

5. Storage Architecture

Este capítulo apresenta a arquitetura de armazenamento do DBMaker. A arquitetura de armazenamento do DBMaker é composta pelo nível lógico e pelo nível físico.

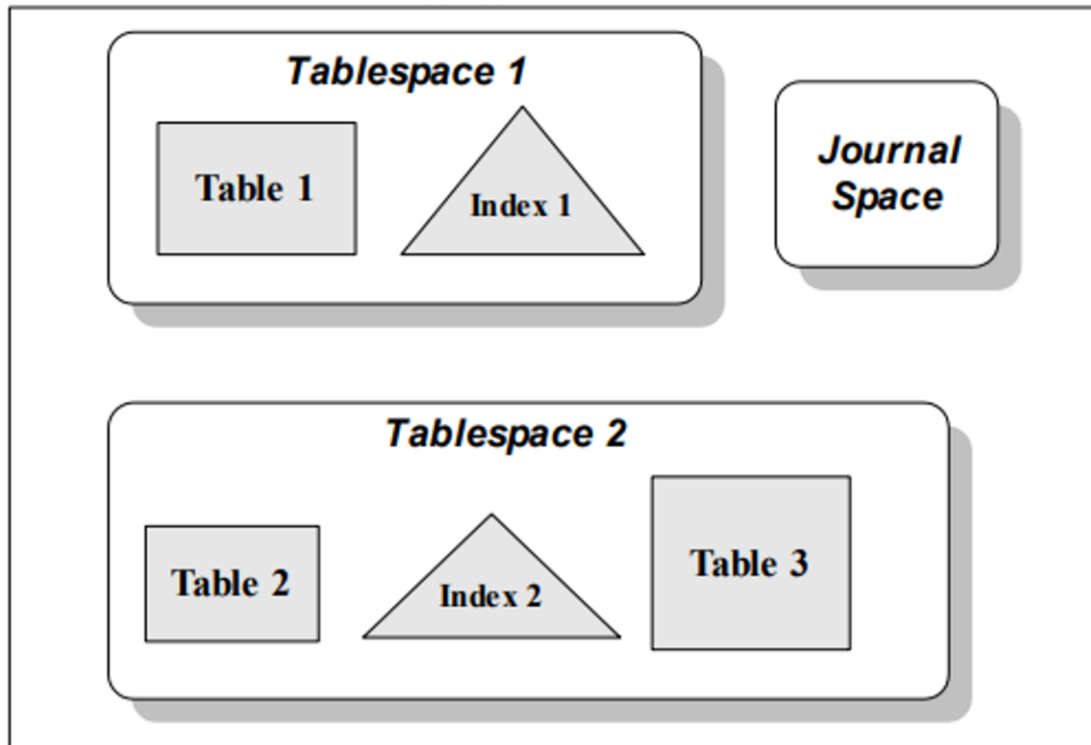
O nível lógico é a visão apresentada aos usuários e organiza os dados no banco de dados de uma forma fácil de entender. O nível físico consiste em arquivos do sistema operacional que correspondem às informações nos tablespaces, mas que são gerenciados pelo DBMaker e ocultados do usuário.

Este capítulo também explica como controlar a alocação de armazenamento de um banco de dados usando tablespaces e arquivos.

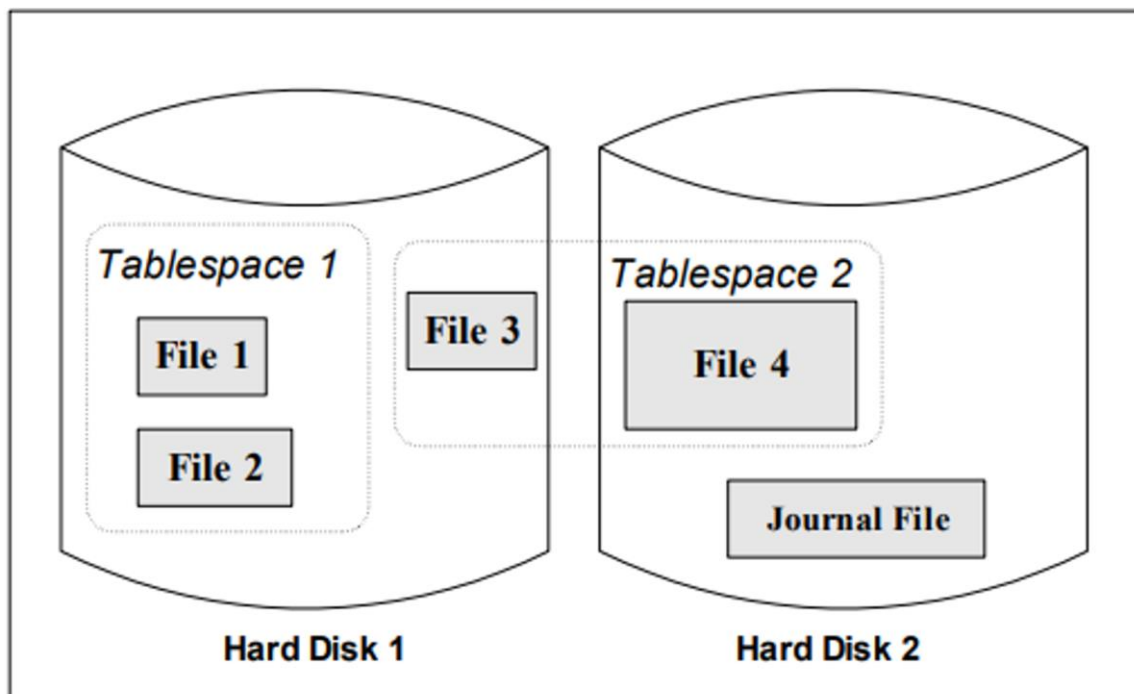
5.1 Architecture

Uma base de dados DBMaker é composta por uma ou mais divisões lógicas conhecidas como **tablespaces**. Tablespaces são a principal estrutura de armazenamento lógico no DBMaker. Do ponto de vista lógico, um tablespace contém uma ou mais tabelas e índices, conforme mostrado na Figura. Na visão física, um tablespace é o armazenamento lógico que consiste em um ou mais arquivos do sistema operacional, conforme mostrado na Figura 5-2.

Database



Database



5.2 File Types

Dez tipos diferentes de arquivos do sistema operacional são usados no DBMaker para armazenar diferentes aspectos de um banco de dados: dados do sistema e arquivos BLOB do sistema, dados do usuário e arquivos BLOB do usuário, arquivos de log do sistema, um arquivo temporário do sistema, arquivos definidos pelo usuário, arquivos de log do DBMaker, arquivos de backup e um arquivo de log de replicação de tabela. O arquivo de dados do sistema, o arquivo BLOB do sistema, os arquivos de dados do usuário e os arquivos BLOB do usuário são de principal importância no que diz respeito à arquitetura de armazenamento de banco de dados e tablespaces. Os arquivos de diário desempenham um papel importante no armazenamento de registros de transações realizadas no banco de dados e são vitais para backup e recuperação do banco de dados.

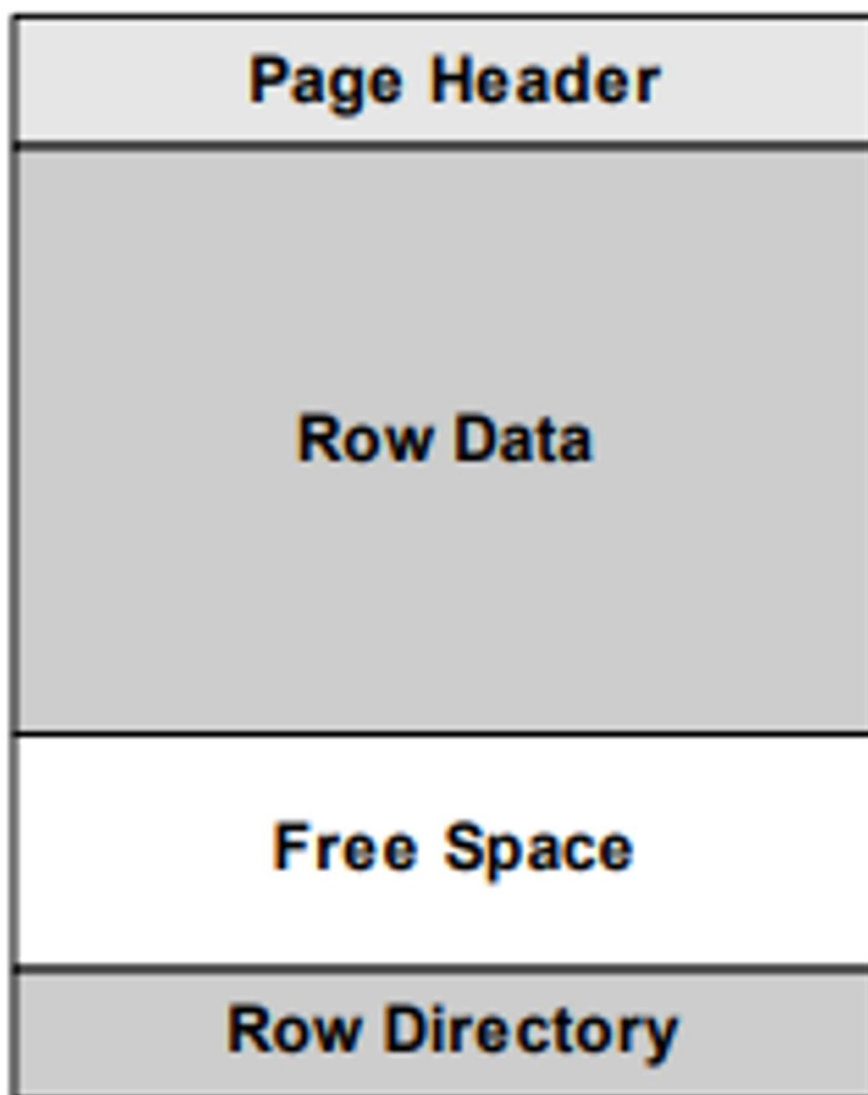
Para melhorar o desempenho do banco de dados, o DBMaker organiza os dados em dois tipos diferentes de arquivos: arquivos de dados e arquivos de Objeto Binário Grande (BLOB). Dados BLOB consistem em objetos de dados grandes, como imagens, voz ou texto extenso, que não podem ser compactados em uma página. O DBMaker armazena os dados BLOB em arquivos BLOB específicos e armazena as linhas de dados e chaves de índice nos arquivos de dados. Para alcançar alto desempenho, o DBMaker gerencia esses dois tipos de arquivo de maneiras diferentes.

User Data Files

Arquivos de dados são compostos por páginas, enquanto arquivos BLOB são compostos por quadros. O tamanho máximo de ambos os arquivos de dados e BLOB é de 8 TB. No entanto, existem duas diferenças principais entre quadros e páginas:

- O tamanho de uma página pode ser de 4 KB, 8 KB, 16 KB ou 32 KB, conforme definido pela palavra-chave `DB_PgSiz` no arquivo `dmconfig.ini` ao criar um banco de dados.
- Uma página pode conter mais de uma tupla, mas um quadro contém apenas um único item de dados BLOB.

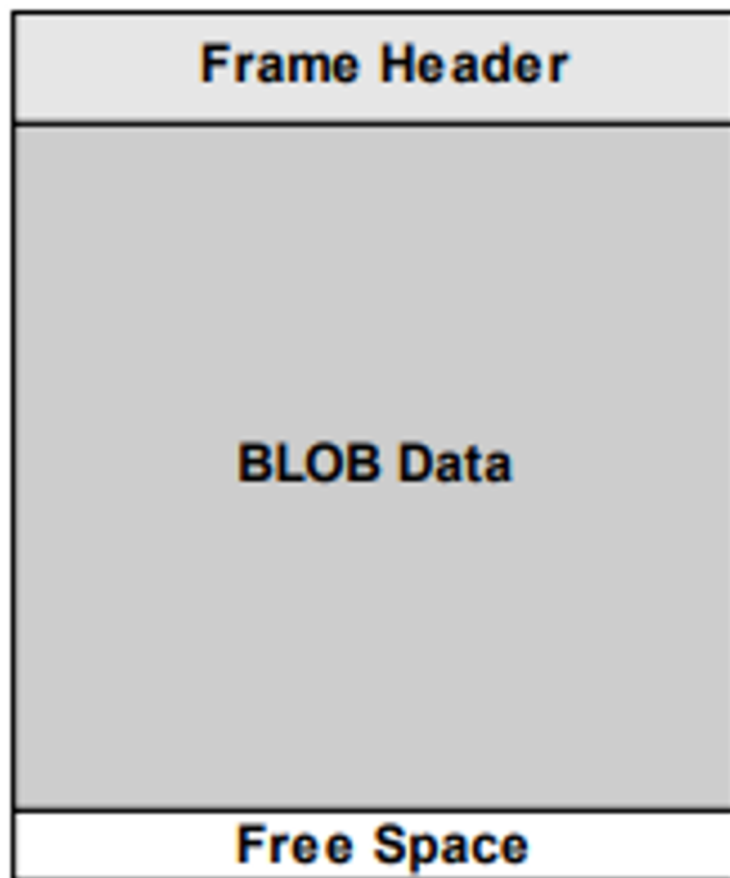
Uma página de dados é a menor unidade de armazenamento usada por arquivos de dados. O formato da página de dados é semelhante, independentemente de a página de dados armazenar dados de tabela ou de índice. Uma página de dados contém quatro seções: o cabeçalho da página, os dados das linhas, o espaço livre e o diretório de linhas.



O cabeçalho da página contém informações gerais da página para o sistema DBMaker. A área de dados das linhas contém os dados reais da tabela ou do índice que são exibidos como linhas e colunas ao visualizar uma tabela ou índice, e o diretório de linhas contém informações sobre as linhas na página. O espaço livre é o espaço disponível naquela página que ainda não foi utilizado para armazenar dados.

User BLOB Files

Um quadro BLOB é a menor unidade de armazenamento usada por arquivos BLOB. O tamanho do quadro BLOB só pode ser definido para um valor diferente do padrão antes da criação de um banco de dados. O tamanho mínimo do quadro é de 8 KB, e o tamanho máximo é de 256 KB. Um quadro BLOB contém três seções: o cabeçalho do quadro, os dados BLOB e o espaço livre. Para mais informações sobre arquivos BLOB, consulte o Capítulo 7, Gerenciamento de Grandes Objetos.



Assim como o cabeçalho de página, o cabeçalho do quadro contém informações gerais do quadro para o sistema DBMaker. A área de dados BLOB contém os dados BLOB, e cada quadro pode conter apenas um único item BLOB. No entanto, dados BLOB que são maiores do que o tamanho do quadro podem ser distribuídos por vários quadros. O espaço livre é o espaço disponível naquela página que não foi utilizado para armazenar dados BLOB..

Journal Files

O journal do DBMaker é composto por um ou vários arquivos físicos de journal. Internamente, um arquivo de journal é composto por blocos, onde cada bloco possui 512 bytes. Toda ação que causa uma mudança no sistema de banco de dados é registrada por um registro de journal. Registros de journal são os elementos lógicos no journal, e vários registros de journal podem ser compactados em um bloco de journal, ou um único registro de journal pode se estender por vários blocos. Um registro de journal pertencente a uma transação ativa não pode ser reutilizado.

Todos os arquivos de journal formam um anel de registros de journal; os registros de journal são gravados em blocos de journal sequenciais do início ao fim do arquivo. Se o banco de dados foi configurado para ter mais de um arquivo de journal, o DBMaker muda automaticamente para um novo arquivo de journal quando o arquivo atual se enche. Caso contrário, os registros de journal serão gravados sobre os blocos de journal no início do arquivo de journal. Quando todos os arquivos de journal estão preenchidos por transações ativas, a transação atual será abortada porque não há blocos disponíveis; isso é chamado de journal cheio.

Além dos registros de journal, um arquivo de journal contém alguns blocos para registrar o status do journal, chamados blocos de status do journal. Estes são usados durante a recuperação ou restauração do banco de dados. A recuperação e restauração serão descritas em seções posteriores.

O DBMaker mantém buffers de blocos de journal na memória para acelerar o acesso aos arquivos. Antes que os dados realmente modificados sejam gravados em disco, o registro de journal é gravado em disco usando o protocolo Write-Ahead-Log (WAL). Quando o buffer de journal está cheio ou uma transação é confirmada, o buffer será despejado nos arquivos de journal de acordo com o protocolo WAL.

JOURNAL PARAMETERS IN DMCONFIG.INI

Vários parâmetros de arquivos de journal podem ser configurados para melhorar o desempenho do banco de dados.

- **DB_JnFil** - Especifica os nomes dos arquivos de journal. Pode-se especificar de um a oito nomes de arquivos de journal. Cada nome de arquivo de journal é separado por uma vírgula ou um espaço.

Exemplo: O banco de dados terá sete arquivos de journal especificados em unidades diferentes para melhorar o desempenho.

```
DB_JnFil = myDb.jn1, myDb.jn2, myDb.jn3, /disk1/usr/myDb.jn4,
myDb.jn5,
/disk2/usr/myDb.jn6, myDb.jn7
```

- **DB_JnISz** - Especifica o tamanho de um arquivo de journal. As unidades podem ser **M** (megabytes), **G** (gigabytes) e **page** (página), sendo que a unidade padrão é **page** (o tamanho da página de journal é definido por **DB_PgSiz**). O tamanho do arquivo de journal é:

Decida um tamanho razoável para os arquivos de journal ao criar um banco de dados. Como mencionado na seção anterior, quando todos os arquivos de journal estiverem preenchidos, a transação atual pode ser abortada devido a um journal cheio. Portanto, um tamanho pequeno para o arquivo de journal pode fazer com que uma transação longa seja abortada pelo sistema. Se as operações do banco de dados envolverem transações longas, escolha um tamanho maior para o arquivo de journal ou utilize mais arquivos de journal.

- **DB_NJnIB** - Especifica o tamanho de um buffer de journal como um múltiplo de páginas de journal (o tamanho da página de journal é definido por **DB_PgSiz**).

RESIZING JOURNAL SPACE

Se mensagens de journal cheio forem frequentemente encontradas enquanto um banco de dados está em execução, aumentar o tamanho dos arquivos de journal melhorará o desempenho do banco de dados. No DBMaker 3.0, backups anteriores não podem ser usados para restaurar um banco de dados a um ponto específico no tempo após redimensionar os arquivos de journal. No entanto, em versões posteriores à 3.0, isso é permitido. Para proteger o banco de dados contra falhas de disco, faça um backup completo imediatamente após redimensionar os arquivos de journal.

Para redimensionar um arquivo de journal, um DBA deve realizar o seguinte:

1. **Determine o número e o tamanho dos arquivos de journal necessários**, estimando o espaço em disco necessário para lidar com as maiores transações.
2. **Desligue o banco de dados.**
3. **Atualize o arquivo dmconfig.ini** e reespecifique os seguintes dois parâmetros: **DB_JnFil** e **DB_JnISz**.
 - a. **Nota:** Essas configurações também podem ser alteradas nas configurações avançadas – na página de armazenamento do assistente de início de banco de dados do JServer Manager.
4. **Defina o modo de início para o novo modo de journal** no dmconfig.ini: **DB_SMode = 2**.
 - a. **Nota:** Esta configuração também pode ser alterada nas configurações avançadas – na página de início do banco de dados do assistente de início do JServer Manager.
5. **Reinicie o banco de dados.**

6. **Redefina o modo de início para normal** no dmconfig.ini: **DB_SMode = 1**.
 - a. **Nota:** Esta configuração também pode ser alterada na página de início do banco de dados da ferramenta JConfiguration.
7. **Realize um backup completo online** se o banco de dados estiver no modo **BACKUP-DATA** ou **BACKUP-DATA-AND-BLOB**.

Tablespaces

Um banco de dados DBMaker é particionado em áreas lógicas menores conhecidas como tablespaces. Tablespaces são áreas lógicas de armazenamento que permitem que o banco de dados seja subdividido em áreas mais gerenciáveis. Cada tablespace contém um ou mais arquivos do sistema operacional. Antes de começar a usar tablespaces e arquivos no DBMaker, é importante se familiarizar com os termos abaixo.

TIPOS DE TABLESPACES

Os tablespaces podem ser de tamanho fixo ou automaticamente extensíveis. Tablespaces de tamanho fixo são chamados de **tablespaces regulares**, e tablespaces que podem ter seu tamanho automaticamente estendido são chamados de **tablespaces autoextend**. O DBMaker também possui um tablespace especial chamado **system tablespace**.

O SYSTEM TABLESPACE

Todos os bancos de dados DBMaker têm pelo menos dois tablespaces: um system tablespace (**SYSTABLESPACE**) e um default tablespace (**DEFTABLESPACE**). O DBMaker gera um system tablespace para registrar a tabela de catálogo do sistema sempre que um banco de dados é criado. As tabelas de catálogo do sistema armazenam informações sobre todo o banco de dados.

O DEFAULT TABLESPACE

O default tablespace armazena as tabelas de usuário quando os usuários não especificam qual tablespace deve ser alocado. No entanto, criar tablespaces adicionais para o armazenamento de tabelas de usuário é mais flexível e eficiente.

O TEMPORARY TABLESPACE

O temporary tablespace (**TMPTABLESPACE**) é usado apenas para armazenar tabelas temporárias externas (**ETT**). O temporary tablespace também é um tablespace autoextend. Ele possui exatamente dois tipos de arquivos: arquivos de dados e

arquivos BLOB. O nome lógico dos arquivos de dados é **DB_TMPDB**, e o nome físico é **DB_TMPDir/DBNAME.TDB**; o nome lógico dos arquivos BLOB é **DB_TMPBB**, e o nome físico é **DB_TMPDir/DBNAME.TBB**.

Quando os usuários executam a instrução "create temporary table" ou "select into", as ETTs serão geradas e armazenadas no **TMPTABLESPACE**. Os usuários podem criar tabelas temporárias no TMPTABLESPACE (o sistema, por padrão, armazenará as ETTs no TMPTABLESPACE), mas não podem criar nenhuma tabela permanente no TMPTABLESPACE. Os usuários podem executar "ALTER TABLESPACE TMPTABLESPACE SET AUTOEXTEND OFF/ON" e "ALTER DATAFILE DB_TMPDB/DB_TMPBB ADD n PAGES;", mas não podem adicionar ou remover arquivos do TMPTABLESPACE. O TMPTABLESPACE é criado quando um banco de dados é criado, e o tamanho será redefinido para o tamanho padrão quando o banco de dados for iniciado.

- Os usuários não podem criar tabelas temporárias em nenhum outro tablespace que não seja o TMPTABLESPACE.
- Os usuários não podem criar tabelas permanentes no TMPTABLESPACE.
- Os usuários não podem adicionar ou remover arquivos do TMPTABLESPACE.
- Os usuários não podem excluir o TMPTABLESPACE.

REGULAR TABLESPACES

Um **regular tablespace** tem tamanho fixo e contém um ou mais arquivos de dados. Se um arquivo em um regular tablespace for pequeno demais para armazenar todos os dados destinados a ele, ele pode ser ampliado manualmente. O número máximo de arquivos que pode ser contido em um regular tablespace é 32.767. O número total de páginas em todos os arquivos de um tablespace não deve exceder 8 TB.

AUTOEXTEND TABLESPACES

Os **autoextend tablespaces** crescem automaticamente conforme necessário. Arquivos em um autoextend tablespace se expandem automaticamente; o DBMaker os expande na ordem inversa à de inserção. Isso significa que o último arquivo de dados adicionado será o primeiro a se expandir se for necessário mais espaço.

Qualquer autoextend tablespace pode ser alterado para um regular tablespace para evitar que ele continue a se expandir. Da mesma forma, um regular tablespace pode ser alterado para um autoextend tablespace se o espaço estiver esgotado. Alternativamente, novos arquivos podem ser adicionados ou arquivos existentes podem ser ampliados para expandir um regular tablespace. Arquivos de dispositivos

brutos só podem ser usados com regular tablespaces e não podem ser usados com autoextend tablespaces.

O DBMaker cria automaticamente um autoextend tablespace chamado **system tablespace** ao criar um banco de dados. Ao criar qualquer outro tablespace, os regular tablespaces serão usados por padrão. Para evitar que o tablespace padrão cresça sem limite, altere-o para um regular tablespace.

O arquivo dmconfig.ini registra o número de páginas para cada arquivo de dados. O número de páginas em um arquivo de dados é o tamanho inicial de um arquivo pertencente a um autoextend tablespace e é o tamanho real de um arquivo pertencente a um regular tablespace.

5.3 Managing Tablespaces and Files

Existem vários aspectos a serem considerados ao gerenciar tablespaces e arquivos para um banco de dados. Por exemplo, o tamanho e o tipo de novos tablespaces devem ser determinados no momento da criação do banco de dados; tablespaces adicionais podem ser criados posteriormente; tablespaces autoextend podem ser alterados para regular tablespaces e vice-versa; arquivos de dados podem ser adicionados a tablespaces; o tamanho dos arquivos em tablespaces pode ser definido e alterado; arquivos de dados e tablespaces podem ser removidos quando não são mais necessários; e tabelas podem ser movidas para outros tablespaces.

Para gerenciar tablespaces, pode-se usar a **JDBA Tool** ou uma combinação de comandos **dmSQL** e modificações no arquivo dmconfig.ini. A JDBA Tool oferece uma interface de usuário intuitiva para todas as rotinas de gerenciamento de tablespaces. Para mais informações sobre como usar a JDBA Tool para gerenciar tablespaces, consulte o **JDBA Tool User's Guide**.

Cada banco de dados DBMaker tem pelo menos um tablespace chamado **system tablespace**. Quando um banco de dados é criado, o DBMaker gera cinco arquivos: um arquivo de dados do sistema, um arquivo de dados do usuário, um arquivo BLOB do sistema, um arquivo BLOB do usuário e um arquivo de journal. O arquivo de dados do sistema, o arquivo BLOB do sistema e o arquivo de journal são colocados no system tablespace. Esses três arquivos registram as tabelas de catálogo do sistema para todo o banco de dados. O arquivo de dados do usuário e o arquivo BLOB do usuário são colocados no **default user tablespace**.

As tabelas de usuário são armazenadas no default user tablespace, a menos que tablespaces adicionais sejam criados. Criar tablespaces adicionais para armazenar tabelas de usuário é uma abordagem mais flexível e eficiente.

Initial Setting of System Files and Tablespace

O DBMaker gera o **system tablespace** e os três arquivos do sistema (o arquivo de dados do sistema, o arquivo BLOB do sistema e o arquivo de journal) ao criar um novo banco de dados. Esses arquivos são usados para manter um registro do esquema do banco de dados e das transações. O DBMaker concatena o nome do banco de dados com as extensões de arquivo **.SDB**, **.SBB** e **.JNL** para nomear, respectivamente, os arquivos de dados do sistema, BLOB e journal. Se os tamanhos dos arquivos de dados do sistema, BLOB e journal não forem especificados, eles serão criados com tamanhos padrão de **200 × DB_PgSiz KB**, **20 KB** e **4.000 KB**, respectivamente. Para usar nomes diferentes para os arquivos do sistema, especifique-os no arquivo dmconfig.ini ou na página de armazenamento da **JConfiguration Tool**.

Exemplo:

Para especificar os nomes dos arquivos do sistema no arquivo dmconfig.ini:

```
[MY_DB] ;database name
DB_DbDir = /disk1/usr ;database directory
DB_DbFil = datafile.sdb ;data file
DB_BbFil = blobfile.sbb ;BLOB file
DB_JnFil = jrnlfilfile.jnl ;journal file
```

Se esses valores estiverem no arquivo dmconfig.ini no momento em que o comando **CREATE DB** for executado, o DBMaker criará os três arquivos do sistema como antes, mas dessa vez usará os nomes fornecidos acima em vez dos nomes padrão. Nesse caso, o arquivo de dados do sistema será nomeado **datafile.sdb**, o arquivo BLOB do sistema será nomeado **blobfile.sbb**, e o arquivo de journal será nomeado **jrnlfilfile.jnl**.

O **system tablespace** é criado como **autoextend** por padrão; portanto, o tamanho do system tablespace é apenas um tamanho inicial, não uma limitação. Para limitar o espaço em disco usado pelo system tablespace, altere-o para um **regular tablespace** usando o comando **ALTER TABLESPACE**.

Uma vez que todo o espaço em um regular system tablespace esteja esgotado, as únicas maneiras de ampliá-lo são adicionar arquivos ao regular system tablespace, aumentar o tamanho dos arquivos do sistema adicionando páginas ou alterar o tipo do tablespace para autoextend.

Initial Setting of Default User Files and Tablespace

O DBMaker gera o **default user tablespace** e os dois arquivos (o arquivo de dados do usuário e o arquivo BLOB do usuário) ao criar um novo banco de dados. Esses arquivos são usados para armazenar dados de usuários. O DBMaker concatena o nome do banco de dados com as extensões de arquivo **.DB** e **.BB** para nomear, respectivamente, os arquivos de dados e BLOB do usuário.

A menos que seus tamanhos sejam especificados antecipadamente, os arquivos de dados do usuário e BLOB do usuário serão criados com tamanhos padrão de **200 × DB_PgSiz KB** e **20 KB**, respectivamente. Para usar nomes diferentes para os arquivos padrão do usuário, especifique-os no arquivo `dmconfig.ini` ou na página de armazenamento da **JConfiguration Tool**.

Exemplo: Para especificar os nomes dos arquivos padrão do usuário no arquivo `dmconfig.ini`:

```
[MY_DB] ;database name
DB_UsrDb = /disk1/usr/f1.db 200 ;data file
DB_UsrBb = /disk1/usr/f1.bb 20 ;blob file
```

Se um banco de dados for criado com esses valores no arquivo `dmconfig.ini`, o DBMaker criará os dois arquivos usando os nomes fornecidos acima em vez dos nomes padrão. Nesse caso, o arquivo de dados padrão será nomeado **f1.db** com um tamanho de **200 páginas**, e o arquivo BLOB padrão será nomeado **f1.bb** com um tamanho de **20 frames**.

O **default tablespace** é inicialmente criado como um **autoextend tablespace**, então seu tamanho inicial não é uma limitação.

Creating Tablespaces

Tablespaces adicionais podem ser criados para conter outros arquivos de dados e BLOBs. Um tablespace pode ser criado usando **dmSQL** ou a **JDBA Tool**. Detalhes sobre como criar tablespaces com **dmSQL** podem ser encontrados no *SQL Command and Function Reference*. Detalhes sobre como criar tablespaces com a **JDBA Tool** podem ser encontrados no *JDBA Tool User's Guide*.

Um tablespace deve conter pelo menos um arquivo de dados, mas arquivos adicionais no tablespace podem ser arquivos de dados ou arquivos BLOB. O DBMaker cria um novo arquivo como um arquivo de dados por padrão; o tipo de arquivo deve ser especificado como BLOB para criar um arquivo BLOB.

Antes de criar um novo tablespace, especifique o tamanho e os nomes dos arquivos de dados associados ao tablespace no arquivo dmconfig.ini.

Exemplo 1: As seguintes entradas são necessárias no dmconfig.ini para especificar três arquivos nomeados **f1**, **f2** e **f3** com nomes de arquivos do sistema operacional e tamanhos de página:

```
[MY_DB] ;database name
f1 = /disk1/usr/f1.dat 1000 ;a data file with 1000 pages
f2 = /disk2/usr/f2.dat 500 ;a data file with 500 pages
f3 = /disk1/usr/f3.blb 1000 ;a blob file with 1000 frames
```

Para criar um regular tablespace **ts_reg** com dois arquivos de dados e um arquivo BLOB, com os arquivos de dados colocados em discos diferentes:

```
dmSQL> CREATE TABLESPACE ts_reg DATAFILE f1, f2, f3 TYPE=BLOB;
```

Exemplo 2:

Para criar um tablespace autoextensível com um arquivo de dados e um arquivo BLOB. O tamanho inicial do arquivo de dados é de 500 páginas e o tamanho inicial do arquivo BLOB é de 20 páginas. Se o arquivo de dados ou o arquivo BLOB ficar cheio, ele será expandido automaticamente.

```
[MY_DB] ;database name
f4 = /usr/f4.dat 500 ;a data file with initial 500 pages
f5 = /usr/f5.blb 20 ;a blob file with initial 20 pages
```

Para criar um novo tablespace que usa esses arquivos:

```
dmSQL> CREATE AUTOEXTEND TABLESPACE ts_aut DATAFILE f4 TYPE=DATA, f5
TYPE=BLOB;
```

RAW DEVICE FILES

Nos sistemas UNIX, se o prefixo do nome do arquivo físico for **/dev/**, o DBMaker considerará isso como um arquivo de dispositivo raw. Um arquivo de dispositivo raw suporta um acesso mais rápido do que um arquivo normal, o que pode melhorar o desempenho do banco de dados. Crie um arquivo de dispositivo raw em um disco

antes de associá-lo a um tablespace. Apenas **regular tablespaces** podem conter arquivos de dispositivo raw.

Exemplo: Para especificar um arquivo de dispositivo raw, **f2**, com o nome de arquivo do sistema operacional **/dev/rawf2** e **5000 páginas**, adicione o seguinte ao arquivo dmconfig.ini:

```
[MY_DB] ;database name  
f2 = /dev/rawf2 5000 ;a raw device file with 5000 pages
```

Para criar um tablespace regular ts_raw, contendo o arquivo de dispositivo raw mencionado anteriormente.

```
dmSQL> CREATE TABLESPACE ts_raw DATAFILE f2;
```

Expanding a Regular Tablespace

Há três maneiras de expandir um **regular tablespace**:

- Adicionar novos arquivos a um regular tablespace
- Adicionar páginas a arquivos existentes em um regular tablespace
- Definir **autoextend** como **ON**

Todas essas funções podem ser realizadas com a **JDBA Tool** ou uma combinação de comandos SQL e modificações no arquivo dmconfig.ini. O exemplo a seguir mostra como expandir um regular tablespace editando o arquivo dmconfig.ini e usando comandos SQL.

Exemplo: Antes de emitir um comando, informe ao DBMaker o nome do arquivo físico que corresponde ao nome lógico do arquivo **file_blob** adicionando uma declaração ao arquivo dmconfig.ini na seção correspondente a esse banco de dados. Neste caso, **file_blob** é o nome lógico que será usado no banco de dados, e **file.blb** é o nome do arquivo físico utilizado pelo sistema operacional:

```
file_blob = file.blb 120
```

Para adicionar um novo arquivo BLOB chamado file_blob a um tablespace regular com 120 quadros no tablespace chamado ts_app:

```
dmSQL> ALTER TABLESPACE ts_app ADD DATAFILE file_blob TYPE = BLOB;
```

Para adicionar 100 páginas a um arquivo de dados existente chamado file_data no tablespace regular denominado ts_app.

```
dmSQL> ALTER DATAFILE file_data ADD 100 PAGES;
```

Após alterar o tamanho do arquivo adicionando as páginas extras, o DBMaker atualizará o número de páginas para o arquivo no arquivo dmconfig.ini para refletir o novo valor.

Expanding an Autoextend Tablespace evenly

Há três maneiras de expandir um **autoextend tablespace**:

1. **Expandir sempre a partir do primeiro arquivo:** Pode proporcionar um bom desempenho, mas todos os arquivos no tablespace não serão expandidos de maneira uniforme.
2. **Expandir sempre a partir do menor arquivo:** Pode manter todos os arquivos em um tablespace com expansão uniforme, mas o desempenho pode ser ruim porque todas as linhas de uma tabela podem ser distribuídas entre todos os arquivos.
3. **Expandir primeiramente o menor arquivo:** Depois, continue expandindo este arquivo e não passe para o segundo menor arquivo até que o tamanho do arquivo atual seja maior do que a soma do tamanho do segundo menor arquivo e o valor de **DB_ExtHd**. Dessa forma, tanto o desempenho quanto o equilíbrio do tamanho dos arquivos podem ser considerados.

Para cada uma das opções acima, o DBMaker irá para o próximo arquivo para expandir caso o arquivo selecionado não possa ser expandido devido a disco cheio, limitação do sistema de arquivos ou limitação de armazenamento do DBMaker. Se o próximo arquivo também não puder ser expandido pelos mesmos motivos, o DBMaker continuará para o próximo arquivo até que todos os arquivos sejam expandidos. Consulte **DB_ExtHd** para mais informações.

Todas essas funções podem ser realizadas com a **JConfiguration Tool** ou uma combinação de comandos SQL e modificações no arquivo dmconfig.ini. O exemplo a seguir mostra como expandir um autoextend tablespace editando o arquivo dmconfig.ini e usando comandos SQL

Exemplo:

Use a palavra-chave **DB_ExtNp** e **DB_ExtHd** ou chame SETSYSTEMOPTION('EXTHD','novoValor') para definir a estratégia de expansão de

um autoextend tablespace. O DBMaker expandirá automaticamente o autoextend tablespace de acordo com a estratégia definida pelos usuários.

No arquivo de configuração dmconfig.ini, especifique que o tamanho do DBMaker para expandir um autoextend tablespace é **30 páginas**, e o valor limite para expandir um arquivo repetidamente é **100 páginas**:

```
DB_ExtNp=30  
DB_ExtHd=100
```

Adicione os cinco arquivos seguintes modificando o arquivo dmconfig.ini:

```
D1=/home/dbmaker/testdb/D1 10  
B1=/home/dbmaker/testdb/B1 30  
D2=/home/dbmaker/testdb/D2 50  
B2=/home/dbmaker/testdb/B2 70  
D3=/home/dbmaker/testdb/D3 100
```

Digite os seguintes comandos no prompt dmSQL para criar o tablespace autoextensível TS.

```
dmSQL> CREATE AUTOEXTEND TABLESPACE TS DATAFILE D1 TYPE=DATA, B1  
TYPE=BLOB, D2  
TYPE=DATA, B2 TYPE=BLOB, D3 TYPE=DATA;
```

Crie a tabela tb_t1 com a coluna c1 char(5000) no tablespace TS e insira várias linhas na tabela tb_t1. O tablespace TS será expandido automaticamente. De acordo com a nova regra, os arquivos serão expandidos da seguinte forma:

```
1st, extend the smallest file D1, add 30 pages. Now, D1=40, D2=50,  
D3=100.  
2nd, extend D1, add 30 pages. Now, D1=70, D2=50, D3=100.  
3rd, extend D1, add 30 pages. Now, D1=100, D2=50, D3=100.  
4th, extend D1, add 30 pages. Now, D1=130, D2=50, D3=100.  
5th, extend D1, add 30 pages. Now, D1=160, D2=50, D3=100.  
6th, extend D2, because D1 > D2+EXTHD. Add 30 pages, D1=160, D2=80,  
D3=100.  
7th, extend D2, until D2 > D3(the smallest)+EXTHD, then extend D3.
```

NOTA: Os arquivos **B1** e **B2** não serão expandidos, pois não há campo BLOB em **tb_t1**.


```
dmSQL> CALL SETSYSTEMOPTION('EXTHD','1000'); // changing EXTHD to 10000 pages
```

Durante a execução, os usuários podem chamar o procedimento armazenado do sistema GETSYSTEMOPTION para exibir o valor de EXTHD.

```
dmSQL> CALL GETSYSTEMOPTION('EXTHD',?); //reporting the current value of EXTHD
```

Adding Files to Tablespaces

Aumente o tamanho de um tablespace regular ou de um tablespace com autoextensão e, conseqüentemente, do banco de dados, criando e adicionando novos arquivos a ele. Para aumentar o espaço disponível para inserir ou atualizar linhas de dados, adicione arquivos de dados a um tablespace regular ou com autoextensão. Para aumentar o espaço disponível para armazenar dados BLOB, adicione arquivos BLOB. Os arquivos podem ser adicionados a um tablespace usando a ferramenta JDBC ou modificando o arquivo dmconfig.ini e inserindo comandos no prompt do dmSQL. A seguir, uma diretriz para adicionar arquivos modificando o arquivo dmconfig.ini e inserindo comandos no prompt do dmSQL.

Certifique-se de primeiro adicionar linhas ao arquivo dmconfig.ini que especifiquem o tamanho e os nomes dos novos arquivos ao adicionar arquivos de dados a um tablespace. Além disso, especifique o tipo de arquivo como BLOB ao adicionar arquivos BLOB; caso contrário, o DBMaker criará um arquivo de dados por padrão.

Exemplo 1:

Para especificar no arquivo dmconfig.ini um arquivo de dados chamado f7 com 3.000 páginas, onde o nome do arquivo do sistema operacional é /disk1/usr/f7.dat:

```
[MY_DB] ;database name  
f7 = /disk1/usr/f7.dat 3000 ;a data file with 3000 pages
```

Para adicionar o arquivo de dados f7 ao tablespace ts_reg:

```
dmSQL> ALTER TABLESPACE ts_reg ADD DATAFILE f7;
```

Exemplo 2:

Para especificar no arquivo dmconfig.ini um arquivo BLOB chamado f8 com 5.000 páginas, onde o nome do arquivo do sistema operacional é /disk1/usr/f8.blb:

```
[MY_DB] ;database name  
f8 = /disk1/usr/f8.blb 5000 ;a blob file with 5000 frames
```

Para adicionar o arquivo BLOB ao tablespace ts_reg:

```
dmSQL> ALTER TABLESPACE ts_reg ADD DATAFILE f8 TYPE=BLOB;
```

O tipo de arquivo deve ser especificado, caso contrário, ele será adicionado como um arquivo de dados por padrão.

Adding Pages to Files in Tablespaces

Além de adicionar novos arquivos a um tablespace regular para expandir o banco de dados, ele também pode ser aumentado pelo incremento do tamanho de arquivos existentes. O tamanho de arquivos em tablespaces autoextensíveis pode ser aumentado pela adição de páginas, o que pré-aloca espaço em disco para melhorar o desempenho. Quando o tamanho de um arquivo é alterado, o DBMaker atualiza automaticamente a entrada do arquivo no dmconfig.ini para refletir o número aumentado de páginas.

O tamanho do arquivo pode ser alterado usando a ferramenta JDBC ou inserindo o comando ALTER ``DATAFILE no prompt dmSQL. A seguir, orientações para alterar o tamanho do arquivo inserindo comandos no prompt dmSQL.

Exemplo:

Para alterar o tamanho e estender o arquivo f1 adicionando 100 páginas (o arquivo f1 já deve existir e estar associado a um tablespace):

```
dmSQL> ALTER DATAFILE f1 ADD 100 PAGES;
```

Changing Regular to Autoextend Tablespaces

Um administrador de banco de dados pode querer alterar um tablespace de regular para autoextensão quando:

- Está adicionando mais dados a um tablespace regular, mas o tablespace já cresceu ao ponto de preencher todos os arquivos pertencentes a esse tablespace, e o disco ainda possui espaço disponível.
- É desejado que o tablespace ocupe uma quantidade ilimitada de espaço.

Após criar um tablespace regular, o administrador de banco de dados pode alterá-lo para um tablespace com autoextensão usando a ferramenta JDBC ou o comando ALTER TABLESPACE no prompt de comando do dmSQL.

Exemplo:

Para converter o tablespace regular ts_reg em um tablespace autoextensível:

```
dmSQL> ALTER TABLESPACE ts_reg SET AUTOEXTEND ON;
```

Changing Autoextend Tablespaces to Regular

Tablespaces

Um administrador de banco de dados pode querer alterar um tablespace de autoextensão para regular quando:

- Deseja-se restringir a quantidade de espaço que um tablespace irá ocupar. Um tablespace com autoextensão pode crescer até preencher todo o espaço disponível em um disco.

Após criar um tablespace com autoextensão, o administrador de banco de dados pode alterá-lo para um tablespace regular usando a ferramenta JDBC ou o comando ALTER TABLESPACE no prompt de comando do dmSQL.

Exemplo:

```
dmSQL> ALTER TABLESPACE ts_reg SET AUTOEXTEND OFF;
```

Shrinking Tablespaces and Files

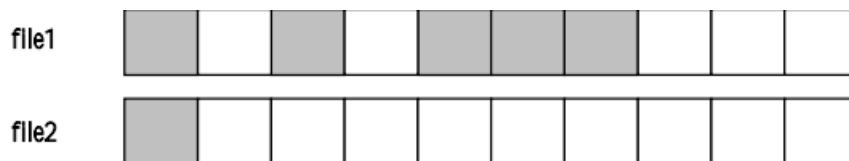
Os tablespaces podem ser reduzidos em tamanho se houver necessidade de alocar espaço em disco para outros usos. Dois comandos do dmSQL podem ser usados para reduzir o tamanho do tablespace: o comando SHRINK DATAFILE e o comando SHRINK TABLESPACE. O comando SHRINK DATAFILE atua em um arquivo especificado pelo usuário, enquanto o comando SHRINK TABLESPACE atua em todos os arquivos no tablespace especificado pelo usuário. Essas operações podem ser realizadas usando a ferramenta JDBC. As seções a seguir descrevem como usar comandos no prompt de comando do dmSQL para reduzir o tamanho do tablespace.

TRUNCATEONLY OPTION

O comando SHRINK com a opção TRUNCATEONLY remove páginas livres contíguas no final de qualquer arquivo de dados no qual seja executado. Ele não comprime o arquivo; se houver páginas livres entre as páginas usadas, elas permanecerão no arquivo. O administrador de banco de dados pode optar por truncar todas as páginas livres no final (sem a opção WITH n FREE PAGES) ou truncar páginas livres permitindo que um determinado número de páginas livres permaneça (WITH n FREE PAGES). A seguir, são apresentados exemplos de ambas as opções.

Without WITH n FREE PAGES Option

O comando SHRINK com a opção TRUNCATEONLY (sem a opção WITH n FREE PAGES) apenas trunca páginas livres contíguas no final de um arquivo. Por exemplo, o tablespace ts_shrink contém file1 e file2. Os diagramas a seguir, onde blocos cinza representam páginas usadas e blocos brancos representam páginas livres, representam o status das páginas de file1 e file2.



As páginas livres no final de ambos os arquivos podem ser removidas executando o comando SHRINK TABLESPACE em todo o tablespace ou executando o comando SHRINK DATAFILE em ambos os arquivos. A opção TRUNCATEONLY deve ser especificada. Os exemplos a seguir demonstram como fazer isso.

Exemplo1:

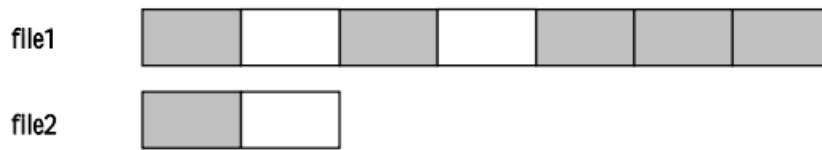
```
dmSQL> SHRINK TABLESPACE ts_shrink TRUNCATEONLY;
```

Exemplo 2:

```
dmSQL> SHRINK DATAFILE file1 TRUNCATEONLY;  
dmSQL> SHRINK DATAFILE file2 TRUNCATEONLY;
```

Após o truncating as páginas no final de ambos os arquivos foram removidas. A seguir, uma representação gráfica do status das páginas de ambos os arquivos:

Result



O arquivo file1 tem cinco páginas usadas que não são adjacentes.

Exemplo:

Para torná-las contíguas

```
dmSQL> SHRINK TABLESPACE ts_shrink COMPRESSONLY;
```

Result 2:



O comando SHRINK deve ser executado com o autocommit ativado. A opção COMPRESSONLY pode ser revertida. Se o banco de dados falhar, a operação COMPRESSONLY será totalmente concluída ou totalmente falhada após a recuperação da falha.

Existem alguns conflitos entre o comando SHRINK com a opção COMPRESSONLY e o uso de backup. O DBMaker não permite que esses dois comandos sejam executados ao mesmo tempo.

LIMITATIONS TO SHRINKING AND COMPRESSING TABLESPACES

As limitações gerais para esses comandos são:

- O comando SHRINK pode ser usado em arquivos de dados e BLOB, mas não em arquivos de journal.
- Apenas um usuário com autoridade DBA pode executar o comando SHRINK.
- O comando SHRINK requer que o autocommit esteja ativado.
- O comando SHRINK foi adicionado no DBMaker 3.7; versões anteriores do DBMaker não reconhecem este comando. Portanto, uma vez que um DBA execute o comando SHRINK e depois realize um backup incremental, versões anteriores do DBMaker não poderão restaurar o arquivo de backup do journal.
- A opção TRUNCATEONLY não pode ser revertida.
- A opção COMPRESSONLY não pode comprimir o tablespace SYSTABLESPACE.
- A opção COMPRESSONLY não verifica se as tabelas do usuário possuem uma coluna OID ou não. Uma coluna OID é usada para referenciar um registro em

outro lugar no banco de dados. Após usar COMPRESSONLY, uma coluna OID pode não apontar para o registro correto se o registro referenciado estiver no tablespace ou arquivo comprimido. A opção COMPRESSONLY não modifica as colunas OID nas tabelas do usuário.

- A opção COMPRESSONLY e o comando de backup não podem ser executados ao mesmo tempo.

Dropping Tablespaces

Se um tablespace estiver vazio ou contiver informações que não são mais necessárias, um administrador de banco de dados pode removê-lo do banco de dados. Qualquer tablespace em um banco de dados DBMaker, exceto o tablespace do sistema, pode ser removido. Para remover um tablespace, primeiro remova todas as tabelas no tablespace ou certifique-se de que ele já esteja vazio de tabelas. Para mais informações sobre como remover tabelas de um tablespace, consulte o Capítulo 6, Gerenciamento de Esquemas e Objetos de Esquema.

Remover um tablespace eliminará automaticamente todos os arquivos associados a ele, mas não os removerá do sistema de arquivos do sistema operacional. Esses arquivos ainda existirão no sistema de arquivos e só poderão ser removidos usando comandos do sistema operacional para recuperar o espaço em disco que ocupam. Os dados armazenados nos arquivos físicos correspondentes a um tablespace não são recuperáveis uma vez que os arquivos físicos tenham sido removidos do sistema de arquivos. Tenha cuidado ao remover arquivos associados a tablespaces, pois dados valiosos podem ser perdidos.

Os tablespaces podem ser removidos usando a ferramenta JDBC ou pelo comando DROP TABLESPACE no prompt de comando do dmSQL.

Exemplo: Para remover o tablespace ts_aut e todos os arquivos associados a ele:

```
dmSQL> DROP TABLESPACE ts_aut;
```

Dropping Files From a Tablespace

Os usuários podem remover arquivos de dados indesejados de um tablespace usando a ferramenta JDBC ou o comando ALTER TABLESPACE tablespace-name DROP DATAFILE file-name no prompt de comando do dmSQL. No entanto, ao usar o último método, o usuário deve remover fisicamente os arquivos de dados e atualizar manualmente as informações no dmconfig.ini após emitir e confirmar o comando ALTER TABLESPACE tablespace-name DROP DATAFILE.

Os arquivos de dados indesejados podem ser removidos de um tablespace com as seguintes condições:

- Os usuários não podem remover um arquivo de dados de um tablespace se ele for o único arquivo de dados nesse tablespace.
- O arquivo de dados a ser removido deve estar vazio.
- Os usuários não podem remover o arquivo de dados do sistema ou o arquivo de dados padrão do tablespace do sistema ou do tablespace padrão.

Exemplo: Para remover o arquivo de dados f4 do tablespace ts_aut.

```
dmSQL> ALTER TABLESPACE ts_aut DROP DATAFILE f4;
```

Read Only Tablespace

O tablespace somente leitura é um tablespace que não permite nenhuma atualização ou criação de novos objetos no tablespace.

Os tablespaces somente leitura têm várias vantagens:

- Elimina a necessidade de realizar backups contínuos. Um único backup após torná-lo somente leitura é suficiente.
- A recuperação se torna mais fácil.
- Tablespaces somente leitura têm menos sobrecarga do que tablespaces atualizáveis (sem bloqueios).
- Reduz o I/O.

Exemplo: Para tornar o tablespace ts_reg somente leitura.

```
dmSQL> ALTER TABLESPACE ts_reg SET READ ONLY;
```

Para tornar o tablespace ts_reg leitura e escrita.

```
dmSQL> ALTER TABLESPACE ts_reg SET READ WRITE;
```

Getting Information about Tablespaces and Files

Usando a ferramenta JDBA, é simples visualizar a estrutura dos tablespaces e arquivos dentro de um determinado tablespace. Os tablespaces são exibidos como parte da estrutura lógica de todos os objetos do banco de dados. Selecionar o nó de tablespaces na árvore expandirá a árvore para exibir todos os tablespaces no banco

de dados. Selecionar um tablespace na árvore exibirá todos os arquivos no tablespace, bem como detalhes sobre os arquivos, como tamanho, localização física, tipo de dado ou se o tablespace é extensível.

Alternativamente, use o dmSQL para selecionar todas as colunas da tabela do sistema SYSTABLESPACE para informações sobre tablespaces, ou SYSFILE para informações sobre arquivos de dados e BLOB do usuário. Exemplo 1: Para obter informações sobre tablespaces, como nomes dos tablespaces, se são tablespaces regulares ou com autoextensão, o número de arquivos associados aos tablespaces e o número total de páginas, consulte a tabela do sistema SYSTABLESPACE no catálogo do sistema:

```
dmSQL> SELECT * from SYSTABLESPACE;
```

Exemplo 2:

Para obter informações sobre arquivos de maneira semelhante, consulte a tabela do sistema SYSFILE para obter informações sobre nomes de arquivos, tipos de arquivos, identificação interna dos arquivos no banco de dados, a quais tablespaces os arquivos estão associados e quantas páginas cada arquivo contém.

```
dmSQL> SELECT * from SYSFILE;
```

Para obter mais informações sobre as tabelas do catálogo do sistema SYSTABLESPACE e SYSFILE, consulte a Referência do Catálogo do Sistema.

Checking File and Tablespace Consistency

O DBMaker oferece seis comandos para verificar a consistência de diferentes partes de um banco de dados. Esses comandos podem ser demorados em bancos de dados grandes, pois irão adquirir bloqueios, e devem ser usados somente quando necessário. A consistência de arquivos e tablespaces pode ser verificada usando um desses comandos. O comando CHECK FILE verifica se um arquivo está corrompido ou se um tablespace contém as tabelas corretas.

CHECKING FILES

O DBMaker permite a verificação do conteúdo de cada página ou quadro em um arquivo de dados. Qualquer corrupção encontrada durante a verificação de arquivos geralmente é causada por erros de disco.

Exemplo:

Para verificar a consistência do arquivo de dados FILE1.

```
dmSQL> CHECK FILE FILE1;
```

CHECKING TABLESPACES

O DBMaker permite a verificação de arquivos e tabelas associados a um tablespace. Ao verificar arquivos e tabelas, o DBMaker utiliza os mesmos métodos dos comandos check file e check table, retornando os mesmos resultados como se esses comandos fossem executados diretamente. Exemplo: Para verificar a consistência do tablespace ts_reg.

```
dmSQL> CHECK TABLESPACE ts_reg;
```