

4. Basic Database Administration

Este capítulo descreve a administração básica de banco de dados, incluindo a criação, inicialização, conexão e encerramento de um banco de dados. Para realizar as operações descritas neste capítulo, os administradores de banco de dados podem optar por usar a ferramenta de linha de comando dmSQL e editar o arquivo dmconfig.ini, ou então utilizar a JConfiguration Tool e o utilitário JServer Manager.

As seções a seguir descrevem parâmetros de configuração e comandos essenciais para a administração básica do banco de dados. A primeira seção descreve a função e o formato do arquivo de configuração. As seções subsequentes descrevem a função de configurações específicas e como elas afetam o desempenho do banco de dados.

4.1 Configuration File - dmconfig.ini

A operação do DBMaker requer muitos parâmetros de configuração. O mecanismo DBMaker usa esses parâmetros para especificar como um banco de dados é executado. Locais de armazenamento de arquivos, alocação de memória em tempo de execução e conexões de rede são apenas algumas das características de um banco de dados definidas usando parâmetros de configuração. Esses parâmetros são armazenados como variáveis de configuração no arquivo dmconfig.ini. Uma variável de configuração é uma palavra-chave que aceita um valor (consulte Formato mais adiante nesta seção). Os usuários podem personalizar o banco de dados definindo parâmetros no arquivo dmconfig.ini ou usando a JConfiguration Tool. A interface gráfica do usuário da JConfiguration Tool simplifica o gerenciamento dos parâmetros de configuração. Mais informações sobre a JConfiguration Tool podem ser encontradas na Referência da JConfiguration Tool.

Certos parâmetros (ou seja, palavras-chave) devem ser definidos antes da criação de um banco de dados, enquanto outros precisam ser definidos apenas antes da inicialização do banco de dados. Além disso, certos parâmetros de configuração não devem ser alterados após a criação do banco de dados, caso contrário, um erro será retornado. As seções a seguir descrevem como gerenciar configurações editando diretamente as palavras-chave no arquivo de configuração dmconfig.ini. Consulte a seção Palavras-chave em dmconfig.ini para obter uma referência completa das opções do dmconfig.ini.

dmconfig.ini Location

O DBMaker procura pelo arquivo dmconfig.ini nos seguintes três locais, na ordem listada, quando executado em plataformas UNIX:

- Diretório atual
- Diretório especificado pela variável de ambiente DBMAKER
- Diretório de instalação do DBMaker: ~DBMaker/Version

Se a seção de banco de dados relevante não for encontrada no arquivo dmconfig.ini de um local, o DBMaker procura no próximo local.

No entanto, para sistemas Microsoft Windows, a regra é diferente. O DBMaker procurará pelo arquivo dmconfig.ini apenas em dois locais:

- Diretório especificado pela variável de ambiente DBMAKER
- Diretório de instalação do DBMaker. Em uma instalação típica do Windows, este é C:\DBMaker\Version

Quando o DBMaker precisa do valor de um parâmetro de configuração para um banco de dados específico, ele examina os três diretórios acima (ou o diretório de instalação em sistemas Microsoft Windows) para encontrar um dmconfig.ini que contenha uma seção com o mesmo nome da seção do banco de dados. Use qualquer editor de texto para editar este arquivo e adicionar ou modificar os valores dos parâmetros em dmconfig.ini para que o DBMaker os utilize durante a execução.

Quando uma seção correspondente não pode ser encontrada em nenhum dos arquivos dmconfig.ini, o DBMaker cria uma nova seção para o banco de dados, usando valores padrão, no primeiro arquivo dmconfig.ini encontrado ou em um novo arquivo dmconfig.ini no diretório local (ou no diretório de instalação em sistemas Microsoft Windows).

Quando um banco de dados é iniciado, o DBMaker retornará um erro a menos que a seção correspondente no dmconfig.ini seja encontrada. Embora várias seções possam ser colocadas em diferentes arquivos dmconfig.ini e diferentes arquivos dmconfig.ini possam estar em diferentes diretórios, isso não é recomendado. Um único arquivo global dmconfig.ini tornará a manutenção mais fácil.

A ferramenta JConfiguration exibe todas as seções de banco de dados listadas no arquivo dmconfig.ini. Em sistemas UNIX, a ferramenta JConfiguration exibe todas as seções de todos os arquivos dmconfig.ini mostrados nos locais listados anteriormente.

dmconfig.ini Format

O arquivo dmconfig.ini é dividido em seções. A primeira seção lista as definições dos keywords (palavras-chave) mais usados. As seções subsequentes começam com um nome de cabeçalho correspondente ao nome de um banco de dados. Os keywords sob cada seção definem a configuração desse banco de dados. Qualquer string após um ponto e vírgula é considerada um comentário.

Exemplos: Formato generalizado de um arquivo dmconfig.ini:

```
[Section_name1]
<key_word1> = <value1>
<key_word2> = <value2> <value3> ; this is a comment;
...
[Section_name2]
<key_word3> = <value4> <value5>
<key_word4> = <value6>
...
```

FILE NAME AND SIZE

Um banco de dados consiste em arquivos do sistema operacional. Esses arquivos são definidos no arquivo dmconfig.ini usando palavras-chave. O parâmetro <filename> é usado no lugar do parâmetro <value>. O parâmetro <filename> pode ser um nome de arquivo simples como firstdb.sdb, um caminho relativo como mydb/firstdb.sdb ou um caminho completo como /disk1/mydb/firstdb.sdb ("/" para UNIX e "" para Microsoft Windows).

O parâmetro <np> representa um número de páginas. O tamanho de página padrão é 8 KB, a menos que seja especificado de outra forma pela palavra-chave DB_PgSiz.

Além de usar um número de páginas, o usuário pode especificar M (megabytes) ou G (gigabytes) como unidade. Se M ou G não forem usados, a unidade é página. Quando M ou G são utilizados, o tamanho real é uma página menor do que o valor especificado. Por exemplo, se o tamanho da página for 16 KB e o tamanho do arquivo for definido como 8 MB, o tamanho será 8.176 KB em vez de 8.192 KB.

Exemplo:

Formato generalizado para indicar nomes e tamanhos de arquivos:

```
[Section_name1]

<key_word1> = <filename>

<key_word2> = <filename> <filename>

<key_word1> = <np>
```

FILE LOCATIONS

Uma base de dados pode ser acessada por usuários que executam o DBMaker de diferentes diretórios. Como consequência, o diretório atual é diferente para cada usuário. Nesse caso, todos os nomes de arquivo em dmconfig.ini devem ser caminhos completos.

Como alternativa, o parâmetro de configuração DB_DbDir pode ser utilizado. Esta palavra-chave indica o diretório inicial (ou seja, o diretório do banco de dados) de uma base de dados.

Exemplo 1: O seguinte define o nome do diretório do banco de dados como "db", em vez do padrão "DB1" indicado pelo cabeçalho da seção. Além disso, outros arquivos do banco de dados são colocados em locais alternativos e em outros discos.

```
[DB1]

DB_DbDir = /disk1/db

DB_DbFil = mydb1

DB_JnFil = /disk2/usr/DB1.JNL

Os nomes resultantes dos arquivos físicos são:

DB_DbFil -- /disk1/db/mydb1

DB_JnFil -- /disk2/usr/DB1.JNL

DB_BbFil -- /disk1/db/DB1.SBB (using default file name)
```

Exemplo 2:

Utilizando a palavra-chave DB_DbFil:

```
[DB2] DB_DbFil = mydb2

DB_JnFil = /disk2/usr/DB2.JNL
```

Os nomes resultantes dos arquivos físicos são:

DB_DbFil -- mydb2 (in current directory)

DB_JnFil -- /disk2/usr/DB2.JNL

DB_BbFil -- DB2.SBB (in current directory)

NOTA: A regra também se aplica a arquivos definidos pelo usuário.

Some Important dmconfig.ini Keywords

Esta lista apresenta algumas das palavras-chave mais importantes do arquivo dmconfig.ini. As palavras-chave essenciais para a criação e inicialização do banco de dados são abordadas em seções subsequentes deste capítulo. Uma lista completa de palavras-chave pode ser encontrada no capítulo "Palavras-chave no dmconfig.ini".

Exemplos de palavras-chave válidas que podem aparecer no dmconfig.ini:

- **DB_DbDir = <nome do arquivo>** - Especifica o diretório onde residem os arquivos do banco de dados.
- **DB_DbFil = <nome do arquivo>** - Nome do arquivo para o arquivo de banco de dados do sistema como <nome do arquivo>.
- **DB_PgSiz = <4, 8, 16, 32>** - Tamanho da página (4 KB, 8 KB, 16 KB ou 32 KB).
- **DB_JnFil = <nome do arquivo>** - Nome do arquivo para o arquivo de log do sistema como <nome do arquivo>.
- **DB_JnISz = <np>** - Tamanho do arquivo de log do sistema em <np> (número de páginas).
- **<arquivo_lógico> = <nome do arquivo> <np>** - Especifica que o arquivo definido pelo usuário com o nome <arquivo_lógico> será mapeado para <nome do arquivo> com <np> páginas. Em outras palavras, <nome do arquivo> é o nome físico do arquivo para <arquivo_lógico>.
- **DB_NBufs = <np>** - Tamanho do buffer de dados de tempo de execução em <np> (número de páginas).
- **DB_SvAdr = <endereço IP> ou <nome do host>** - Endereço IP do servidor de banco de dados ou seu nome de host. Em um sistema cliente/servidor, esta opção deve ser definida no lado do cliente.
- **DB_PtNum = <número da porta>** - Número da porta TCP/IP usada para comunicação entre o cliente do banco de dados e o servidor do banco de dados.
- **DB_MaxCo = <número>** - Número máximo de conexões que o banco de dados pode manipular.

NOTA: A comparação de padrões não diferencia maiúsculas e minúsculas, exceto para <nome_lógico>.

Default Values

Algumas opções possuem valores padrão. Portanto, se uma palavra-chave não estiver presente no dmconfig.ini, seu valor padrão será utilizado. Consulte o capítulo "Palavras-chave no dmconfig.ini" para uma descrição mais detalhada das palavras-chave e seus valores padrão.

Support Environment variables

Para bancos de dados somente leitura armazenados em CD-ROM, pode ser difícil para o usuário especificar o caminho no arquivo dmconfig.ini. Será mais fácil para o usuário se o DBMaker puder suportar as variáveis de ambiente padrão \$APP_HOME e \$APP_DRIVE.

- **\$APP_HOME:** Diretório de instalação do DBMaker. Ele sempre obtém as informações do HOME do DBMaker do registro. Por exemplo, se o DBMaker estiver instalado no diretório D:\dbmaker\5.4, o DBMaker substituirá automaticamente \$APP_HOME por "D:\dbmaker\5.4" ao ler o arquivo dmconfig.ini.
- **\$APP_DRIVE:** Esta variável retorna uma string vazia no sistema operacional Linux e retorna o diretório de instalação onde está a letra da unidade no sistema operacional Windows. Por exemplo, se o DBMaker estiver instalado no diretório D:\dbmaker\5.4, o DBMaker retornará a letra da unidade "D:" e substituirá automaticamente "D:\dbmaker\5.4" por "D:" ao ler o arquivo dmconfig.ini.

O DBMaker também suporta variáveis de ambiente do sistema, como \$TEMP = "C:\TEMP", que são definidas nas variáveis de ambiente do sistema operacional. O DBMaker substituirá automaticamente \$TEMP por "C:\TEMP" ao ler o arquivo dmconfig.ini.

Se a variável de ambiente padrão \$APP_HOME ou \$APP_DRIVE estiver definida nas variáveis de ambiente do sistema, o DBMaker não encontrará o valor definido das variáveis de ambiente do sistema ao ler o arquivo dmconfig.ini, mas dará prioridade ao valor da variável de ambiente padrão.

Exemplo:

Se o usuário tiver um CD-ROM, ele poderá colocar o software DBMaker e o banco de dados no CD-ROM com a seguinte configuração:

```
dmconfig.ini

[DBSAMPLE5]

DB_DbDir=$APP_DRIVE\database
DB_FoDir=$APP_DRIVE\database\fo
DB_TpFil=$TEMP\DBSAMPLE5.tmp
DB_SMode=6
```

Um arquivo dmconfig.ini típico:

```
[Personnel]

DB_DbFil = /disk1/bin/PERSONNEL.DB
DB_JnFil = /disk1/bin/PERSONNEL.JNL
f1.os = /disk1/bin/PERSONNEL.OS 100
f1.blob = /disk1/bin/PERSONNEL.BLOB 1000

DB_NBufs = 0 ; auto-configure number of data buffers
DB_NJnlB = 100 ; number of journal buffers
DB_MaxCo = 100 ; maximum number of connections
DB_JnlSz = 2000 ; size of journal file (pages)
DB_RTime = 0 ; restoration target time
DB_SvAdr = 192.72.116.130 ; server's IP address
DB_PtNum = 21000 ; and port number [LIBRARY]
DB_DbFil=/disk3/usr/lib/library.db
DB_JnFil=/disk3/usr/lib/library.jnl
DB_SvAdr = 192.72.116.137
DB_PtNum = 26999
DB_JnlSz = 2000
```

4.2 Creating a Database

Criar um novo banco de dados requer algum planejamento. Existem vários parâmetros de configuração que devem ser considerados antes de criar um novo banco de dados, alguns desses parâmetros não podem ser alterados após a criação do banco de

dados. Os parâmetros que devem ser definidos durante a criação do banco de dados são:

- Nome do banco de dados
- Segurança do banco de dados (se o banco de dados possui diferentes níveis de autoridade do usuário)
- Sensibilidade a maiúsculas e minúsculas (determina a diferenciação entre maiúsculas e minúsculas de determinados objetos de esquema dentro do banco de dados)
- Tamanho do quadro BLOB (a quantidade de espaço em disco alocada para cada quadro BLOB)
- Configuração de idioma (determina o conjunto de caracteres a ser usado - ASCII, Big5, etc.)
- Classificação de código de idioma (o padrão usado para classificar dados do tipo caractere)

Todos os outros parâmetros de configuração podem ser alterados após a criação do banco de dados. No entanto, é importante considerar outros parâmetros do banco de dados antes de sua criação. Esses parâmetros incluem:

- Nome, localização, tamanho e extensibilidade do tablespace
- Número de arquivos de log
- Nome, tamanho e localização do arquivo de log
- Nomes, tamanhos e localizações dos arquivos de dados e BLOB do sistema
- Nomes, tamanhos e localizações dos arquivos de dados e BLOB do usuário padrão
- Nome e localização do arquivo temporário do sistema
- Nomes, tamanhos e localizações de arquivos definidos pelo usuário
- Localizações dos arquivos de log do DBMaker
- Localização do diretório de backup
- Localização do diretório de log de replicação de tabela
- Permitir objetos de arquivo do usuário
- Habilitar o uso de dispositivos raw (apenas para plataforma UNIX)
- Habilitar banco de dados cliente/servidor
- Endereço IP e número da porta do banco de dados (para bancos de dados cliente/servidor)
- ID do usuário e senha padrão
- Alocação de memória

O DBMaker oferece um assistente fácil de usar para criar bancos de dados na ferramenta JServer Manager. Um administrador de banco de dados pode facilmente criar um banco de dados editando o arquivo dmconfig.ini e usando o dmSQL. As

subseções a seguir descrevem o processo de criação de um banco de dados. As subseções também seguem aproximadamente a sequência de etapas usadas no assistente de criação de banco de dados do JServer Manager.

Naming the Database

Antes de nomear um banco de dados, esteja ciente das seguintes regras de nomenclatura:

- Os nomes de banco de dados podem conter no máximo 128 caracteres.
- Os nomes de banco de dados podem conter qualquer caractere alfanumérico, incluindo o sublinhado.
- O caractere pode estar em qualquer posição.
- Os nomes de banco de dados não diferenciam maiúsculas e minúsculas.
- Os nomes de banco de dados devem ser exclusivos entre todos os computadores que se conectarão ao banco de dados.

Os bancos de dados podem ser nomeados no assistente de criação de banco de dados do JServer Manager ou usando o dmSQL.

Exemplo:

Para criar um banco de dados usando dmSQL:

```
dmSQL> CREATE DB <database name>;
```

```
dmSQL> TERMINATE DB;
```

```
dmSQL> QUIT;
```

Schema Object Name Case Sensitivity

É possível especificar a diferenciação entre maiúsculas e minúsculas para todos os identificadores em um banco de dados. No modo insensível a maiúsculas, todos os identificadores aparecem em letras maiúsculas quando definidos. Uma vez que um banco de dados é criado, esta configuração não pode ser alterada.

- Definir a palavra-chave como zero torna o banco de dados sensível a maiúsculas e minúsculas.
- Por padrão, a palavra-chave é definida como um, portanto, um banco de dados criado sem alterar esta configuração não diferenciará maiúsculas e minúsculas.

A seguinte variável dmconfig.ini especifica a diferenciação entre maiúsculas e minúsculas do banco de dados:

DB_IDCap = <valor> (valor padrão = 1)

Setting Storage Parameters

Um banco de dados está associado a dez tipos de arquivos do sistema operacional: dados do sistema e arquivos BLOB, dados do usuário padrão e arquivos BLOB, 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. Quando um banco de dados é criado pela primeira vez, o usuário pode atribuir nomes e locais para cada arquivo, ou o DBMaker atribuirá valores padrão a eles. É importante, antes de criar um banco de dados, ter um bom entendimento da função que esses arquivos desempenham dentro do banco de dados.

Muitos dos parâmetros discutidos nesta seção podem ser modificados na página "Armazenamento" da ferramenta JConfiguration. Para saber mais sobre como usar a ferramenta JConfiguration para alterar parâmetros do banco de dados, consulte a Referência da Ferramenta JConfiguration. Mais informações sobre o gerenciamento de arquivos estão disponíveis na seção 5.2 Tipos de Arquivo.

Ao criar um banco de dados, o DBMaker cria o arquivo de banco de dados do sistema, o arquivo de log e o arquivo BLOB do sistema de acordo com as configurações relacionadas no arquivo dmconfig.ini. Se nenhuma configuração DB_DbFil, DB_JnFil ou DB_BbFil estiver definida, os valores padrão serão usados.

Os valores padrão são:

- **DB_DbFil** -- nome_do_banco_de_dados + '.SDB'
- **DB_JnFil** -- nome_do_banco_de_dados + '.JNL'
- **DB_BbFil** -- nome_do_banco_de_dados + '.SBB'

SPECIFYING THE DATABASE DIRECTORY

O diretório do banco de dados é o local padrão onde os arquivos associados a um banco de dados são criados e armazenados. Se o arquivo for especificado com um caminho completo, o DBMaker usa esse nome para referenciá-lo. Se for usado um nome de arquivo sem um caminho completo, o DBMaker procura o diretório do banco de dados. Se não for encontrado, o DBMaker usa o nome do arquivo e assume que ele está localizado no diretório atual.

Ao criar um novo banco de dados no Windows, o DBMaker atribui um diretório de banco de dados padrão de (Diretório de Instalação do DBMaker)/bin. É necessário criar um novo diretório para armazenar os arquivos do banco de dados. Vários bancos

de dados não devem ser criados no mesmo diretório. A seguinte palavra-chave dmconfig.ini especifica o diretório do banco de dados:

DB_DbDir = <caminho> (padrão: <diretório de instalação do DBMaker>\bin<Nome do Banco de Dados>)

Exemplo:

Definindo o Diretório do Banco de Dados para /disk1/db

```
[DB1]
DB_DbDir = /disk1/db
```

CREATING THE SYSTEM TABLESPACE

Os bancos de dados DBMaker são compostos por várias divisões lógicas conhecidas como tablespaces. Com os tablespaces, o banco de dados pode ser dividido em áreas gerenciáveis. Na visão lógica, um tablespace contém uma ou mais tabelas e índices. Na visão física, um tablespace é o armazenamento físico que consiste em um ou mais arquivos. Um banco de dados recém-criado tem dois tablespaces: o tablespace do sistema e o tablespace padrão do usuário.

O tablespace do sistema consiste em um arquivo de dados do sistema e um arquivo BLOB do sistema. O tablespace do sistema registra o catálogo do sistema para todo o banco de dados. O administrador do banco de dados pode especificar o local inicial dos arquivos de dados e BLOB do sistema no tablespace do sistema.

O tablespace do sistema não pode ser removido (ou seja, excluído), embora outros tablespaces do usuário possam ser. O tamanho inicial do arquivo de dados do sistema é de 200 páginas (200 × DB_PgSiz KB). As seguintes palavras-chave do dmconfig.ini definem o tablespace do sistema:

Arquivo de dados do sistema: DB_DbFil = <filename> (padrão: <Nome do Banco de Dados>.SDB)

Arquivo BLOB do sistema: DB_BbFil = <filename> (padrão: <Nome do Banco de Dados>.SBB)

O parâmetro <filename> pode ser um nome de arquivo simples como firstdb.sdb, um caminho relativo como mydb/firstdb.sdb ou um caminho completo como /disk1/mydb/firstdb.sdb ("/" para UNIX e "" para Microsoft Windows).

Exemplo:

Inserir as seguintes linhas no arquivo dmconfig.ini resulta na armazenagem dos arquivos do tablespace do sistema no diretório /disk1/mydb/:

```
DB_DbFil = /disk1/mydb/firstdb.sdb
DB_BbFil = /disk1/mydb/firstdb.sbb
```

CREATING THE USER DEFAULT TABLESPACE

O tablespace padrão do usuário inicialmente contém um arquivo de dados e um arquivo BLOB. Os dados do usuário são armazenados nesses arquivos. O administrador do banco de dados pode especificar o tamanho inicial e o local dos arquivos de dados e BLOB no tablespace padrão do usuário. O tamanho do arquivo de dados é especificado em páginas (uma página pode ter 4 KB, 8 KB, 16 KB ou 32 KB). O tamanho do arquivo BLOB é especificado em quadros. O tamanho do quadro pode ser definido pelo usuário e será discutido em "Especificando o Tamanho do Quadro BLOB" mais adiante neste capítulo. Por padrão, o tablespace padrão do usuário é configurado para autoexpansão. Isso significa que, se o tablespace estiver cheio de dados, o DBMaker aumentará automaticamente os arquivos (e, portanto, o tablespace). No entanto, é mais flexível e eficiente criar tablespaces adicionais para armazenar tabelas do usuário.

O JDBC Tool pode ajudar a criar novos tablespaces e gerenciar os existentes. Se arquivos de dados ou BLOB forem adicionados a um tablespace sem especificar um caminho completo, o arquivo será criado no diretório do banco de dados. O tablespace padrão do usuário não pode ser removido (ou seja, excluído), embora outros tablespaces do usuário possam ser. As seguintes palavras-chave do dmconfig.ini definem os arquivos de dados e BLOB do usuário padrão:

Arquivo de dados do usuário: DB_UsrDb = <filename> (padrão: <Nome do Banco de Dados>.DB)

Arquivo BLOB do usuário: DB_UsrBb = <filename> (padrão: <Nome do Banco de Dados>.BB)

O parâmetro <filename> pode ser um nome de arquivo simples como firstdb.sdb, um caminho relativo como mydb/firstdb.sdb ou um caminho completo como /disk1/mydb/firstdb.sdb ("/" para UNIX e "" para Microsoft Windows).

CREATING JOURNAL FILES

Os arquivos de jornal fornecem um registro histórico em tempo real de todas as alterações feitas em um banco de dados e o status de cada alteração. Até oito arquivos

de jornal podem ser criados. Cada arquivo de jornal tem um tamanho fixo. Quando todos os arquivos de jornal são preenchidos por transações ativas (ou seja, transações não comprometidas e cujos blocos de jornal ocupados não podem ser liberados), a transação atual é abortada devido à falta de espaço; isso é chamado de "jornal cheio". Certifique-se de que a transação mais longa não use todos os registros de jornal em todos os arquivos de jornal. Se os arquivos de jornal forem criados sem especificar um caminho completo, eles serão criados no diretório do banco de dados. Os arquivos de jornal não podem ser modificados após o banco de dados ser iniciado. Para reduzir o número de arquivos de jornal, adicionar mais arquivos de jornal ou alterar o tamanho dos arquivos de jornal, reinicie o banco de dados em modo de novo jornal. Mais informações sobre o modo de novo jornal estão disponíveis na seção 4.3, Iniciando um Banco de Dados. Além disso, consulte a seção 5.2, Tipos de Arquivos, para mais informações sobre arquivos de jornal. As seguintes palavras-chave do dmconfig.ini definem os nomes, locais e tamanhos dos arquivos de jornal:

Nome(s) do arquivo de jornal: DB_JnFil = <filename> (padrão: "<Nome do Banco de Dados>.JNL")

Tamanho do arquivo de jornal (páginas): DB_JnISz = <size> onde size = 100 páginas até 8G

Exemplo: As seguintes linhas no arquivo dmconfig.ini instruem o DBMaker a criar dois arquivos de jornal de 500 páginas em dois discos separados no diretório /mydb:

```
DB_JnFil = /disk1/mydb/firstdb1.jnl/disk2/mydb/firstdb2.jnl
DB_JnISz = 500
```

CREATING SYSTEM TEMPORARY FILES

Os arquivos temporários do sistema são usados pelo DBMaker para armazenar informações sobre o banco de dados, como resultados de ordenação, enquanto o banco de dados está ativo. Esses arquivos são gerados conforme necessário e excluídos quando o banco de dados é encerrado. Se os arquivos temporários forem criados sem especificar um caminho completo, eles serão criados no diretório do banco de dados. Até oito arquivos temporários do sistema podem ser especificados. Cada arquivo temporário pode conter até 2 GB. Os arquivos temporários podem estar localizados em discos diferentes para melhorar o desempenho de entrada/saída em disco. Os usuários devem reservar espaço suficiente no disco para um arquivo temporário inteiro (máximo de 2 GB para um único arquivo), ou podem ocorrer erros. Os arquivos temporários do sistema podem ser especificados usando o JConfiguration Tool ou editando o dmconfig.ini antes de iniciar o banco de dados. A

seguinte palavra-chave do dmconfig.ini define os nomes e locais dos arquivos temporários do sistema:

DB_TpFil = <filename>[<filename>...] (default: <Database Name>.TMP)

SPECIFYING THE BLOB FRAME SIZE

Um bloco BLOB é a menor unidade de armazenamento usada por arquivos BLOB, que são usados para armazenar dados de objetos grandes, como LONG VARCHAR ou LONG VARBINARY. O tamanho do bloco BLOB não pode ser alterado após a criação do banco de dados. O tamanho mínimo do bloco é 8 KB e o tamanho máximo é 256 KB.

Determinar o tamanho do bloco é um equilíbrio entre utilização do disco e desempenho. Se BLOBs inteiros forem recuperados com frequência, ajustar o tamanho do bloco para conter um BLOB inteiro fornece melhor desempenho porque apenas um acesso ao disco é necessário. No entanto, pode haver grandes variações no tamanho dos dados BLOB.

Se o tamanho do bloco for grande o suficiente para conter o maior BLOB, ele poderá desperdiçar espaço em disco, pois outros blocos que contêm BLOBs menores conterão espaço em disco não utilizado. Por outro lado, se os blocos forem grandes o suficiente apenas para conter os menores BLOBs, o desempenho se degradará ao buscar BLOBs maiores armazenados em vários blocos.

A seguinte palavra-chave dmconfig.ini define o tamanho do bloco BLOB:

DB_BFrSz = <nk> O parâmetro <nk> é o tamanho do bloco em kilobytes. O tamanho do arquivo BLOB do sistema é (tamanho da página + (número de blocos - 1) × nk). Consulte o Capítulo 7, Gerenciamento de Objetos Grandes, para obter informações mais detalhadas.

Exemplo: Para definir o tamanho do bloco BLOB para 10 KB, siga estas etapas:

```
DB_BFrSz = 10
```

SETTING THE NUMBER OF PAGES TO EXTEND AN AUTOEXTEND

TABLESPACE

Quando todas as páginas no arquivo de dados ou arquivo BLOB de um tablespace com autoexpansão estiverem cheias, o DBMaker permite que o tablespace cresça estendendo automaticamente o número de páginas ou blocos no arquivo. Essa

configuração informa ao DBMaker quantas páginas ou blocos devem ser adicionados ao arquivo cheio caso ele seja preenchido.

Se o administrador do banco de dados espera que o banco de dados cresça muito rapidamente, um número maior deve ser selecionado para diminuir a frequência com que o arquivo é anexado. Esse número pode ser ajustado antes de iniciar um banco de dados usando a ferramenta JConfiguration ou editando o arquivo dmconfig.ini.

A seguinte palavra-chave dmconfig.ini especifica o número de páginas/blocos para expandir um tablespace com autoexpansão:

DB_ExtNp = <np>, onde <np> é o número de páginas a serem expandidas (padrão: 20 páginas/blocos)

ENABLING USER FILE OBJECTS

Dados do tipo "FILE" podem ser armazenados como objetos de arquivo de usuário ou objetos de arquivo do sistema. Objetos de arquivo de usuário são arquivos externos acessados por meio da máquina onde o banco de dados reside. Em outras palavras, um objeto de arquivo de usuário é apenas um link para um arquivo externo fora do banco de dados.

Habilitar objetos de arquivo de usuário permite que uma coluna do tipo "FILE" se vincule a arquivos externos, que serão acessados pelo servidor do banco de dados. Esse recurso pode ser desabilitado e habilitado conforme necessário. Objetos de arquivo de usuário inseridos podem ser acessados mesmo se a configuração estiver desativada.

Os objetos de arquivo de usuário podem ser habilitados por meio da página de armazenamento da ferramenta JConfiguration antes de iniciar o banco de dados. O valor da palavra-chave pode ser modificado antes de iniciar o banco de dados.

Definir a palavra-chave como 0 impede a inserção de objetos de arquivo.

Definir a palavra-chave como 1 permite a inserção de objetos de arquivo.

A seguinte variável dmconfig.ini ativa ou desativa a funcionalidade de objeto de arquivo:

DB_UsrFo = <valor> (padrão: 0 / desabilitado)

CREATING A DIRECTORY FOR SYSTEM FILE OBJECTS

Objetos de arquivo do sistema são criados, excluídos e gerenciados pelo DBMaker. Todos os objetos de arquivo do sistema são colocados em subdiretórios do diretório de objetos de arquivo do sistema. Alterar o diretório de objetos de arquivo do sistema não altera o local onde objetos de arquivo do sistema inseridos anteriormente residem.

O nome e o local do objeto de arquivo do sistema podem ser definidos na página de armazenamento da ferramenta JConfiguration antes de iniciar o banco de dados, ou durante a execução com as configurações do JServer Manager ou JDBC Tool Run Time. O valor da palavra-chave pode ser modificado antes de iniciar o banco de dados.

A seguinte variável dmconfig.ini define o local dos objetos de arquivo do sistema:

DB_FoDir = <caminho> (padrão: <diretório do banco de dados>\fo)

CREATING A DIRECTORY FOR USER-DEFINED FUNCTION DLL FILES

O administrador do banco de dados pode especificar o diretório onde as bibliotecas de link dinâmico (DLLs) de funções definidas pelo usuário (UDFs) serão colocadas. As UDFs são funções compiladas armazenadas em uma biblioteca de link dinâmico (.DLL para o sistema operacional Windows ou .so para sistemas operacionais UNIX) que o usuário deseja usar no DBMaker. As DLLs armazenadas no diretório de arquivos DLL de funções definidas pelo usuário ficam acessíveis ao DBMaker e podem ser usadas em instruções SQL ou aplicativos ODBC. As UDFs devem ser carregadas quando o banco de dados é iniciado. A seguinte palavra-chave especifica o local dos arquivos DLL da UDF:

DB_LbDir = <caminho> (padrão: diretório de trabalho atual)

Turning On the Log System

O DBMaker fornece um sistema de log para registrar informações úteis, incluindo conexões, usuários, tempos de execução e comandos SQL. O sistema pode ser usado para registrar informações adicionais do banco de dados e é muito útil para resolver erros gerados pelo ambiente de execução.

O formato do log será CSV (comma-separated values), permitindo que o usuário visualize os dados em um visualizador de planilhas eletrônicas após renomear o arquivo de .log para .csv. A tabela a seguir lista todas as colunas do arquivo de log e uma breve descrição do que cada coluna contém.

Nome da coluna	Descrição
----------------	-----------

LOG_TIME	Hora em que o log foi criado
BEG_TIME	Hora de início do comando.
STATE	Estado do comando. Existem quatro estados possíveis: “_” - Desconhecido ”O” - OK (concluído sem erros) ”X” - Erro ”S” - Lento (a execução demorou mais do que o normal) A determinação do estado é feita verificando primeiro o código de retorno (RETCODE) e depois o tempo de execução.
RETCODE	Código de retorno do comando: 0 indica sucesso, qualquer outro valor indica erro.
EXE_TIME	Tempo de execução do comando.
SV_FUNC	Função do servidor sendo executada no momento do registro do log.
CONNECT_ID	ID da conexão.
USERNAME	Nome de usuário.
LOGIN_TIME	Hora de login do usuário.
LOGIN_ADDR	Endereço IP de login do usuário.
STMT_ID	ID da instrução SQL.
NUM_STMT	Número de instruções SQL executadas em uma única chamada.
ERROR_ARG	Argumento do erro (caso haja erro).
OTHER_INFO	Se outras configurações de log (por exemplo, LgPln) estiverem ativadas, essas informações serão registradas no arquivo LOGNAME.TXT. O usuário pode marcar "[INFO_XXX]" no arquivo .TXT para verificar essas informações.
SQL_CMD	A instrução SQL mais recentemente executada.

O Sistema de Log pode ser ativado configurando a palavra-chave DB_LgSvr no arquivo dmconfig.ini antes de iniciar o banco de dados ou chamando a procedure armazenada SETSYSTEMOPTION() em tempo de execução.

As informações de log são divididas por nível. Quais operações são registradas e quando podem ser especificadas individualmente para cada nível. Quando o log está ativado, o DBMaker registra a operação do servidor de acordo com as opções de registro e armazena o log no diretório especificado por DB_LgDir. O DBMaker atribui o nome do log de acordo com o DBNAME e o número de índice do log. Como o log do servidor pode incluir a data atual no nome do arquivo de log, o nome do arquivo

de log será único e não será sobrescrito. O usuário pode especificar o número de dias que deseja manter os arquivos de log disponíveis. O arquivo de log expirado será removido por um serviço em segundo plano. Essa configuração é especificada pela palavra-chave `DB_LgDay` no arquivo `dmconfig.ini`. No entanto, como o número de arquivos de log pode crescer, compactá-los (em formato ZIP) seria necessário para economizar espaço de armazenamento. Essa configuração corresponde à palavra-chave `DB_LgZip` no arquivo `dmconfig.ini`. Algumas informações do sistema são registradas em `DBNAME.LOG` e as informações de log são registradas em `DBNAME_dataatual_1.LOG` quando o log é iniciado pela primeira vez. Quando o tamanho do arquivo de log atinge o padrão de 100 MB ou o tamanho especificado por `DB_LgFSz`, as informações subsequentes são armazenadas em `DBNAME_dataatual_2.LOG`, Ex: `DBNAME_20080706_2.LOG`, `DBNAME_20080708_3.LOG`, `DBNAME_dataatual_n.LOG`; onde `n` é 20 por padrão, a menos que especificado de forma diferente por `DB_LgDay` e `DB_LgFNo`.

Qualquer informação adicional do log gerada quando `DB_LgPln`, `DB_LgPar`, `DB_LgLck`, `DB_LgDay` ou `DB_LgZip` são iniciados, ou quando `DMERROR.LOG` contém informações, é registrada em `DBNAME_currentdate_n.TXT`. Com relação ao tamanho padrão do arquivo e ao recurso de log rotativo, o `DBNAME_currentdate_n.TXT` opera de maneira idêntica ao `DBNAME_currentdate_n.LOG` descrito anteriormente.

As informações adicionais são registradas como `INFO_connection_id_number` no campo `OTHER_INFO` no arquivo de log `DBNAME_currentdate_n.LOG` e também são registradas no arquivo de log `DBNAME_currentdate_n.TXT`. O número de identificação da conexão (`connection_id_number`) pode ser usado para rastrear a origem das informações de log adicionais. Observe que a nomeação de `DBNAME_currentdate_n.LOG` e `DBNAME_currentdate_n.TXT` é mantida em sincronia para que as informações sejam sempre registradas no arquivo de log respectivo com valores de `n` iguais. Isso é obtido somando-se o tamanho do arquivo de `DBNAME_currentdate_n.LOG` e `DBNAME_currentdate_n.TXT`. Quando essa soma atinge o tamanho máximo do arquivo de log, ambos os logs são considerados cheios e as informações subsequentes são imediatamente registradas em `DBNAME_currentdate_n+1.LOG` e `DBNAME_currentdate_n+1.TXT`. Informações adicionais do sistema são registradas quando `DB_LgSys` é ativado. As palavras-chave `dmconfig.ini` a seguir afetam a configuração do sistema de log:

- **DB_LgDir** = <string> (padrão: `DB_DbDir\lgdir`)
- **DB_LgSvr** = <valor> (valor padrão = 0/desabilitado)
- **DB_LgErr** = <valor> (valor padrão = 3)
- **DB_LgSTm** = <valor> (valor padrão = 5 segundos)
- **DB_LgFSz** = <valor> (valor padrão = 100 MB)
- **DB_LgFNo** = <valor> (valor padrão = 20)

- **DB_LgPln = <valor>** (valor padrão = 0/desabilitado)
- **DB_LgSys = <valor>** (valor padrão = 0)
- **DB_LgSql = <valor>** (valor padrão = 0/desabilitado)
- **DB_LgPar = <valor>** (valor padrão = 0/desabilitado)
- **DB_LgLck = <valor>** (valor padrão = 0/desabilitado)
- **DB_LgDay = <valor>** (valor padrão = 30)
- **DB_LgZip = <valor>** (valor padrão = 1/habilitado)

Exemplo:

Para registrar consultas lentas que levam mais de 10 segundos e possuem um código de erro maior que 10000, além de manter os arquivos de log por cinco dias e compactar os arquivos de log fechados, configure o dmconfig.ini da seguinte maneira antes de iniciar o banco de dados:

```
[DBNAME]
...
DB_LgSvr=3;
DB_LgErr=2;
DB_LgSTm=10;
DB_LgDay=5;
DB_LgZip=1;
```

De maneira alternativa, o mesmo resultado pode ser obtido chamando o comando SETSYSTEMOPTION:

```
dmSQL> CALL SETSYSTEMOPTION('LGSVR', '3');
dmSQL> CALL SETSYSTEMOPTION('LGERR', '2');
dmSQL> CALL SETSYSTEMOPTION('LGSTM', '10');
dmSQL> CALL SETSYSTEMOPTION('LGDAY', '5');
dmSQL> CALL SETSYSTEMOPTION('LGZIP', '1');
```

O sistema de log é usado no lado do servidor, portanto erros do cliente ou da rede não serão registrados. O desempenho do servidor é afetado quando o log está habilitado. Isso é especialmente perceptível quando o nível de registro é alto. Deve haver espaço em disco suficiente para armazenar o log do servidor, caso contrário, informações do log serão perdidas.

Raw Devices

O sistema de armazenamento físico do DBMaker é bastante flexível. Em sistemas UNIX, o DBMaker permite aos usuários criar um banco de dados usando somente arquivos UNIX, somente arquivos de dispositivo raw ou arquivos de ambos os sistemas de arquivos. No dmconfig.ini, se um nome de arquivo começar com /dev/, esse arquivo será tratado como um dispositivo raw.

Operações de entrada/saída (I/O) em dispositivos raw serão mais rápidas do que em arquivos UNIX regulares. Portanto, é recomendado aos administradores de banco de dados que usem dispositivos raw como arquivos de banco de dados. Para usar dispositivos raw como arquivos de banco de dados, o administrador do sistema deve criar esses dispositivos antes de criar qualquer banco de dados. Consulte o manual do sistema UNIX para obter o procedimento de criação de dispositivos raw.

Vários arquivos podem ser colocados em um dispositivo raw sem a necessidade de particioná-lo. No entanto, para fazer isso, é preciso considerar as seguintes restrições:

- É impossível configurar vários arquivos com autoextensão em um único dispositivo raw.
- Ao configurar vários arquivos em um dispositivo raw, o tamanho do arquivo não pode ser alterado após a configuração inicial.
- O tamanho total de todos os arquivos em um único dispositivo raw é limitado a 8 TB ou menos.
- Se um arquivo com autoextensão for colocado em um dispositivo raw, nenhum outro arquivo poderá ser colocado nesse dispositivo. Além dos arquivos que você define como autoextensão, os arquivos DB_DbFil, DB_BbFil, DB_UsrDb, DB_UsrBb e DB_TpFil são todos arquivos com autoextensão.
- Se DB_DbFil, DB_BbFil, DB_UsrDb, DB_UsrBb e DB_TpFil forem configurados para um dispositivo raw, eles só poderão ter um parâmetro: número de páginas. Você não pode definir um deslocamento para eles. Como esses arquivos são arquivos autoextensíveis, eles só podem ocupar um dispositivo raw que não pode ser compartilhado com outros arquivos e não precisam de deslocamento. Por exemplo:

```
DB_DbFil = /dev/sda 500; Creating a file with 500 pages
```

Isso também é válido, mas criará um arquivo com 30 páginas. O parâmetro 500 será ignorado.

```
DB_BbFil = /dev/sdb 30 500;
```

NOTA: O Microsoft Windows não oferece suporte a dispositivos raw.

Exemplo 1:

```
[MYDB]
```

```
f1 = /dev/sda 0 500
```

```
f2 = /dev/sda 500 200
```

```
f3 = /dev/sdb 300
```

Para criar um tablespace regular, ts_raw, contendo os arquivos de dispositivo raw mencionados acima.

```
dmSQL> CREATE TABLESPACE ts_raw DATAFILE f1, f2, f3 TYPE=DATA;
```

Então, você criará três arquivos nos dispositivos raw. Suponha que você tenha selecionado 8 K para o tamanho da página. Nesse caso, o primeiro arquivo terá o tamanho de $500 \times 8 \text{ K} = 4000 \text{ K}$, começando no endereço 0 em /dev/sda. O segundo arquivo terá o tamanho de $200 \times 8 \text{ K} = 1600 \text{ K}$, começando no endereço $500 \times 8 \text{ K} = 4000 \text{ K}$ em /dev/sda. Por último, o terceiro arquivo terá o tamanho de $300 \times 8 \text{ K} = 2400 \text{ K}$, iniciando no endereço 0 (o que parece ser um erro, pois já existe um arquivo nesse endereço. Provavelmente deveria ser o endereço subsequente ao do segundo arquivo).

Exemplo 2:

```
[MYDB2]
```

```
DB_JnlSz = 1000
```

```
DB_JnFil = dev/sda/j1.jnl 1000 /dev/sda/j2.inl 2000
```

Suponha também que você tenha selecionado 8 K para o tamanho da página. Nesse caso, você criará dois arquivos de diário normais, J1.jnl e J2.jnl, e dois arquivos de diário em dispositivos raw, sendo que um começará no endereço 8.000 K do /dev/sda e o outro no endereço 16.000 K do /dev/sda.

Enabling Client/Server Database

Qualquer banco de dados pode ser iniciado como um banco de dados de usuário único ou multiusuário. Antes de criar o banco de dados, determine qual é a função principal do banco de dados e qual modo de usuário é mais adequado.

Se o banco de dados for principalmente multiusuário, configure-o para usar um endereço IP ou nome DNS apropriado para a rede na qual o servidor DBMaker será executado. Além disso, especifique o número da porta TCP/IP que o servidor de banco

de dados usará. O banco de dados do cliente também usará essas informações para se conectar ao banco de dados.

Essa configuração pode ser alterada a qualquer momento antes de iniciar o banco de dados, mas é altamente recomendável definir esses parâmetros antes da criação do banco de dados para garantir uma operação suave. Os clientes não poderão se conectar a um banco de dados do servidor configurado incorretamente.

Se ambas as configurações estiverem desabilitadas, o banco de dados será iniciado no modo de usuário único. Esses parâmetros podem ser alterados na página de conexões da Ferramenta de Configuração J ou editando as seguintes palavras-chave dmconfig.ini:

Endereço IP/Nome do Servidor: DB_SvAdr = (padrão: nome do host local ou 127.0.0.1)

Número da Porta: DB_PtNum = (padrão: 2.300, varia de 1.024 a 65.535)

Default User and Password

O nome de usuário e a senha padrão já devem existir no banco de dados. Essas duas configurações não são verificadas ao iniciar o banco de dados, mas sim ao se conectar a ele.

Exemplo: Para especificar um nome de usuário e senha padrão para conexão com o banco de dados

```
DB_UsrId = <user name>  
DB_PasWd = <*****>
```

Changing Language Code Order

O DBMaker oferece várias classificações para ordenação de dados, definidas pela palavra-chave DB_WsorT. Por exemplo, DB_WsorT pode definir se a ordenação diferencia maiúsculas e minúsculas. O padrão é a ordenação binária.

O DBMaker suporta diferentes conjuntos de caracteres (códigos de idioma), como US-ASCII para inglês, BIG5 para chinês tradicional, GBK para chinês simplificado e JIS para japonês. A palavra-chave DB_LCode no arquivo dmconfig.ini especifica qual conjunto de caracteres o DBMaker usará. Para cada conjunto de caracteres, pode haver várias classificações.

Por exemplo, para o chinês tradicional, a ordenação pode ser de acordo com a sequência do código, número de traços ou equivalente fonético. A ordenação padrão do DBMaker é a sequência binária. Ao criar um novo banco de dados, o arquivo de definição de ordenação definido pelo usuário, especificado pela palavra-chave `DB_Order`, pode alterar o comportamento da ordenação. O parâmetro de código de idioma também pode ser definido na página "Criar Banco de Dados" da Ferramenta de Configuração J.

SETTING THE SORTING ORDER

A ordem de classificação é um conjunto de regras que especifica como o DBMaker apresenta dados em resposta a consultas ao banco de dados e instruções do DBMaker envolvendo cláusulas `GROUP BY`, `ORDER BY` e `DISTINCT`. A ordem de classificação também determina como determinadas consultas são resolvidas, como aquelas que envolvem cláusulas `WHERE` e `DISTINCT`.

Você pode especificar a palavra-chave `DB_WsorT` do `dmconfig.ini` para definir a ordem de classificação de palavras. O DBMaker considera valores de caracteres que diferem apenas em maiúsculas e minúsculas como iguais quando uma ordem de classificação sem distinção entre maiúsculas e minúsculas é especificada (por exemplo, 'João' = 'joão'). Muitas vezes, é necessário obter resultados de consulta com distinção entre maiúsculas e minúsculas ao usar uma ordem de classificação sem distinção entre maiúsculas e minúsculas.

A próxima seção especifica a sensibilidade a maiúsculas e minúsculas da ordem de classificação de palavras:

`DB_WsorT` = <valor> (padrão: 0 / ordem de classificação binária)

1. **ordem de classificação sem distinção de maiúsculas e minúsculas** (This literally translates to "sort order without distinction of uppercase and lowercase")
2. **ordem de classificação com distinção de maiúsculas e minúsculas** (This literally translates to "sort order with distinction of uppercase and lowercase")

O seguinte exemplo demonstra como definir o idioma local e o arquivo de ordem de classificação antes da criação de um novo banco de dados.

Exemplo: Para definir o tipo de idioma para o chinês tradicional, use BIG5.

```
[MY_DB] ...  
DB_LCode =1 ; BIG5 for traditional Chinese
```

```
DB_Order = big5_stroke.ord ; order definition file
```

...

A palavra-chave DB_Order indica o arquivo de definição de ordem definido pelo usuário, chamado big5_stroke.ord. Este arquivo deve ser colocado no subdiretório shared/codeorder do diretório de instalação do DBMaker. O arquivo de definição de ordem é um arquivo de texto puro que afeta o resultado da classificação no DBMaker. A palavra-chave é usada quando o banco de dados é criado, sendo então registrada no banco de dados e não mais utilizada. Sem essa palavra-chave, durante a criação do banco de dados, a sequência de classificação seria binária. Uma vez especificado um arquivo de definição, ele deve existir sempre, caso contrário o banco de dados falhará ao iniciar.

USER-DEFINED ORDER DEFINITION FILE

Definição de ordem é um arquivo de texto puro criado pelo usuário. Ele define a sequência de caracteres válidos. Um exemplo do esquema de nomes é codename_ordertype.ord, onde codename é o código do idioma e ordertype é o tipo de ordenação, por exemplo, big5_stroke.ord. Exemplo:

Um arquivo de definição de ordem:

```
Comment: Write information here.

[BEGIN] // begin to arrange the character sequence
c // ASCII 0x63
0x62 // Character 'b'
a // ASCII 0x61

[SINGLE] // Single-Byte Character Default Order
[DOUBLE] // Double-Byte Character Default Order
0xA440 // one of Chinese characters
0xA441 // one of Chinese characters
0xA442 // one of Chinese characters
```

*Todas as linhas antes da palavra-chave [BEGIN] são consideradas comentários. Todas as palavras após // ou / também são consideradas comentários.** Após o [BEGIN], cada linha representa um caractere e deve ocupar a primeira posição da linha, seguida por pelo menos um espaço ou uma nova linha. No exemplo acima, o caractere c é menor que b e b é menor que a. Se o editor de texto não puder ser usado para editar alguns

caracteres, represente-os com hexadecimal. Por exemplo, o caractere a pode ser escrito como "a" ou seu valor de código 0x61. Isso também é muito útil para caracteres invisíveis.

O criador da ordem de classificação pode estar interessado apenas em alguns caracteres e deixar que outros sejam classificados por padrão, ou seja, de forma binária. As palavras-chave [SINGLE] e [DOUBLE] podem ser usadas para representar o conjunto de caracteres de um único byte e o conjunto de caracteres de dois bytes, ambos não especificados no arquivo de definição. Se não houver um [SINGLE], os caracteres ausentes de um byte aparecerão antes de todos os caracteres no arquivo de definição. Se o [DOUBLE] estiver ausente, os caracteres ausentes de dois bytes aparecerão após os caracteres no arquivo de definição.

O DBMaker ignora erros encontrados no arquivo de definição. Por exemplo, se o [BEGIN] estiver ausente, o DBMaker usará a ordem de classificação padrão para todos os caracteres. Se o mesmo caractere aparecer duas ou mais vezes, apenas o primeiro será processado; os caracteres subsequentes serão ignorados. Após criar um banco de dados, os administradores devem verificar cuidadosamente o comportamento da ordem de classificação para garantir que está correto.

Em ambientes de banco de dados distribuídos, todos os bancos de dados devem usar o mesmo arquivo de definição de ordem de classificação. Ao copiar ou mover um banco de dados para outra máquina, não se esqueça de copiar quaisquer arquivos de definição de ordem de classificação existentes.

A Área de Comunicações e Controle de Dados (DCCA)

A Área de Comunicações e Controle de Dados (DCCA) é um bloco de memória no qual quase todas as informações e dados são colocados. Para bancos de dados multiusuário, a DCCA é alocada da memória compartilhada e usada para comunicação entre processos. Quando um banco de dados é iniciado, ele aloca uma DCCA para armazenar todas as informações sobre esse banco de dados. A DCCA pode ser dividida em três partes: buffers de página, buffers de log e a área de controle do sistema.

Existem várias palavras-chave no dmconfig.ini relacionadas ao uso da DCCA:

- **DB_NBufs = <np>** - Número de buffers de página que o DBMaker utilizará. O valor padrão é 0 (configuração automática).
- **DB_NJnlB = <np>** - Número de buffers de log que o DBMaker utilizará. O valor padrão é 64.
- **DB_ScaSz = <np>** - Número de páginas na área de controle do sistema. O valor padrão é 200.

- **DB_MaxCo = <número>** - Número máximo de transações concorrentes que o banco de dados pode manipular. DB_MaxCo também é usado para formatar o arquivo de log quando o banco de dados é criado ou iniciado com um novo log.

O tamanho da DCCA pode ser estimado somando-se o tamanho dos buffers de página, dos buffers de log e da área de controle do sistema. Quando o tamanho especificado da DCCA não for grande o suficiente, o DBMaker alocará automaticamente o espaço mínimo necessário para armazenar as informações necessárias na DCCA, em vez do tamanho padrão usado no cálculo acima.

Em um ambiente multiusuário no UNIX, o tamanho da DCCA não pode exceder o tamanho permitido de memória compartilhada do sistema, pois nesse caso a DCCA é alocada da memória compartilhada. Os usuários podem consultar seus manuais do UNIX para obter instruções sobre como aumentar o tamanho da memória compartilhada, o que geralmente requer a recompilação do kernel. O DBMaker funcionará de forma mais suave com mais buffers e uma área de controle do sistema maior.

A relação entre a DCCA, buffers de página, buffers de log e a área de controle do sistema é explicada com mais detalhes no Capítulo 18, Ajuste de Performance.

Os parâmetros da DCCA também podem ser configurados na página Cache e Controle da Ferramenta JConfiguration. Para obter detalhes, consulte a Referência da Ferramenta JConfiguration.

4.3 Starting a Database

O objetivo de iniciar um banco de dados é alocar os recursos necessários do sistema operacional, inicializá-los e aguardar a conexão dos usuários. Antes de iniciar um banco de dados, é preciso considerar as configurações de alguns parâmetros:

- **Modo de inicialização do banco de dados**
- **Habilitar banco de dados cliente/servidor**
- **Endereço IP e número da porta do banco de dados (para bancos de dados cliente/servidor)**
- **ID de usuário e senha padrão**
- **Alocação de memória**
- **Método de relatório de erros**

Um banco de dados pode ser iniciado usando o dmSQL ou o JServer Manager. Para obter mais informações sobre como iniciar um banco de dados usando o dmSQL,

consulte as seções seguintes. Para saber como usar o JServer Manager para iniciar um banco de dados, consulte o Guia do Usuário do JServer Manager.

Single-User

Um banco de dados single-user (usuário único) precisa ser iniciado toda vez que um usuário quiser se conectar e finalizado após o término do uso. Exemplo: Iniciando um Banco de Dados Single-User com dmSQL:

```
dmSQL> START DB <database name> <user name> <password>;  
  
.  
  
< do DML here >  
  
.  
  
dmSQL> TERMINATE DB;
```

NOTA: *Apenas usuários com privilégio de DBA podem iniciar um banco de dados. Para informações sobre privilégios de banco de dados, consulte o Capítulo 8, Gerenciamento de Segurança. Se um banco de dados for iniciado no modo de usuário único, apenas um usuário pode acessar o banco de dados por vez.*

Client/Server

O Administrador do Banco de Dados (DBA) deve iniciar o banco de dados cliente/servidor na máquina servidora para que todos os clientes em máquinas remotas (ou na mesma máquina) possam se conectar ao banco de dados do servidor através da rede. Duas variáveis de configuração devem ser definidas no servidor antes que o banco de dados seja iniciado.

Iniciar um banco de dados cliente/servidor é um pouco mais complexo do que iniciar um banco de dados single-user. Primeiramente, precisamos conhecer o endereço IP da máquina servidora. O único identificador para distinguir cada máquina em uma rede é o endereço IP. A palavra-chave DB_SvAdr no dmconfig.ini especifica o endereço IP do servidor.

O segundo item é o número da porta. O programa do servidor se vinculará a um determinado número de porta, especificado por DB_PtNum no dmconfig.ini, para aguardar conexões. Todos os programas clientes devem se conectar a esse número de porta para se comunicar com o servidor do banco de dados.

Exemplo1:

Para especificar o endereço IP do Servidor e as portas do Servidor e Cliente

```
DB_SvAdr = <server IP address> (on client side)
DB_PtNum = <port number> (on both server and client sides)
```

Exemplo 2:

Para iniciar um banco de dados cliente/servidor na máquina servidora usando o DmServer, siga estas etapas:

```
UNIX> dmserver <database name>
```

Exemplo 3:

Digite o nome de usuário e senha. O DmServer iniciará o banco de dados e aguardará a conexão dos clientes.

```
UNIX> dmserver [-f] [-t port_number] [-u username [-p password]]
database_name
```

Descrição dos parâmetros do DmServer no UNIX:

O DmServer oferece algumas opções para iniciar o banco de dados cliente/servidor no modo UNIX. Veja a seguir a descrição de cada parâmetro:

- **f** - Executar o programa do servidor em modo foreground. Normalmente, o DmServer é executado em background.
- **t** - Usar a porta especificada por este parâmetro ao invés da porta definida no dmconfig.ini.
- **u <nome_de_usuario>** - Nome de usuário para login.
- **p <senha>** - Senha para o nome de usuário informado.

Caso o nome de usuário e senha não sejam especificados na linha de comando, o DmServer buscará os valores nas variáveis DB_UsrId e DB_PasWd do arquivo dmconfig.ini. Se não forem encontrados, o DmServer solicitará que o usuário digite o nome de usuário e senha.

Start Mode

O modo de inicialização de um banco de dados é especificado pela palavra-chave DB_SMode no arquivo dmconfig.ini. A palavra-chave DB_SMode pode ter seis valores, correspondendo a seis modos de inicialização diferentes.

- **1 - Inicialização Normal (Normal start):** Inicia o sistema normalmente. Se o banco de dados sofreu uma falha na sessão anterior, o DBMaker realizará automaticamente a recuperação de falhas para trazer o banco de dados a um estado consistente e estável.
- **2 - Novo Jornal (New Journal):** O modo Novo Jornal deve ser selecionado se novos nomes e/ou locais para o arquivo de log (journal) foram definidos no arquivo dmconfig.ini. Novos nomes e locais para o arquivo de log também podem ser especificados na página Armazenamento da ferramenta JConfiguration. Todos os registros antigos serão sobrescritos se os nomes anteriores dos arquivos de log forem mantidos. Esta opção deve ser selecionada caso o usuário queira alterar o tamanho do arquivo de log, adicionar mais arquivos de log ou alterar o nome do arquivo de log. Recomendamos realizar um backup antes de selecionar esta opção.
- **3 - Restaurar Banco de Dados de Backup (Restore Backup Database):** Utiliza os arquivos de backup do banco de dados (incluindo o arquivo de log) para iniciar o banco de dados. O DBMaker usará os arquivos de backup incremental para reverter as operações até o ponto especificado por DB_RTime. Se nenhum valor for especificado ou a data especificada for posterior à hora do último backup incremental, DB_RTime retornará ao seu valor padrão. Consulte o Capítulo 15, Recuperação, Backup e Restauração do Banco de Dados para obter informações mais detalhadas sobre rollover (reversão).
- **4 - Fonte para Banco de Dados de Destino (Source for Target Database):** Utilizado para replicação de banco de dados. Iniciar um sistema neste modo o torna um banco de dados primário (fonte). Para obter mais informações sobre replicação de banco de dados, consulte o Capítulo 17, Replicação de Dados.
- **5 - Alvo da Replicação de Banco de Dados (Target of Database Replication):** Utilizado para replicação de banco de dados. Iniciar um sistema neste modo o torna um banco de dados secundário (slave). Para obter mais informações sobre replicação de banco de dados, consulte o Capítulo 17, Replicação de Dados.
- **6 - Somente Leitura (Database is Read Only):** Inicia o sistema normalmente, mas o banco de dados fica somente leitura ou concede apenas privilégio de leitura a todos os usuários. Iniciar um banco de dados primário no modo somente leitura impede que os usuários o modifiquem.

O modo de inicialização também pode ser especificado na página Iniciar Banco de Dados da ferramenta JConfiguration ou na janela Configurações Avançadas de Iniciar Banco de Dados do JServer Manager.

Forced Startup

Ao tentar iniciar um banco de dados corrompido, é possível que uma mensagem de erro seja sempre retornada. A única solução é usar a "Inicialização Forçada" fornecida pelo DBMaker. Defina a variável de configuração DB_ForcS como um (1) e o DBMaker forçará a inicialização do banco de dados. Consulte o Capítulo 15, Recuperação, Backup e Restauração do Banco de Dados para obter informações mais detalhadas.

Email Error Report System

Normalmente, todas as mensagens de erro são armazenadas no arquivo dmerror.log. A menos que o administrador do banco de dados verifique o arquivo dmerror.log constantemente, alguns erros do banco de dados podem passar despercebidos. O DBMaker fornece um sistema de relatório de erros por e-mail para garantir que os administradores de banco de dados estejam cientes dos erros no sistema.

O sistema de relatório de erros pode ser ativado definindo duas palavras-chave no arquivo de configuração, usando a ferramenta JConfiguration ou durante a inicialização do banco de dados com o JServer Manager. As palavras-chave que governam o comportamento do sistema de relatório por e-mail são DB_ERMRv e DB_ERMSv. Use DB_ERMRv para especificar os endereços de e-mail para receber relatórios de erro e DB_ERMSv para definir o endereço de um servidor SMTP para roteamento de e-mail. Para obter mais informações sobre como configurar o sistema de relatório de erros por e-mail com a ferramenta JConfiguration ou o JServer Manager, consulte a Referência da Ferramenta JConfiguration e o Guia do Usuário do JServer Manager, respectivamente.

4.4 Connecting to a Database

Esta seção descreve como se conectar a um banco de dados cliente/servidor em execução. Um usuário precisa primeiro se conectar a um banco de dados antes de realizar operações de manipulação de dados (DML). Após a desconexão, um banco de dados cliente/servidor permanece ativo. Os usuários podem continuar a se conectar até que o banco de dados seja desligado.

Existem alguns parâmetros para conexões cliente/servidor, incluindo número da porta, endereço do servidor, intervalo de tempo limite de conexão e intervalo de tempo limite de bloqueio. Os parâmetros de conexão são definidos alterando os valores das palavras-chave no arquivo dmconfig.ini ou usando a ferramenta JConfiguration.

Um banco de dados single-user (usuário único) só permite a conexão de um único usuário por vez. Sempre que um usuário for utilizar o banco de dados, ele precisará iniciá-lo, não sendo necessário fazer uma conexão separada. Consulte a seção Iniciando um Banco de Dados para obter mais informações.

Client/Server Database

As palavras-chave DB_SvAdr e DB_PtNum devem ser configuradas no arquivo dmconfig.ini. Se as palavras-chave DB_UsrId e DB_PasWd estiverem definidas no dmconfig.ini, as opções e no comando CONNECT podem ser ignoradas. Exemplo: Para conectar e desconectar de um banco de dados cliente/servidor com dmsqlc:

```
dmSQL> CONNECT TO <database name> <username> <password>;
.
< do DML here >
.
dmSQL> DISCONNECT;
dmSQL> QUIT;
```

Connection Time-Out

Em um modelo de banco de dados cliente/servidor, às vezes um cliente não consegue se conectar ao servidor porque a máquina do servidor está desligada ou o endereço IP da máquina do servidor está incorreto. Nestes casos, os usuários devem aguardar o estabelecimento da conexão. Para definir o valor do tempo limite de conexão, os usuários podem definir o parâmetro DB_CTimeOut (em segundos). O valor padrão para esta palavra-chave é cinco segundos.

Lock Time-Out

Bloqueios são necessários para controle de simultaneidade entre múltiplas transações nos mesmos objetos do banco de dados. Para obter mais informações sobre transações e controle de simultaneidade, consulte o Capítulo 9, Controle de Simultaneidade. Ao se conectar a um banco de dados, uma palavra-chave de tempo limite de bloqueio, DB_LTimO, deve ser definida no arquivo dmconfig.ini para indicar

por quanto tempo (em segundos) um usuário aguardará por um bloqueio que não pode ser adquirido.

Por exemplo, se DB_LTimO = 10, o DBMaker retornará um erro de "tempo limite de bloqueio" se o usuário esperar por um bloqueio por mais de 10 segundos. DB_LTimO pode ser definido como zero, indicando que o usuário não quer esperar nada. Definir DB_LTimO como -1 desativa esse recurso. Nesse caso, o usuário aguardará pelo bloqueio até que ele seja liberado. Cada usuário pode ter um valor DB_LTimO.

Compressing Data

Acessar o conteúdo do banco de dados (ou seja, dados) é a principal causa do tráfego de rede. Compactar os dados antes da entrega na rede reduz a quantidade realmente transmitida, resultando em um aumento de desempenho.

Defina a palavra-chave DB_NetZc antes de conectar a um banco de dados para habilitar a função de compactação de rede. Quando ativa, esta função compacta os dados transmitidos do servidor e descompacta os dados quando eles são recebidos pelo cliente.

Exemplo: Defina o dmconfig.ini antes de conectar a um banco de dados para ativar a compactação de rede:

```
[DBNAME]
DB_NetZc = 1;
```

4.5 Shutting Down a Database

Um banco de dados deve ser encerrado após a finalização de todas as operações. O DBMaker irá liberar todos os recursos, como o DCCA, para o sistema operacional. Se ainda houver transações ativas no mecanismo de banco de dados, o DBMaker as abortará.

No entanto, se ainda houver conexões ativas com o mecanismo de banco de dados, o DBMaker encerrará o banco de dados sem encerrar os processos dessas conexões. Nesse caso, o administrador do banco de dados deve encerrar os processos manualmente; caso contrário, a mensagem de erro "Não é possível bloquear o rollback da transação do arquivo" ocorrerá ao iniciar o banco de dados na próxima vez.

Portanto, os administradores de banco de dados (usuários DBA) devem garantir que todos os usuários estejam desconectados antes de encerrar o banco de dados. Para encerrar um banco de dados, um DBA precisa se conectar primeiro e emitir o

comando apropriado. Apenas um DBA tem privilégio para encerrar um banco de dados.

Exemplo:

Para finalizar bancos de dados single-user ou cliente/servidor, utilize o dmSQL:

```
dmSQL> CONNECT TO <database name> <DBA username> <password>;  
dmSQL> TERMINATE DB;  
dmSQL> QUIT;
```