

Depois de projetar o aplicativo cliente/servidor, você está pronto para criar e migrar um protótipo local. Protótipo local é um modelo de trabalho do seu aplicativo, que utiliza tabelas, visualizações e bancos de dados do Visual FoxPro para representar dados que serão acessados em um servidor remoto. Utilize os assistentes de upsizing para mover bancos de dados, tabelas e visualizações do sistema para um SQL Server remoto ou um servidor Oracle®.

Este capítulo aborda os tópicos a seguir:

- [Objetivos da criação de protótipos](#)
- [Criando um protótipo local para um aplicativo](#)
- [Utilizando os assistentes de upsizing](#)
- [Migrando para o SQL Server](#)
- [Migrando para um Oracle](#)

Objetivos da criação de protótipos

Ao utilizar o Visual FoxPro para criar um protótipo de um aplicativo, você estará aproveitando o máximo dos recursos de formulários visuais, assistentes, construtores, criadores e o **Gerenciador de projetos** para desenvolver rapidamente um aplicativo de trabalho. Sua meta final é implementar o aplicativo em todas as plataformas cliente/servidor e para tanto são muitas as vantagens de se criar um protótipo eficaz.

Reduzindo o tempo de desenvolvimento

Ao criar um protótipo rápido, você pode aprimorar o projeto e a arquitetura local do aplicativo de forma rápida e fácil, sem ter que acessar o servidor remoto para reconstruir tabelas e bancos de dados do servidor. Você pode também testar e depurar os formulários dos seus aplicativos utilizando áreas de armazenamento de dados pequenas, permitindo que você corrija e aperfeiçoe rapidamente a interface de usuário do seu aplicativo. Como não precisa trabalhar em arquiteturas diferentes, você poupa tempo de desenvolvimento com reconstrução, reindexação e reconexão de dados remotos apenas para testar o protótipo.

Diminuindo os custos de desenvolvimento para aumentar a satisfação do cliente

Como o protótipo local está contido no seu computador, desde o início do ciclo de desenvolvimento, você poderá facilmente fazer uma demonstração do modelo de trabalho do aplicativo para o usuário final. O fato de poder visualizar o aplicativo em andamento faz com que o cliente confie na sua capacidade de fornecer uma solução que atenda às suas necessidades. Também dá a você a oportunidade de obter o retorno do cliente sobre a interface e relatórios de usuário antes que você invista recursos na implementação em um servidor remoto.

À medida que os usuários vêem e interagem com o protótipo, podem começar a identificar as áreas que gostariam de alterar, assim como observar o potencial de adição de maior funcionalidade ao aplicativo. Você pode implementar alterações e redemonstrar o aplicativo em um processo iterativo até que você e o cliente fiquem satisfeitos com o projeto e a função do protótipo de aplicativo. O seu protótipo serve então como uma especificação de trabalho para o aplicativo cliente/servidor final.

Contribuindo para uma implementação bem-sucedida

Você pode também proporcionar aos usuários uma demonstração do protótipo do aplicativo, permitindo que eles testem o modelo de trabalho, à medida que você avança no processo de implementação do aplicativo propriamente dito. Ao ganhar experiência com o protótipo, as dificuldades dos usuários diminuem e eles tornam-se melhores parceiros no processo de aprimorar e ajustar o aplicativo com você. Terão assim mais condições de obter maior produtividade e satisfação na etapa de implementação final porque já compreendem a estrutura básica do aplicativo.

A utilização de um modelo de trabalho ajuda a aumentar a margem de tempo que o usuário final terá para conhecer e ficar à vontade com o aplicativo. Também fornece uma estrutura para os funcionários da sua empresa ou do cliente criarem e desenvolverem um plano de treinamento para o aplicativo. O protótipo pode ainda ser utilizado para treinar usuários finais antes que o aplicativo final seja entregue, contribuindo assim para uma implementação bem-sucedida do aplicativo cliente/servidor final.

Criando um protótipo local de um aplicativo

Quando você cria um protótipo local do aplicativo, pode partir do zero ou converter um aplicativo do Visual FoxPro existente em um aplicativo cliente/servidor. A principal diferença entre a criação de um protótipo local de um aplicativo cliente/servidor e o desenvolvimento de qualquer outro aplicativo do Visual FoxPro está na utilização de visualizações e tabelas locais para representar dados que serão migrados.

► Para criar e migrar um protótipo local

- 1 Crie o aplicativo, utilizando visualizações e tabelas locais para representar os dados que deseja mover para um servidor remoto.
- 2 Utilize as visualizações locais nos formulários e relatórios de dados no ambiente de dados do aplicativo.
- 3 Migre as visualizações e tabelas locais utilizando o **Assistente de upsizing do SQL Server** ou **Assistente de upsizing do Oracle**:

- Na etapa **Definir opções de migração**, na área **Alterações a serem feitas localmente**, selecione **Redirecionar visualização a dados remotos**.

Quando você seleciona esta opção, o **Assistente de upsizing** copia as tabelas locais selecionadas para o servidor remoto e redireciona as visualizações locais para utilizar dados remotos, onde aplicável.

Para obter maiores informações sobre como criar visualizações, consulte o capítulo 8, [Criando visualizações](#) Para obter informações sobre como criar formulários e utilizar um ambiente de dados, consulte o capítulo 9, [Criando formulários](#) Para obter informações sobre como desenvolver um aplicativo, consulte o capítulo 2, [Desenvolvendo um aplicativo](#).

Utilizando os assistentes de upsizing

O Visual FoxPro fornece dois Assistentes de upsizing: o **Assistente de upsizing do SQL Server** e o **Assistente de upsizing do Oracle**. Esses assistentes criam bancos de dados do SQL Server ou Oracle que duplicam, o máximo que podem, a funcionalidade de um conjunto de tabelas em um banco de dados do Visual FoxPro. Você pode também optar por redirecionar as visualizações do Visual FoxPro para que utilizem os dados remotos recém-criados em vez de dados locais. É possível utilizar os assistentes de upsizing para:

- Mover dados locais para um servidor remoto.
- Transformar tabelas-base e visualizações locais em tabelas-base e visualizações remotas.
- Migrar um aplicativo local para um aplicativo cliente/servidor.

Observação Embora o **Assistente de upsizing** acesse servidores do SQL Server ou Oracle, você pode criar um aplicativo cliente/servidor para qualquer fonte de dados ODBC remota. Para servidores que não sejam o SQL Server ou Oracle, você pode utilizar as funções de passagem SQL para criar tabelas remotas, depois utilizar o Visual FoxPro para criar visualizações remotas que acessem tabelas do servidor. Para obter maiores informações sobre como utilizar as funções de passagem SQL, consulte o capítulo 21, [Implementando um aplicativo cliente/servidor](#) Para obter informações sobre como criar visualizações remotas, consulte o capítulo 8, [Criando visualizações](#)

Migrando para um SQL Server

Antes de executar o **Assistente de upsizing do SQL Server**, você deve preparar os lados do

cliente e do servidor.

Preparando o lado SQL Server

Antes da migração, você deve garantir que possui as autorizações necessárias no servidor, estimar o tamanho do banco de dados e verificar se o servidor possui espaço em disco suficiente. Também existem preparativos especiais para a migração para vários discos ou dispositivos.

Verificando o espaço livre em disco

Certifique-se de que possui espaço em disco suficiente no servidor.

Importante Se o **Assistente de upsizing do SQL Server** ficar sem espaço em disco no servidor, ele pára, deixando um banco de dados parcial e os dispositivos que criou no servidor. Você pode remover dispositivos, bancos de dados e tabelas com a ferramenta SQL Server Administration.

Definindo permissões em bancos de dados do SQL Server

Para executar o **Assistente de upsizing do SQL Server**, você deve possuir determinadas permissões no servidor do SQL Server para o qual estará migrando. As permissões necessárias dependem das tarefas que você deseja realizar.

- Para migrar para um banco de dados existente, você precisa das permissões CREATE TABLE e CREATE DEFAULT.
- Para construir um novo banco de dados, você precisa das permissões CREATE DATABASE e SELECT nas tabelas do sistema no banco de dados mestre.
- Para criar novos dispositivos, você deve ser um administrador de sistema.

Para obter maiores informações sobre como conceder permissões de servidor, consulte a documentação do SQL Server.

Estimando o tamanho do dispositivo e do banco de dados do SQL Server

Quando você cria um novo banco de dados, o **Assistente de upsizing do SQL Server** solicita que você selecione dispositivos para o banco de dados e log. Também pede que você defina o tamanho do banco de dados e dos dispositivos.

Estimando o tamanho do banco de dados do SQL Server

Quando o SQL Server cria um banco de dados, ele reserva uma quantidade fixa de espaço para esse banco de dados em um ou mais dispositivos. O espaço inteiro não será necessariamente utilizado pelo banco de dados—o tamanho estabelecido para o banco de dados simplesmente indica o quanto ele pode crescer antes de ficar sem espaço.

Observação Você pode aumentar o tamanho de um banco de dados do SQL Server depois da sua criação. Para obter maiores informações, consulte o comando ALTER DATABASE na documentação do SQL Server.

Para calcular o espaço necessário para o banco de dados, observe o tamanho dos arquivos .DBF do Visual FoxPro para as tabelas que você deseja migrar e calcule o ritmo em que o novo banco de dados do SQL Server vai crescer. Em geral, cada megabyte de dados do Visual FoxPro requer pelo menos de 1,3 a 1,5 megabytes no SQL Server.

Se tiver muito espaço em disco no seu servidor, multiplique o tamanho das tabelas do Visual FoxPro por dois. Isso garante que o **Assistente de upsizing do SQL Server** tenha espaço suficiente para migrar o banco de dados e também deixar espaço para crescimento. Se você espera adicionar muitos dados ao banco de dados, aumente o múltiplo.

Estimando o tamanho dos dispositivos do SQL Server

Todos os bancos de dados e logs do SQL Server são inseridos em dispositivos. Um dispositivo é uma localização lógica na qual são inseridos bancos de dados e logs, sendo também um arquivo físico. Quando um dispositivo é criado, o SQL Server cria um arquivo, reservando assim uma quantidade definida de espaço em disco para seu próprio uso.

O **Assistente de upsizing do SQL Server** exibe a quantidade de espaço livre disponível nos dispositivos existentes do SQL Server. Selecione um dispositivo que tenha espaço livre em disco suficiente, pelo menos para o tamanho estimado do banco de dados.

Se nenhum dispositivo existente tiver espaço livre suficiente, você pode utilizar o **Assistente de upsizing do SQL Server** para criar um novo dispositivo. Os novos dispositivos devem ser pelo menos do tamanho estimado do banco de dados. Se possível, faça o dispositivo ser maior do que o banco de dados, para que você possa expandi-lo mais tarde ou colocar outros bancos de dados ou logs no mesmo dispositivo.

Importante O tamanho do dispositivo não pode ser alterado. Certifique-se de criar dispositivos com tamanho suficiente.

Utilizando vários discos ou dispositivos do SQL Server

Na maioria dos casos, o **Assistente de upsizing do SQL Server** fornece controle mais do que suficiente sobre os dispositivos do SQL Server. No entanto, se o servidor tiver vários discos ou se você desejar inserir um banco de dados ou log em vários dispositivos, convém criar dispositivos antes de executar o **Assistente de upsizing do SQL Server**.

Servidores com vários discos físicos

Se o servidor tiver mais de um disco rígido físico, convém inserir o banco de dados em um disco e o log de banco de dados em um disco diferente. No caso de uma falha de disco, será mais provável recuperar o banco de dados se o log e o banco de dados forem armazenados em discos físicos diferentes.

O **Assistente de upsizing do SQL Server** permite que você crie novos dispositivos mas apenas em um disco físico —o mesmo disco do dispositivo de banco de dados principal.

Para inserir um banco de dados e log em discos separados, certifique-se de que possui dispositivos com tamanho suficiente nos dois discos, criando novos dispositivos, se necessário. Em seguida, execute o **Assistente de upsizing do SQL Server**.

Inserindo bancos de dados ou logs em vários dispositivos

O SQL Server permite que bancos de dados e logs utilizem vários dispositivos. No entanto, o **Assistente de upsizing do SQL Server** permite que você especifique apenas um dispositivo para o banco de dados e um dispositivo para o log.

Para especificar vários dispositivos para um banco de dados ou log, torne estes dispositivos (e nenhum outro) dispositivos padrão. Em seguida, execute o **Assistente de upsizing do SQL Server** e escolha Padrão para o dispositivo do log ou banco de dados.

Observação Se o novo tamanho do banco de dados ou log do SQL Server não exigir a utilização de todos os dispositivos padrão, o SQL Server só utilizará os dispositivos necessários para conter o banco de dados ou log.

Preparando o cliente

Antes da migração, você deve acessar um SQL Server através de uma fonte de dados ODBC ou conexão definida. Também deve ter um banco de dados do Visual FoxPro, com uma cópia de reserva antes de executar o **Assistente de upsizing do SQL Server**.

Criando uma fonte de dados ODBC ou conexão definida

Ao criar um novo banco de dados remoto, certifique-se de que possui uma fonte de dados ODBC ou conexão definida no banco de dados do Visual FoxPro que acesse o SQL Server para o qual deseja migrar. Pelo fato de você não poder prosseguir pelo **Assistente de upsizing** até selecionar uma fonte de dados ou conexão definida, você deve criar a fonte de dados ou conexão definida apropriada antes de iniciar o processo de migração.

Para obter informações sobre como criar uma conexão definida, consulte o capítulo 8, [Criando visualizações](#). Se desejar criar uma fonte de dados ODBC, execute o Administrador ODBC. Para obter informações sobre como configurar fontes de dados ODBC, consulte o capítulo 1, [Instalando o Visual FoxPro](#), no *Guia de Instalação e Índice Principal*.

Criando um backup do banco de dados

Recomenda-se criar uma cópia de reserva do banco de dados (arquivos .DBC, .DCT e .DCX) antes da migração. O **Assistente de upsizing do SQL Server** não modifica os arquivos .DBF, mas atua no .DBC diretamente, abrindo-o às vezes como uma tabela, e indiretamente, renomeando as tabelas e visualizações ao criar novas visualizações remotas. Se você criar um backup do banco de dados, poderá reverter o banco de dados ao seu estado original pré-migração, sobrescrevendo os arquivos .DBC, .DCT e .DCX migrados com o backup original, o que inverte o processo de renomear e criar novas visualizações.

Fechando tabelas

O **Assistente de upsizing do SQL Server** tenta abrir de forma exclusiva todas as tabelas no banco de dados a ser migrado; se houver tabelas abertas e compartilhadas, o assistente as fecha e as reabre de forma exclusiva. Abrir as tabelas de forma exclusiva antes da migração as protege contra outros usuários que tentem modificar os seus registros durante a exportação de dados. Se alguma tabela não puder ser aberta de forma exclusiva, o **Assistente de upsizing do SQL Server** exibirá uma mensagem; essas tabelas não estarão disponíveis para migração.

Iniciando o Assistente de upsizing do SQL Server

Depois de criar uma fonte de dados ODBC e completar os preparativos necessários para o cliente e o servidor, você estará pronto para iniciar a migração.

► Para iniciar o Assistente de upsizing do SQL Server

- 1 No menu **Ferramentas**, escolha **Assistentes** e, em seguida, **Upsizing**.
- 2 Na caixa de diálogo **Seleção de assistente**, selecione **Assistente de upsizing do SQL Server**.
- 3 Siga as instruções apresentadas nas telas do assistente, conforme descrito nas seções a seguir.
Você pode escolher o botão **Cancelar** a qualquer momento para sair do assistente; o assistente não executará ações no servidor até você escolher o botão **Concluir**.
- 4 Quando estiver pronto para migrar, escolha o botão **Concluir**.

Depois de escolher o botão Finalizar, o **Assistente de upsizing do SQL Server** começa a exportar o banco de dados para o servidor.

O botão **Finalizar** está disponível depois que você fornece as informações básicas necessárias para a migração. Se você escolher o botão **Concluir** antes de completar todas as telas do assistente, o **Assistente de upsizing do SQL Server** usará os valores padrão para as telas restantes.

Como Funciona o Assistente de upsizing do SQL Server

O **Assistente de upsizing do SQL Server** faz a migração de um banco de dados do Visual FoxPro para o SQL Server praticamente transparente. Esta seção explica exatamente o que acontece depois que você escolhe o botão **Concluir** — como o **Assistente de upsizing do SQL Server** exporta dados e mapeia os objetos do Visual FoxPro para objetos do SQL Server.

Métodos de exportação de dados

O **Assistente de upsizing do SQL Server** exporta dados utilizando um dos dois métodos disponíveis. O primeiro método cria um procedimento armazenado que executa várias inserções de linha. Esse método pode ser muito rápido, porque os procedimentos armazenados são pré-compilados e rapidamente executados.

No entanto, os procedimentos armazenados não podem aceitar variáveis binárias de tamanho variável como parâmetros. Se você estiver exportando dados que devem ser armazenados em tabelas do SQL Server utilizando dados de texto ou imagem ou tabelas com mais de 250 campos, o **Assistente de upsizing do SQL Server** utilizará um método de exportação diferente. O segundo método cria uma instrução SQL INSERT para cada linha na tabela e depois executa a instrução.

Se o **Assistente de upsizing de SQL do Server** encontrar erros durante a exportação de dados utilizando o método SQL INSERT e o número de erros exceder 10% do número de registros na tabela ou 100 registros (o que for maior), o assistente cancelará a exportação para a tabela e salvará o número de erros de exportação para o relatório de erros. A tabela do servidor exportada não é abandonada, no entanto, e os registros que foram exportados com sucesso são deixados na tabela do servidor.

Visão geral do mapeamento de objetos

Para migrar um banco de dados do Visual FoxPro para um servidor, o **Assistente de upsizing do SQL Server** cria objetos de servidor que, na medida do possível, fazem tudo o que o banco de dados do Visual FoxPro fazia. Mapear alguns objetos do Visual FoxPro para objetos do servidor é um processo bastante direto. Os bancos de dados, tabelas, campos, padrões e índices do Visual FoxPro são mapeados para bancos de dados, tabelas, campos, padrões e índices do SQL Server em um mapeamento direto de um para um.

No entanto, nem todos os objetos locais são mapeados diretamente para objetos do servidor. As regras de validação e a integridade referencial no Visual FoxPro fazem parte do dicionário de dados e são executadas pelo próprio mecanismo do banco de dados. As regras de validação e a integridade referencial do SQL Server não fazem parte do dicionário de dados e são executadas por código ligado a uma tabela. Essas diferenças, assim como as decisões de projeto tomadas pelo **Assistente de upsizing do SQL Server**, significam que boa parte do dicionário de dados do Visual FoxPro não pode ser mapeado diretamente para construtos do SQL Server.

A tabela a seguir resume a forma como os objetos são mapeados do Visual FoxPro para o SQL Server:

Objetos do Visual FoxPro	Objetos do SQL Server
Banco de dados	Banco de dados
Tabela	Tabela
Índices	Índices
Campo	Campo
Padrão	Padrão
Regra de validação de tabela	procedimentos armazenados de SQL server, chamados dos disparadores UPDATE e INSERT
Regra de validação de campo	Procedimentos armazenados do SQL Server, chamados dos disparadores UPDATE e INSERT
Relacionamentos permanentes (quando utilizados para restrições de integridade referencial)	Disparadores UPDATE, INSERT e DELETE ou restrições da tabela

As seções a seguir abrangem cada objeto do Visual FoxPro e o objeto do SQL Server (ou objetos) para o qual são mapeados.

Convenções para nomes de objetos migrados

Ao migrar objetos para uma fonte de dados, o **Assistente de upsizing do SQL Server** cria objetos definidos no servidor. O assistente utiliza prefixos para os objetos que precisam de nomes novos porque não existia um objeto independente desse tipo no Visual FoxPro (padrões e regras, por exemplo). Depois do prefixo vem um nome de tabela e depois um nome de campo, se necessário. Essas convenções para nomes permitem que todos os objetos do mesmo tipo tenham o mesmo prefixo e sejam classificados juntos quando visualizados com as ferramentas de administração da fonte de dados. Os objetos criados na mesma tabela também são agrupados quando visualizados.

Objetos de bancos de dados e tabelas

Um banco de dados do Visual FoxPro é mapeado diretamente para um banco de dados do SQL Server. Uma tabela do Visual FoxPro, excluindo parte do seu dicionário de dados, é mapeada para uma tabela do SQL Server.

Os nomes de bancos de dados, tabelas, índices e campos podem ser alterados durante o processo de migração, se violarem as convenções para nomes do SQL Server. Os nomes no SQL Server devem ter até 30 caracteres, e o primeiro caractere deve ser uma letra ou o símbolo "@". Os caracteres restantes podem ser números, letras ou os símbolos "\$", "#" e "_"; espaços não são aceitos. O **Assistente de upsizing do SQL Server** substitui qualquer caractere ilegal pelo símbolo "_".

Qualquer nome que seja igual às palavras reservadas do SQL Server recebe um sufixo "_". Por exemplo, FROM e GROUP tornam-se FROM_ e GROUP_. O **Assistente de upsizing do SQL Server** também coloca o símbolo "_" na frente dos nomes de objeto que começam com um número.

Tabelas

O **Assistente de upsizing do SQL Server** dá a cada tabela migrada o mesmo nome da tabela local, a não ser que o nome da tabela contenha um espaço ou seja uma palavra-chave para a fonte de dados.

Visualizações das novas tabelas do servidor

Se você selecionar **Criar visualizações remotas em tabelas**, o **Assistente de upsizing do SQL Server** criará visualizações remotas e fornecerá a elas muitas das propriedades dos campos na tabela local original.

Mapeando nomes de campos e tipos de dados do Visual FoxPro para os equivalentes do SQL Server

Os nomes de campos e tipos de dados são automaticamente convertidos em campos do SQL Server quando uma tabela do Visual FoxPro é exportada pelo **Assistente de upsizing do SQL Server**.

Os tipos de dados do Visual FoxPro são mapeados para tipos de dados do SQL Server como se segue:

Abreviação	Tipo de dado do Visual FoxPro	Tipo de dado do SQL Server
C	Caractere	char
Y	Moeda	money
D	Data	datetime
T	DataHora	datetime

B	Duplo	float
F	Flutuante	float
G	Geral	image
I	Inteiro	int
L	Lógico	bit
M	Memo	text
M (binário)	Memo binário	image
C (binário)	Caractere binário	binary
N	Numérico	float

Colunas de identidade e marca de data e hora

Colunas de marca de data e hora são criadas utilizando dados tipo marca de data e hora Transact-SQL. Ao selecionar a caixa de verificação da coluna **Marca de data e hora** para uma tabela específica no Passo 4 - **Mapear tipos de dados de campo**, o **Assistente de upsizing do SQL Server** cria um campo de marca de data e hora para a tabela.

Se uma tabela contiver um ou mais campos memo (M) ou figura (P), o **Assistente de upsizing do SQL Server** selecionará a caixa de verificação da coluna **Marca de data e hora** para a tabela como padrão e criará um campo de marca de data e hora na versão migrada da tabela.

Colunas de identidade são criadas utilizando os campos de propriedade Transact-SQL IDENTITY.

Índices

Os índices do SQL Server e do Visual FoxPro são muito semelhantes. A tabela a seguir mostra como os tipos de índice do Visual FoxPro são convertidos em tipos de índice do SQL Server:

Conversão de tipo de índice

Tipo de índice do Visual FoxPro	Tipo de índice do SQL Server
Primário	Clustered Unique
Candidato	Unique
Único, Regular	Non-unique

O **Assistente de upsizing do SQL Server** utiliza os nomes de marca do Visual FoxPro para os índices no SQL Server. Se o nome da marca for uma palavra reservada no servidor, o assistente alterará o nome da marca anexando o caractere “_”.

Observação O SQL Server não aceita índices ascendentes ou descendentes, nem permite expressões dentro dos índices do servidor. O **Assistente de upsizing do SQL Server** remove expressões do Visual FoxPro das expressões do índice à medida que o índice é migrado; apenas os nomes de campos são enviados para o servidor.

Padrões do SQL Server

Uma expressão padrão do Visual FoxPro é mapeada diretamente para um único padrão do SQL Server. O **Assistente de upsizing do SQL Server** tenta criar um padrão do SQL Server com base na expressão padrão para um campo do Visual FoxPro. Se o padrão for criado com sucesso, o assistente liga-o ao campo adequado do SQL Server. O relatório de migração sobre os campos indica se o **Assistente de upsizing do SQL Server** foi bem-sucedido ao converter a expressão do Visual FoxPro em Transact-SQL do SQL Server. Para obter detalhes sobre a conversão, consulte [Mapeamento de expressões](#) mais adiante neste capítulo.

Os padrões do SQL Server e do Visual FoxPro são muito semelhantes, mas existem algumas diferenças na forma como os padrões são criados e se comportam nos dois produtos. Os padrões do SQL Server são objetos isolados e independentes de qualquer campo ou tabela específicos. Uma vez tendo sido criado um padrão, este pode ser utilizado por (ou ligado a) qualquer número de

campos diferentes.

Convenções para nomes de padrões

O **Assistente de upsizing do SQL Server** nomeia os padrões utilizando o prefixo `Dflt_` mais o nome da tabela e do campo. Por exemplo, um valor padrão para o campo `ordamt` na tabela `Customer` pode se chamar `Dflt_Customer_Ordamt` no servidor. Se a combinação do prefixo com os nomes de tabela e campo fizer com que o nome padrão exceda 30 caracteres, o Visual FoxPro truncará os caracteres excedentes.

Os campos com uma expressão padrão com valor zero se ligam a um padrão chamado `UW_ZeroDefault`. Se dois ou mais campos tiverem a mesma expressão padrão não-zero, o **Assistente de upsizing do SQL Server** criará dois padrões, com dois nomes diferentes, que são funcionalmente idênticos.

Valores padrão para campos lógicos do Visual FoxPro

Os campos lógicos no SQL Server proíbem valores nulos; os campos lógicos do Visual FoxPro os aceitam. Para gerenciar essa diferença, o **Assistente de upsizing do SQL Server** cria automaticamente um valor padrão chamado `UW_ZeroDefault` e o liga a cada campo lógico exportado, quer você escolha ou não exportar padrões. Este padrão definirá o valor do campo do servidor como 0 (ou falso (.F.)), se você olhar para o campo no Visual FoxPro) quando você não fornecer um valor.

Se a tabela local do Visual FoxPro contiver um valor padrão para um campo lógico que defina o campo como igual a verdadeiro (.T.), o **Assistente de upsizing do SQL Server** não ligará o padrão `UW_ZeroDefault` à tabela do servidor. Em vez disso, o assistente criará um padrão que define o campo como igual a 1, nomeando o padrão de acordo com as convenções para nomes apresentadas anteriormente neste capítulo.

Os padrões do SQL Server têm um comportamento diferente dos padrões do Visual FoxPro. Para obter maiores informações, consulte a seção [Valores padrão](#), mais adiante neste capítulo.

Disparadores do SQL Server

Um **Disparador do SQL Server** é uma série de instruções Transact-SQL associadas a determinada tabela do SQL Server. Quando você optar por fazer a migração das Regras de validação e relacionamentos no Passo 8, o **Assistente de upsizing do SQL Server** converterá as regras de validação de campos e registros e os relacionamentos de tabela permanentes do Visual FoxPro em procedimentos armazenados, que são chamados a partir de disparadores do SQL Server. Cada disparador do servidor pode conter um código para emular a funcionalidade de várias regras de integridade referencial e de validação.

Observação O **Assistente de upsizing do SQL Server** não faz a migração de disparadores do Visual FoxPro.

Uma tabela do servidor pode ter três disparadores, um para cada um dos comandos que podem modificar dados na tabela: UPDATE, INSERT e DELETE. O disparador é automaticamente acionado quando o comando associado é executado.

A tabela a seguir descreve os disparadores criados pelo **Assistente de upsizing do SQL Server**. Qualquer disparador específico poderá conter um código para emular algumas ou todas as funcionalidades do Visual FoxPro listadas.

Disparador	Funcionalidade do Visual FoxPro emulada
UPDATE	Regras de validação (validação de campos e registros) Integridade referencial
INSERT	Regras de validação (validação de campos e

	registros)
	Integridade referencial (apenas disparadores de tabela filho)
DELETE (apenas tabela pai)	Integridade referencial

Convenções para nomes de disparadores

O **Assistente de upsizing do SQL Server** nomeia os disparadores do servidor, combinando um prefixo que indica o tipo de disparador sendo criado com o nome da tabela do SQL Server, à qual o disparador pertence. O prefixo (“TrigU_” para disparadores UPDATE, “TrigD_” para disparadores DELETE e “TrigI_” para disparadores INSERT) é colocado na frente do nome da tabela. Por exemplo, o disparador UPDATE na tabela Customer pode se chamar `TrigU_Customer`.

Regras de validação

O **Assistente de upsizing do SQL Server** pode exportar regras de validação de campos e registros do Visual FoxPro, que são convertidas em procedimentos armazenados no SQL Server. O assistente nomeia as regras de campos, combinando um prefixo “vrf” (do inglês “validation rule, field”) com os nomes da tabela e do campo; um exemplo pode ser `vrf_customer_lname`. As regras de validação das tabelas são chamadas com o prefixo “vrt” (do inglês “validation rule, table”) mais o nome da tabela, para criar um nome como `vrt_customer`.

O **Assistente de upsizing do SQL Server** utiliza disparadores que chamam procedimentos armazenados em vez de regras do SQL Server para garantir a validação de campos porque as regras do SQL Server não permitem que você exiba mensagens de erro personalizadas. Para obter maiores informações sobre as regras do SQL Server, consulte o comando CREATE RULE na documentação do seu SQL Server.

Integridade referencial

O aplicativo do Visual FoxPro aceita integridade referencial por disparadores nos eventos UPDATE, DELETE e INSERT em relacionamentos de tabelas permanentes executadas pelo próprio mecanismo do banco de dados. Você pode optar por implementar restrições à integridade referencial no SQL Server utilizando um dos dois métodos a seguir:

- Integridade referencial baseada em disparadores
 - Ou –
- Integridade referencial declarativa

Ao escolher integridade referencial baseada em disparadores, o **Assistente de upsizing do SQL Server** criará disparadores que incluem o código Transact-SQL exigido para duplicar as restrições de integridade referencial do Visual FoxPro. Se escolher implementar integridade referencial declarativa, o **Assistente de upsizing do SQL Server** criará restrições ao SQL Server utilizando o comando ALTER TABLE com a palavra-chave CONSTRAINT.

Integridade referencial baseada em disparadores

No método baseado em disparadores, a integridade referencial é executada no SQL Server pelo código Transact-SQL em disparadores. Você pode utilizar disparadores para fornecer restrições em instruções UPDATE, DELETE e INSERT e para dispor em cascata as alterações resultantes das instruções DELETE e INSERT.

O **Assistente de upsizing do SQL Server** cria disparadores SQL Server avaliando os disparadores do Visual FoxPro utilizados para executar integridade referencial em relacionamentos permanentes no banco de dados do Visual FoxPro. A tabela a seguir lista o mapeamento entre as restrições de integridade referencial do Visual FoxPro e os disparadores do SQL Server gerados pelo **Assistente de upsizing do SQL Server**.

Restrições de integridade referencial do Visual FoxPro		Disparador do SQL Server
UPDATE	Cascata	Disparador UPDATE em cascata
	Restrito	Disparador UPDATE restrito
	Ignorar	Nenhum disparador gerado
DELETE	Cascata	Disparador DELETE em cascata
	Restrito	Disparador DELETE restrito
	Ignorar	Nenhum disparador gerado
INSERT	Restrito	Disparador INSERT restrito
	Ignore	Nenhum disparador gerado

Um relacionamento permanente do Visual FoxPro que é utilizado em uma restrição de integridade referencial pode se tornar até quatro disparadores em uma fonte de dados do SQL Server: dois para a tabela pai e dois para a tabela filho.

Observação Se apenas uma das tabelas em um relacionamento for migrada ou a integridade referencial não for executada no Visual FoxPro, o relacionamento não será exportado.

Tabela pai

O **Assistente de upsizing do SQL Server** cria um **Disparador UPDATE** que impede o usuário de alterar a chave primária da tabela pai ou que passa a alteração para a tabela filho, dependendo do tipo de relacionamento que foi criado no Visual FoxPro.

O assistente também cria um **Disparador DELETE** que impede o usuário de excluir um registro com registros filho relacionados ou que exclui também os registros filho, novamente dependendo do tipo de relacionamento existente entre as tabelas do Visual FoxPro.

Tabela filho

O **Assistente de upsizing do SQL Server** cria um **Disparador UPDATE** que evita que o usuário faça alterações na chave estrangeira que tornaria o registro órfão. Da mesma forma, um **Disparador INSERT** é criado para evitar que o usuário adicione um novo registro sem pai.

Valores de erro personalizados

Se, durante a execução, a integridade referencial estabelecida pelos disparadores criados pelo **Assistente de upsizing do SQL Server** for violada, o SQL Server colocará um valor de erro personalizado na variável @@ERROR. Os valores de erro potenciais são definidos pelo **Assistente de upsizing do SQL Server** como parte do código de disparador. O valor de erro específico retornado em tempo de execução depende da ação que o usuário estava tentando executar: atualização, inserção ou exclusão.

A tabela a seguir lista os números de erro gerados para cada ação:

Ação	Erro do SQL Server
Regra de validação violada	44444
Tentativa de exclusão	44445
Tentativa de atualização	44446
Tentativa de inserção	44447
Instrução de atualização ou exclusão afetou mais de uma linha; a instrução sofre um rollback automático	44448

Integridade referencial declarativa

Se você optar por implementar integridade referencial declarativa, o **Assistente de upsizing do SQL Server** criará restrições do SQL Server utilizando o comando ALTER TABLE com a palavra-chave CONSTRAINT. A restrição de tabela pai utiliza a palavra-chave PRIMARY KEY. A restrição de tabela filho utiliza as palavras-chave FOREIGN KEY e REFERENCES. A integridade referencial declarativa é aceita nos níveis RESTRICT, atualizações RESTRICT e exclusões RESTRICT.

Utilize restrições do SQL Server para fornecer restrições em instruções UPDATE, DELETE e INSERT.

Mapeamento de expressões

Embora o Visual FoxPro e o Transact-SQL tenham algumas funções em comum, muitas funções do Visual FoxPro não são aceitas pelo SQL Server. O **Assistente de upsizing do SQL Server** tenta converter em Transact-SQL as expressões do Visual FoxPro de regras de validação de campos e registros e os valores padrão, utilizando o seguinte mapa de expressões:

Expressão do Visual FoxPro	Expressão do SQL Server
Verdadeiro (.T.)	1
Falso (.F.)	0
#	<>
.AND.	AND
.NOT.	NOT
.NULL.	NULL
.OR.	OR
=<	<=
=>	>=
ASC()	ASCII()
AT()	CHARINDEX()
CDOW()	DATENAME(dw, ...)
CHR()	CHAR()
CMONTH()	DATENAME(mm, ...)
CTOD()	CONVERT(datetime, ...)
CTOT()	CONVERT(datetime, ...)
DATE()	GETDATE()
DATETIME()	GETDATE()
DAY()	DATEPART(dd, ...)
DOW()	DATEPART(dw, ...)
DTOC()	CONVERT(varchar, ...)
DTOR()	RADIANS()
DTOT()	CONVERT(datetime, ...)
HOUR()	DATEPART(hh, ...)
LIKE()	PATINDEX()
MINUTE()	DATEPART(mi, ...)
MONTH()	DATEPART(mm, ...)
MTON()	CONVERT(money, ...)
NTOM()	CONVERT(float, ...)
RTOD()	DEGREES()
SUBSTR()	SUBSTRING()

TTOC()	CONVERT(char, ...)
TTOD()	CONVERT(datetime, ...)
YEAR()	DATEPART(yy, ...)

As expressões a seguir são iguais no Visual FoxPro e no SQL Server.

Expressões mapeadas diretamente do Visual FoxPro para o SQL Server

CEILING()	LOG()	LOWER()
LTRIM()	RIGHT()	RTRIM()
SOUNDEX()	SPACE()	STR()
STUFF()	UPPER()	

Arquivos criados pelo Assistente de upsizing do SQL Server

O **Assistente de upsizing do SQL Server** cria tabelas para sua própria utilização durante o processo de migração. Esses arquivos serão removidos do disco rígido, a menos que:

- Você escolha gerar um relatório de migração.
- Você queira salvar a SQL gerada.
- Ocorram erros durante a migração e você escolha salvar as informações de erro.

Se qualquer uma das condições acima for verdadeira, o **Assistente de upsizing do SQL Server** criará um projeto (chamado Report, Report1, Report2 e assim por diante) e um banco de dados (chamado Upsize, Upsize1 e assim por diante) em um subdiretório (chamado UPSIZE) do diretório definido pelo comando SET DEFAULT para a sua sessão do Visual FoxPro. O assistente adiciona ao banco de dados as tabelas utilizadas para gerar o Relatório de migração, uma tabela para armazenar a SQL gerada e qualquer tabela de erro. A tabela a seguir lista os arquivos que podem ser criados pelo processo de migração.

Tabelas locais criadas durante a migração

Objetivo do arquivo	Nome da tabela	Conteúdo
Tabelas de relatório	Errors_uw	Informações sobre qualquer erro ocorrido durante a migração.
	Fields_uw	Informações sobre todas as tabelas migradas.
	Indexes_uw	Informações sobre todos os índices migrados.
	Misc_uw	Informações variadas sobre a migração.
	Relations_uw	Informações sobre todas as restrições de integridade referencial armazenadas no banco de dados do Visual FoxPro.
	Tables_uw	Informações sobre todas as tabelas do banco de dados escolhido para migração.
	Views_uw	Informações sobre as visualizações locais redirecionadas para acessar dados remotos.

Tabela Script	SQL_uw	Um campo memo contendo todo o código SQL gerado pelo Assistente de upsizing do SQL Server .
Tabelas de erro de exportação de dados	ExportErrors_ <i>tabela_nome</i>	Para cada tabela que apresenta um erro de exportação de dados durante a migração, o Assistente de upsizing do SQL Server gera uma tabela contendo os registros que não são exportados com sucesso.

Se o assistente for cancelado durante o processamento ou se o assistente for interrompido devido a um erro, nenhuma tabela será deixada no seu disco rígido.

Utilizando a SQL gerada

A tabela Script armazenada no seu disco rígido contém todo o código SQL gerado pelo **Assistente de upsizing do SQL Server**, quer seja executado sem erros no servidor ou não. Se você deseja utilizar este código, a melhor abordagem é examinar a SQL gerada, copiar as partes que deseja utilizar, executar as partes extraídas do código e repetir o processo para obter os resultados desejados. Você não pode executar todo o script SQL em lugar de executar o **Assistente de upsizing do SQL Server**, porque o assistente executa etapas adicionais que não são refletidas no código SQL gerado.

Completando o processo de migração do SQL Server

Você pode agora executar etapas adicionais, tanto no servidor quanto no aplicativo do Visual FoxPro, para garantir que o aplicativo e os dados estejam seguros e funcionando de forma adequada.

Você pode também utilizar as informações desta seção ao criar um aplicativo a partir de visualizações remotas, em vez de através da migração. Independentemente de como você cria tabelas remotas, você pode executar certas etapas para garantir que o servidor e o cliente estejam preparados para trabalhar juntos em seu aplicativo cliente/servidor.

Passos do SQL Server

Você pode concluir o processo de migração no seu servidor:

- Certificando-se de que as tabelas que deseja editar a partir do Visual FoxPro são atualizáveis.
- Definindo permissões no banco de dados de modo que os usuários possam acessar os objetos necessários.
- Protegendo o seu trabalho ao tornar o novo banco de dados recuperável, caso seja danificado ou perdido.

Adicionando índices únicos para atualização

Uma tabela remota deve ter um índice único e atualizável no Visual FoxPro. O **Assistente de upsizing do SQL Server** pode exportar um índice único existente, mas não cria um onde não existam índices. Certifique-se de que as tabelas que deseja editar a partir do Visual FoxPro sejam atualizáveis.

Definindo permissões

O novo banco de dados do SQL Server e seus objetos recebem um conjunto de permissões padrão do SQL Server. Defina as permissões no banco de dados remoto de modo que os usuários tenham acesso aos objetos necessários.

Permissões de login de banco de dados

As permissões padrão de um novo banco de dados o tornam acessível apenas aos administradores de sistema e ao proprietário do banco de dados.

Você pode adicionar novos usuários e grupos, utilizando o Gerenciador de segurança do SQL Server ou os procedimentos de sistema `sp_adduser` e `sp_addgroup`.

Para obter maiores informações sobre como adicionar usuários e grupos, consulte a **Ajuda do Gerenciador de segurança do SQL Server** e a documentação dos procedimentos de sistema `sp_adduser` e `sp_` no manual *Transact-SQL Reference* do SQL Server.

Permissões de objeto

Todos os objetos criados pelo Visual FoxPro para o **Assistente de upsizing do SQL Server**, incluindo tabelas, disparadores e padrões, são acessíveis inicialmente apenas para o proprietário do banco de dados e os administradores de sistema. Isso ocorre na migração para um banco de dados novo ou já existente. Se você sobrescrever objetos existentes, também sobrescreverá todas as permissões de objeto.

Para fornecer permissões em tabelas, utilize o SQL Enterprise Manager ou os comandos GRANT e REVOKE. Para obter maiores informações sobre como definir permissões de objeto, consulte a seção “Managing Object Permissions” no manual *Microsoft SQL Server Administrator's Companion*, e os comandos GRANT e REVOKE no manual *Microsoft SQL Server Transact-SQL Reference*.

Garantindo a recuperação

Proteja o seu trabalho, fazendo com que seus novos bancos de dados possam ser recuperados em caso de perda ou dano.

Descarregando o banco de dados principal

Quando um banco de dados é criado em um SQL Server, novos registros são adicionados às tabelas do sistema no banco de dados principal. Descarregar o banco de dados principal fornece um backup, incluindo todas as alterações mais recentes. Para obter maiores informações sobre como descarregar o banco de dados principal, consulte “Backing Up the master Database” no manual *Microsoft SQL Server Administrator's Companion*, e a instrução DUMP e “Dumping the master Database” no manual *Microsoft SQL Server Transact-SQL Reference*.

Programando backups

Programe backups regulares do banco de dados para que possa restaurá-lo caso ocorra um problema sério. Para obter detalhes sobre como fazer um backup dos bancos de dados do SQL Server, consulte “Database Maintenance Plan Wizard” e “Backup and Recovery” in *What's New in SQL Server 6.5* and “Database Design and Backup Strategy” in the *Microsoft SQL Server Transact-SQL Reference*.

Espelhamento de dispositivos

Ao se espelhar um dispositivo, duplicam-se continuamente as informações de um dispositivo do SQL Server para outro. No caso de um dispositivo falhar, o outro contém uma cópia atualizada de todas as transações.

Se você prever que muitas alterações serão feitas em um banco de dados entre os backups e não

puder perder essas alterações, avalie a possibilidade de espelhamento de dispositivos. O espelhamento é mais eficaz quando os dispositivos estão localizados em discos separados, pois os dois dispositivos poderão ser perdidos se estiverem no mesmo disco e o disco falhar.

Para obter maiores informações sobre o espelhamento de dispositivos, consulte “Mirroring a Database Device”, “About SQL Server Device Mirroring” e “Using SQL Server Mirroring” no manual *Microsoft SQL Server Administrator's Companion*.

Passos do cliente do Visual FoxPro

Depois de ter transferido os objetos do Visual FoxPro para um SQL Server, você provavelmente vai precisar modificar o código do banco de dados original do Visual FoxPro, de modo que ele funcione de forma adequada com o novo banco de dados do SQL Server.

Otimizando visualizações

As visualizações criadas pelo **Assistente de upsizing do SQL Server** não são parametrizadas e, portanto, não são otimizadas. Para obter um processamento mais eficiente, adicione parâmetros às visualizações criadas pelo **Assistente de upsizing do SQL Server** para transferir apenas os dados necessários. Para obter informações sobre como adicionar um parâmetro a uma visualização, consulte o capítulo 8, [Criando visualizações](#)

Algumas funções do Visual FoxPro não são aceitas pelo SQL Server. Se a visualização remota criada pelo **Assistente de upsizing do SQL Server** utilizar funções que não podem ser mapeadas para as funções Transact-SQL, a visualização não funcionará. Para obter maiores informações sobre como mapear expressões do Visual FoxPro para expressões Transact-SQL, consulte a seção [Mapeamento de expressões](#) anteriormente neste capítulo.

Criando procedimentos armazenados e disparadores

O **Assistente de upsizing do SQL Server** não faz a migração dos procedimentos armazenados e disparadores do Visual FoxPro. Se você deseja criar procedimentos armazenados ou disparadores do SQL Server, poderá utilizar o Transact-SQL no servidor ou utilizar a passagem SQL no Visual FoxPro. Para obter maiores informações sobre como utilizar o Transact-SQL, consulte a documentação do SQL Server. Para obter informações sobre como utilizar a passagem SQL, consulte o capítulo 21, [Implementando um aplicativo cliente/servidor](#)

Comparando a ordem de eventos

No Visual FoxPro, alguns eventos ocorrem em uma ordem diferente, dependendo se o seu aplicativo está utilizando dados do SQL Server ou do Visual FoxPro. Essas diferenças podem exigir alterações no seu código.

Valores padrão

Os valores de campo padrão do Visual FoxPro aparecem quando você começa a editar um novo registro. Os valores padrão gerados pelos padrões do SQL Server aparecem apenas depois que um registro foi inserido. Você precisa alterar qualquer código que dependa da existência de valores antes que o registro seja gravado, como o código para as pesquisas.

Regras de validação

No Visual FoxPro, a validação de campo ocorre quando o campo não está mais destacado. Quando você edita os dados do SQL Server em tabelas anexas, os disparadores e regras não são acionados até que você deixe o registro. Você talvez precise modificar qualquer regra de validação de registro baseada na validação de campo que ocorre quando se sai de um campo.

Manipulando expressões não convertidas

O relatório de migração indica se cada regra de validação da tabela, regra de validação de campo e expressão padrão do Visual FoxPro foi convertida com sucesso. Se uma expressão ou regra de validação padrão não foi convertida, você deve reescrevê-la no Transact-SQL.

Você pode também executar uma validação de formulário no Visual FoxPro. No entanto, se os dados do servidor forem alterados em seguida, sem utilizar um formulário específico, a validação não será aplicada e dados inválidos poderão ser inseridos.

Para obter maiores informações sobre a conversão de expressões, consulte a seção [Mapeamento de expressões](#) anteriormente neste capítulo. Para obter maiores informações sobre as funções Transact-SQL, consulte a documentação do SQL Server.

Bloqueio de registros

O Visual FoxPro utiliza bloqueios otimistas internamente quando acessa tabelas em um servidor do SQL Server. Bloqueio otimista significa que a linha só é bloqueada quando o valor editado é gravado e o processo de atualização ocorre—geralmente um intervalo muito pequeno.

No SQL Server, o bloqueio otimista é utilizado em vez do pessimista porque este é fornecido pelo bloqueio de página, bloqueando potencialmente muitos registros de cada vez. O bloqueio de página evita que outros usuários alterem o mesmo registro que você está editando, mas também pode impedir que os usuários acessem muitos outros registros na mesma página (bloqueada). O bloqueio otimista fornece o melhor acesso multiusuário para um aplicativo cliente/servidor do Visual FoxPro.

Você pode otimizar atualizações e controlar como os conflitos de atualização são gerenciados com a propriedade SQL WhereType. Para obter maiores informações sobre como controlar os conflitos de atualização, consulte o capítulo 8, [Criando visualizações](#)

Migrando para o Oracle

O **Assistente de upsizing do Oracle** apresenta um comportamento semelhante ao do **Assistente de upsizing do SQL Server**. Para obter instruções passo-a-passo, consulte [Assistente de upsizing do Oracle](#). Para obter informações específicas sobre servidores Oracle, consulte a documentação do Oracle.

Iniciando o Assistente de upsizing do Oracle

Após criar uma conexão definida ou uma fonte de dados ODBC em conexão com um servidor Oracle e completar os preparativos necessários no cliente e no servidor, você estará pronto para começar a migração.

► Para iniciar o Assistente de upsizing do Oracle

- 1 No menu **Ferramentas**, selecione **Assistentes** e, em seguida, selecione **Upsizing**.
- 2 Na caixa de diálogo **Seleção de assistente**, selecione **Assistente de upsizing do Oracle**.
- 3 Siga as instruções nas telas do assistente.
Você pode selecionar **Cancelar** em qualquer momento para sair do assistente; o assistente não executa nenhuma ação no servidor até você selecionar **Finalizar**.
- 4 Quando estiver pronto para migrar, selecione **Finalizar**.
Finalizar está disponível depois que você fornecer as informações básicas necessárias para a migração. Se você selecionar **Finalizar** antes de completar todas as telas do assistente, o **Assistente de upsizing do Oracle** usará valores padrão para as telas restantes.

Depois de selecionar **Finalizar**, o **Assistente de upsizing do Oracle** começa a exportar o banco de dados para o servidor.