

No Visual FoxPro, você utiliza [bancos de dados](#) para organizar e relacionar [tabelas](#) e [visualizações](#). Os bancos de dados fornecem a arquitetura para armazenar seus dados, além de possuir vantagens adicionais. Quando você utiliza um banco de dados, pode criar extensões de nível de tabela, como regras de nível de campo e de registro, valores de campo padrão e [disparadores](#). Você pode criar, também, [procedimentos](#) armazenados e relações [permanentes](#) de tabelas. Você pode utilizar seu banco de dados para acessar conexões com origens de dados remotas e para criar [visualizações](#) de tabelas locais e remotas.

Este capítulo apresenta diretrizes para o planejamento de tabelas destinadas a um banco de dados do Visual FoxPro. Ele o orienta através da estruturação do banco de dados de exemplo da Tasmanian Traders e oferece estruturações adicionais de exemplos de bancos de dados. O banco de dados de exemplo da Tasmanian Traders, TASTRADE.DBC, está localizado em VFP\SAMPLES\TASTRADE\DATA.

Para obter informações quanto à criação de bancos de dados do Visual FoxPro depois de tê-los estruturado, consulte o capítulo 6, [Criando bancos de dados](#). Para obter informações sobre a criação de tabelas do Visual FoxPro, consulte o capítulo 7, [Trabalhando com tabelas](#).

Este capítulo examina:

- Utilizando um processo de estruturação de banco de dados
- Analisando requisitos de dados
- Agrupando requisitos em tabelas
- Determinando os campos necessários
- Identificando relacionamentos
- Refinando a estruturação
- Diagramas de bancos de dados de exemplo

## Utilizando um processo de estruturação de banco de dados

Se você utilizar um processo de estruturação de banco de dados estabelecido, poderá criar rápida e efetivamente um banco de dados bem estruturado que oferecerá o acesso conveniente às informações desejadas. Com uma estruturação sólida, você levará menos tempo na construção do banco de dados e obterá resultados mais rápidos e precisos.

**Observação** Os termos “banco de dados” e “tabela” não são sinônimos no Visual FoxPro. O termo [banco de dados](#) (arquivo .DBC) refere-se a um banco de dados relacional que é um recipiente de informações sobre uma ou mais [tabelas](#) (arquivos .DBF) ou visualizações.

A chave para uma estruturação eficiente de banco de dados está em compreender exatamente quais informações você quer armazenar e o modo como um sistema de gerenciamento de banco de dados relacional, como o Visual FoxPro, armazena dados. Para fornecer as informações de modo eficiente e preciso, o Visual FoxPro precisa ter os dados relativos a assuntos diferentes organizados em tabelas separadas. Por exemplo, você pode ter uma tabela que armazena apenas os fatos sobre funcionários e outra que armazena apenas os fatos sobre vendas.

Quando você organiza seus dados adequadamente, cria flexibilidade em seu banco de dados e adquire a capacidade de combinar e apresentar os fatos de várias maneiras diferentes. Por exemplo, você pode imprimir relatórios que combinam fatos sobre funcionários e sobre vendas.

### Separar fatos em tabelas adiciona flexibilidade ao seu banco de dados.

Armazene todos os dados sobre funcionários...



Emp_id	Last_name	First_name	Title
5	Buchanan	Steven	Gerente comercial
6	Suyama	Michael	Representante comercial
7	King	Robert	Representante comercial
8	Callahan	Laura	Coordenador de vendas internas
9	Dodsworth	Anne	Representante comercial
10	Hellstern	Albert	Gerente comercial
11	Smith	Tim	Responsável pelo malote
12	Patterson		

Order_id	Cust_id	Emp_id	Item	To name	To address
----------	---------	--------	------	---------	------------

...e todos os dados sobre vendas...

Armazene todos os dados sobre funcionários...

Emp_id	Last_name	First_name	Title
5	Buchanan	Steven	Gerente comercial
6	Suyama	Michael	Representante comercial
7	King	Robert	Representante comercial
8	Callahan	Laura	Coordenador de vendas internas
9	Dodsworth	Anne	Representante comercial
10	Hellstern	Albert	Gerente comercial
11	Smith	Tim	Responsável pelo malote
12	Patterson		

Order_id	Cust_id	Emp_id	To_name	To_address
10000	FRANS	6	Franchi S.p.A.	Via Monte Bianco 34
10001	MEREP	8	Mère Paillarde	43 rue St. Laurent
10002	FOLKO	3	Folk och få HB	Åkergatan 24
10003	SIMOB	8	Simons bistro	Vinbæltet 34
10004	VAFFE	3	Vaffeljernet	Smagsloget 45
10005	WARTH	5	Wartian Herkku	Torikatu 38
10006	FRANS	8	Franchi S.p.A.	Via Monte Bianco 34
10007	MORGK	4	Morgenstern Gesundkost	Heerstr. 22

Ao organizar um banco de dados, você primeiro divide as informações que quer manter como assuntos separados e, em seguida, informa o Visual FoxPro sobre como os assuntos se relacionam entre si, para que este possa reunir as informações corretas quando você precisar. Mantendo as informações em tabelas separadas, você facilita a organização e manutenção dos seus dados, assim como a construção de um aplicativo de alto desempenho.

Estas são as etapas do processo de estruturação de um banco de dados. Cada etapa é examinada em detalhes nas seções seguintes deste capítulo.

- 1 Determine a finalidade do seu banco de dados** Conhecer a finalidade ajudará na decisão de quais fatos você quer que o Visual FoxPro armazene.
- 2 Determine as tabelas necessárias** Quando você tiver uma finalidade clara para o seu banco de dados, você pode dividir suas informações em assuntos separados, como Funcionários (Employees) ou Pedidos (Orders). Cada assunto será uma tabela do seu banco de dados.
- 3 Determine os campos necessários** Decida quanto às informações que você quer manter em cada tabela. Cada categoria de informação em uma tabela é chamada de um [campo](#) e é exibida como uma coluna quando você [pesquisa](#) a tabela. Por exemplo, um campo em uma tabela Funcionários (Employees) poderia ser Last\_name (Sobrenome); outro poderia ser Hire\_date (Data\_admissão).
- 4 Determine os relacionamentos** Observe cada tabela e decida quanto ao modo em que os dados em uma tabela estão relacionados com os dados de outras. Adicione campos às tabelas ou crie novas, para esclarecer os relacionamentos conforme necessário.
- 5 Refine sua criação** Analise sua criação quanto a erros. Crie as tabelas e adicione alguns registros de dados de exemplo. Verifique se consegue obter os resultados que você quer das suas tabelas. Faça os ajustes à estruturação conforme necessário.

Não se preocupe se cometer erros ou se alguma coisa ficar fora da sua estruturação inicial. Considere-a como um esboço inicial que poderá aprimorar depois. Faça experiências com os dados de exemplo e protótipos dos seus [formulários](#) e relatórios. Com o Visual FoxPro, é fácil fazer alterações na estruturação do seu banco de dados enquanto você o cria. Porém, é muito mais difícil fazer alterações nas tabelas depois de elas estarem cheias de dados ou depois de você ter construído formulários e relatórios. Por esse motivo, assegure-se de ter uma estruturação sólida antes de ter avançado demais na construção do seu aplicativo.

## Analisando os requisitos dos dados

Sua primeira etapa na estruturação de um banco de dados do Visual FoxPro é analisar os requisitos dos seus dados, determinando a finalidade do banco de dados e como ele deve ser utilizado. Isso indica quais informações você quer do banco de dados. A partir daí, você pode determinar os assuntos a respeito dos quais precisa armazenar fatos (as tabelas) e quais os fatos que precisa armazenar a respeito de cada assunto (os campos nas tabelas).

Converse com as pessoas que utilizarão o banco de dados. Peça sugestões quanto a questões que você gostaria que o banco de dados respondesse. Esboce os relatórios que gostaria que ele gerasse. Reuna os formulários que você utiliza atualmente para registrar seus dados. Você usará todas essas informações nos passos seguintes do processo de criação.

### Exemplo: Controle de vendas e estoque

Suponha que a Tasmanian Traders, uma empresa de importação/exportação que vende alimentos especiais de todas as partes do mundo, deseje um banco de dados que possa controlar as informações sobre as vendas e o estoque da empresa.

Comece escrevendo uma lista de perguntas que o banco de dados seja capaz de responder. Quantas vendas do nosso produto em promoção atingimos no mês passado? Onde moram nossos melhores clientes? Quem é o fornecedor do produto que mais vendemos?

Em seguida, reuna todos os formulários e relatórios que contenham informações que o banco de dados deveria ser capaz de gerar. Atualmente a empresa utiliza um relatório impresso para controlar os produtos que estão sendo pedidos e um formulário para a colocação de novos pedidos. A ilustração abaixo mostra esses dois documentos.

**Formulários e relatórios mostram alguns requisitos de dados para o seu banco de dados.**

Produtos solicitados					
14-Jun-95					
Categoria	Nome do produto	Em estoque	Solicitados	Nome do fornecedor	Telefone
Bebidas	15% do total do inventário				
	Beer	10	1	Tasmanian Traders	(011) 555-1234
	Whisky	10	1	Tasmanian Traders	(011) 555-1234
	Wine	10	1	Tasmanian Traders	(011) 555-1234
	Tea	10	1	Tasmanian Traders	(011) 555-1234
	Coffee	10	1	Tasmanian Traders	(011) 555-1234
	Soft drink	10	1	Tasmanian Traders	(011) 555-1234
	Alcoholic	10	1	Tasmanian Traders	(011) 555-1234
	Non-alcoholic	10	1	Tasmanian Traders	(011) 555-1234
	Organic	10	1	Tasmanian Traders	(011) 555-1234
	Gluten-free	10	1	Tasmanian Traders	(011) 555-1234
	Low-sodium	10	1	Tasmanian Traders	(011) 555-1234
	Low-sugar	10	1	Tasmanian Traders	(011) 555-1234
	Low-fat	10	1	Tasmanian Traders	(011) 555-1234
	Low-calorie	10	1	Tasmanian Traders	(011) 555-1234
		559	81		
Condimentos	15% do total do inventário				
Mustard	10	1	Tasmanian Traders	(011) 555-1234	

FORMULÁRIO DE PEDIDO				
<b>Tasmanian Traders</b>				
One Tasmanian Way, Twin Peaks, WA 98000				
Fone: 1-206-555-8888 Fax: 1-206-555-8889				
Enviar para: <u>Tasmanian Traders</u>	Enviar para: <u>Tasmanian Traders</u>			
<u>1234567890</u>	<u>1234567890</u>			
Data do pedido: <u>14-Jun-95</u>				
Código do produto	Nome do produto	Quant.	Quantidade	Preço total
1	Beer	10	10	10.00
2	Whisky	10	10	10.00
3	Wine	10	10	10.00
Subtotal:				30.00
Fatura:				30.00
Total:				30.00

**Produtos solicitados**  
14-Jan-95

Categoria	Nome do produto	Em estoque	Solicitados	Nome do fornecedor	Telefone
Bebidas	10% do total do inventário				
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
	Beer	10	1	La cornu d'abondance	011 2 555 1234
Condimentos	10% do total do inventário				
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234
	Mustard	10	1	La cornu d'abondance	011 2 555 1234

559 80

**Tasmanian Traders**  
One Tasmanian Way, Twin Pines, WA 08000  
Phone: 1-202-555-8888 Fax: 1-202-555-8888

Customer name: Tasmanian Traders Enviar para: Tasmanian Traders  
 Customer address: 10000 1st Ave  
 Customer city: 10000 1st Ave  
 Customer state: 10000 1st Ave  
 Customer zip: 10000 1st Ave  
 Date da pedido: 14-Jan-95

Código da pedido	Nome da pedido	Quant.	Quantidade	Preço total
1	Beer	10	10	10.00
2	Mustard	10	10	10.00
3	Mustard	10	10	10.00
4	Mustard	10	10	10.00
5	Mustard	10	10	10.00
6	Mustard	10	10	10.00
7	Mustard	10	10	10.00
8	Mustard	10	10	10.00
9	Mustard	10	10	10.00
10	Mustard	10	10	10.00
11	Mustard	10	10	10.00
12	Mustard	10	10	10.00
13	Mustard	10	10	10.00
14	Mustard	10	10	10.00
15	Mustard	10	10	10.00
16	Mustard	10	10	10.00
17	Mustard	10	10	10.00
18	Mustard	10	10	10.00
19	Mustard	10	10	10.00
20	Mustard	10	10	10.00
21	Mustard	10	10	10.00
22	Mustard	10	10	10.00
23	Mustard	10	10	10.00
24	Mustard	10	10	10.00
25	Mustard	10	10	10.00
26	Mustard	10	10	10.00
27	Mustard	10	10	10.00
28	Mustard	10	10	10.00
29	Mustard	10	10	10.00
30	Mustard	10	10	10.00
31	Mustard	10	10	10.00
32	Mustard	10	10	10.00
33	Mustard	10	10	10.00
34	Mustard	10	10	10.00
35	Mustard	10	10	10.00
36	Mustard	10	10	10.00
37	Mustard	10	10	10.00
38	Mustard	10	10	10.00
39	Mustard	10	10	10.00
40	Mustard	10	10	10.00
41	Mustard	10	10	10.00
42	Mustard	10	10	10.00
43	Mustard	10	10	10.00
44	Mustard	10	10	10.00
45	Mustard	10	10	10.00
46	Mustard	10	10	10.00
47	Mustard	10	10	10.00
48	Mustard	10	10	10.00
49	Mustard	10	10	10.00
50	Mustard	10	10	10.00
51	Mustard	10	10	10.00
52	Mustard	10	10	10.00
53	Mustard	10	10	10.00
54	Mustard	10	10	10.00
55	Mustard	10	10	10.00
56	Mustard	10	10	10.00
57	Mustard	10	10	10.00
58	Mustard	10	10	10.00
59	Mustard	10	10	10.00
60	Mustard	10	10	10.00
61	Mustard	10	10	10.00
62	Mustard	10	10	10.00
63	Mustard	10	10	10.00
64	Mustard	10	10	10.00
65	Mustard	10	10	10.00
66	Mustard	10	10	10.00
67	Mustard	10	10	10.00
68	Mustard	10	10	10.00
69	Mustard	10	10	10.00
70	Mustard	10	10	10.00
71	Mustard	10	10	10.00
72	Mustard	10	10	10.00
73	Mustard	10	10	10.00
74	Mustard	10	10	10.00
75	Mustard	10	10	10.00
76	Mustard	10	10	10.00
77	Mustard	10	10	10.00
78	Mustard	10	10	10.00
79	Mustard	10	10	10.00
80	Mustard	10	10	10.00
81	Mustard	10	10	10.00
82	Mustard	10	10	10.00
83	Mustard	10	10	10.00
84	Mustard	10	10	10.00
85	Mustard	10	10	10.00
86	Mustard	10	10	10.00
87	Mustard	10	10	10.00
88	Mustard	10	10	10.00
89	Mustard	10	10	10.00
90	Mustard	10	10	10.00
91	Mustard	10	10	10.00
92	Mustard	10	10	10.00
93	Mustard	10	10	10.00
94	Mustard	10	10	10.00
95	Mustard	10	10	10.00
96	Mustard	10	10	10.00
97	Mustard	10	10	10.00
98	Mustard	10	10	10.00
99	Mustard	10	10	10.00
100	Mustard	10	10	10.00

Subtotal: 10.00  
 Total: 10.00  
 Total: 10.00

A Tasmanian Traders também precisa imprimir etiquetas postais para clientes, funcionários e fornecedores.

Depois de reunir essas informações, você está pronto para a próxima etapa.

## Agrupando os requisitos em tabelas

A determinação das [tabelas](#) do seu banco de dados pode ser a etapa mais ardilosa do seu processo de estruturação. Isso porque os resultados que você quer do seu banco de dados — os relatórios que quer imprimir, os formulários que quer utilizar e as perguntas as serem respondidas — nem sempre dão pistas quanto à estrutura das tabelas que os produzem. Podem indicar o que você quer saber, mas não como categorizar as informações em tabelas.

Observe o pedido acima como exemplo. Ele inclui fatos sobre o cliente — seu endereço e número de telefone — além de fatos sobre o pedido. Este formulário fornece vários fatos que você sabe que quer armazenar no seu banco de dados. Embora eles estejam todos no mesmo formulário, você pode facilmente evitar problemas comuns de integridade de dados armazenando-os em tabelas separadas.

**Armazenar informações uma só vez reduz a probabilidade de erro** Por exemplo, se você utilizasse apenas uma tabela para armazenar as informações de um formulário de pedido, suponha que um cliente colocasse três pedidos diferentes. Você poderia adicionar o endereço e o número do telefone do cliente ao seu banco de dados três vezes, uma para cada pedido. Mas isso multiplicaria a probabilidade de erros de entrada de dados.

**A tabela Customer armazena as informações de endereço uma vez.**

Customer_orders				
Order_id	Order_date	Company	Address	
10927	01/27/95	La cornu d'abondance	67, avenue de l'Europe	Qual endereço está correto?
10972	02/15/95	La cornu d'abondance	6, avenue De L' Europe	
10973	02/15/95	La cornu d'abondance	67, ave de l'Europe	
10923	01/25/95	La maison d'Asie	1 rue Alsace-Lorraine	
11051	03/21/95	La maison d'Asie	1 rue Alsace-Lorraine	
10891	01/11/95	Lehmanns Marktstand	Magazinweg 7	

Order_id	Order_date	Company	Address
10927	01/27/95	La corne d'abondance	67, avenue de l'Europe
10972	02/15/95	La corne d'abondance	6, avenue De L' Europe
10973	02/15/95	La corne d'abondance	67, ave de l'Europe
10923	01/25/95	La maison d'Asie	1 rue Alsace-Lorraine
11051	03/21/95	La maison d'Asie	1 rue Alsace-Lorraine
10891	01/11/95	Lehmans Marktstand	Magazinweg 7
10934	01/31/95	Lehmans Marktstand	Magazinweg 7

Qual endereço está correto?

Cust_id	Company	Address	City
LACOR	La corne d'abondance	67, avenue de l'Europe	Versailles
LAMAI	La maison d'Asie	1 rue Alsace-Lorraine	Toulouse
LAUGB	Laughing Bacchus Wine Cellars	1900 Oak St.	Vancouver
LAZYK	Lazy K Kountry Store	12 Orchestra Terrace	Walla Walla
LEHMS	Lehmans Marktstand	Magazinweg 7	Frankfurt a.
LETSS	Let's Stop N Shop	87 Polk St., Suite 5	San Francis
LILA S	LILA-Supermercado	Carrera 52 con Ave. Bolívar #65-98 Llano Barquisimel	

Por motivos de precisão, armazene cada dado apenas uma vez.

Além disso, se o cliente se mudasse, você teria de aceitar informações contraditórias ou localizar e alterar cada um dos registros de vendas desse cliente na tabela. É bem melhor criar uma tabela Customer (Cliente) que armazene o endereço do cliente no seu banco de dados uma só vez. Então, se você precisar alterar os dados, só os mudaria uma vez.

**Evitando a exclusão de informações valiosas** Suponha que um novo cliente faça um pedido e, em seguida, o cancele. Ao excluir o pedido da tabela que contém as informações sobre ambos, os clientes e seus pedidos, você excluiria, também, o nome e o endereço do cliente. Mas você quer manter o nome deste novo cliente no banco de dados para enviar-lhe o seu próximo catálogo. Novamente, é melhor colocar as informações sobre o cliente em uma tabela Customer separada. Assim você poderia excluir o pedido sem excluir as informações do cliente.

Observe as informações que você quer obter do seu banco de dados e divida-as em assuntos fundamentais que queira controlar, como clientes, funcionários, produtos que vende, serviços que fornece e assim por diante. Cada um desses assuntos é um candidato a uma tabela separada.

**Dica** Uma estratégia para dividir as informações em tabelas é observar os fatos individuais e determinar a que cada fato realmente se refere. Por exemplo, no formulário de pedido da Tasmanian Traders, o endereço do cliente não se refere à venda; mas ao cliente. Isso sugere que você precisa de uma tabela separada para clientes. No relatório Products On Order (Produtos encomendados), número do telefone do fornecedor não se refere ao produto em estoque; refere-se ao fornecedor. Isso sugere que você precisa de uma tabela separada para os fornecedores.

### Exemplo: Criando tabelas no banco de dados da Tasmanian Traders

O relatório Order Form e Products On Order (Formulário de pedidos e produtos encomendados) da Tasmanian Traders inclui informações sobre estes assuntos:

- Funcionários
- Clientes
- Fornecedores
- Produtos
- Pedidos

Desta lista, você pode extrair um esboço inicial das tabelas do banco de dados e dos registros para

cada tabela.

### Esboço inicial das tabelas e campos necessários para o banco de dados da Tasmanian Traders

BANCO DE DADOS TASMANIAN TRADERS			
	Funcionários	Cientes	Fornecedores
	Nome	Empresa	Empresa
	Endereço	Endereço	Endereço
		Contato	Contato
			Telefone
	Produtos	Pedidos	
	Nome	Data do Pedido	
	Preço da Unidade	Endereço de Entrega	
	Unidades em Estoque		
	Unidades Encomendadas		

Embora o banco de dados final da Tasmanian Traders contenha outras tabelas, esta lista é um bom começo. Adiante, neste capítulo, você verá como adicionar outras tabelas para aprimorar sua estruturação.

## Determinando os campos necessários

Para determinar os campos de uma tabela, decida o que você precisa saber sobre as pessoas, coisas ou eventos registrados na tabela. Você pode considerar os campos como atributos da tabela. Cada registro (ou linha) na tabela contém o mesmo conjunto de campos ou atributos. Por exemplo, um campo de endereço em uma tabela de clientes contém endereços de clientes. Cada registro na tabela contém dados sobre um cliente e o campo de endereço contém o endereço desse cliente.

## Identificando campos

Eis algumas dicas para determinar seus campos:

**Relacione cada campo diretamente ao assunto da tabela** Um campo que descreve o assunto de uma outra tabela pertence a essa outra tabela. Depois, quando você definir os relacionamentos entre as suas tabelas, verá como os dados de campos de várias tabelas podem ser combinados. Por enquanto, certifique-se de que cada campo em uma tabela descreva diretamente o assunto dela. Se você perceber que está repetindo a mesma informação em várias tabelas, é um sinal de que possui campos desnecessários em algumas delas.

**Não inclua dados derivados ou calculados** Na maioria dos casos, você não desejará armazenar o resultado de cálculos nas tabelas. Ao contrário, você poderá deixar que o Visual FoxPro faça os cálculos quando quiser ver o resultado. Por exemplo, o formulário de pedidos, mostrado anteriormente neste capítulo, exibe o preço estendido para cada linha do pedido no banco de dados da Tasmanian Traders. Porém, não há nenhum campo com o subtotal Extended Price (Preço estendido) em qualquer tabela da Tasmanian Traders. Em lugar disso, a tabela Order\_Line\_Items (Itens\_Linha\_Pedido) inclui um campo de quantidade que armazena as unidades pedidas para cada produto individual, assim como o preço unitário de cada item pedido. Utilizando esses dados, o Visual FoxPro calcula o subtotal a cada vez que você imprimir um formulário. O subtotal em si não precisa ficar armazenado em uma tabela.

**Inclua todas as informações necessárias** É fácil ter uma visão geral das informações importantes. Retorne para as informações que você reuniu na primeira etapa do processo de estruturação. Observe seus formulários e relatórios impressos para assegurar-se de que as



informações requeridas no passado estejam incluídas nas suas tabelas do Visual FoxPro ou possam ser derivadas delas. Pense nas perguntas que você fará ao Visual FoxPro. Ele pode encontrar todas as respostas utilizando as informações nas suas tabelas? Você identificou os campos que armazenarão dados únicos, como o customer ID (ID do cliente)? Quais tabelas incluem informações que você combinará em um relatório ou formulário? Para maiores informações sobre como identificar campos-chave e relacionar tabelas, consulte as seções [Utilizando campos da chave primária](#) e [Identificando relacionamentos](#), adiante neste capítulo.

**Armazene as informações em partes lógicas menores** Você pode sentir-se tentado a ter um único campo para nomes inteiros ou para nomes de produtos, em conjunto com a descrição deles. Se você combinar mais de um tipo de informação em um campo, será difícil recuperar fatos individuais depois. Tente dividir as informações em partes lógicas; por exemplo, crie campos separados para o primeiro nome e o sobrenome ou para nome, categoria e descrição do produto.

### Exemplo: Adicionando campos à tabela Products

A Tasmanian Traders vende alimentos especiais importados de todo o mundo. Os funcionários utilizam um relatório Products On Order (Produtos encomendados) para controlar os produtos que estão sendo pedidos.

#### Relatório para controlar o estoque de produtos

Produtos solicitados					
14-Jun-95					
Categoria	Nome do produto	Em estoque	Solicitados	Nome do fornecedor	Telefone
Bebidas	18% do total do inventário				
	Caviar	39	0	Exotic Liquids	(71) 555-2222
	Chang	17	40	Exotic Liquids	(71) 555-2222
	Chartreuse Verte	69	0	Aux joyeux ecclésiastiques	(1) 03 38 00 68
	Côte de Blaye	17	0	Aux joyeux ecclésiastiques	(1) 03 38 00 68
	Cornus Fantastica	20	0	Refrescos Americanos LTDA	(11) 555 4640
	Ikool Coffee	17	10	Leiz Trading	(85) 8237
	Lula's Hotpot	17	0	Surf'd Oy	(85) 30 956
	Lugling Lumberjack	17	0	Bigfoot Breweries	(805) 555-9981
	Luger	17	10	Pavlova, Ltd.	(06) 444-2545
	Outback Lager	125	0	Flanagan's	(069) 992 755
	Woburn's Winterbier	111	0	Lebensmittel Importe GmbH	(03) 555-9981
	Sandwich Ale	20	0	Bigfoot Breweries	(805) 555-9981
	Steeleye Stout	20	0	Bigfoot Breweries	(805) 555-9981
Condimentos	16% do total do inventário				
	Atwood Syrup	15	70	Exotic Liquids	(71) 555-2222
	Clef Anton's Cajun	15	0	New Orleans Cajun	(408) 555-4882
	Seasoning	0	0	Delight's	(408) 555-4882
	Clef Anton's Combo Mix	39	0	New Orleans Cajun	(06) 451-7877
	Green Shrimp			Delight's	
				Mexican's	

O relatório indica que a tabela Products, contendo os fatos sobre os produtos vendidos, precisa incluir campos para o nome do produto, unidades no estoque e as unidades encomendadas, entre outros. Mas e quanto aos campos para o nome e o telefone do fornecedor? Para gerar o relatório, o Visual FoxPro precisa saber qual fornecedor acompanha cada produto.

#### Esboço da tabela Supplier contendo o nome e o telefone do fornecedor

Produtos		Fornecedores
○	Nome do Produto	Empresa
	Preço da Unidade	Endereço
	Unidades em Estoque	Contato
	Unidades Encomendadas	Telefone

Você pode solucionar isto sem armazenar dados redundantes em suas tabelas, criando uma tabela Supplier com campos separados para o nome e telefone do fornecedor. Na próxima etapa, você

adicionará um campo à tabela Products que identifica as informações do fornecedor necessárias.

## Utilizando campos da chave primária

A potência de um sistema de gerenciamento de banco de dados relacional como o Visual FoxPro origina-se da sua habilidade de encontrar e reunir rapidamente informações armazenadas em tabelas separadas. Para que o Visual FoxPro trabalhe com a eficiência máxima, cada tabela do seu banco de dados deve incluir um campo ou conjunto de campos que identifiquem com exclusividade cada registro individual armazenado na tabela. Geralmente isso é um número de identificação único, como um número ID de um funcionário ou um número de série. Na terminologia dos bancos de dados, essa informação é denominada chave primária da tabela. O Visual FoxPro utiliza campos da chave primária para associar rapidamente os dados de várias tabelas e reuni-los para você.

Se você já possui um identificador exclusivo para uma tabela, como um conjunto de números de produtos que você desenvolveu para identificar os itens no seu estoque, poderá utilizar este identificador como a chave primária da tabela. Mas certifique-se de que os valores neste campo sempre serão diferentes para cada registro — o Visual FoxPro não permite valores duplicados em um campo da chave primária. Por exemplo, não utilize os nomes de pessoas como chave primária, pois os nomes não são únicos. Seria fácil ter duas pessoas com o mesmo nome na mesma tabela.

Ao escolher campos da chave primária, considere os seguintes pontos:

- O Visual FoxPro não permite valores nulos ou duplicados em um campo da chave primária. Por isso, você não deve escolher uma chave primária que possa conter tais valores.
- Você pode utilizar o valor no campo da chave primária para procurar registros, portanto ele não deve se grande demais para ser lembrado ou digitado. É recomendável que tenha um certo número de letras ou dígitos ou estar dentro de uma determinada faixa de valores.
- O tamanho da chave primária afeta a velocidade das operações no seu banco de dados. Ao criar campos da chave primária, utilize o menor tamanho que acomode os valores que você precisa armazenar no campo.

### Exemplo: Definindo a chave primária para a tabela Products

A chave primária da tabela Products da Tasmanian Traders contém números de Product\_id (ID de produtos). Como cada número de produto identifica um produto diferente, não deve haver dois produtos com o mesmo número.

**A chave primária para a tabela Products no campo Product\_id.**



The screenshot shows a table window titled 'Products'. The table has four columns: Product\_id, Product\_name, English\_name, and Unit\_cost. The Product\_id column is highlighted with a key icon, indicating it is the primary key. Below the table, a status bar message reads: 'Não existe dois códigos iguais de produtos... ...mas outros campos podem conter valores duplicados.'

Product_id	Product_name	English_name	Unit_cost
1	Chai	Chá Dharamsala	12.6000
2	Chang	Cerveja tibetana Barley	13.3000
3	Aniseed Syrup	Licor	7.0000
4	Chef Anton's Cajun Seasoning	Condimentos Cajun Chef Anton's	15.4000
5	Chef Anton's Gumbo Mix	Sopa de quiabo com galinha Chef Anton's	14.9450

Em alguns casos, é recomendável que você utilize dois ou mais campos que, juntos, forneçam a chave primária de uma tabela. Por exemplo, a tabela Order\_Line\_Items (Itens\_Linha\_Pedido) no banco de dados da Tasmanian Traders utiliza dois campos como sua chave primária: Order\_id e Product\_id (Pedido\_id e Produto\_id). Na próxima etapa você verá o porquê.

## Identificando relacionamentos

Agora que você dividiu suas informações em tabelas, precisa indicar ao Visual FoxPro como



reuni-las novamente de modos significativos. Por exemplo, o [formulário](#) abaixo inclui informações de várias tabelas.

O formulário **Entrada de pedido** utiliza informações de várias tabelas.

As informações são provenientes da tabela Customer...

...da tabela Product...

...da tabela Order Line Items...

...e da tabela Orders.

**Entrada de pedidos**

**Cliente** B's Beverages

**Enviar para** B's Beverages

**Endereço** Fauntleroy Circus

**Cidade** London **CEP** EC2 5NT

**Bairro** **País** Itália

**Pedido nº** 1

**Data do pedido** 05/09/92

**Informações sobre a entrega**

Speedy Express

**Pagável em** 06/06/92

Produto	Quantidade	Preço unitário	Total
Ikura	5.000	31.0000	155.0000
Chartreuse verte	1.000	18.0000	18.0000

**Obs.** Clique na grade para acessar menu

**Último pedido**

**Crédito disponível:** \$11,000.00

**Pago** ☒

**Ajuda**

**Subtotal do item** 173.00

**% de desconto** 10 17.30

**Frete** 0.00

**Total da fatura** 155.70

O Visual FoxPro é um sistema de gerenciamento de banco de dados relacional. Isso significa que você armazena dados relacionados em tabelas separadas. Em seguida você define os relacionamentos entre as tabelas e o Visual FoxPro utiliza-os para localizar as informações associadas no seu banco de dados.

Por exemplo, suponha que você queira telefonar para um funcionário para fazer perguntas sobre uma venda que ele efetuou. Os telefones de funcionários são registrados na tabela Employee; as vendas são registradas na tabela Orders. Quando você informa ao Visual FoxPro qual é a venda em que está interessado, o Visual FoxPro pode procurar o número de telefone baseado no relacionamento entre as duas tabelas. Isso funciona porque Employee\_id (Funcionário\_id), a [chave primária](#) para a tabela Employee é, também, um campo na tabela Orders. Na terminologia dos bancos de dados, o campo Employee\_id (Funcionário\_id) na tabela Orders é chamado [chave estrangeira](#), porque refere-se a uma chave primária de uma tabela diferente ou estrangeira.

**O campo Employee\_id como a chave primária para a tabela Employee e a chave estrangeira para a tabela Orders**

Chave primária

Emp_id	Last_name	First_name	Title
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative

O campo Emp\_id aparece em ambas as tabelas.

Chave estrangeira

Order_id	Cust_id	Emp_id	To_name	To_address
10005	WARTH	5	Wartian Herkku	Torikatu 38
10006	FRANS	8	Franchi S.p.A.	Via Monte Bianco 34

Chave primária

Emp_id	Last_name	First_name	Title
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative

O campo Emp\_id aparece em ambas as tabelas.

Chave estrangeira

Order_id	Cust_id	Emp_id	To_name	To_address
10005	WARTH	5	Wartian Herkku	Torikatu 38
10006	FRANS	8	Franchi S.p.A.	Via Monte Bianco 34
10007	MORIGK	4	Mormenstern Gesundkost	Heerstr. 22

Assim, para configurar um relacionamento entre duas tabelas — Tabela A e Tabela B — você adiciona a chave primária de uma tabela à outra tabela, para que ela apareça em ambas. Mas como decidir qual é a chave primária da tabela a ser usada? Para configurar o relacionamento corretamente, você deve determinar, primeiro, a natureza dele. Há três tipos de relacionamentos entre tabelas:

- [Relacionamentos um-para-n](#)
- [Relacionamentos n-para-n](#)
- [Relacionamentos um-para-um](#)

O resto desta seção apresenta um exemplo de cada tipo de relacionamento e explica como criar suas tabelas de modo que o Visual FoxPro possa associar os dados corretamente. A finalidade de cada exemplo é explicar como você determina os relacionamentos entre as suas tabelas e como você decide quanto a quais campos pertencem às tabelas para suportar esses relacionamentos — não é apresentada uma descrição sobre como utilizar a interface do Visual FoxPro para relacionar tabelas.

### Exemplo: Criando um relacionamento um-para-n

O relacionamento [um-para-n](#) é o tipo de relacionamento mais comum em um banco de dados relacional. Nele, um registro na Tabela A pode ter mais de um registro correspondente na Tabela B, mas um registro na Tabela B tem, no máximo, um registro correspondente na Tabela A.

Por exemplo, as tabelas Category e Products no banco de dados da Tasmanian Traders têm um relacionamento um-para-n.

**As tabelas Category e Products representam um relacionamento um-para-n.**

Category_id	Name	Description	Picture_file	Picture
1	Bebidas	Memo	Memo	Gen
2	Condimentos	Memo	Memo	Gen
3	Confeções	Memo	Memo	Gen
4	Produtos do dia-a-dia	Memo	Memo	Gen

Uma categoria pode estar relacionada a vários produtos.

Product_id	Supplier_id	Category_id	Product_name
43	20	1	Ipoh Coffee
67	16	1	Laughing Lumberjack Lager
70	7	1	Outback Lager
75	12	1	Rönbräu Klosterbier
76	23	1	Lakkalikööri
3	7	2	Aniseed Syrup

Category				
Category_id	Name:	Description	Picture_file	Picture
1	Bebidas	Memo	Memo	Gen
2	Condimentos	Memo	Memo	Gen
3	Confeccões	Memo	Memo	Gen
4	Produtos do dia-a-dia	Memo	Memo	Gen

Uma categoria pode estar relacionada a vários produtos.

Products				
Product_id	Supplier_id	Category_id	Product_name	
43	20	1	Ipoh Coffee	
67	16	1	Laughing Lumberjack Lager	
70	7	1	Outback Lager	
75	12	1	Rhönbräu Klosterbier	
76	23	1	Lakkaikööri	
3	7	2	Aniseed Syrup	

Para configurar o relacionamento, você adiciona o campo ou campos que compõem a [chave primária](#) no lado “um” do relacionamento à tabela no lado “n” do relacionamento. Você utiliza uma chave de índice primário ou de [índice candidato](#) para o lado “um” do relacionamento e uma chave de [índice normal](#) para o lado “n”. Neste caso, você adicionaria o campo Category\_id (ID da Categoria) da tabela Category à tabela Products, pois uma categoria inclui vários produtos. O Visual FoxPro utiliza o número ID da categoria para localizar a categoria correta para cada produto.

Para obter informações sobre como criar chaves de índice, consulte o capítulo 7, [Trabalhando com tabelas](#).

### Exemplo: Criando um relacionamento n-para-n

Em um relacionamento [n-para-n](#), um registro na Tabela A pode ter mais de um registro correspondente na Tabela B, e um registro na Tabela B pode ter mais de um na Tabela A. Este tipo de relacionamento requer alterações na estruturação do seu banco de dados antes que você possa especificar corretamente o relacionamento para o Visual FoxPro.

Para detectar relacionamentos n-para-n entre as suas tabelas, é importante que você observe as duas direções do relacionamento. Por exemplo, considere o relacionamento entre pedidos e produtos nas transações da Tasmanian Traders. Um pedido pode incluir mais de um produto. Assim, para cada registro na tabela Orders, pode haver vários registros na tabela Products. Mas isso não é tudo. Cada produto pode aparecer em vários pedidos. Assim, para cada registro na tabela Products, pode haver vários registros na tabela Orders.

As tabelas Orders e Products representam um relacionamento n-para-n.

Entrada de pedidos

Cliente

Alfreds Futterkiste

Enviar para

Alfreds Futterkiste

Endereço

Obere Str. 57

Cidade

Berlin

Bairro

CEP

12209

País

Alemanha

Pedido nº

1142

Data do pedido

11/12/96

Informações sobre a entrega

Speedy Express

Pagável em

11/19/96

Produto	Quantidade	Preço unitário	Total
Manjimup Dried Apples	12.000	53.0000	636.0000
Pâté chinois	20.000	24.0000	480.0000

Obs.

Clique

Entrada de pedidos

Cliente

Centro comercial Moctezuma

Pedido nº

1142

**\$ Entrada de pedidos**

**Cliente** Alfreds Futterkiste **Pedido nº** 1142

**Enviar para** Alfreds Futterkiste **Data do pedido** 11/12/96

**Endereço** Obere Str. 57

**Cidade** Berlin **CEP** 12209

**Bairro** **País** Alemanha

**Informações sobre a entrega**

Speedy Express

**Pagável em** 11/19/96

Produto	Quantidade	Preço unitário	Total
Manjimup Dried Apples	12.000	53.0000	636.0000
Pâté chinois	20.000	24.0000	480.0000

**Obs. Clique**

Um pedido pode ter vários produtos.  
Cada produto pode aparecer em vários produtos.

**\$ Entrada de pedidos**

**Cliente** Centro comercial Moctezuma **Pedido nº** 1142

**Enviar para** Centro comercial Moctezuma **Data do pedido** 11/12/96

**Endereço** Sierras de Granada 9993

**Cidade** México D.F. **CEP** 05022

**Bairro** **País** Mexico

**Informações sobre a entrega**

United Package

**Pagável em** 08/06/92

Produto	Quantidade	Preço unitário	Total
Cher Anton's Cajun Seasoning	8.000	22.0000	176.0000
Manjimup Dried Apples	1.000	53.0000	53.0000
Filo Mix	8.000	7.0000	56.0000

**Obs. Clique na grade para acessar menu**

**Último pedido**

**Subtotal do item** 678.60

**Crédito disponível:** 5 **% de desconto** 33.93

\$1,000.00 **Pago** ☒ **Frete** 0.00

**Ajuda** **Total da fatura** 644.67

Os assuntos das duas tabelas — pedidos e produtos — têm um relacionamento n-para-n. Isso propõe um desafio para a estruturação do banco de dados. Para compreender o problema, imagine o que aconteceria se você quisesse configurar o relacionamento entre as duas tabelas adicionando o campo `Product_id` (`Produto_id`) à tabela `Orders`. Para ter mais de um produto por pedido, você precisaria de mais de um registro na tabela `Orders`, por pedido. Você ficaria repetindo as informações do pedido várias vezes para cada registro que se relacionasse com um único pedido — uma criação ineficiente que poderia levar a dados imprecisos. O mesmo problema ocorreria se o campo `Order_id` (`Pedido_id`) fosse colocado na tabela `Products` — você teria mais de um registro na tabela `Products` para cada produto. Como solucionar este problema?

A resposta é criar uma terceira tabela que divida o relacionamento n-para-n em dois relacionamentos um-para-n. Esta terceira tabela é chamada *tabela de associação*, pois age como uma associação entre as outras duas. Você coloca a chave primária de cada uma das duas tabelas na tabela de associação.

**A tabela `Order_Line_Items` cria um vínculo um-para-n entre `Orders` e `Products`.**

Chave primária da tabela `Orders`.

Chave primária da tabela `Products`.

Line_no	Order_id	Product_id	Unit_price	Quantity
1	10000	17	27.0000	4.000
1	10001	25	9.8000	42.000
2	10001	40	12.8000	36.000
3	10001	59	38.5000	24.000
4	10001	64	23.0000	12.000



Chave primária da tabela Orders.

Chave primária da tabela Products.

Informações referentes ao pedido e ao produto..

Line_no	Order_id	Product_id	Unit_price	Quantity
1	10000	17	27.0000	4.000
1	10001	25	9.8000	42.000
2	10001	40	12.8000	36.000
3	10001	59	38.5000	24.000
4	10001	64	23.0000	12.000
1	10002	31	8.0000	15.000
2	10002	39	12.6000	19.000
3	10002	71	15.0000	15.000
1	10003	18	13.7000	12.000

Uma [tabela de associação](#) pode conter apenas as duas [chaves primárias](#) das tabelas que ele vincula entre si ou, como na tabela Order\_Line\_Items, poderia conter informações adicionais.

Cada [registro](#) na tabela Order\_Line\_Items representa uma linha de item em um pedido. A [chave primária](#) da tabela Order\_Line\_Items consiste em dois [campos](#) — as [chaves estrangeiras](#) das tabelas Orders e Products. O campo Order\_id (Pedido\_id) sozinho não funciona como a chave primária para essa tabela, pois um pedido pode ter vários itens de linha. O ID do produto é repetido para cada linha de um pedido, de modo que o campo não contém valores exclusivos. O campo Order\_id sozinho também não funciona, pois um produto pode aparecer em vários pedidos diferentes. Mas juntos, os dois campos na [tabela de associação](#) sempre produzem um valor exclusivo para cada registro. A tabela de associação não requer sua própria [chave primária](#).

No banco de dados da Tasmanian Traders, a tabela Orders e a tabela Products não estão diretamente relacionadas entre si. Em lugar disso, são indiretamente relacionadas através da tabela Order\_Line\_Items. O relacionamento n-para-n entre pedidos e produtos é representado no banco de dados utilizando-se dois relacionamentos um-para-n:

- As tabelas Orders e Order\_Line\_Items têm um relacionamento um-para-n. Cada pedido pode ter mais de um item de linha, mas cada item de linha é ligado a apenas um pedido.
- As tabelas Products e Order\_Line\_Items têm um relacionamento um-para-n. Cada produto pode ter vários itens de linha associados a ele, mas cada item de linha refere-se a apenas um produto.

### Exemplo: Criando um relacionamento um-para-um

Em um relacionamento [um-para-um](#), um registro na Tabela A pode ter no máximo um registro coincidente na Tabela B e um registro na Tabela B pode ter no máximo um registro coincidente na Tabela A. Esse tipo de relacionamento é raro e pode exigir algumas alterações na estruturação do seu banco de dados.

Os relacionamentos um-para-um entre tabelas são raros porque, em muitos casos, as informações nas duas tabelas podem ser simplesmente combinadas em uma. Por exemplo, suponha que você tenha criado uma tabela, chamada Ping-Pong Players (Jogadores de pingue-pongue), para controlar as informações sobre um evento de pingue-pongue da Tasmanian Traders para levantamento de fundos. Como os jogadores de pingue-pongue são todos funcionários da Tasmanian Traders, essa tabela tem um relacionamento um-para-um com a tabela Employee no banco de dados da Tasmanian Traders.

**As tabelas Employee e Ping\_Pong\_Players representam um relacionamento um-para-um.**

Emp_id	Player_nickname	Preferred_date	Skill_level	Pledge
6	Ace	07/07/96	1	2.0000
7	King John	07/09/96	2	2.0000
8	Calmeister	07/07/96	1	2.0000
9	Slammin' Nan	07/07/96	1	2.0000

— Cada jogador de pingue-pongue possui um registro na tabela Employee.

Emp_id	Player_nickname	Preferred_date	Skill_level	Pledge
6	Ace	07/07/96	1	2.0000
7	King John	07/09/96	2	2.0000
8	Calmeister	07/07/96	1	2.0000
9	Slammin' Nan	07/07/96	1	2.0000

— Cada jogador de pingue-pongue possui um registro na tabela Employee.

Emp_id	Last_name	First_name	Title
6	Suyama	Michael	Representante comercial
7	King	Robert	Representante comercial
8	Callahan	Laura	Coordenador de vendas internas
9	Dodsworth	Anne	Representante comercial

— Este conjunto de valores é um subconjunto do campo Emp\_id na tabela Employee.

Você poderia adicionar todos os campos da tabela Ping\_Pong\_Players à tabela Employee. Mas a tabela Ping-Pong Players controla um evento de ocorrência única, e você não precisará dessas informações depois de ele ter-se encerrado. Além disso, nem todos os funcionários jogam pingue-pongue e, assim, se esses campos estivessem na tabela Employee, estariam vazios para vários registros. Por estes motivos, faz sentido criar uma tabela separada.

Quando você necessita um relacionamento [um-para-um](#) no seu banco de dados, considere se você pode reunir as informações em uma tabela. Por exemplo, na tabela Employee, um funcionário pode ter um gerente, que também é um funcionário. Você pode adicionar um campo para o número ID do gerente. Para reunir as informações depois, você pode utilizar auto-associação na sua [consulta](#) ou [visualização](#). Não é preciso uma tabela separada para resolver o relacionamento um-para-um. Se por algum motivo não quiser fazer isso, você pode configurar um relacionamento um-para-um entre duas tabelas desta maneira:

- Se as duas tabelas tiverem o mesmo assunto, provavelmente você só pode configurar o relacionamento utilizando o mesmo campo da [chave primária](#) em ambas as tabelas.
- Se as duas tabelas tiverem assuntos diferentes com chaves primárias diferentes, escolha uma das tabelas (qualquer uma) e coloque seu campo da chave primária na outra tabela como uma [chave estrangeira](#).

## Refinando a estruturação

Quando você tiver as tabelas, campos e relacionamentos necessários, é o momento de estudar sua estruturação e detectar ajustes que precisem ser feitos.

Você pode encontrar várias armadilhas ao criar o seu banco de dados. Esses problemas comuns podem dificultar a utilização ou manutenção dos seus dados:

- Você tem uma [tabela](#) com um grande número de [campos](#) que não se relacionam todos com o mesmo assunto? Por exemplo, uma tabela pode conter campos que pertencem aos seus clientes assim como campos com informações sobre vendas. Tente assegurar-se de que cada tabela contenha dados sobre apenas um assunto.
- Você tem [campos](#) que foram intencionalmente deixados em branco em vários [registros](#) por não se aplicarem a esses registros? Geralmente isso significa que os campos deveriam estar em outra [tabela](#).



- Você tem um número grande de [tabelas](#), muitas delas contendo os mesmos [campos](#)? Por exemplo, você tem tabelas separadas para as vendas de janeiro e fevereiro ou para clientes locais e clientes distantes, nas quais armazena o mesmo tipo de informações. Tente consolidar todas as informações relativas a um único assunto em uma tabela. Pode ser necessário adicionar mais um campo, por exemplo, para identificar a data da venda.

Crie suas tabelas, especifique os relacionamentos entre as tabelas e insira alguns registros de dados em cada tabela. Verifique se você pode utilizar o banco de dados para obter as respostas que deseja. Crie esboços iniciais dos seus formulários e relatórios e verifique se eles mostram os dados que você espera. Procure por duplicações de dados dispensáveis e as elimine.

À medida que experimenta o seu banco de dados inicial, é provável que descubra aperfeiçoamentos que podem ser feitos. Eis algumas coisas que você pode verificar:

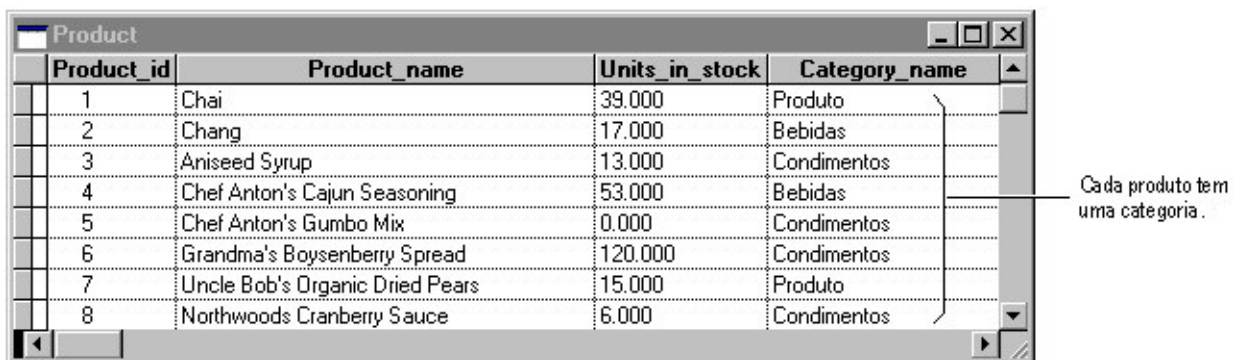
- Você esqueceu de algum [campo](#)? Há informações necessárias que não foram incluídas? Em caso afirmativo, elas pertencem às [tabelas](#) existentes? Se a informação for sobre alguma outra coisa, pode ser preciso criar outra tabela.
- Você escolheu uma boa [chave primária](#) para cada [tabela](#)? Se você a utilizar para procurar por [registros](#) específicos, elas são fáceis de lembrar e digitar? Assegure-se de que não precisará introduzir um valor em um campo da chave primária que duplique outro valor no campo.
- Você está introduzindo informações duplicadas repetidamente em uma das suas [tabelas](#)? Em caso afirmativo, é provável que você precise dividir a tabela em duas, com um relacionamento [um-para-n](#).
- Você tem [tabelas](#) com muitos [campos](#), um número limitado de [registros](#) e vários campos vazios em registros individuais? Em caso afirmativo, considere a reestruturação da sua tabela para que tenha menos campos e mais registros.

À medida que identificar as alterações que deseja fazer, você pode alterar suas tabelas e campos para que reflitam a estruturação aperfeiçoada. Para informações sobre a modificação de tabelas, consulte o capítulo 7, [Trabalhando com tabelas](#).

### Exemplo: Refinando a tabela Products

Cada produto no estoque da Tasmanian Traders fica sob uma categoria geral, como Beverages, Condiments ou Seafood (Bebidas, Condimentos ou Frutos do mar). A tabela Products poderia incluir um [campo](#) que mostre a categoria de cada produto.

#### Tabela Products com o campo Category\_name



Product_id	Product_name	Units_in_stock	Category_name
1	Chai	39.000	Produto
2	Chang	17.000	Bebidas
3	Aniseed Syrup	13.000	Condimentos
4	Chef Anton's Cajun Seasoning	53.000	Bebidas
5	Chef Anton's Gumbo Mix	0.000	Condimentos
6	Grandma's Boysenberry Spread	120.000	Condimentos
7	Uncle Bob's Organic Dried Pears	15.000	Produto
8	Northwoods Cranberry Sauce	6.000	Condimentos

Suponha que ao examinar e refinar o [banco de dados](#), a Tasmanian Traders decida armazenar uma descrição da categoria junto com seu nome. Se você adicionar um campo Category Description (Descrição da categoria) à tabela Products, deverá repetir cada descrição de categoria para cada produto que fique sob ela — o que não é uma boa solução.

Uma solução melhor seria transformar Category em um novo assunto para o banco de dados controlar, com sua própria tabela e sua própria [chave primária](#). Você pode, então, adicionar a chave primária da tabela Category à tabela Products como uma [chave estrangeira](#).

A tabela **Category** fornece um local para armazenar as informações de categoria com eficiência.

Category					
Category_id	Name	Description	Picture_file	Picture	
1	Bebidas	Memo	Memo	Ger	
2	Condimentos	Memo	Memo	Ger	
3	Confeções	Memo	Memo	Ger	
4	Produtos do dia-a-dia	Memo	Memo	Ger	
5	Grãos/Cereais	Memo	Memo	Ger	
6	Carnes/Áves	Memo	Memo	Ger	
7					
8					

Products				
Product_id	Product_name	English_name	Category_id	
1	Chai	Chá Dharamsala	7	
2	Chang	Cerveja tibetana Barley	1	
3	Aniseed Syrup	Licor	2	
4	Chef Anton's Cajun Seasoning	Condimentos Cajun Chef Anton's	1	
5	Chef Anton's Gumbo Mix	Sopa de quiabo com galinha Chef Anton's	2	
6	Grandma's Boysenberry Spread	Pasta de amora Grandma's	2	
7	Uncle Bob's Organic Dried Pears	Peras secas orgânicas Uncle Bob's	7	
8	Northwoods Cranberry Sauce	Calda de uva-do-monte Northwoods	2	
9	Mishi Kobe Niku	Carne do Japão	6	

Chave primária

Chave estrangeira

As tabelas **Category** e **Products** têm um relacionamento [um-para-n](#): uma categoria pode ter mais de um produto nela, mas qualquer produto individual pode pertencer a apenas uma categoria.

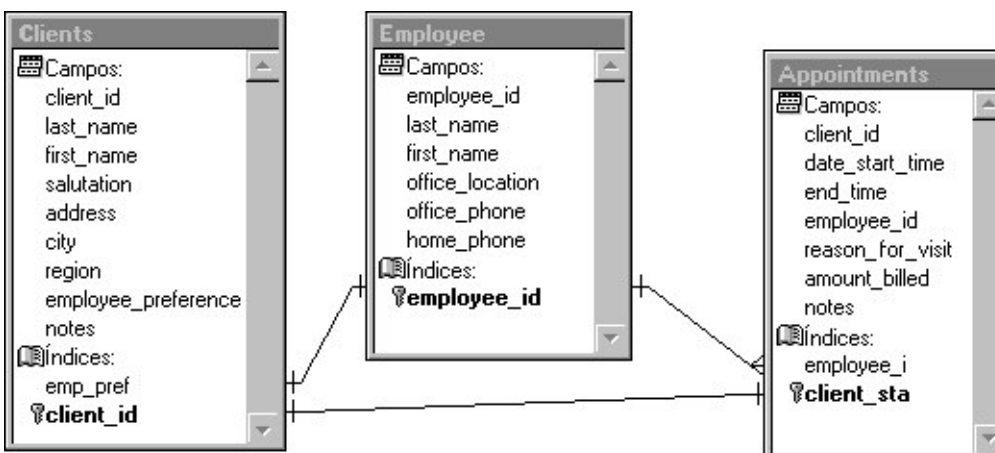
## Diagramas de bancos de dados de exemplo

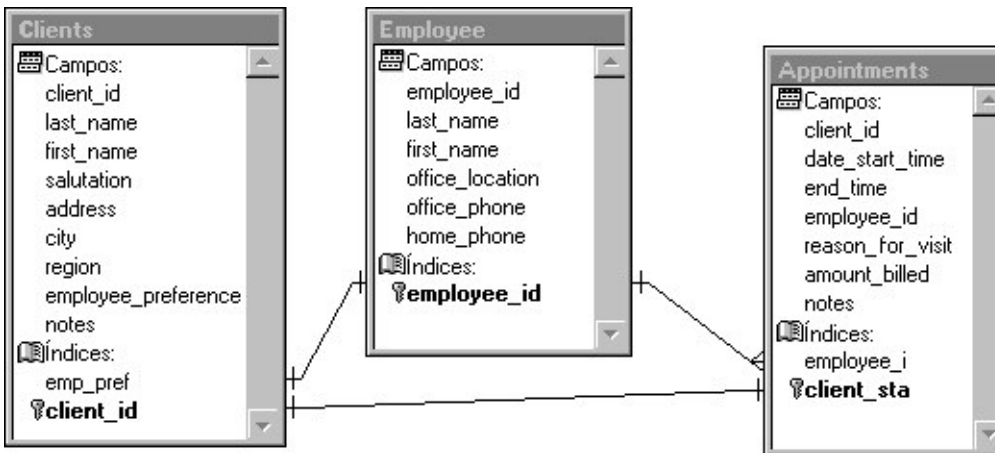
Os diagramas de bancos de dados nesta seção podem proporcionar idéias para a estruturação do seu próprio banco de dados. Estes não estão incluídos no Visual FoxPro; estão incluídos aqui como exemplos dos tipos de [bancos de dados](#) e [tabelas](#) que você pode criar.

### Banco de dados de compromissos

Esta estrutura de banco de dados armazena compromissos para um escritório profissional e pode ser facilmente modificado para utilização em consultórios de médicos e dentistas ou em escritórios de advogados ou contadores. A tabela **Appointments** (Compromissos) tem uma [chave primária](#) de múltiplos campos para identificar com exclusividade cada compromisso. Esta chave primária, o índice "cliente\_ini", é criada indexando-se uma [expressão](#) que combina os campos `client_id` e `date_start time` (`cliente_id` e `data_início`).

#### Exemplo de um banco de dados de compromissos

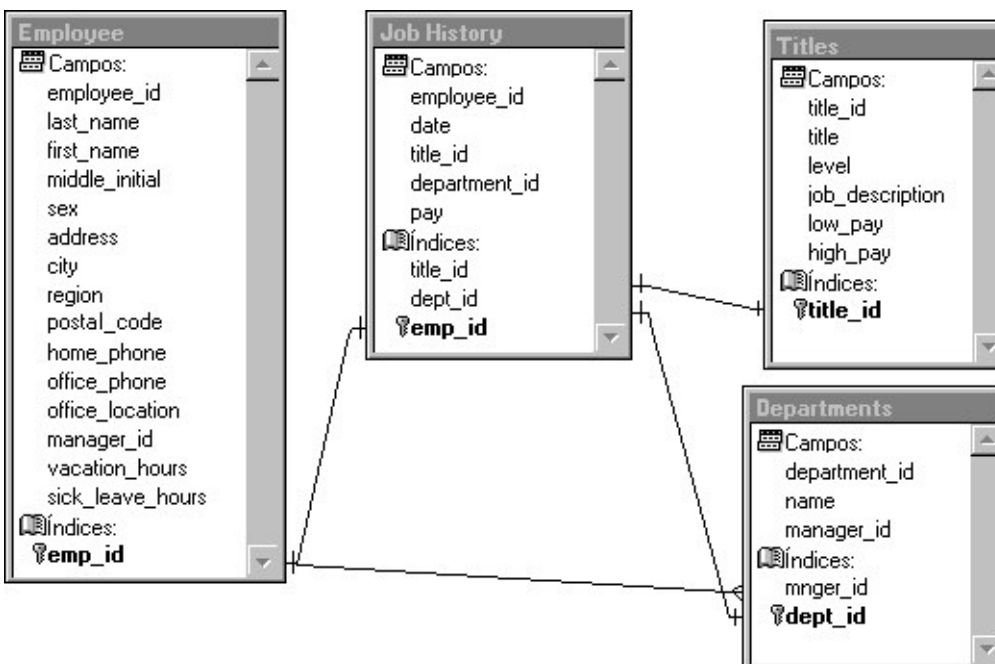




## Banco de dados do pessoal

Este banco de dados armazena as informações sobre recursos humanos. A tabela Job History (Histórico de cargos) armazena as informações sobre cada admissão ou promoção, podendo ter vários registros para cada funcionário.

### Exemplo de um banco de dados de pessoal



## Banco de dados de biblioteca

Este banco de dados armazena as informações sobre os livros da biblioteca e as retiradas dos sócios. Observe os relacionamentos [n-para-n](#) entre as tabelas Books e Authors (Livros e Autores) e entre as tabelas Books e Subjects (Livros e Assuntos).

### Exemplo de um banco de dados de biblioteca

