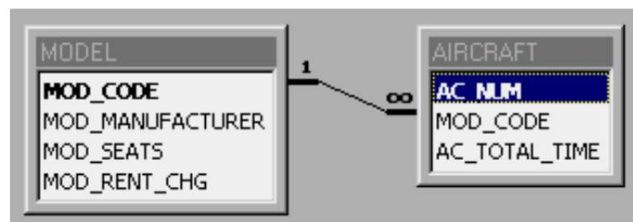
Em contraste, as estruturas de tabela são boas quando excluem a possibilidade de produzir redundâncias de dados não controladas.

Podemos produzir uma circunstância tão feliz dividindo a tabela AIRCRAFT_1 conforme mostrado nas duas Figuras a seguir, conectando as duas tabelas resultantes por meio da chave estrangeira MOD_CODE da tabela AIRCRAFT_1.

	AC_NUM	MOD_CODE	AC_TOTAL_TIME
►	1178R	C-172	4512.6
	205IY	C-172	5325.5
	2087V	C-152	4889.8
	226BR	C-172	4299.6
	2867W	PA28-181	3267.4
	3213R	PA28-181	2517.9
	4112E	PA28-181	5211.3
	45ZU	C-152	7003.1
	5725Y	C-172	3968.2

	MOD_CODE	MOD_MANUFACTURER	MOD_SEATS	MOD_RENT_CHG
►	C-152	Cessna	2	\$51.75
	C-172	Cessna	4	\$58.50
	PA28-181	Piper	4	\$64.00

Observe que uma alteração da taxa de aluguel precisa ser feita em apenas um local, uma descrição é fornecida em apenas um local e assim por diante. Não há mais atualização de dados e anomalias de exclusão e não há mais problemas de integridade de dados. O esquema relacional na Figura a seguir mostra como as duas tabelas estão relacionadas.



A primeira forma normal (1FN)

Uma tabela na qual a interseção de cada coluna e registro contém apenas um valor. Proíbe aninhar ou repetir grupos na tabela. A interseção deve ser atômica.

Por exemplo, a coluna telNos contém vários valores.

branchNo	branchAddress	telNos
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

↑ Primary key

More than one value, so not in 1NF

Branch (Not 1NF)

branchNo	branchAddress	telNos
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

Take copy of branchNo column to new table to become foreign key

Remove telNos column and create new column called telNo in the new table

Branch (1NF)

branchNo	branchAddress
B001	8 Jefferson Way, Portland, OR 97201
B002	City Center Plaza, Seattle, WA 98122
B003	14 – 8th Avenue, New York, NY 10012
B004	16 – 14th Avenue, Seattle, WA 98128

↑ Primary key

BranchTelephone (1NF)

branchNo	telNo
B001	503-555-3618
B001	503-555-2727
B001	503-555-6534
B002	206-555-6756
B002	206-555-8836
B003	212-371-3000
B004	206-555-3131
B004	206-555-4112

↑ Becomes foreign key

↑ Becomes primary key

A segunda forma normal (2FN)

2NF APENAS se aplica a tabelas com **chaves primárias compostas** (mais de uma chave primária). Uma tabela que está na 1NF e na qual os valores de cada **coluna de chave não primária** podem ser calculados a partir dos valores em TODAS as colunas que compõem a chave primária.

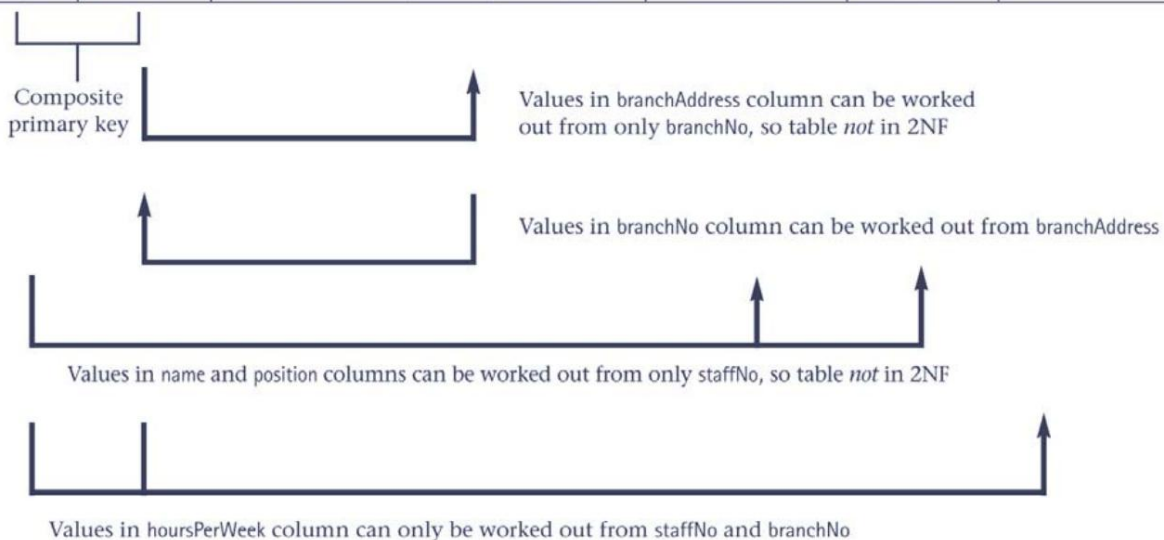
Uma tabela está em 2NF se **cada coluna não chave (não primária e/ou chaves candidatas) depende de TODAS as chaves candidatas, NÃO de um subconjunto de QUALQUER chave candidata.**

A violação 2NF ocorre quando a Dependência Funcional (FD) em que parte da chave (em vez das chaves inteiras) determina uma não chave. Um FD contendo uma única coluna Left Hand Side (LHS) não pode violar 2NF.

Por exemplo, a tabela TempStaffAllocation na figura a seguir está em 2NF porque branchAddress pode depender de branchNo apenas não de staffNo AND branchNo

(staffNo & branchNo são chaves candidatas e ao mesmo tempo podem ser chaves primárias e ao mesmo tempo são chaves compostas porque mais de uma chave primária). Outra é que os valores nas colunas de nome e posição podem depender (podem ficar por conta própria) do staffNo SOMENTE, não de staffNo e branchNo. O que queremos é a coluna hoursPerWeek, que depende de staffNo e branchNo. Em outras palavras, devemos evitar as dependências parciais das chaves candidatas.

staffNo	branchNo	branchAddress	name	position	hoursPerWeek
S4555	B002	City Center Plaza, Seattle, WA 98122	Ellen Layman	Assistant	16
S4555	B004	16 – 14th Avenue, Seattle, WA 98128	Ellen Layman	Assistant	9
S4612	B002	City Center Plaza, Seattle, WA 98122	Dave Sinclair	Assistant	14
S4612	B004	16 – 14th Avenue, Seattle, WA 98128	Dave Sinclair	Assistant	10



A dependência funcional

Os **relacionamentos particulares que mostramos entre as colunas de uma tabela** são mais formalmente chamados de **dependências funcionais (FDs)**. FD descreve o relacionamento entre colunas em uma tabela.

Outro exemplo, considere uma tabela com as colunas A e B, onde B é funcionalmente dependente de A (denominado A \rightarrow B). Se soubermos o valor de A, encontraremos apenas um valor de B em todos os registros que possuem esse valor de A, em qualquer momento.

Definição de FD

$X \rightarrow Y$.

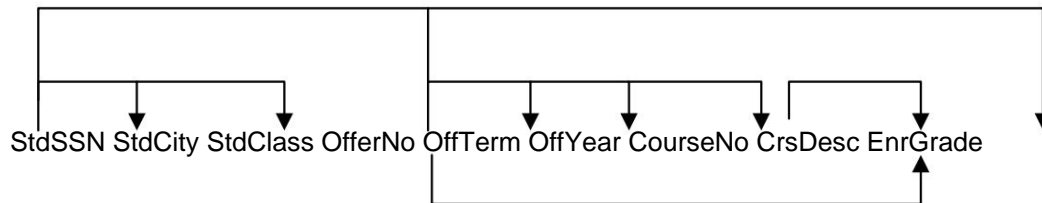
X (funcionalmente) determina que Y ou Y é funcionalmente dependente de X.

X: lado esquerdo (LHS) ou determinante.

Para cada valor X, há no máximo um valor Y.

Semelhante às chaves candidatas.

Por exemplo (observe o fluxo de setas!):



Os FDs são (outra notação usada para escrever FDs):

StdSSN \rightarrow StdCity, StdClass

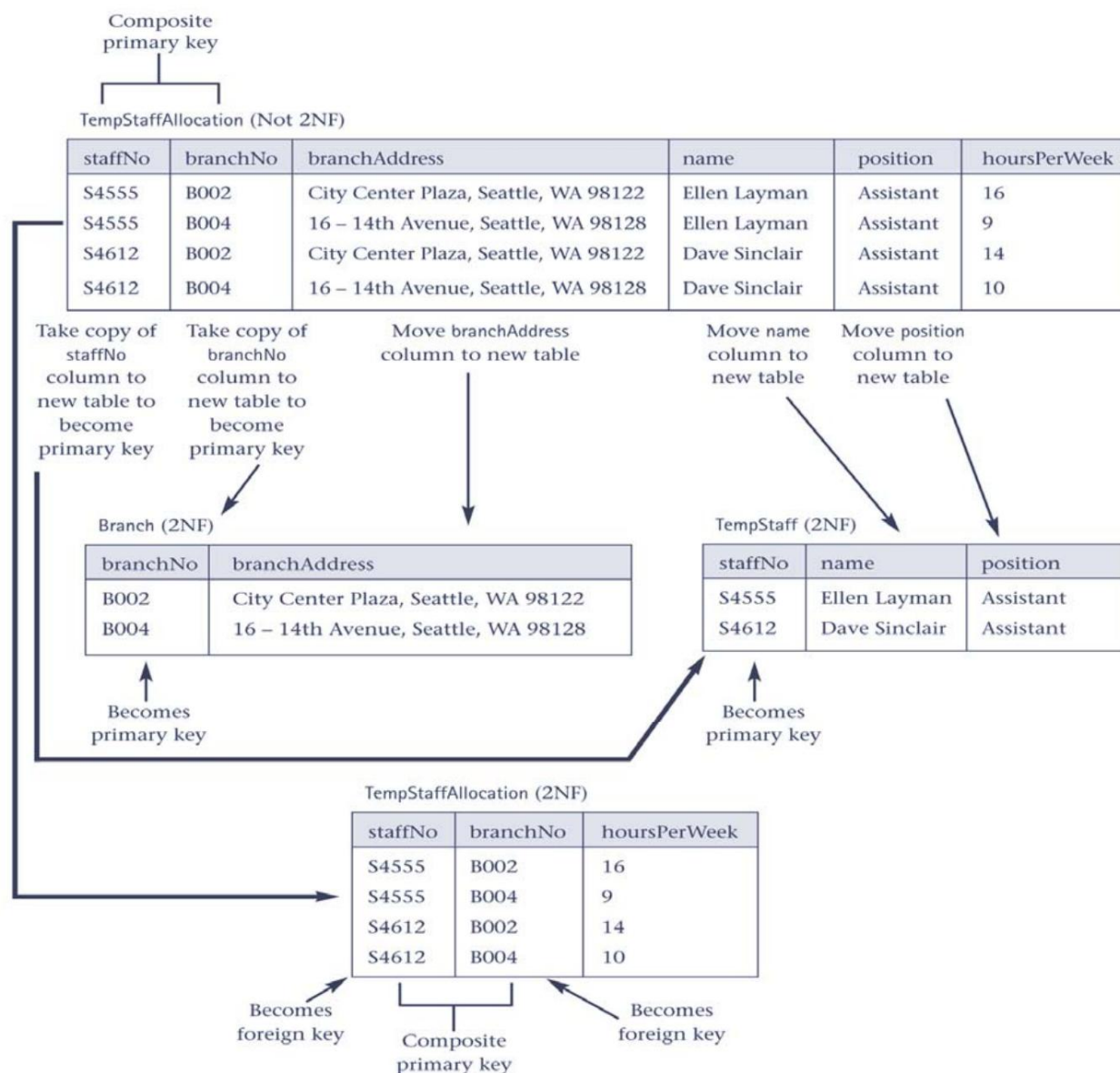
OfferNo \rightarrow OffTerm, OffYear, CourseNo, CrsDesc

CursoNão \rightarrow CrsDesc

StdSSN, OfferNo \rightarrow EnrGrade

A definição formal de 2NF é uma tabela que está em 1NF e cada coluna de chave não primária é **totalmente funcional dependente** da chave primária.

A dependência funcional completa indica que, se A e B são colunas de uma tabela, B é totalmente dependente de A se B é funcionalmente dependente de A, mas não de qualquer subconjunto adequado de A. Considere os exemplos a seguir.



Identificando Dependências Funcionais

Os designers de banco de dados devem ser capazes de identificar FD ao coletar os requisitos do banco de dados.

Em narrativas de problemas, alguns DF podem ser identificados por declarações sobre singularidade. Por exemplo, um usuário pode declarar que cada oferta de curso tem um número de oferta exclusivo junto com o ano e o prazo da oferta. A partir desta declaração, o designer deve afirmar que OfferNo \rightarrow OffYear e OffTerm.

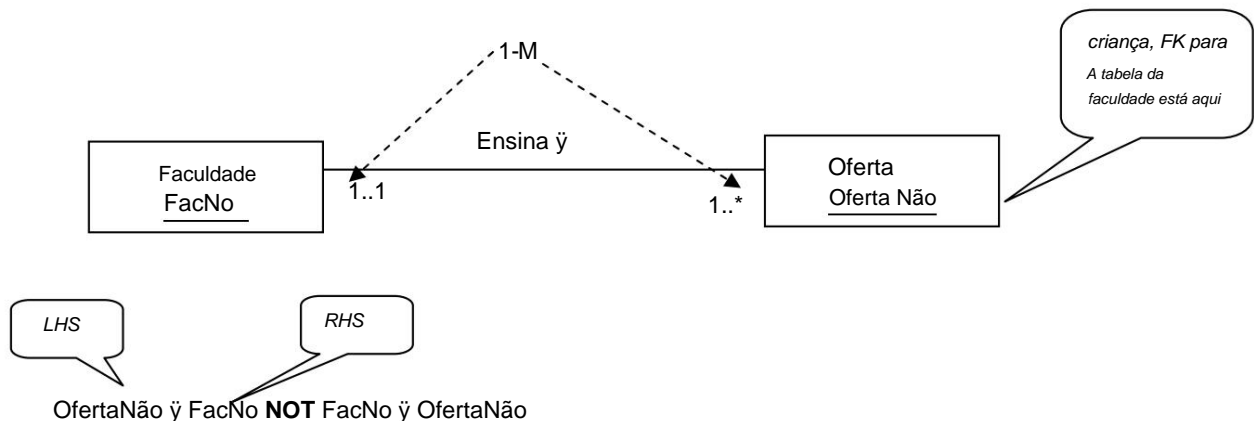
Você também pode identificar dependências funcionais em um design de tabela resultante da conversão de um ERD. FD seria afirmado para cada coluna exclusiva (PK ou outra chave candidata) com a coluna exclusiva como LHS e outras colunas na tabela no RHS.

Embora o DF derivado de declarações sobre relacionamentos 1-M possa ser identificado, o FD derivado de declarações sobre relacionamento 1-M pode ser confuso para identificar.

Quando você vê uma declaração sobre um relacionamento 1-M, o FD é derivado da direção de filho para pai, não da direção de pai para filho.

Por exemplo, a afirmação “Um corpo docente ensina muitas ofertas, mas uma oferta é ensinada por um corpo docente” define um FD de uma coluna única de oferta para uma coluna única de corpo docente, como $\text{OfertaNo} \twoheadrightarrow \text{FacNo}$.

Designers novatos às vezes afirmam incorretamente que FacNo determina uma coleção de OfferNo valores. Esta afirmação não está correta porque um **FD deve permitir no máximo um valor associado**, não uma coleção de valores.



FD em que o LHS não é uma PK ou chave candidata também pode ser difícil de identificar.

Esses FDs são especialmente importantes para identificar após a conversão de um ERD em um design de tabela.

Você deve procurar cuidadosamente por FDs em que o LHS não seja uma chave candidata ou chave primária.

Você também deve considerar FDs em tabelas com um PK combinado ou chave candidata em que o LHS faz parte de uma chave, mas não da chave inteira. As NFs desses tipos de FDs podem levar a anomalias de modificação.

Outra consideração importante na afirmação do DF é o minimalismo do LHS. É importante distinguir quando uma coluna sozinha é o determinante versus uma combinação de colunas.

Um FD em que o LHS contém mais de uma coluna geralmente representa uma relação MN. Por exemplo, a declaração “A quantidade do pedido é coletada para cada produto adquirido em um pedido” se traduz em FD $\text{OrdNo}, \text{ProdNo} \twoheadrightarrow \text{OrdQty}$.

A quantidade do pedido depende da combinação do número do pedido e do número do produto, não apenas uma dessas colunas.

Parte da confusão sobre o minimalismo do LHS se deve ao significado das colunas no lado esquerdo versus o lado direito de uma dependência. O determinante mínimo significa que o determinante (coluna(s) que aparecem no LHS de FD) não deve conter colunas extras e esse requisito de minimalismo é semelhante ao requisito de minimalismo para chaves candidatas.

Por exemplo, para registrar que o Número do Seguro Social do aluno determina a cidade e a classe, você pode escrever:

$\text{StdSSN} \twoheadrightarrow \text{StdCity}, \text{StdClass}$ (mais compacto) ou

$\text{StdSSN} \twoheadrightarrow \text{StdCity}$ e $\text{StdSSN} \twoheadrightarrow \text{StdClass}$ (menos compacto)

Se você assumir que o endereço de e-mail também é exclusivo para cada aluno, poderá escrever:

$\text{E-mail} \twoheadrightarrow \text{StdCity}, \text{StdClass}$

Você não deve escrever:

StdSSN, Email \rightarrow StdCity, StdClass

Isso ocorre porque esses FDs implicam que a combinação de StdSSN e Email é o determinante.

Assim, você deve escrever FDs para que o LHS não contenha colunas desnecessárias.

A proibição contra colunas desnecessárias para determinantes é a mesma que a proibição contra colunas desnecessárias em chaves candidatas. Tanto os determinantes quanto as chaves candidatas devem ser mínimos.

A existência de um FD não pode ser comprovada examinando as linhas de uma tabela. No entanto, você pode falsificar um FD (ou seja, provar que um FD não existe) examinando o conteúdo de uma tabela.

Por exemplo, no banco de dados da universidade, podemos concluir que StdClass não determina StdCity porque existem duas linhas com o mesmo valor para StdClass, mas um valor diferente para StdCity.

Assim, às vezes é útil examinar linhas de amostra em uma tabela para eliminar possíveis dependências funcionais.

Existem várias ferramentas de design de banco de dados comercial que automatizam o processo de eliminação de dependências por meio do exame de linhas de amostra. Em última análise, o designer de banco de dados deve tomar a decisão final sobre os FDs que existem em uma tabela.

Terceira forma normal (3FN)

Uma tabela que está em 1NF e 2NF e na qual **todas as colunas de chave não primária podem ser trabalhadas apenas a partir da(s) coluna(s) de chave primária e nenhuma outra coluna.**

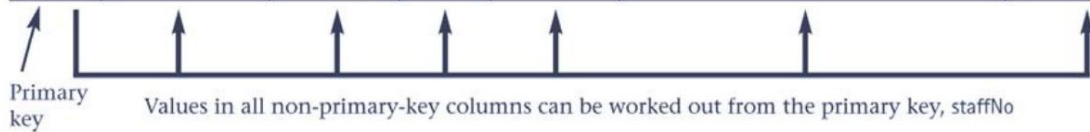
Nesse nível, a definição combinada de 2NF e 3NF é que uma tabela está em 3NF se cada coluna não chave depende de todas as chaves candidatas, chaves candidatas inteiras e nada além de chaves candidatas.

Para a 2FN devemos remover **a dependência parcial** e para a 3FN devemos remover **a dependência transitiva.**

Por exemplo, a tabela StaffBranch não está na 3NF.

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000



Values in branchAddress and telNo columns can be worked out from branchNo, so table *not* in 3NF



Values in branchNo and telNo columns can be worked out from branchAddress, so table *not* in 3NF



Values in branchNo and branchAddress columns can be worked out from telNo, so table *not* in 3NF



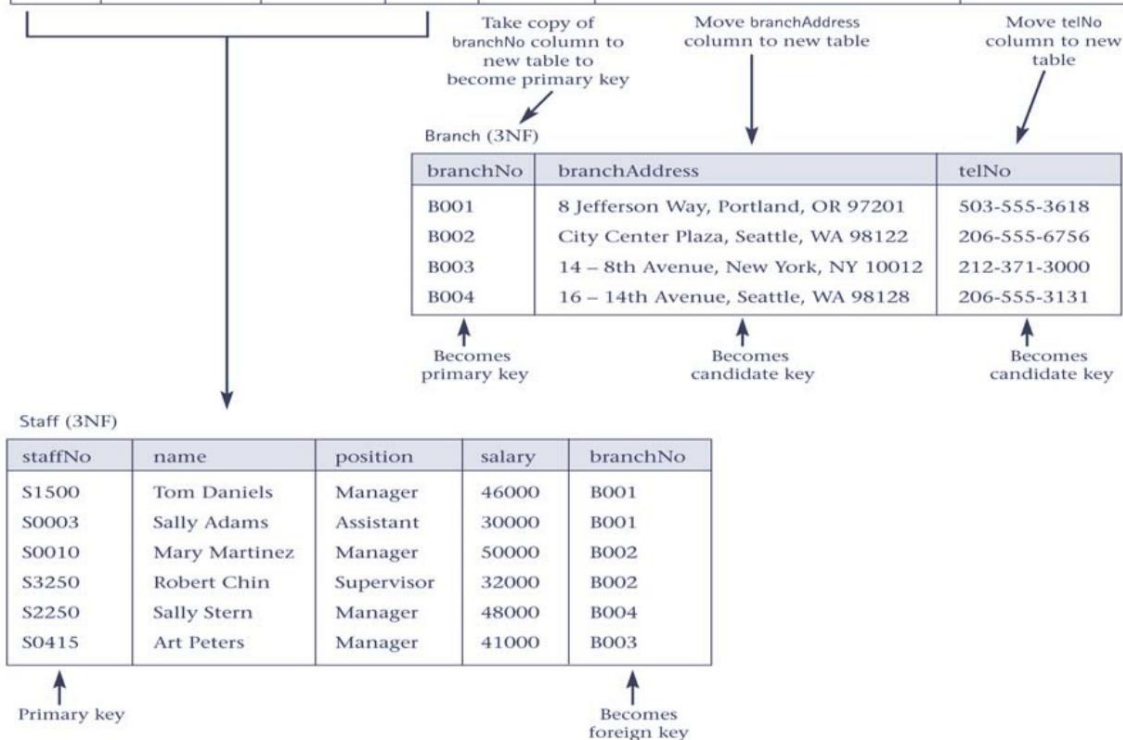
A definição formal de 3NF é uma tabela que está em 1NF e 2NF e na qual nenhuma coluna de chave não primária é dependente transitivamente da chave primária.

Por exemplo, considere uma tabela com A, B e C. Se B é funcionalmente dependente de A ($A \twoheadrightarrow B$) e C é funcionalmente dependente de B ($B \twoheadrightarrow C$), então C é **transitivamente dependente** de A via B (desde que A não é funcionalmente dependente de B ou C).

Se existir uma dependência transitiva na chave primária, a tabela não está na 3NF.

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000



Comentários Perguntas e Respostas

1. Descreva as características de uma tabela que viola a primeira forma normal (1NF) e depois descreva como essa tabela é convertida para 1NF.

A regra para a primeira forma normal (1NF) é uma tabela na qual a interseção de cada coluna e registro contém apenas um valor. Em outras palavras, uma tabela que contém mais de um valor atômico na interseção de uma ou mais colunas para um ou mais registros não está em 1NF. A tabela não 1NF pode ser convertida em 1NF reestruturando a tabela original removendo a coluna com os valores múltiplos junto com uma cópia da chave primária para criar uma nova tabela. A vantagem dessa abordagem é que as tabelas resultantes podem estar em formas normais depois da 1NF.

2. Qual é a forma normal mínima que uma relação deve satisfazer? Dê uma definição para isso forma normal.

Somente a primeira forma normal (1NF) é crítica na criação de tabelas apropriadas para bancos de dados relacionais. Todas as formas normais subsequentes são opcionais. No entanto, para evitar as anomalias de atualização, normalmente é recomendado que você prossiga para a terceira forma normal (3NF).

A primeira forma normal (1NF) é uma tabela na qual a interseção de cada coluna e registro contém apenas um valor.

3. Descreva uma abordagem para converter uma tabela de primeira forma normal (1NF) em segunda forma normal (2NF) tabela(s).

A segunda forma normal aplica-se apenas a tabelas com chaves primárias compostas, ou seja, tabelas com uma chave primária composta por duas ou mais colunas. Uma tabela 1NF com uma chave primária de coluna única está automaticamente em pelo menos 2NF.

Uma segunda forma normal (2NF) é uma tabela que já está na 1NF e na qual os valores em cada coluna de chave não primária podem ser calculados a partir dos valores em todas as colunas que compõem a chave primária.

Uma tabela em 1NF pode ser convertida em 2NF removendo as colunas que podem ser trabalhadas de apenas parte da chave primária. Essas colunas são colocadas em uma nova tabela junto com uma cópia da parte da chave primária a partir da qual elas podem ser trabalhadas.

4. Descreva as características de uma tabela na segunda forma normal (2NF).

A segunda forma normal (2NF) é uma tabela que já está na 1NF e na qual os valores de cada coluna não primária só podem ser calculados a partir dos valores de todas as colunas que compõem a chave primária.

5. Descreva o que se entende por dependência funcional completa e descreva como esse tipo de dependência se relaciona com a 2NF. Dê um exemplo para ilustrar sua resposta.

A definição formal de segunda forma normal (2NF) é uma tabela que está na primeira forma normal e cada coluna de chave não primária é totalmente funcionalmente dependente da chave primária. A dependência funcional completa indica que, se A e B são colunas de uma tabela, B é totalmente funcionalmente dependente de A, se B não depende de nenhum subconjunto de A. Se B depende de um subconjunto de A, isso é chamado de dependência parcial. Se existir uma dependência parcial na chave primária, a tabela não está na 2NF. A dependência parcial deve ser removida para que uma tabela atinja 2NF.

6. Descreva as características de uma tabela na terceira forma normal (3FN).

A terceira forma normal (3NF) é uma tabela que já está na 1NF e na 2NF, e na qual os valores em todas as colunas de chave não primária podem ser calculados apenas a partir da(s) coluna(s) de chave primária (ou chave candidata) e não outras colunas.

7. Descreva o que se entende por dependência transitiva e descreva como esse tipo de dependência se relaciona com a 3FN. Dê um exemplo para ilustrar sua resposta.

A definição formal para a terceira forma normal (3NF) é uma tabela que está na primeira e na segunda formas normais e na qual nenhuma coluna de chave não primária é dependente transitivamente da chave primária.

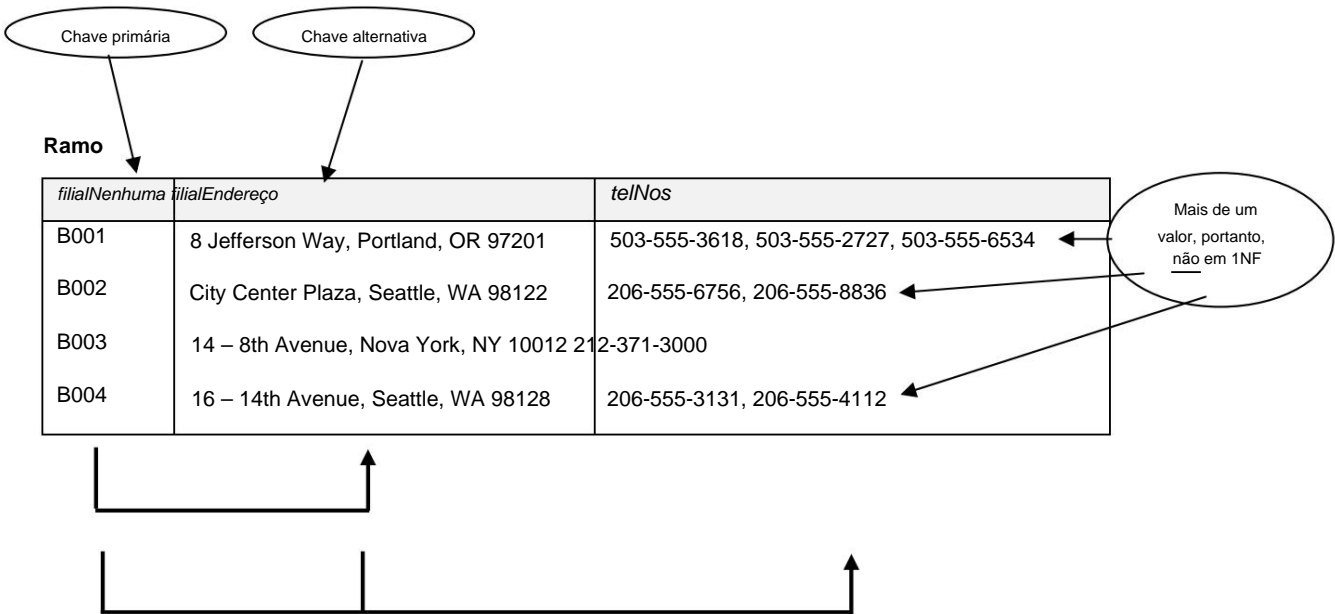
A dependência transitiva é um tipo de dependência funcional que ocorre quando um tipo específico de relacionamento é mantido entre as colunas de uma tabela. Por exemplo, considere uma tabela com as colunas A, B e C. Se B for funcionalmente dependente de A ($A \rightarrow B$) e C for funcionalmente dependente de B ($B \rightarrow C$), então C é transitivamente dependente de A via B (desde que A não seja funcionalmente dependente de B ou C). Se existir uma dependência transitiva na chave primária, a tabela não está na 3NF. A dependência transitiva deve ser removida para que uma tabela atinja 3NF.

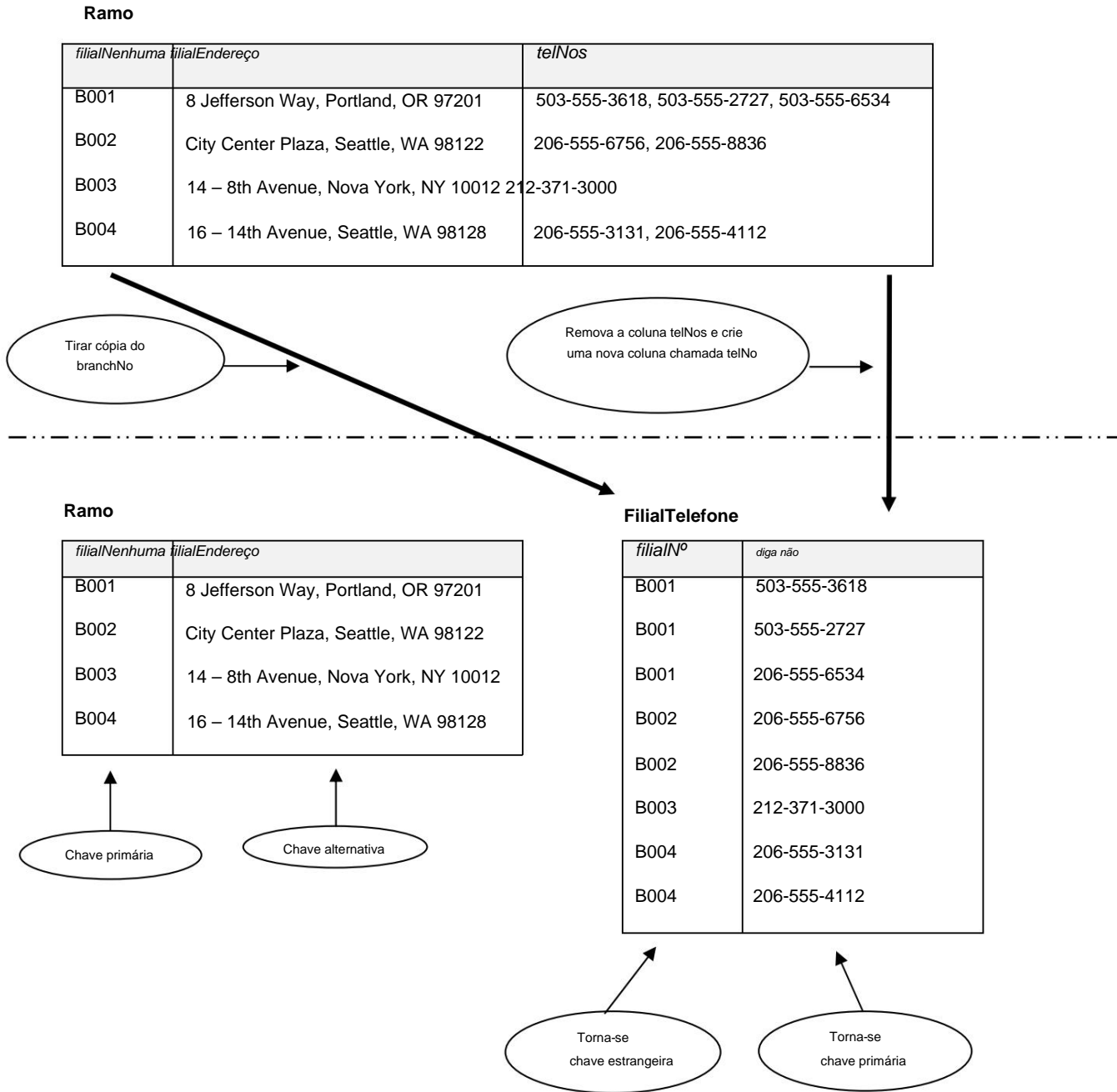
8. Examine a tabela mostrada abaixo.

<i>filial</i> <i>Nenhuma</i>	<i>filial</i> <i>Endereço</i>	<i>tel</i> <i>Nos</i>
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 – 8th Avenue, Nova York, NY 10012 212-371-3000	
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

- (a) Por que esta tabela não está na 1FN?
- (b) Descreva e ilustre o processo de normalização dos dados mostrados nesta tabela para a terceira forma normal (3NF).
- (c) Identifique as chaves primárias, alternativas e estrangeiras em suas relações 3NF.

Responda:





(Observação: há uma abordagem alternativa para alterar a tabela de ramais original – colunas podem ser adicionadas à tabela de ramais original para manter os valores individuais de cada número de telefone, por exemplo, telNo1, telNo2 e telNo3.)

9. Examine a tabela mostrada abaixo.

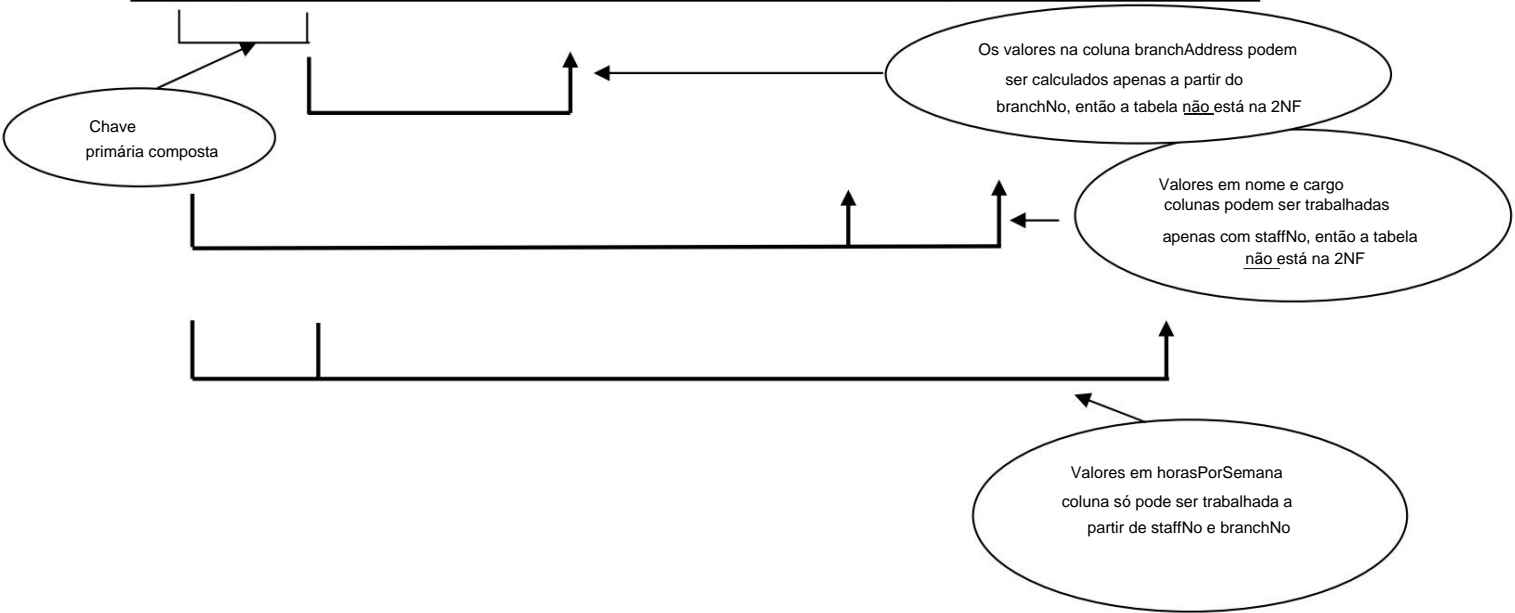
staffSem	filial	Nenhuma filial	Endereço	nome	posição	horas por semana
S4555	B002		City Center Plaza, Seattle, WA 98122	Ellen Layman	Assistant	16
S4555	B004		16 – 14th Avenue, Seattle, WA 98128	Ellen Layman	Assistant 9	
S4612	B002		City Center Plaza, Seattle, WA 98122	Dave Sinclair	Assistente	14
S4612	B004		16 – 14th Avenue, Seattle, WA 98128	Dave Sinclair	Assistente	10

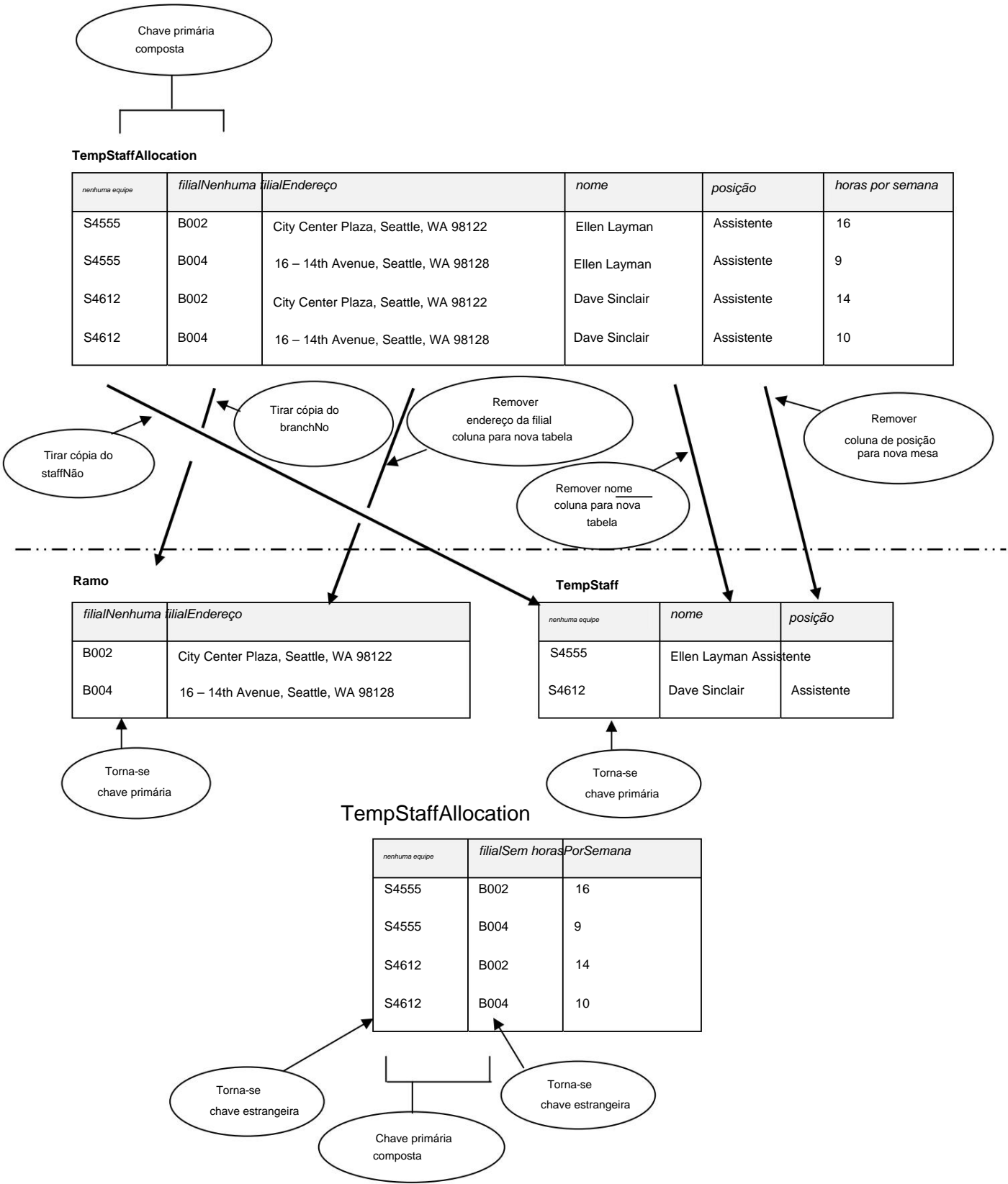
- (a) Por que esta tabela não está na 2FN?
- (b) Descreva e ilustre o processo de normalização dos dados mostrados nesta tabela para a terceira forma normal (3NF).
- (c) Identifique as chaves primária, (alternativa) e estrangeira em suas relações 3NF.

Responda:

TempStaffAllocation

staffSem	filial	Nenhuma filial	Endereço	nome	posição	horas por semana
S4555	B002		City Center Plaza, Seattle, WA 98122	Ellen Layman	Assistant	16
S4555	B004		16 – 14th Avenue, Seattle, WA 98128	Ellen Layman	Assistant	9
S4612	B002		City Center Plaza, Seattle, WA 98122	Dave Sinclair	Assistente	14
S4612	B004		16 – 14th Avenue, Seattle, WA 98128	Dave Sinclair	Assistente	10





10. Examine a tabela mostrada abaixo.

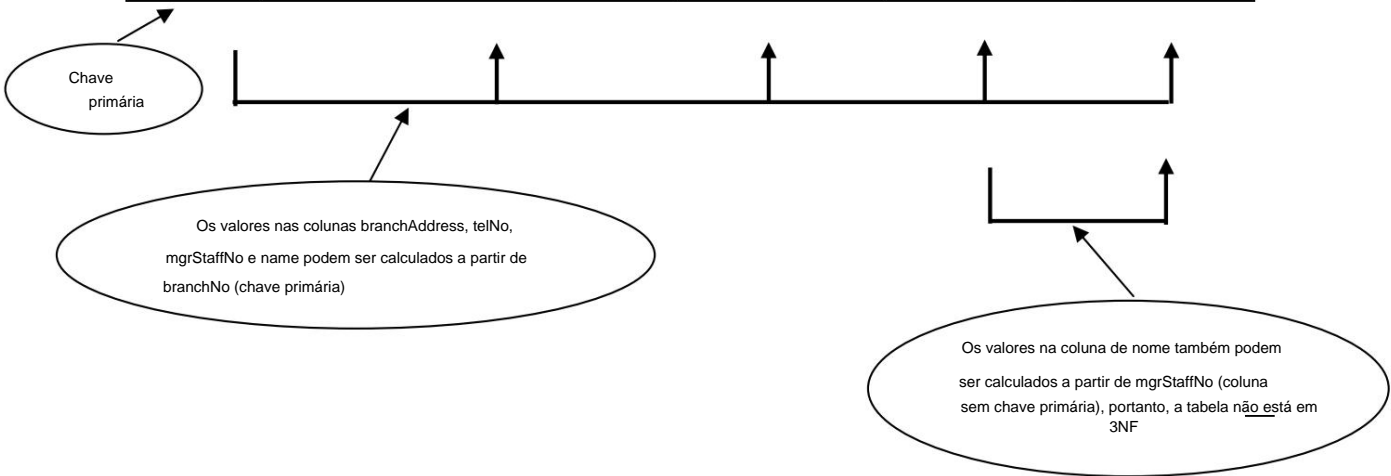
<i>filialNenhuma</i>	<i>filialEndereço</i>	<i>diga não</i>	<i>mgrStaffNo</i>	<i>nome</i>
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618	S1500	Tom Daniels
B002	City Center Plaza, Seattle, WA 98122	206-555-6756	S0010	Maria Martinez
B003	14 – 8th Avenue, Nova York, NY 10012 212-371-3000		S0145	Arte Peters
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131	S2250	Sally Stern

- (a) Por que esta tabela não está na 3FN? (b) Descrever e ilustrar o processo de normalização dos dados mostrados nesta tabela para a terceira forma normal (3NF). (c) Identifique as chaves primária, (alternativa) e estrangeira em suas relações 3NF.

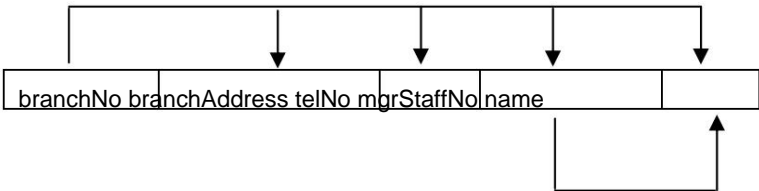
Responda:

Gerente de Filial

<i>filialNenhuma</i>	<i>filialEndereço</i>	<i>diga não</i>	<i>mgrStaffNo</i>	<i>nome</i>
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618	S1500	Tom Daniels
B002	City Center Plaza, Seattle, WA 98122	206-555-6756	S0010	Maria Martinez
B003	14 – 8th Avenue, Nova York, NY 10012 212-371-3000		S0145	Arte Peters
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131	S2250	Sally Stern

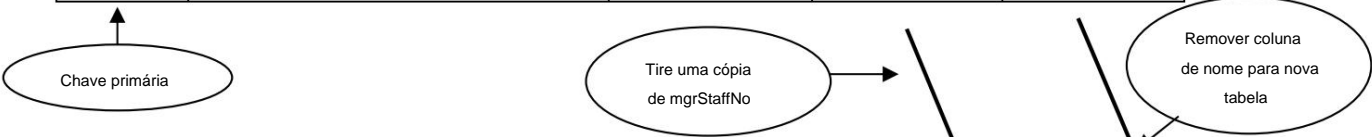


Usando o diagrama de dependência, ele pode ser representado conforme mostrado na figura a seguir:



Gerente de Filial

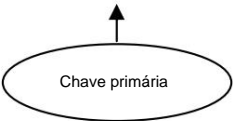
<i>filial</i> Nenhuma	<i>filial</i> Endereço	<i>diga não</i>	<i>mgrStaffNo</i>	<i>nome</i>
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618	S1500	Tom Daniels
B002	City Center Plaza, Seattle, WA 98122	206-555-6756	S0010	Maria Martinez
B003	14 – 8th Avenue, Nova York, NY 10012	212-371-3000	S0415	Arte Peters
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131	S2250	Sally Stern



A tabela BranchManger é renomeada para Branch

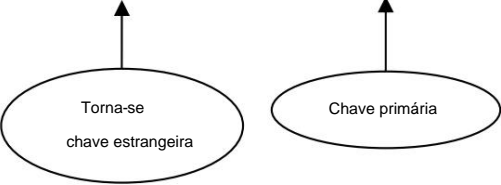
Ramo

<i>filial</i> Nenhuma	<i>filial</i> Endereço	<i>diga não</i>	<i>mgrStaffNo</i>
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618	S1500
B002	City Center Plaza, Seattle, WA 98122	206-555-6756	S0010
B003	14 – 8th Avenue, Nova York, NY 10012	212-371-3000	S0415
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131	S2250



GerentePessoal

<i>mgrStaffNo</i>	<i>nome</i>
S1500	Tom Daniels
S0010	Maria Martinez
S0415	Arte Peters
S2250	Sally Stern



Mais perguntas e respostas

1. O que é normalização?

A normalização é o processo de atribuição de atributos a entidades. Corretamente executado, o processo de normalização elimina redundâncias descontroladas de dados, eliminando assim as anomalias de dados e os problemas de integridade de dados que são produzidos por tais redundâncias.

A normalização não elimina a redundância de dados; em vez disso, produz a redundância cuidadosamente controlada que nos permite vincular adequadamente as tabelas do banco de dados.

2. Quando uma mesa está na 1FN?

Uma tabela está na 1NF quando todos os atributos de chave são definidos (sem grupos repetidos na tabela) e quando todos os atributos restantes são dependentes da chave primária. No entanto, uma tabela na 1NF ainda pode conter dependências parciais, ou seja, dependências baseadas apenas em parte da chave primária.

3. Quando uma tabela está na 2FN?

Uma tabela está na 2FN quando está na 1FN e não inclui dependências parciais. No entanto, uma tabela em 2NF ainda pode ter dependências transitivas, ou seja, dependências baseadas em atributos que não fazem parte da chave primária.

4. Quando é uma tabela na 3FN?

Uma tabela está na 3FN quando está na 2FN e não contém dependências transitivas.

5. Quando é uma tabela na BCNF?

Uma tabela está na Forma Normal de Boyce-Codd (BCNF) quando está na 3FN e cada determinante na tabela é uma chave candidata. Por exemplo, se a tabela estiver na 3NF e contiver um atributo não principal que determine um atributo principal, os requisitos da BCNF não serão atendidos. Esta descrição produz claramente as seguintes conclusões:

Se uma tabela está na 3NF e contém apenas uma chave candidata, 3NF e BCNF são equivalentes.

O BCNF pode ser violado somente se a tabela contiver mais de uma chave candidata. Colocando de outra forma, não há como o requisito BCNF ser violado se houver apenas uma chave candidata.

6. Dado o diagrama de dependência mostrado na Figura a seguir, responda às seguintes perguntas.

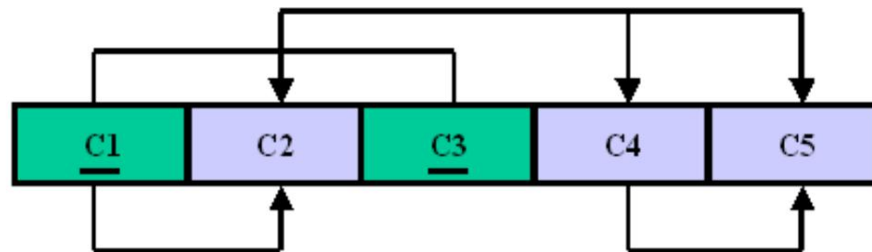


Figura: O Diagrama de Dependência

uma. Identifique e discuta cada uma das dependências indicadas.

C1 e C2 representa uma dependência parcial, pois C2 depende apenas de C1, e não de toda a chave primária composta por C1 e C3.

C4 e C5 representa uma dependência transitiva, pois C5 depende de um atributo (C4) que não faz parte de uma chave primária.

C1, C3 e C2, C4, C5 representa uma dependência funcional, pois C2, C4 e C5 dependem da chave primária composta por C1 e C3.