Após ter projetado o banco de dados, você pode criá-lo utilizando a interface ou com a linguagem. É possível que você queira adicionar tabelas existentes ao banco de dados e depois modificá-las para aproveitar os recursos do dicionário de dados do Visual FoxPro. Se você estiver trabalhando em um projeto no **Gerenciador de projetos**, poderá adicionar tabelas à medida que são criadas.

Para obter maiores informações sobre a criação de um banco de dados para um ambiente de vários usuários, consulte o capítulo 17, Programando para acesso compartilhado

Este capítulo aborda os seguintes tópicos:

- Criando um banco de dados
- Visualizando e modificando a arquitetura dos bancos de dados
- Gerenciando um banco de dados
- Fazendo referência a vários bancos de dados
- Gerenciando erros de banco de dados

Criando um banco de dados

Quando você cria um banco de dados, as tabelas são reunidas em um conjunto e ganham os benefícios dos recursos do dicionário de dados.

Um dicionário de dados possibilita uma maior flexibilidade no projeto e modificação do banco de dados, além de poupar você da necessidade de escrever um código para criar validações de campos e de linhas, ou para assegurar que os valores dentro dos campos de chave primária sejam exclusivos. O dicionário de dados do Visual FoxPro permite que você crie ou especifique:

- Chaves primárias e candidatas.
- Relacionamentos permanentes entre tabelas de banco de dados.
- Nomes extensos para tabelas e campos.
- Legendas para campos que são exibidas como cabeçalhos em janelas **Pesquisar** e colunas de Grade.
- Valores padrão para os campos.
- A classe de controle padrão utilizada em formulários.
- Máscaras de entrada e formatos de exibição para os campos.
- Regras de campos e de registros.
- Disparadores.
- Procedimentos armazenados.
- Conexões a fontes de dados remotas.
- Visualizações locais e remotas.
- Comentários para cada campo, tabela, e banco de dados.

Alguns dos recursos do dicionário de dados, como nomes de campos extensos, chaves primárias e candidatas, valores padrão, regras de campos e de registros, além dos disparadores, são armazenados no arquivo .DBC, porém são criados como parte do processo de construção de uma tabela ou visualização. Para obter informações sobre esses recursos, consulte o capítulo 7, Trabalhando com tabelas, e o capítulo 8, Criando visualizações

Reunindo tabelas em um banco de dados

Para reunir tabelas em um banco de dados, é necessário que você crie um recipiente de bancos de dados para armazenar todos os objetos como visualizações, conexões e procedimentos armazenados associados às tabelas que formam o seu banco de dados.

Para criar um novo banco de dados

- No **Gerenciador de projetos**, selecione a guia **Dados**, depois **Bancos de dados** na lista e, em seguida, escolha **Novo**.

  – Ou –

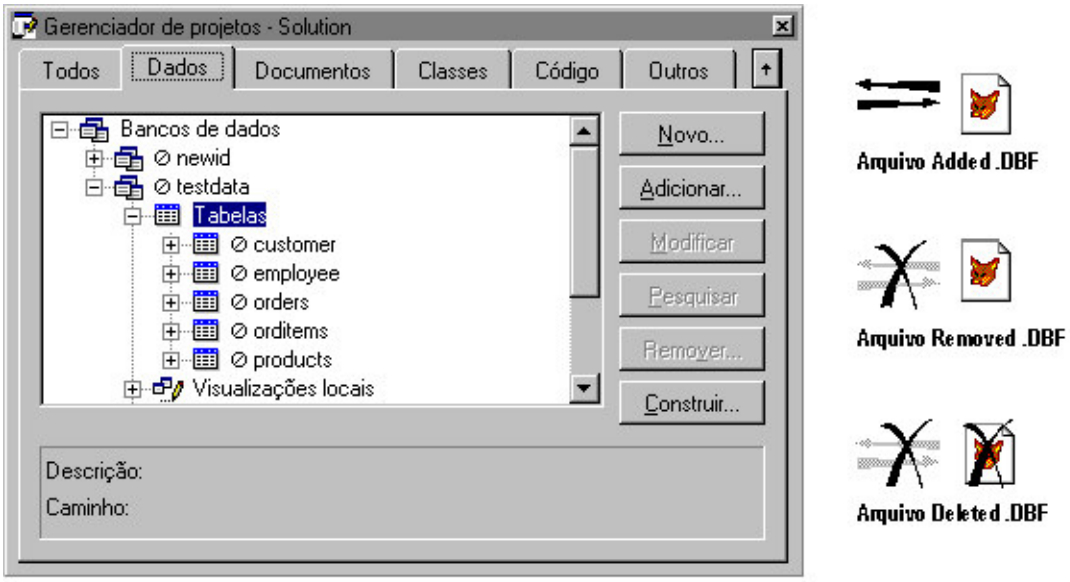
- Utilize o comando CREATE DATABASE .

Por exemplo, o código a seguir cria e abre de forma exclusiva um novo banco de dados, chamado `sample`:

```
CREATE DATABASE sample
```

Ao criar um novo banco de dados, ele estará vazio, sem nenhuma tabela ou qualquer outro objeto associado a ele. Quando se adiciona uma tabela, são criados vínculos entre o arquivo desta tabela e o recipiente de bancos de dados. As informações sobre uma tabela de vínculo armazenadas no banco de dados consistem em um vínculo prospectivo. As informações sobre o recipiente de bancos de dados de vínculo armazenadas na tabela representam o vínculo retroativo.

Os vínculos especificam as associações entre um recipiente de bancos de dados e as tabelas.



É possível usar os comandos e funções a seguir para trabalhar com um banco de dados e seus objetos utilizando a linguagem de programação.

Comandos e funções de manipulação de bancos de dados e de seus objetos

ADATABASES( )	CREATE VIEW	MODIFY CONNECTION
ADBOBJECTS( )	DBC( )	MODIFY DATABASE
ADD TABLE	DBGETPROP( )	MODIFY PROCEDURE
ALTER TABLE	DBSETPROP( )	MODIFY STRUCTURE
APPEND PROCEDURES	DELETE CONNECTION	MODIFY VIEW
CLOSE DATABASE	DELETE DATABASE	OPEN DATABASE
COPY PROCEDURES	DELETE VIEW	PACK DATABASE
CREATE CONNECTION	DISPLAY DATABASE	RENAME TABLE

CREATE DATABASE	DROP TABLE	REMOVE TABLE
CREATE SQL VIEW	INDBC( )	SET DATABASE
CREATE TABLE	LIST DATABASE	VALIDATE DATABASE

Adicionando tabelas a um banco de dados

Cada tabela do Visual FoxPro pode aparecer em duas formas: seja como uma *tabela livre,* que é um arquivo .DBF não associado a um banco de dados, ou como uma *tabela de banco de dados,* que é um arquivo .DBF associado a um banco de dados. As tabelas associadas a um banco de dados podem ter propriedades que as tabelas fora de um banco de dados não têm, como regras de campos e de registros, disparadores e relacionamentos permanentes.

Para associar tabelas a um banco de dados, basta criá-las dentro de um banco de dados que esteja aberto ou acrescentar tabelas já existentes a um banco de dados. Para obter informações sobre como criar novas tabelas, consulte o capítulo 7, Trabalhando com tabelas

Para adicionar uma nova tabela livre a um banco de dados

- No **Gerenciador de projetos**, selecione **Tabelas livres** na guia **Todos** ou na guia **Dados** e, em seguida, escolha **Adicionar**.

  – Ou –

- Utilize o comando ADD TABLE.

Por exemplo, o código abaixo abre o banco de dados testdata e adiciona a tabela `orditems`:

```
OPEN DATABASE testdata
ADD TABLE orditems
```

Você deve adicionar explicitamente uma tabela livre já existente a um banco de dados para que ela se torne parte dele. Modificar a estrutura de uma tabela livre não faz com que o Visual FoxPro adicione a tabela livre a um banco de dados, mesmo que este esteja aberto quando o comando MODIFY STRUCTURE for emitido.

Utilizando tabelas livres

Só é possível associar cada tabela a apenas um banco de dados. Contudo, você pode utilizar os dados de um arquivo .DBF sem incorporá-lo a um banco de dados.

Para acessar uma tabela em um outro banco de dados

- Crie uma visualização em um banco de dados que faça referência à tabela.

  – Ou –

- Acesse a tabela ao comando **USE** e o símbolo"!".

Utilize o símbolo "!" para fazer referência a uma tabela em um banco de dados que não seja o atual. Por exemplo, se quiser pesquisar a tabela `orditems` no banco de dados `testdata`, você pode digitar:

```
USE testdata!orditems
BROWSE
```

No exemplo acima, o banco de dados `testdata` será aberto automaticamente quando você emitir o comando USE, porém o Visual FoxPro não define `testdata` como o banco de dados atual. Um banco de dados aberto automaticamente, como no exemplo acima, será fechado automaticamente quando a tabela for fechada, a menos que, antes de fechá-la, você abra o banco de dados de forma explícita.

Para obter informações sobre como usar uma visualização para acessar informações externas a seu banco de dados, consulte o capítulo 8, Criando visualizações

Removendo uma tabela de um banco de dados

Quando você adiciona uma tabela a um banco de dados, o Visual FoxPro modifica o registro de cabeçalho do arquivo da tabela para documentar o caminho e o nome de arquivo do banco de dados que possui a tabela. Esta informação de caminho e nome de arquivo é chamada de *vínculo retroativo,* já que vincula a tabela de volta ao banco de dados que é seu proprietário. Remover uma tabela de um banco de dados faz com que não só ela e suas informações do dicionário de dados sejam retiradas do arquivo de banco de dados , mas também atualiza as informações do vínculo retroativo para refletir o novo status da tabela como livre.

É possível remover uma tabela de um banco de dados através da interface ou utilizando o comando REMOVE TABLE. Quando se faz isso, pode-se também optar por apagar fisicamente do disco o arquivo da tabela.

Para remover uma tabela de um banco de dados

- No **Gerenciador de projetos**, selecione o nome da tabela e escolha **Remover**.

  – Ou –

- No **Criador de bancos de dados**, selecione a tabela e escolha **Remover** no menu **Bancos de dados**.

  – Ou –

- Utilize o comando REMOVE TABLE.

Por exemplo, o código a seguir abre o banco de dados `testdata` e remove a tabela `orditems`:

```
OPEN DATABASE testdata
REMOVE TABLE orditems
```

Remover uma tabela de um banco de dados não exclui o arquivo da tabela automaticamente. Se você desejar remover a tabela do banco de dados e também apagar o arquivo .DBF do disco, utilize a cláusula DELETE do comando REMOVE TABLE ou o comando DROP TABLE. Por exemplo, o código a seguir abre o banco de dados `testdata` e exclui a tabela `orditems` do disco:

```
OPEN DATABASE testdata
REMOVE TABLE orditems DELETE
```

O código a seguir também abre o banco de dados `testdata` e, em seguida, exclui a tabela `orditems` sem deslocar uma cópia para a Lixeira do Windows:

```
OPEN DATABASE testdata
DROP TABLE orditems NORECYCLE
```

Atualizando vínculos de tabela e de banco de dados

Se você mover arquivos de bancos de dados (.DBC, .DCT e .DCX), ou uma tabela associada a um banco de dados, os caminhos relativos mudam e podem quebrar os vínculos retroativos e prospectivos utilizados pelo Visual FoxPro para associar arquivos de bancos de dados e de tabela:

- O vínculo retroativo vincula a tabela de volta para o banco de dados que possui a tabela. Consiste no nome do arquivo e seu caminho relativo .DBC associado à tabela. É armazenado no cabeçalho do arquivo (.DBF) de tabela do Visual FoxPro.
- O *vínculo prospectivo* informa ao banco de dados as tabelas que pertencem a ele. Os vínculos prospectivos são armazenados no arquivo de banco de dados (.DBC) e consistem no nome de arquivo e seu caminho relativo para cada arquivo de tabela associado.

Você pode restabelecer vínculos e atualizar informações de caminho relativo que revelem a nova localização do arquivo.

Para atualizar vínculos depois de mover uma tabela ou um banco de dados

- Utilize a cláusula RECOVER do comando VALIDATE DATABASE .

Por exemplo, o código a seguir abre o banco de dados `testdata` e exibe caixas de diálogo que permitem a localização de tabelas que não estejam nas localizações contidas no banco de dados.

```
OPEN DATABASE testdata
VALIDATE DATABASE RECOVER
```

Dica Se você deseja utilizar uma tabela poupando o tempo de restabelecer os vínculos de todas as tabelas no banco de dados, poderá abri-la com o comando USE. O Visual FoxPro exibe a caixa de diálogo Abrir , permitindo que você localize o banco de dados proprietário ou exclua os vínculos.

Para obter informações sobre como remover o vínculo retroativo de uma tabela cujo banco de dados proprietário foi excluído acidentalmente do disco, consulte FREE TABLE.

Criando relacionamentos permanentes

É possível criar relacionamentos permanentes entre tabelas de um banco de dados. Tratam-se de relacionamentos entre tabelas de banco de dados armazenados no arquivo de banco de dados e são:

- Utilizados automaticamente como condições padrão de associação nos **Criadores de consultas e visualizações**.
- Exibidos no **Criador de bancos de dados** como linhas relacionando índices de tabelas.
- Exibidos no **Criador de ambiente de dados** como relacionamentos padrão para formulários e relatórios.
- Utilizados para armazenar informações sobre integridade referencial.

Ao contrário dos relacionamentos temporários criados com o comando SET RELATION, os relacionamentos permanentes não precisam ser recriados a cada vez que as tabelas são utilizadas. Contudo, uma vez que os relacionamentos permanentes não controlam o relacionamento entre os ponteiros de registro nas tabelas, você irá utilizar tanto os relacionamentos temporários definidos com SET RELATION quanto os permanentes, ao desenvolver aplicativos do Visual FoxPro.

No Visual FoxPro, os índices são utilizados para estabelecer relacionamentos permanentes entre tabelas de um banco de dados. Desta forma, você cria um relacionamento permanente entre índices, e não entre campos, possibilitando que as tabelas sejam relacionadas com base em uma expressão de índice simples ou complexa.

Para criar relacionamentos permanentes entre tabelas

- No **Criador de bancos de dados**, escolha o nome do índice que você deseja relacionar e arraste-o até o nome do índice da tabela relacionada.

  – Ou –

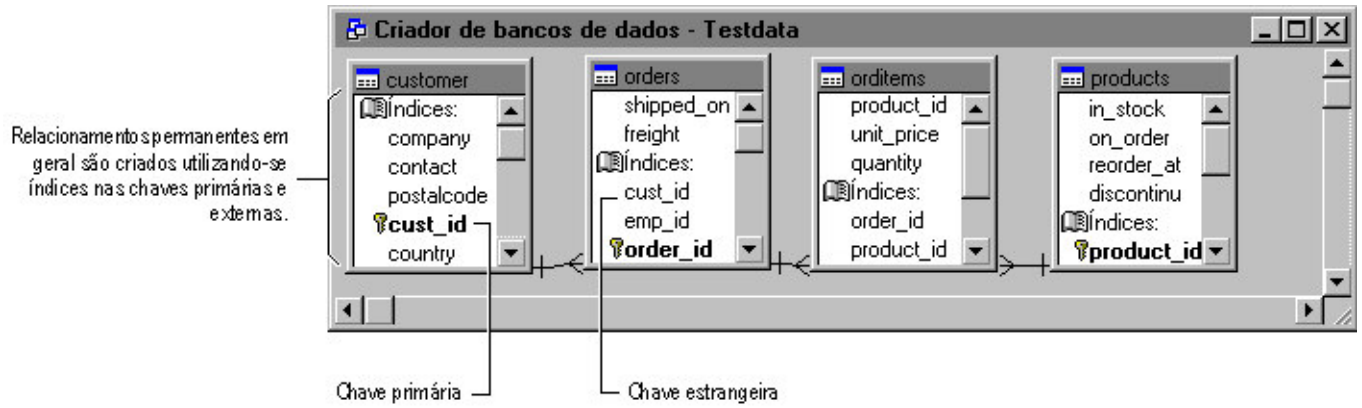
- Utilize a cláusula FOREIGN KEY com os comandos CREATE TABLE ou ALTER TABLE.

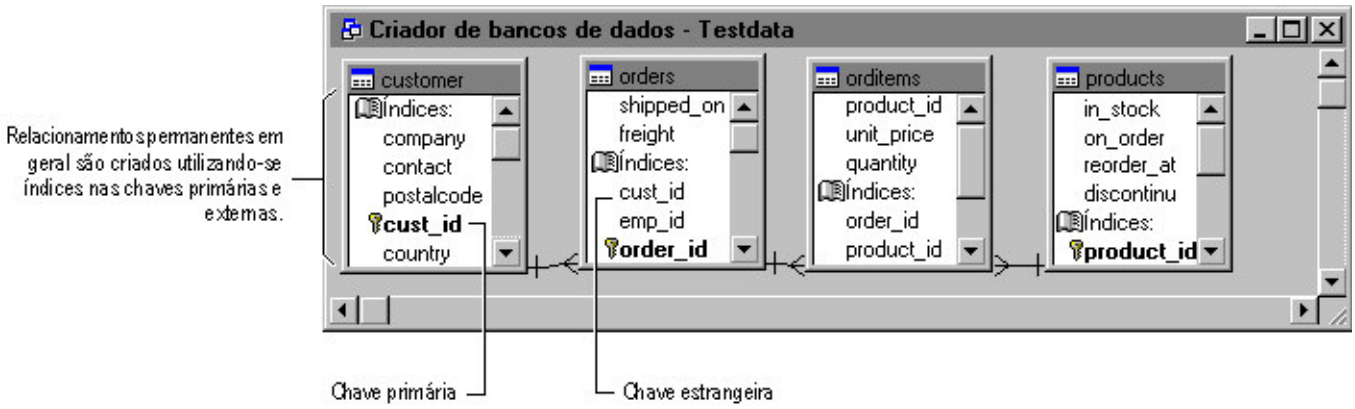
Por exemplo, o comando a seguir adiciona um relacionamento permanente um-para-n entre as tabelas `customer` e `orders`, com base na chave primária `cust_id` da tabela `customer` e uma nova chave estrangeira, `cust_id`, da tabela `orders`:

```
ALTER TABLE orders;
   ADD FOREIGN KEY cust_id TAG ;
      cust_id REFERENCES customer
```

Se examinar o esquema do banco de dados no **Criador de bancos de dados**, você verá uma linha associando `orders` e `customer`, representando o novo relacionamento permanente.

Índices fornecem a base para relacionamentos permanentes





O tipo de marca ou chave de índice determina o tipo de relacionamento permanente que pode ser criado. Você deve utilizar uma marca ou chave de índice primária ou candidata para o lado “um” de um relacionamento um-para-n; para o lado ‘n’, utilize uma marca ou chave de índice comum. Para obter maiores informações sobre tipos e criação de índices, consulte o capítulo 7, Trabalhando com tabelas

Para excluir um relacionamento permanente entre tabelas

1. No **Criador de bancos de dados**, clique sobre a linha de relacionamento entre as duas tabelas.

   A largura da linha de relacionamento aumenta para indicar que o relacionamento foi selecionado.
2. Pressione a tecla DELETE .

   – Ou –

   Utilize a cláusula DROP FOREIGN KEY com o comando ALTER TABLE .

Por exemplo, o comando a seguir exclui um relacionamento permanente entre as tabelas `customer` e `orders`, com base na chave primária `cust_id` da tabela `customer` e uma chave estrangeira, `cust_id`, da tabela `orders`:

```
ALTER TABLE orders DROP FOREIGN KEY TAG cust_id SAVE
```

Criando integridade referencial

O estabelecimento de integridade referencial envolve a criação de um conjunto de regras para preservar os relacionamentos definidos entre tabelas quando você insere ou exclui registros.

Caso aplique a integridade referencial, o Visual FoxPro irá impedir que você:

- Acrescente registros a uma tabela relacionada quando não houver registros associados na tabela principal.
- Modifique valores em uma tabela principal que resultem em registros órfãos em uma tabela relacionada.
- Exclua registros de uma tabela principal quando houver registros relacionados correspondentes.

É possível escrever seus próprios disparadores e código de procedimentos armazenados para executar a integridade referencial. Contudo o **Construtor de integridade referencial** (RI) do Visual FoxPro permite a você determinar os tipos de regras que deseja executar, as tabelas onde deseja aplicar as regras e os eventos do sistema que farão com que o Visual FoxPro verifique as regras de integridade referencial.

O **Construtor de integridade referencial** pode lidar com níveis múltiplos de cascateamento para exclusões e atualizações em cascata e é recomendado como ferramenta para a criação de integridade referencial.

Para abrir o Construtor de integridade referencial

1. Abra o **Criador de bancos de dados**.

2 No menu **Banco de dados**, selecione **Editar integridade referencial**.

Quando você utiliza o **Construtor de integridade referencial** para criar regras para seu banco de dados, o Visual FoxPro, salva o código gerado para executar as regras de integridade relacional como disparadores que fazem referência a procedimentos armazenados. Para visualizar este código, abra o editor de texto de procedimentos armazenados para seu banco de dados. Para obter informações sobre como criar disparadores utilizando a linguagem de programação, consulte "Utilizando disparadores" no capítulo 7, Trabalhando com tabelas

---

Cuidado Ao alterar o projeto de um banco de dados, por exemplo, mudando as tabelas do banco de dados ou alterando os índices utilizados em um relacionamento permanente, você deve aplicar novamente o **Construtor de integridade referencial** antes de utilizar o banco de dados. Isto fará com que o código dos procedimentos armazenados e os disparadores de tabela utilizados para aplicar a integridade referencial, passem a refletir o novo projeto. Caso você não execute o **Construtor de integridade referencial** novamente, poderão ocorrer resultados inesperados, pois os procedimentos armazenados e os disparadores não refletirão as mudanças efetuadas.

Criando procedimentos armazenados

É possível criar procedimentos armazenados para as tabelas do banco de dados. Um *procedimento armazenado* é um código do Visual FoxPro armazenado no arquivo .DBC. Tratam-se de procedimentos de código que operam especificamente nos dados do banco de dados. Eles podem melhorar o desempenho porque são carregados na memória quando um banco de dados é aberto.

Para criar, modificar ou remover um procedimento armazenado

- No **Gerenciador de projetos**, selecione um banco de dados e, em seguida, **Procedimentos armazenados**. Depois escolha **Novo**, **Modificar** ou **Remover**.

  – Ou –

- No **Criador de bancos de dados**, escolha **Editar procedimentos armazenados** no menu **Banco de dados**.

  – Ou –

- Na janela **Comando**, use o comando MODIFY PROCEDURE.

Cada uma destas opções abre o editor de texto do Visual FoxPro para que você crie, modifique ou remova procedimentos armazenados no banco de dados atual.

Os procedimentos armazenados são utilizados basicamente para criar funções definidas pelo usuário às quais você faz referência em regras de validação de campos e registros. Quando você salvar uma função definida pelo usuário como um procedimento armazenado em seu banco de dados, o código para essa função será salvo no arquivo.DBC e acompanhará automaticamente o banco de dados quando este for transportado. O uso de procedimentos armazenados também torna seu aplicativo mais portátil, porque você não precisa gerenciar arquivos de funções definidas pelo usuário separadamente do arquivo de banco de dados.

Visualizando e definindo propriedades de bancos de dados

Cada banco de dados do Visual FoxPro contém as propriedades Comment e Version. É possível visualizar e definir estas propriedades utilizando as funções DBGETPROP( ) e DBSETPROP( ).

Por exemplo, o código a seguir exibe o número da versão do banco de dados `testdata`:

```
? DBGETPROP('testdata', 'database', 'version')
```

O valor retornado representa o número da versão do .DBC do Visual FoxPro, sendo somente para leitura. Utilizando a mesma função, você poderá visualizar o comentário, caso exista, no banco de dados:

```
? DBGETPROP('testdata', 'database', 'comment')
```

Ao contrário da propriedade Version, a propriedade Comment pode ser definida. Utilize a função

DBSETPROP( ) para digitar uma descrição ou outro texto que você deseja armazenar com o banco de dados.

Para definir a propriedade Comment do banco de dados atual

- No **Criador de bancos de dados**, escolha **Propriedades** no menu **Banco de dados** e digite um comentário na caixa **Comentário**.

  – Ou –

- Utilize a opção Comment da função DBSETPROP( ).

Por exemplo, o código a seguir altera o comentário do banco de dados `testdata`:

```
? DBSETPROP('testdata', 'database', 'comment', ;
    'TestData é incluído com o Visual FoxPro')
```

Também é possível usar as funções DBGETPROP( ) e DBSETPROP( ) para visualizar e definir propriedades de outros objetos do banco de dados como Conexões e Visualizações.

Visualizando e modificando a arquitetura dos bancos de dados

Quando um banco de dados é criado, o Visual FoxPro cria e abre de forma exclusiva um arquivo. Recipiente de bancos de dados (DBC, *DataBase Container*). O arquivo .DBC armazena todas as informações sobre o banco de dados, incluindo os nomes de arquivos e objetos associados a ele. O arquivo .DBC não contém fisicamente nenhum objeto de nível superior, como tabelas ou campos. Em vez disso, o Visual FoxPro armazena no arquivo .DBC ponteiros de caminho de arquivo para as tabelas.

Para examinar a arquitetura do seu banco de dados, você pode pesquisar o arquivo de banco de dados, visualizar o esquema, pesquisar seus objetos, validar o banco de dados e até mesmo estender o arquivo .DBC.

Visualizando o esquema do banco de dados

O *esquema* do banco de dados é uma representação visual das estruturas das tabelas e dos relacionamentos permanentes definidos no banco de dados. A janela **Criador de bancos de dados** exibe o esquema do banco de dados aberto.

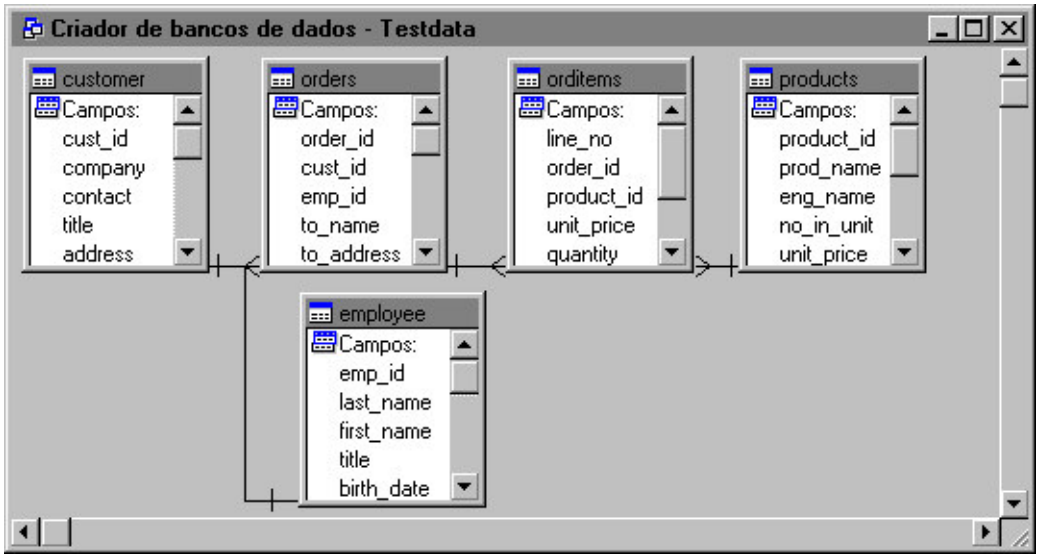
Para visualizar o esquema do banco de dados

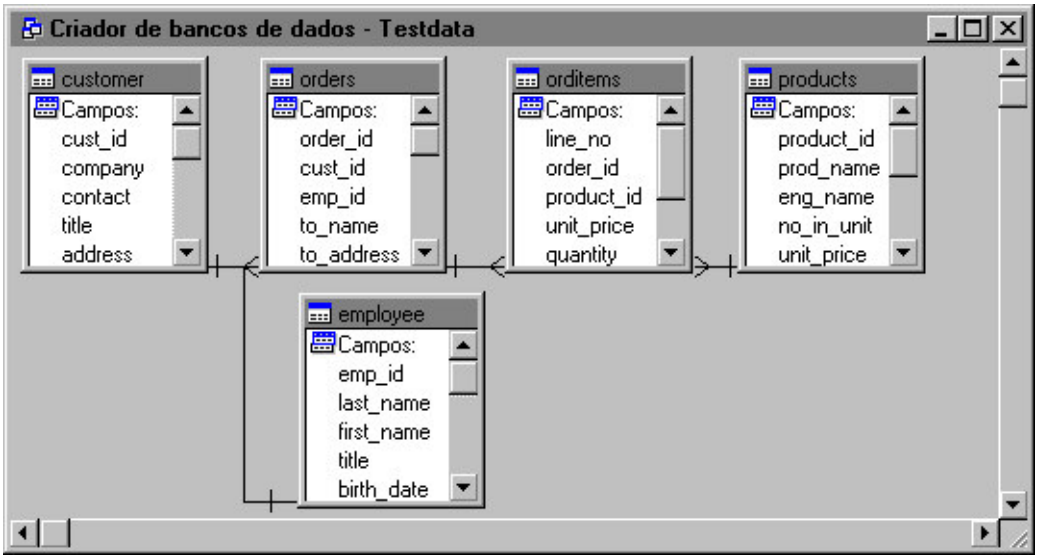
- Utilize o comando MODIFY DATABASE .

Por exemplo, o código a seguir abre o banco de dados `testdata` e exibe o esquema no **Criador de bancos de dados**:

```
MODIFY DATABASE testdata
```

O esquema de um banco de dados é uma representação dos objetos em um banco de dados.





No **Criador de bancos de dados**, é possível utilizar a **Barra de ferramentas banco de dados** para criar uma nova tabela, adicionar uma tabela já existente do banco de dados, remover uma tabela do banco de dados ou modificar a estrutura de uma tabela. Também é possível criar conexões e editar procedimentos armazenados.

Pesquisando o arquivo de banco de dados

O arquivo de banco de dados contém um registro para cada tabela, visualização, índice, marca de índice, relacionamento permanente e conexão associados ao banco de dados, assim como para cada campo da tabela ou de visualização que possuir propriedades estendidas. Inclui ainda um registro único contendo todos procedimentos armazenados no banco de dados. Para obter informações sobre a estrutura do arquivo .DBC, procure o tópico “Estruturas de arquivos” na Ajuda.

Ainda que o **Criador de bancos de dados** forneça uma representação conceitual do esquema do banco de dados, algumas vezes você precisa pesquisar o próprio conteúdo do arquivo de banco de dados. É possível pesquisar um banco de dados fechado emitindo o comando USE para o arquivo .DBC. O exemplo a seguir abre uma janela **Pesquisar** com o conteúdo do banco de dados `sales` em formato de tabela.

```
CLOSE DATABASE sales
USE sales.dbc EXCLUSIVE
BROWSE
```

Cuidado Não utilize o comando BROWSE para alterar o arquivo de banco de dados a menos que você conheça a estrutura do arquivo .DBC. Caso cometa um erro tentando alterar o arquivo .DBC, você poderá invalidar o banco de dados e até mesmo perder dados.

Estendendo arquivos de banco de dados

Cada arquivo .DBC contém um campo Memo, denominado User, que você pode usar para armazenar suas próprias informações sobre cada registro do banco de dados. É possível também estender um arquivo .DBC para adicionar campos a fim de dar conta de suas necessidades como desenvolvedor. Os campos devem ser acrescentados ao final da estrutura. Para modificá-la, você deve ter acesso exclusivo a um arquivo .DBC.

Para adicionar um campo a um arquivo .DBC

1. Abra o arquivo .DBC para uso exclusivo utilizando o comando USE.
2. Utilize o comando MODIFY STRUCTURE.

Por exemplo, o código a seguir abre o **Criador de tabelas** para que seja possível adicionar um campo à estrutura de TESTDATA.DBC:

```
USE TESTDATA.DBC EXCLUSIVE
MODIFY STRUCTURE
```

Quando você adicionar um novo campo a um arquivo de banco de dados, inicie com “U” o nome do campo para designá-lo como definido pelo usuário. Esta designação impedirá que ele entre em conflito com eventuais extensões futuras dos arquivos .DBC.

Cuidado Não altere nenhum dos campos definidos pelo Visual FoxPro em um arquivo .DBC. Qualquer mudança em um arquivo .DBC poderá afetar a integridade de seu banco de dados.

Validando um banco de dados

Quando se torna válido um banco de dados, isso assegura que suas linhas estão armazenando representações corretas dos meta-dados no banco de dados. É possível verificar a integridade do banco de dados atual com o comando VALIDATE DATABASE.

Para validar um banco de dados

- Utilize o comando VALIDATE DATABASE .

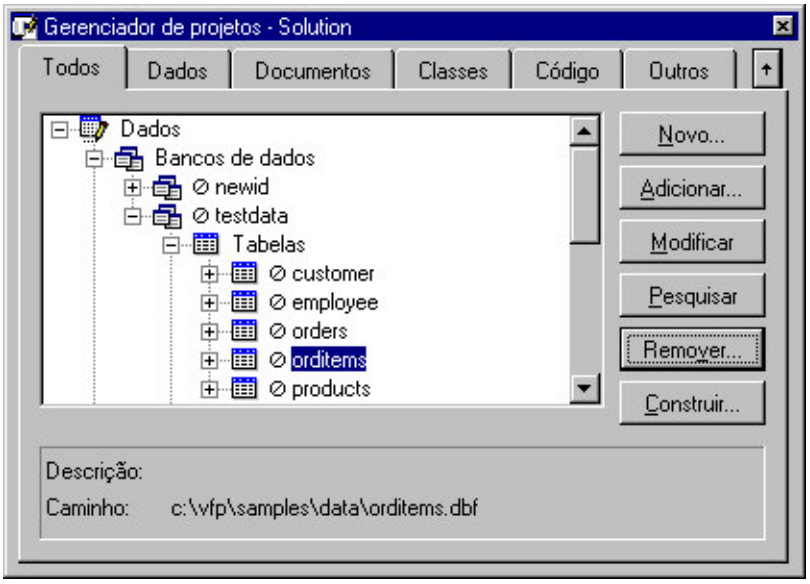
Por exemplo, o código a seguir utiliza e valida o arquivo .DBC para o banco de dados `testdata`:

```
OPEN DATABASE testdata EXCLUSIVE
VALIDATE DATABASE
```

Gerenciando um banco de dados

Após criar um banco de dados, é possível que você queira adicioná-lo a um projeto caso ele ainda não faça parte de um. Se o banco de dados já fizer parte de um projeto, poderá removê-lo dele. Além disso, se ele não for mais útil, você poderá excluí-lo do disco.

Um banco de dados no Gerenciador de projetos



Adicionando um banco de dados a um projeto

Quando um banco de dados é criado com o comando CREATE DATABASE , ele não passa automaticamente a fazer parte de um projeto, mesmo que o **Gerenciador de projetos** esteja aberto. É possível adicionar o banco de dados a um projeto para facilitar a organização, visualização e manipulação de objetos do banco de dados através da interface, além de simplificar o processo de

construção de um aplicativo. Você pode adicionar um banco de dados a um projeto somente através do **Gerenciador de projetos**.

Para adicionar um banco de dados a um projeto

- No **Gerenciador de projetos**, selecione **Bancos de dados** e escolha **Adicionar**.

Removendo um banco de dados de um projeto

Só é possível remover um banco de dados de um projeto através do **Gerenciador de projetos.**

Para remover um banco de dados de um projeto

- No **Gerenciador de projetos,** selecione o banco de dados e escolha **Remover**, depois escolha **Remover** novamente.

Excluindo um banco de dados

É possível excluir um banco de dados do disco utilizando o **Gerenciador de projetos** ou o comando DELETE DATABASE.

Para excluir um banco de dados

- No **Gerenciador de projetos,** selecione o banco de dados e escolha **Remover** e, em seguida, escolha **Excluir**.

  – Ou –

- Utilize o comando DELETE DATABASE.

Por exemplo, o código a seguir exclui o banco de dados sample:

```
DELETE DATABASE sample
```

Utilize sempre um dos métodos mencionados para excluir um banco de dados do disco. Quando o **Gerenciador de projetos** ou o comando DELETE DATABASE são utilizados, o Visual FoxPro pode remover das tabelas os vínculos retroativos para o banco de dados. Se você lançar mão de outro utilitário de manipulação de arquivos para excluir um arquivo de banco de dados, como o Windows Explorer, esses vínculos retroativos não serão removidos.

Observação O comando DELETE DATABASE não exclui do disco as tabelas associadas a um banco de dados; em vez disso, essas tabelas se tornam tabelas livres. Se você desejar excluir do disco um banco de dados e também todas suas tabelas associadas, utilize a cláusula DELETETABLES com o comando DELETE DATABASE.

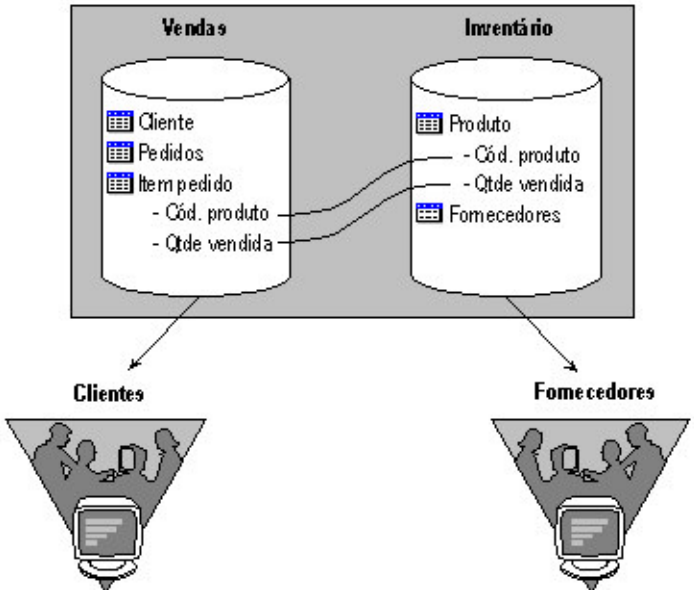
Fazendo referência a vários bancos de dados

Você pode ter diversos bancos de dados do Visual FoxPro em seu sistema para atender às necessidades organizacionais em um ambiente com muitos usuários. Os vários bancos de dados oferecem as seguintes vantagens:

- Controle do acesso do usuário a um subconjunto de tabelas no sistema geral.
- Organização dos dados para atender com melhor eficiência às necessidades de informação dos diversos grupos que utilizam o sistema.
- Permissão do uso exclusivo de um subconjunto de tabelas para a criação de visualizações locais e remotas em tempo de execução.

Por exemplo, é possível que você tenha um banco de dados que mantenha informações de vendas utilizadas principalmente pela equipe de vendas que trabalha com os clientes, e outro banco de dados com informações de inventário utilizadas principalmente pelos compradores que trabalham com os fornecedores. Às vezes, as necessidades de informações desses grupos ficarão sobrepostas. Esses bancos de dados podem ser abertos ao mesmo tempo e acessados à vontade, mas contêm tipos de informações completamente diferentes.

Vários bancos de dados podem adicionar flexibilidade ao seu sistema



É possível utilizar mais de um banco de dados abrindo mais de um deles simultaneamente ou fazendo referência a tabelas de um banco de dados fechado. Após ter aberto diversos bancos de dados, você pode definir o atual e selecionar tabelas dele.

Abrindo mais de um banco de dados

Quando um banco de dados está aberto, as tabelas e os relacionamentos entre elas são controlados pelas informações nele armazenadas. É possível haver mais de um banco de dados aberto ao mesmo tempo. Por exemplo, você pode usar diversos bancos de dados abertos ao executar mais de um aplicativo, cada qual utilizando um banco de dados diferente. Você também pode abrir diversos bancos de dados para utilizar informações, como controles personalizados, que estejam armazenadas em um banco de dados que não o de seu aplicativo.

Para abrir mais de um banco de dados

- No **Gerenciador de projetos,** selecione um banco de dados e escolha **Modificar** ou **Abrir**.

  – Ou –

- Utilize o comando OPEN DATABASE.

Abrir um novo banco de dados não fecha aqueles abertos anteriormente. Os outros continuarão abertos e o banco de dados recém-aberto se tornará o banco de dados atual.

Definindo o banco de dados atual

Ao abrir diversos bancos de dados, o Visual FoxPro define o último banco de dados aberto como sendo o atual. Como padrão, as tabelas e outros objetos que forem criados ou adicionados ao banco de dados farão parte do banco de dados atual. Comandos e funções que manipulem bancos de dados abertos, tais como ADD TABLE e DBC( ), irão operar no banco de dados atual.

É possível definir outro banco de dados como sendo o atual através da interface ou utilizando o comando SET DATABASE.

Para definir o banco de dados atual

- Na **Barra de ferramentas padrão**, selecione um banco de dados na caixa **Bancos de dados**.

  – Ou –

- Utilize o comando SET DATABASE.

Por exemplo, o código a seguir abre três bancos de dados, define o primeiro como o atual e utiliza a função DBC( ) para exibir o nome dele:

```
OPEN DATABASE testdata
OPEN DATABASE tastrade
OPEN DATABASE sample
SET DATABASE TO testdata
? DBC( )
```

Dica O Visual FoxPro pode abrir um ou mais bancos de dados automaticamente quando você executar uma consulta ou um formulário que exija que esses bancos de dados estejam abertos. Para certificar-se de estar usando o banco de dados desejado, defina o banco de dados atual de forma explícita antes de emitir qualquer comando que opere sobre ele.

Selecionando tabelas no banco de dados atual

É possível selecionar uma tabela do banco de dados atual em uma lista, utilizando o comando USE.

Para selecionar uma tabela do banco de dados atual

- Emita o comando USE com um símbolo"?".

  A caixa de diálogo **Utilizar** será exibida para que você possa selecionar uma tabela a ser aberta.

Por exemplo, o código a seguir abre o banco de dados sales e pede que você selecione uma tabela em uma lista de tabelas.

```
OPEN DATABASE SALES
USE ?
```

Se você deseja selecionar uma tabela que não está associada ao banco de dados aberto, escolha Outras na caixa de diálogo **Usar**.

Fechando um banco de dados

Para fechar um banco de dados aberto, use o **Gerenciador de projetos** ou o comando CLOSE DATABASE.

Para fechar um banco de dados

- No **Gerenciador de projetos,** selecione o banco de dados e escolha **Fechar**.

  – Ou –

- Utilize o comando CLOSE DATABASE.

Por exemplo, o código a seguir fecha o banco de dados testdata:

```
SET DATABASE TO testdata
CLOSE DATABASE
```

Ambas as opções fecham o banco de dados automaticamente. Também é possível fechar bancos de dados e todos os outros objetos abertos com a cláusula ALL do comando CLOSE.

Se o comando CLOSE DATABASE for emitido na janela **Comando**, um banco de dados não será fechado caso tenha sido aberto nas seguintes condições:

- Pelo **Gerenciador de projetos** quando você tiver expandido os tópicos para visualizar o conteúdo de um banco de dados.
- Por um formulário que estiver sendo executado em sua própria Sessão de dados.

Nestas circunstâncias, o banco de dados permanecerá aberto até ser fechado pelo **Gerenciador de projetos** ou até que o formulário que estiver utilizando o banco de dados seja fechado.

Resolução de escopo

O Visual FoxPro utiliza o banco de dados atual como escopo primário para objetos definidos, como tabelas. Quando um banco de dados está aberto, o Visual FoxPro procura primeiro dentro desse

banco de dados, qualquer objeto que tenha sido pedido, como tabelas, visualizações, conexões e assim por diante. Se o objeto não estiver no banco de dados, o Visual FoxPro irá procurar no caminho de busca padrão.

Por exemplo, se a tabela customer estiver associada ao banco de dados sales, o Visual FoxPro sempre encontrará essa tabela no banco de dados quando você emitir os comandos a seguir:

```
OPEN DATABASE SALES
ADD TABLE F:\SOURCE\CUSTOMER.DBF
USE CUSTOMER
```

O Visual FoxPro irá procurar a tabela products primeiramente no banco de dados atual, caso você emita o comando a seguir.

```
USE PRODUCTS
```

Se products não estiver no banco de dados atual, o Visual FoxPro irá procurar fora do banco de dados, utilizando o caminho de busca padrão.

Observação Você pode especificar o caminho completo para uma tabela caso você queira poder acessá-la de dentro ou de fora de um banco de dados—por exemplo, caso você preveja uma alteração na localização de uma tabela. Contudo, você aumenta o desempenho ao referir-se apenas ao nome da tabela, já que o Visual FoxPro acessa os nomes de tabelas de banco de dados mais rapidamente do que os nomes especificados com um caminho completo.

Gerenciando erros de banco de dados

Os erros de banco de dados, também denominados “erros de mecanismo”, ocorrem quando há erros em tempo de execução no código de evento de registros. Por exemplo, há um erro de banco de dados quando um usuário tenta armazenar um valor nulo em um campo que não permite valores nulos.

Geralmente, quando ocorre um erro de banco de dados, o mecanismo de banco de dados subjacente exibe uma mensagem de erro. No entanto, a natureza exata da mensagem de erro depende do que está sendo acessado pelo banco de dados — por exemplo, as mensagens de erro produzidas por um servidor de banco de dados remoto (como o Microsoft SQL Server) provavelmente serão diferentes das produzidas se ocorrer um erro de banco de dados em um tabela local do Visual FoxPro.

Além disso, às vezes os erros de mecanismo são bastante genéricos, pois o mecanismo de banco de dados não possui informações sobre o contexto no qual um registro está sendo atualizado. Por isso, as mensagens de erro produzidas por um mecanismo de banco de dados geralmente são menos úteis para o usuário final de um aplicativo do Visual FoxPro.

Para gerenciar erros de banco de dados com ênfase em um aplicativo , você poderá criar disparadores utilizando o comando CREATE TRIGGER. O disparador é solicitado durante a tentativa de atualização de um registro (exclusão, inserção ou atualização). Seu código de disparador personalizado pode procurar condições de erro específicas do aplicativo e reportá-las.

Ao gerenciar erros de banco de dados utilizando disparadores, você deve ativar a utilização de buffer. Desta forma, quando se atualiza um registro, o disparador é solicitado, mas o registro não é enviado imediatamente para o banco de dados subjacente. Por isso, evite a possibilidade de produzir duas mensagens de erro: uma do disparador e outra do mecanismo de banco de dados subjacente.

Para criar mensagens de erro personalizadas utilizando disparadores

1. Em uma função definida pelo usuário ou procedimento armazenado, escreva o seu próprio texto de mensagem.
2. Ative a utilização de buffer com a função CURSORSETPROP( ) para exibir o texto personalizado. Se a utilização de buffer estiver desativada, o usuário verá o texto personalizado e a mensagem de erro de mecanismo.