

# Capítulo 1

## Server Manager

Interbase Server Manager é um aplicativo para administrar os Bancos de Dados locais ou remotos. Este software é executado em uma estação cliente, mas pode administrar bancos de dados no cliente ou em qualquer servidor na rede.

Utilizado para administrar segurança de banco de dados, autorizar usuários novos, senhas de usuário, remover autorizações de usuário, habilitar e desabilitar o banco de dados, executar manutenção de banco de dados, manter a integridade de um banco de dados, consertar um banco de dados corrompido, recuperar transações, fechar e reiniciar um banco de dados.



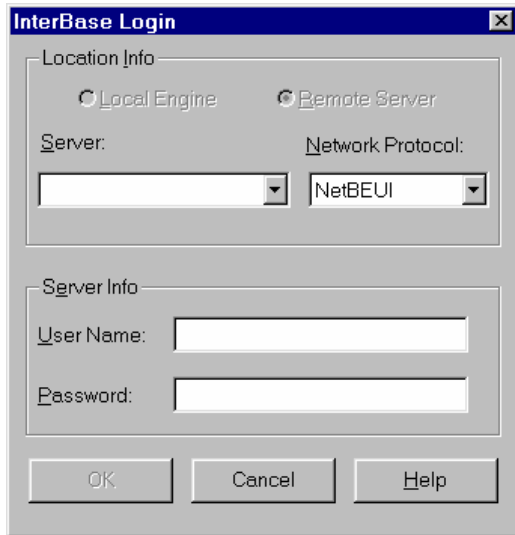
Ao longo das próximas páginas, teremos uma definição de como operar e o que significa cada opção deste gerenciador, objetivando dar-lhe uma visão ampla deste recursos e a sua aplicabilidade.

## File | Server Login

Antes de executar alguma tarefa de banco de dados local ou remoto, você deve se registrar em um servidor .

Se já tiver conectado a mais de um banco de dados, você pode trocar a qualquer momento entre eles. Para tanto, escolha um nome de banco de dados diferente na árvore de servidor ou use o combo.

Para problemas de login, veja **Troubleshooting**.



### Caixa de dialogo Login

Para conectar a um servidor local ou remoto, você tem que usar a caixa de dialogo Login.

1. Escolha o tipo de servidor - local ou remoto - clicando em um dos botões de rádio.
2. Escolhendo um servidor remoto, você tem que entrar com o nome do servidor remoto (a máquina que contem o servidor ) no campo Servidor. Desejando conectar a um servidor distante que tenha sido utilizado anteriormente, simplesmente escolha na lista dropdown. Você também tem que escolher um protocolo (método de conexão) para o servidor selecionado na lista de Protocolos de rede. Se seu servidor é local, o Servidor e Protocolo de rede não são requeridos.
3. Adicione o Nome de Usuário e a Senha, clicando em OK para encerrar o diálogo e conectar ao servidor selecionado.

Após os procedimentos anteriores, um artigo novo aparece na árvore do servidor (painel da esquerda), e o servidor exibe a versão do na área de informação sumária (painel certo).

**Nota:** Inicialmente, o único usuário autorizado de um servidor é SYSDBA, com "masterkey" sendo a senha. Todos os outros usuários devem ser adicionados pelo SYSDBA em **Tasks | User Security**

### Communications troubleshooting

## Troubleshooting Conexões de Servidor Locais

Se você encontra problemas de conexão para um Servidor Local, primeiro consulte o arquivo READIB.TXT para ver se há qualquer restrição geral que afete sua configuração.

Certifique-se que a aplicação está achando as DLLs do . Use Arquivo | Procurar... no Windows para localizar GDS.DLL e assegurar-se que elas sejam da versão do seu Servidor .

Aplicações clientes procurarão diretórios para DLLs na seguinte ordem:

1. O diretório atual.
2. Diretório WINDOWS.
3. Diretório SYSTEM do WINDOWS.
4. O diretório que contém o executável para a tarefa que requer a DLL.

## Troubleshooting Conexões de Servidor Remoto

Certifique-se que a aplicação cliente está usando as DLLs corretas. Certifique-se de que nenhuma DLL permaneça carregada em memória. E também, se a aplicação está encontrando as DLLs corretas. Use Arquivo | Procura... no Windows Gerenciador de Arquivo para localizar REMOTE.DLL e GDS.DLL, a fim de assegurar-se que elas são da versão de seu Servidor .

Aplicações clientes procurarão diretórios para DLLs na seguinte ordem:

1. O diretório atual.
2. Diretório WINDOWS
3. Diretório SYSTEM do WINDOWS.
4. O diretório que contém o executável para a tarefa que requer a DLL.

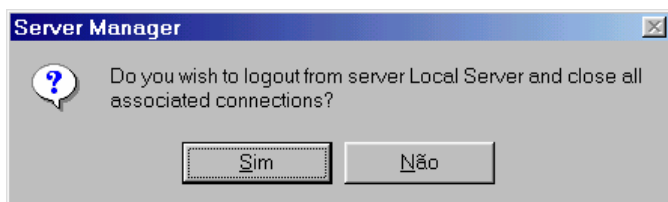
**Nota** O inclui uma ferramenta diagnóstico para ajudar a encontrar a fonte de qualquer problema, estabelecendo conexões de cliente-servidor. Esta ferramenta chama-se WSDiag.

## COMDIAG

Alternativamente, você pode clicar duas vezes em COMDIAG.EXE em .../INTERBAS\BIN. Para mais informação sobre ComDiag e mais detalhes sobre troubleshooting para cada protocolo, veja a ajuda de online de ComDiag:

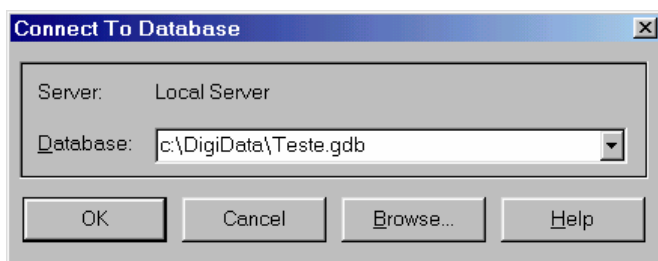
## File | Server Logout

Utilize esta opção para desligar/desativar o Servidor.  
A caixa de dialogo exibida abaixo, é apresentada, permitindo a confirmação ou não da operação.



## File | Database Connect

Exibe a caixa de dialogo abaixo, permitindo conectar um banco de dados baseado na escolha feita na opção Database.



## File | Database Disconnect

Desconecta o banco de dados.

## File | Exit

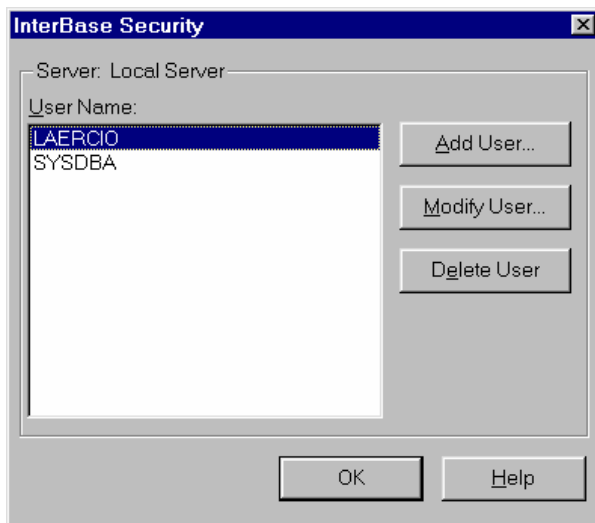
Encerra o gerenciador desconectando automaticamente qualquer banco de dados para o qual esteja conectado.

## Tasks | User Security

Escolha Tasks | User Security para abrir a caixa diálogo de segurança do Interbase e executar as seguintes operações:

- 1 Adicionar um usuário.
- 2 Modificar a senha de usuário e informação opcional. Você não pode modificar o nome de um usuário, mas se necessitar fazê-lo você tem que excluir o nome do usuário e então readicioná-lo novamente selecionando o usuário cuja configuração você quer modificar e então escolher a opção Modifique o Usuário....
- 3 Para excluir um usuário, selecione o nome do usuário que você quer remover e escolher a opção Apague.... Uma caixa de diálogo lhe pedirá que confirme a ação.

**Nota:** Só o usuário SYSDBA é autorizado para as tarefas citadas.



### Caixa de dialogo Security

Segurança do banco de dados é controlada pelo Security que contém informações sobre todos os usuários autorizados para o servidor atual. Esta caixa de diálogo permite ver estes usuários e operar as tarefas citadas no tópico anterior.

Quando um usuário tenta conectar a um banco de dados no servidor, são verificados o nome e a senha no banco de dados de segurança. Se um registro idêntico é achado, a conexão tem sucesso.

### O usuário de SYSDBA

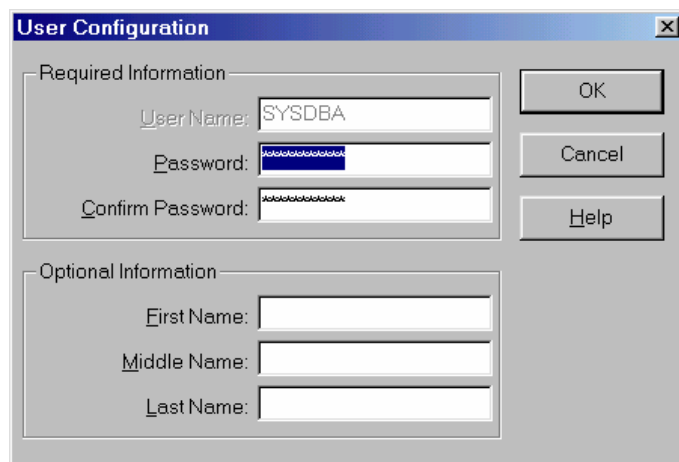
Todo servidor tem um usuário de **SYSDBA**, sendo a senha **masterkey**.

Inicialmente, SYSDBA é o único usuário autorizado em um servidor; o SYSDBA tem que autorizar todos os outros usuários no servidor. SYSDBA é um usuário especial que se sobrepõe à segurança do Servidor, e executa tarefas especiais como paralisação do banco de dados.

Só o usuário de SYSDBA pode atualizar o banco de dados de segurança para adicionar usuários novos, apagar os usuários, ou modificar configurações de usuário.

O SYSDBA pode autorizar um novo usuário nomeando um usuário e uma senha no banco de dados de segurança.

**Importante:** É recomendado mudar a senha o mais rápido possível para SYSDBA. Caso contrário, a conta SYSDBA não será segura já que é de domínio público.



The image shows a 'User Configuration' dialog box with a blue title bar and a close button. It is divided into two sections: 'Required Information' and 'Optional Information'. The 'Required Information' section contains three text input fields: 'User Name' (containing 'SYSDBA'), 'Password' (masked with asterisks), and 'Confirm Password' (masked with asterisks). To the right of these fields are three buttons: 'OK', 'Cancel', and 'Help'. The 'Optional Information' section contains three text input fields: 'First Name', 'Middle Name', and 'Last Name'.

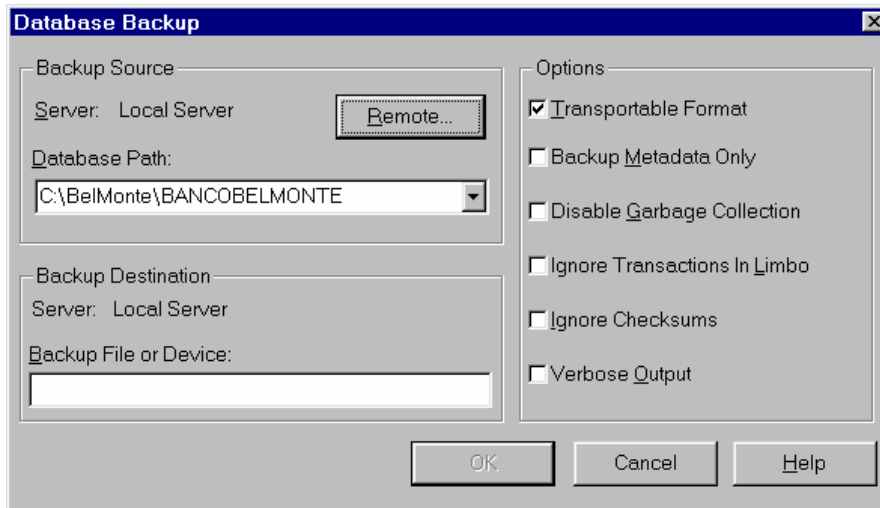
### Caixa de Dialogo User Configuration

Esta caixa de diálogo permite ao SYSDBA adicionar um usuário novo para o Interbase, modificar as senhas de usuários existentes, e informação opcional.

**Nota:** Nomes de usuário podem ter até 31 caracteres, não sendo case-sensitive. Senhas são case-sensitive. Só os primeiros oito caracteres da senha são considerados.

## Tasks | Backup

Auxilia o processo de cópia de um banco de dados para um arquivo em um disco rígido ou outra mídia de armazenamento. Para proteger um banco de dados, caso haja falha em disco ou outra perda de dados potencial, deve-se copiar o banco de dados regularmente. Para segurança adicional, é recomendado armazenar a cópia em uma localização física diferente do servidor de banco de dados de origem.



### Caixa de diálogo Database Backup

Esta caixa de diálogo consiste em três áreas: Fonte (Backup Source), Destino (Backup Destination), e Opções (Options). O nome de servidor atual é exibido na área superior da caixa de diálogo.

#### Backup Source

Digite o nome e o caminho do diretório do banco de dados que você quer copiar. Sendo um banco de dados servidor fora da máquina local, clique no botão Remote.

#### Backup Destination

Identifique nome e endereço para o arquivo backup.

**Nota:** Arquivos de banco de dados e arquivos backup podem ter qualquer nome com qualquer extensão legal em seu sistema operacional. As extensões **.gdb** e **.gbk** são convenções do Interbase.

#### Backup options

##### Transportable Format

Ao copiar um banco de dados para outra máquina que não esteja na rede, inspecione a opção de Formato Transportável. Esta opção escreve dados em um formato genérico e o permite restabelecer a cópia em qualquer máquina que contenha o Interbase.

##### Back Up Metadata Only

Copia um banco de dados vazio, ou seja, só seu metadata. Você também pode extrair o metadata de um banco de dados utilizando o Windows, o que vai produzir um arquivo de script (arquivo texto).

## Disable Garbage Collection

A cópia é procedida sem que os dados excluídos e ainda presentes fisicamente, sejam também copiados para o destino.

## Ignore Transactions in Limbo

Esta opção gera o backup desconsiderando as transações ativas e não encerradas. Antes de copiar um banco de dados que contém transações pendentes, execute o recuperador de transação.

## Ignore Checksums

Um checksum é uma análise “página-por-página” de dados para verificar sua integridade. Erros de Checksum indicam corrupção de dados, e o Interbase lhe impede de copiar um banco de dados se são descobertos checksums ruins. Examine os dados restabelecendo o banco de dados para posteriormente copiá-lo.

## Verbose Output

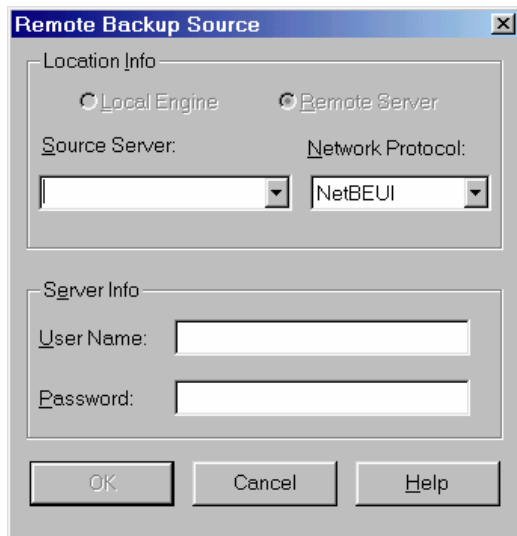
Habilita uma tela para exibição de texto standard para as mensagens de erro. A janela de exibição de texto standard permite procurar texto específico, copiá-lo para um arquivo, e imprimir o texto.

## Backing up a database from a remote server

Permite copiar um banco de dados para um servidor diferente do servidor atual.

- Coloque o banco de dados na máquina onde é freqüentemente usado.
- Feche o servidor original para manutenção enquanto o banco de dados estiver em funcionamento.

## Caixa de dialogo Remote Backup Source



Entre com nome do servidor remoto. Selecione o protocolo de rede desejado. Entre com seu nome de usuário e a senha no servidor remoto, e clique em OK.



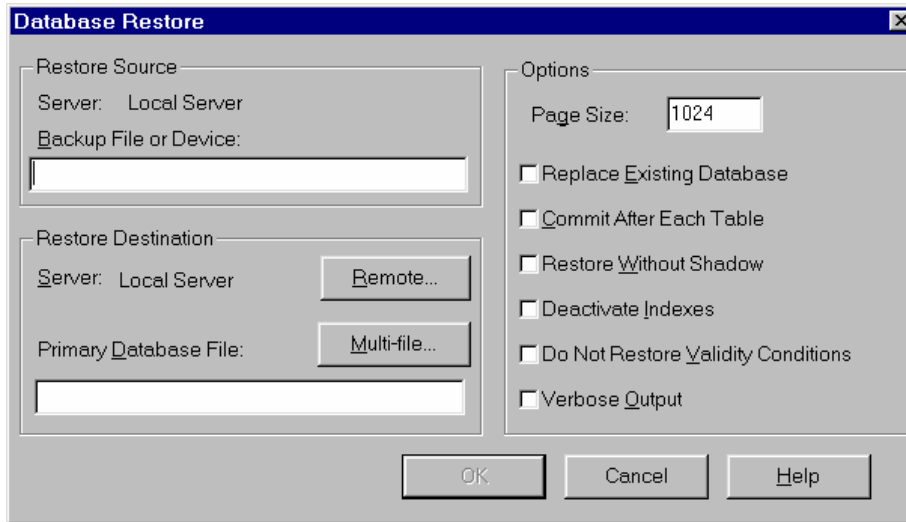
**Nota:** O protocolo de rede para um cópia remota deve ser utilizável pela fonte e servidores de objetivo. Só use NetBEUI se a fonte e servidores de objetivo seja o Windows NT. Use TCP/IP para copiar entre servidores heterogêneos quando ambos tenham software TCP/IP

### Backup options

Idem neste capítulo.

### Tasks | Restore

Opção para restabelecer/restaurar uma cópia de banco de dados.



### Caixa de dialogo Database Restore

Permite restabelecer um banco de dados de um arquivo previamente criado. Esta caixa de diálogo consiste em três áreas: Restore Source, Restore Destination, e Restore Options. O nome de servidor atual é exibido na esquerda superior da área Restore Source. O arquivo posterior do qual o banco de dados está sendo restabelecido é chamado FONTE. O banco de dados que é restabelecido é chamado de DESTINO ou OBJETIVO.

Para Restabelecer um banco de dados:

- Digite o nome do arquivo de fonte ou dispositivo no servidor atual (inclusive caminho do diretório) em Restore Source.
- Quando Restabelecer um banco de dados para múltiplos arquivos clique no botão Multi-file....
- Restabeleça um banco de dados para um servidor remoto (um servidor diferente do servidor atual), clique no botão Remote....
- Entre com tamanho de página de banco de dados, em bytes (1024 através default). Outros valores legais para bancos de dados são: 2048, 4096, e 8192.

Na área Restore Options, estabeleça as opções para iniciar o Restore.

**Nota:** Restauração de Multi-volume não é aceito. Quando o banco de dados reside em mais de uma fita, execute **gbak** no console do servidor para executar a restauração.

## Restore options

### Replace Existing Database

Não será escrito um arquivo de banco de dados existente e sim haverá substituição de um banco de dados existente. Se você tentar restaurar um banco de dados existente, e esta opção não estiver selecionada, o processo *restore* falhará.

### Commit After Each Table

Normalmente, o Gerenciador do Servidor restaura todo o metadata antes de restaurar qualquer dado. Selecionando Commit After Each Table, o Gerenciador restabelece o metadata e os dados junto a cada tabela, e procede o comit em uma tabela de cada vez. Esta opção é útil quando você está com dificuldade em restaurar um arquivo posterior; como por exemplo, se o dados estão corrompidos ou inválidos, de acordo com restrições de integridade.

### Restore Without Shadow

Restabelece um banco de dados sem restaurar sua sombra (Shadow), se:

Está restabelecendo um servidor que suporta Shadow para um tipo de servidor que **não** suporta Shadow.

O banco de dados que você está restabelecendo é uma cópia de um banco de dados existente que tem uma Shadow.

Você está migrando de um banco de dados Interbase para uma versão de servidor que **não** suporta Shadow.

A Shadow terminaria em uma máquina que é indisponível ao banco de dados (devido a uma falha na rede, por exemplo).

### Deactivate Indexes

Normalmente, reconstrói índices quando um banco de dados é restabelecido. Se o banco de dados conter valores duplicados em um índice sem igual quando foi copiado, a restauração falhará. Podem ser introduzidos valores duplicados em um banco de dados, se foram feitos índices temporariamente inativo (por exemplo, permitir inserção de muitos registros ou para restabelecer um índice).

Um índice único deve ser excluído e então deve ser criado novamente. Para mais informação sobre ativar índices, veja Referência de SQL online.

**Nota:** Desativar a opção de Índices também é útil para tornar as operações online de banco de dados mais rápidas. O acesso de dados estará mais lento até que os índices sejam reconstruídos, mas pelo menos o banco de dados estará disponível. Depois que o banco de dados for restabelecido, os usuários podem ter acesso ao banco de dados enquanto você reativa os índices.

### **Do Not Restore Validity Conditions**

Habilitando esta opção, caso já exista dados consolidados, estes dados não estarão sujeitos as restrições do conjunto de restrições que estão sendo restaurados. Mantendo esta opção desabilitada, tentando restabelecer o banco de dados, uma mensagem de erro ocorrerá quando um certo dado não cumprir uma certa restrição..

**Precaução:** Sempre faça uma cópia de metadata antes.

Restabelecer um banco de dados que contém dados nulos, Do Not Restore Validity Conditions. Esta opção apaga a constraint de validação do metadata. Depois que o banco de dados seja restabelecido, mude os dados para tornar isto válido de acordo com as constraints de integridade. Então adicione as constraints que foram excluídas antes.

### **Verbose Output**

Ativa o editor que servirá para acompanhar as mensagens de retorno das operações que acusarem erro.

### **Changing the Database Page Size**

Servidor de Workgroup copia página de banco de dados com tamanhos de 1024, 2048, 4096, e 8192 bytes. O default é 1024 bytes.

Mudar o tamanho da página pode melhorar desempenho porque:

Armazenar e acessar dados Blob é muito eficiente quando o Blob inteiro resida em uma única página de banco de dados. Se uma aplicação armazena muitos Blobs que excedem 1K e usar um tamanho de página maior, reduz o tempo para ter acesso aos dados Blob.

O tem melhor performance ao acessar strings quando acomodadas em uma única página. Considere aumentar o tamanho de página, quando usar campos Blob que armazenam grande massa de texto.

Se um banco de dados tem um índice grande e aumenta o tamanho de página de banco de dados reduz o número de níveis na hierarquia de índice. Índices trabalham mais rapidamente se a profundidade deles/delas é mantida a um mínimo. Escolha estatísticas de Banco de dados para exibir estatísticas de índice, e considere aumentar o tamanho de página se profundidade de índice é maior que dois em qualquer índice freqüentemente usado.

Se a maioria das transações envolvem só algumas linhas de dados, um tamanho de página menor pode ser apropriado, desde que menos necessidades de dados sejam passadas de um lado para outro e menos memória é usada pelo esconderijo de disco.

### **Restoring a database to multiple files**

Você poderia querer restabelecer um banco de dados para múltiplos arquivos distribuídos entre discos diferentes que provêem mais flexibilidade alocando recursos de sistemas.

Restabelecer um banco de dados para banco de dados de arquivo múltiplo clique no botão multiple files.

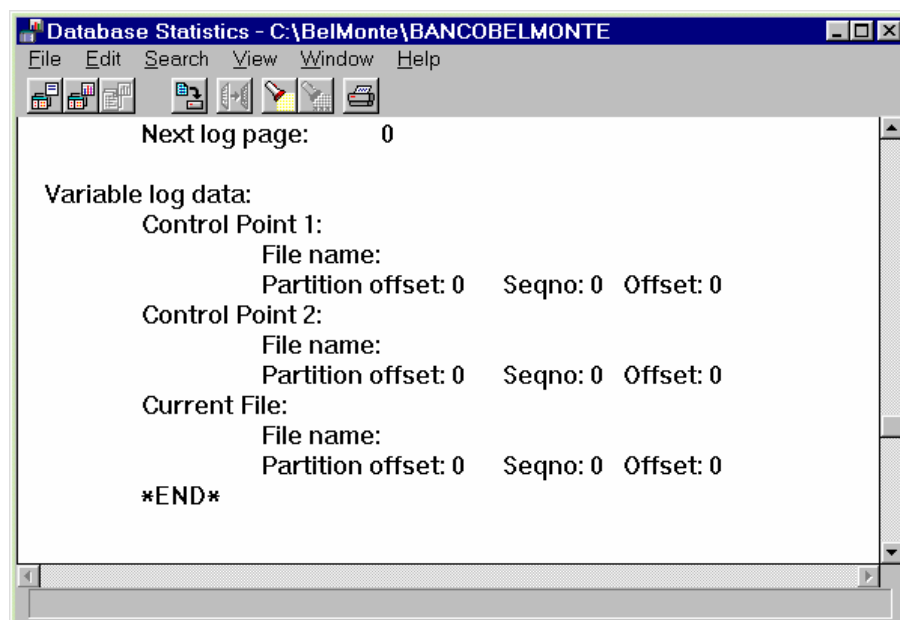
Especifique nomes de arquivos para restabelecer, digitando o nome de cada arquivo para restore, digite também o número de páginas para estes arquivos campo de texto abaixo. O número mínimo de páginas em um arquivo é 200. Escolha salvar, e o nome de arquivo aparecerá à direita na Lista de Arquivo do lado da caixa de diálogo.

## Tasks | Database Statistics

Para obter estatísticas de banco de dados, selecione o banco de dados desejado na árvore de server/database e então escolha Tasks | Database no menu. Isto abre a janela de Estatísticas de Banco de dados.

### Database Statistics window

A janela de Estatísticas de Banco de dados automaticamente exibe várias estatísticas para o banco de dados atualmente selecionado. Você pode localizar informação específica entre as estatísticas, copiar e imprimir.



### Database Statistics menus

#### File Menu

Save As	Copia o texto na janela para um arquivo.
Print To	Imprimir o texto da janela.
Print Setup	Configurar sua impressora.
Close	Fechar a janela.

#### Edit Menu

Copy	Copiar o texto atualmente selecionado para o clipboard.
------	---

#### Search Menu

Find	Procurar um texto específico.
Next	Procurar a próxima ocorrência do último texto pesquisado.

#### Window Menu

Main Window	Altera entre esta janela e a do servidor.
-------------	---

## View Menu

Exibe estatísticas do header de banco de dados, a mais velha transação e número de páginas alocadas. Também podem ser analisadas páginas de banco de dados para determinar o número de páginas e a distribuição de abastecimento de páginas dentro de cada tabela. Opções de visão são:

Database Summary  
Database Analysis  
WAL Statistic(Somente Servidores NetWare)

## View | Database Summary

Exibir estatísticas sumárias sobre o banco de dados atual. A informação seguinte é exibida:

### Database name information

As primeiras exibições de linha no nome e localização do arquivo de banco de dados primário.

### Database header page information

**Checksum:** O header chamam checksum. Este é um valor único computado de todos os dados na página de header. Quando a página de header é armazenada a disco e depois é recobrada, o checksum da página recobrada é recomputado e comparado ao valor armazenado para assegurar que a informação está correta.

**Generation:** Contador incrementou cada página de header de tempo é escrita.

**Page size:** O tamanho de página de banco de dados atual em bytes.

**ODS version:** .A versão da estrutura do banco de dados.

**Oldest transaction:** A transação ID numeram da mais velha transação não marcada cometido (ativo, em limbo, ou retrocedida).

**Oldest active:** A transação que ID numeram da mais velha transação ativa.

**Next transaction:** A transação número de ID que será nomeado à próxima transação. A diferença entre a mais velha transação e a próxima transação determina quando banco de dados varrendo acontecerão. Por exemplo, se a diferença é maior que 20.000, então através de falta o banco de dados será varrido.

**Sequence number:** O número de sucessão da página de header (zero é usado para a primeira página, um para segunda página, e assim por diante).

**Next attachment ID:** Numero do próximo anexo de banco de dados.

**Shadow count:** O número de arquivos de sombra definido para o banco de dados.

**Number of cache buffers:** O número de pára-choques de página no esconderijo de banco de dados.

**Next header:** O ID da próxima página de header.

**Creation date:** A data quando o banco de dados foi criado.

## Variable header data

**Shared Cache file:** Não usado.

**Sweep Interval:** Número de transações entre varreduras de banco de dados.

## Database file sequence

Esta seção lista todos os arquivos associados com o banco de dados e inclui qualquer arquivo secundário.

## Database log page information

Esta seção contém informação sobre a página de tronco de banco de dados. Isto só aplica a bancos de dados com o escrever à frente que o tronco habilitou (só em servidores NetWare).

**Next log page:** Próxima página de tronco: Para uso futuro.

Para cada ponto de controle (posto de fiscalização), a informação seguinte é listada:

Controle ponto 1 é o posto de fiscalização encontrado. Controle ponto 2 é o posto de fiscalização antes do último. Arquivo atual é o arquivo para o qual WAL está escrevendo atualmente.

**Partition offset:** Reservado para uso futuro.

**Seqno:** Número de sucessão do arquivo de tronco atual na sucessão WAL.

**Offset::** O byte compensou no arquivo de tronco atual.

## View | Database Analysis

Escolha Visão | Análise de Banco de dados na janela de Estatísticas de Banco de dados para exibir uma análise cheia do banco de dados atual na área de exibição de texto.

Primeiro, o header de banco de dados chamam informação é exibida. Esta é a mesma informação gerada por Visão | Resumo de Banco de dados. Então informação sobre todas as mesas e índices no banco de dados é exibida.

## Table and index information

Então informação sobre cada mesa e indexa no banco de dados é exibido.

## Primary pointer page

A página que é a primeira página de ponteiro para a mesa.

## Index root page

O número de página que é a primeira página de ponteiro para índices.

## Data pages

O número total de páginas de dados.

## Average fill:

A porcentagem comum para a qual as páginas de dados estão cheias.

## Fill distribution

Um histograma que mostra o número de páginas de dados que são enchidas às porcentagens.

Para cada índice no banco de dados:

**Index**

O nome do índice.

**Depth**

O número de níveis na árvore de página de índice.

**Leaf buckets**

O número de folha (nível de fundo) páginas na árvore de página de índice.

**Nodos**

O número total de páginas de índice na árvore

**Average data length**

A duração comum de cada chave em bytes.

**Total dup**

O número total de filas que têm índices duplicados.

**Max dup**

O número de duplicatas no índice com as demais duplicatas.

**Fill distribution**

Um histograma que mostra o número de páginas de índice encheu às porcentagens especificadas.

Se a profundidade da árvore de página de índice é maior que três, ordenar então podem não ser tão eficientes. Reduza a profundidade da árvore na página de índice, aumente o tamanho da página. Se aumentando o tamanho de página não reduz a profundidade, então devolva ao seu tamanho prévio.

**View | WAL Statistics**

WAL só é apoiado em servidores de NetWare distantes.

Para servidores de NetWare, escolha Visão | Estatísticas de WAL. Para exibir estatísticas, estatísticas de WAL consistem em várias seções separadas por linhas horizontais:

a primeira seção lista o WAL segmentam bandeiras. Estas podem ser úteis a apoio técnico diagnosticando problemas. Logo é o tamanho de página de banco de dados atual, em bytes, e o dono e grupo IDs. As listas de linhas finais: número de conexões e desconexões do banco de dados. A diferença entre as conexões e desconexões menos dois é o número de usuários conectados ao banco de dados, desde que lá seja uma conexão para a linha de esconderijo e para a linha de WAL.

A próxima seção lista o tamanho configurado (em bytes) e número de "pára-choques" de WAL. Em lançamentos futuros em plataformas que apoiam arquivos, os "pára-choques totais" podem aumentar automaticamente se necessário. A informação restante conta o que o sistema está fazendo com os "pára-choques", e pode ser útil ao apoio técnico diagnosticando problemas.

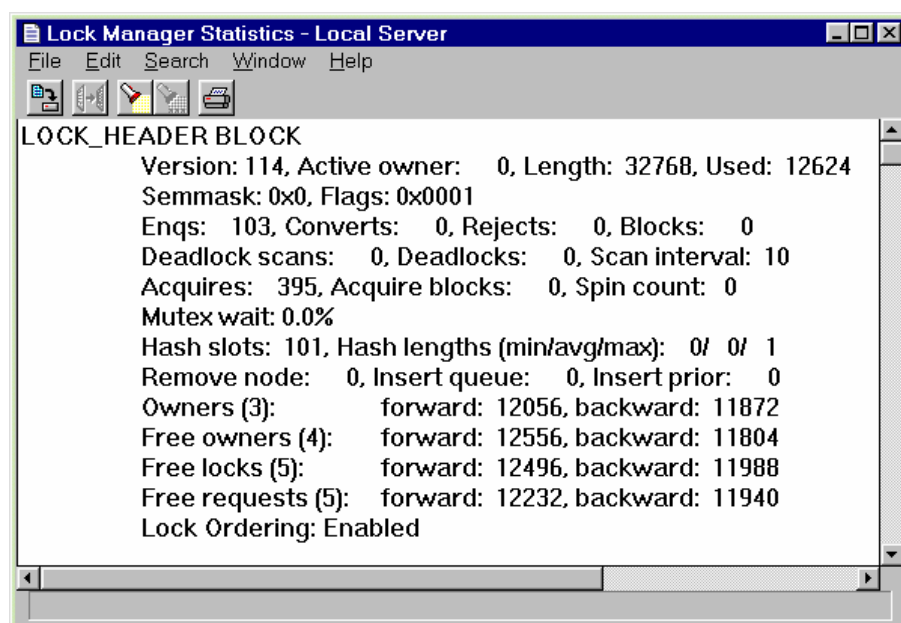
A próxima seção dá o nome do arquivo de tronco atual, no arquivo ao que os registros de tronco estão sendo escritos, e o tamanho do arquivo (rollover).

A próxima seção lista o intervalo de posto de fiscalização máximo em bytes, o intervalo de posto de fiscalização atual em bytes. Para cada arquivo de tronco: número de sucessão de posto de fiscalização atual, nome do arquivo de tronco, que partição de tronco compensou, e posto de fiscalização que compensou.

As próximas duas seções listam o WAL escritor processo ID, o processo ID do último processo em adquirindo o protocolo de WAL, o número total de WAL I/Os, o tamanho comum de cada I/O (em bytes), e o próximo número de sucessão de bloco.

A próxima seção lista grupo de informação. Use o grupo comum de tamanho para determinar o grupo efetivo de valor por seu banco de dados. Se o valor listasse depois de "grpc espera micro-segundos" é zero, então nenhum grupo está acontecendo.

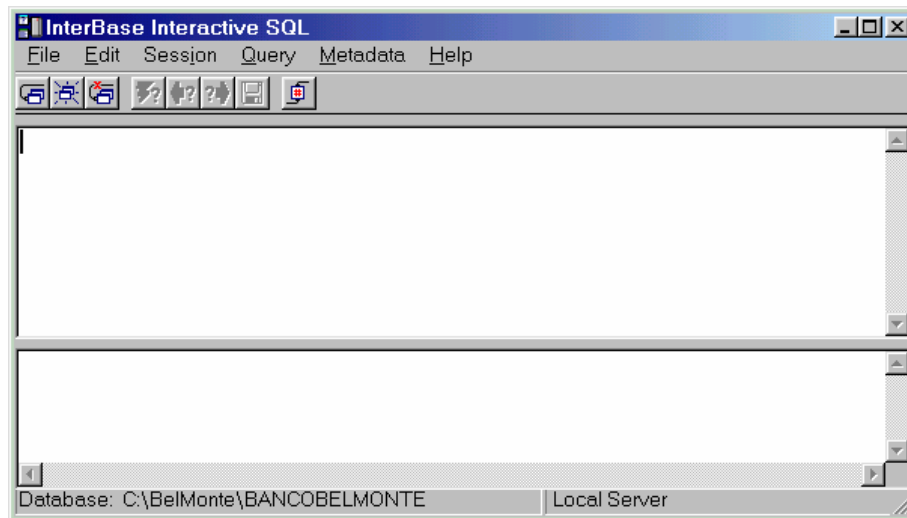
O final duas seções provêem informação que pode ser útil a Borland apoio técnico diagnosticando problemas.





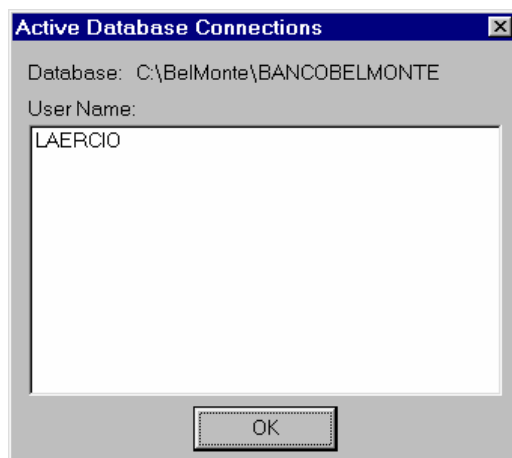
## Tasks | Interactive SQL

Permite estabelecer comunicação e interação com o banco de dados. É utilizado para executar script de um modo geral. Uma alternativa, é utilizar o **SQL Explore** que acompanha o **Delphi**.



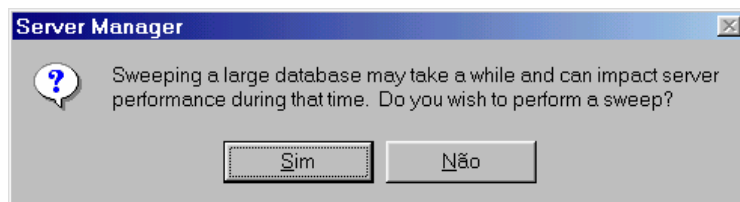
## Maintenance | Database Connection

Exibe todos os bancos de dados ativos.



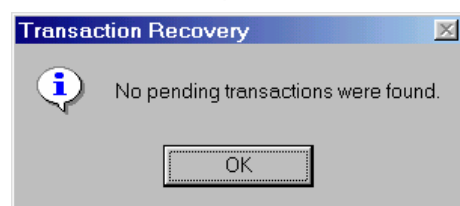
## Maintenance | Database Sweep

Exibe a caixa de dialogo abaixo.



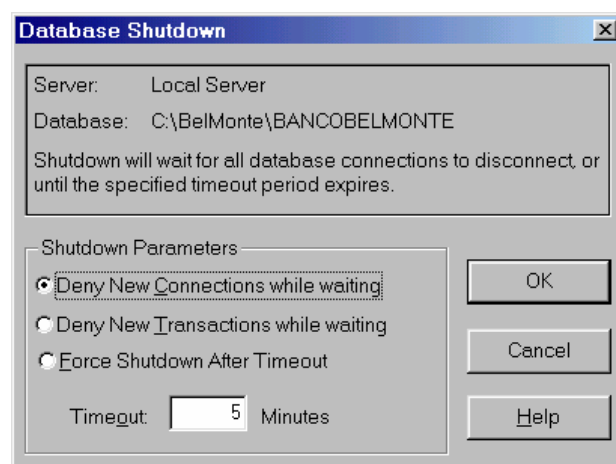
## Maintenance | Transaction Recovery

Retrocede transação pendentes.



## Maintenance | Database Shutdown

Providencia condições para baixar o banco de dados.



**Time-out period in minutes.**

O servidor tentará fechar o banco de dados durante este período. Os parâmetros abaixo, definem de que forma ocorrerá.

### **Deny new connections while waiting**

Conexões de banco de dados existentes podem completar as operações. O banco de dados está disponível para novas conexões. Ao término do período de Time-out, se ainda houver conexões ativas, então o banco de dados não será fechado.

### **Deny new transactions while waiting**

Permite que transações existentes sejam concluídas normalmente. Uma vez processadas as transações, o banco de dados será fechado.

Impedirá transações novas e também conexões novas, desde que uma conexão de banco de dados venha requerer uma transação. Ao término do período de Time-out, se ainda há transações ativas, então o banco de dados não será fechado.

### **Force shutdown after the time-out period**

Durante o período de Time-out, não há nenhuma restrição em transações ou conexões. Ao término do período de Time-out, ou assim que não haja nenhuma transação ativa, o banco de dados será baixado, qualquer transação de uncommitted é retrocedida (Rollback), e os usuários serão desconectados.

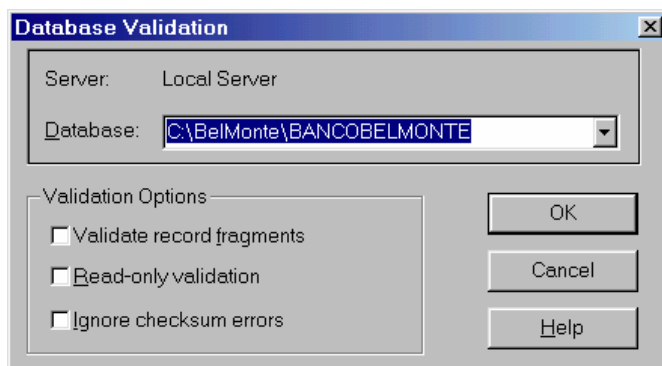
**Precaução:** Forçar paralisação de um banco de dados interferirá em operações de banco de dados normais, e só deveria ser usado depois que todos os usuários forem notificados.

## **Maintenance | Database Restart**

Providencia reativação do banco de dados selecionado na estrutura de árvore.

## **Maintenance | Database Validation**

Gerencia validação verificando a integridade da estruturas de dados. Especificamente:



### **Validate record fragments**

Por default, o banco de dados só valida estruturas de página. Quando assinalado o checkbox Validate record fragments, a validação também ocorre para as estruturas de registros.

### **Read -only validation**

Por default, a execução de validação de um banco de dados afetará os dados em curso por outros usuários. Para prevenir estas validações, assinale o checkbox Read-only validation.

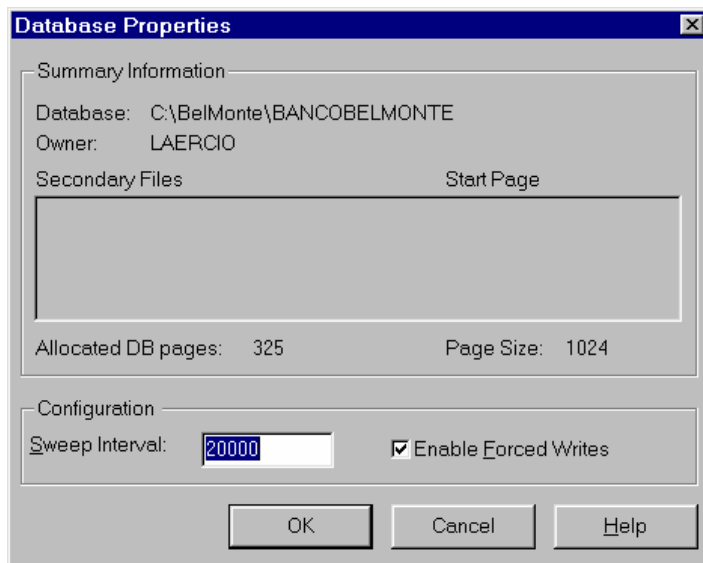
### Ignore Checksum erros

Um checksum é uma análise “página-por-página” de dados para verificar sua integridade. Um modo possível de ocorrência por exemplo, seria uma quebra do sistema. Erros de Checksum indicam corrupção de dados. Para consertar um banco de dados que apresente erros de checksum, selecione o checkbox Ignore checksum erros.

**Nota** Quando for validar um banco de dados, por questões de segurança, promova antes uma cópia do mesmo utilizando o sistema operacional e não o Interbase.

## Maintenance | Database Properties

Esta caixa de diálogo contém uma área de Informação sumária que não permite modificação de parâmetros e uma área de configuração que permite modificação.



Varrer (sweep) um banco de dados é um modo sistemático de remover registros antigos (excluídos e alterados) do mesmo. Este processo periódico impede que um banco de dados venha crescer desnecessariamente. Porém, este recurso também pode reduzir o desempenho do sistema no momento da execução desta tarefa.

O Interbase usa uma arquitetura mult-generational. Isto significa que são armazenadas versões de múltiplos registros de dados diretamente nas páginas de dados. Quando um registro é atualizado ou é apagado, mantém uma cópia do estado antigo do registro e cria uma versão nova. Isto pode aumentar o tamanho de um banco de dados.

O Interbase varre um banco de dados automaticamente a um intervalo fixo: por default, a cada 20.000 transações.

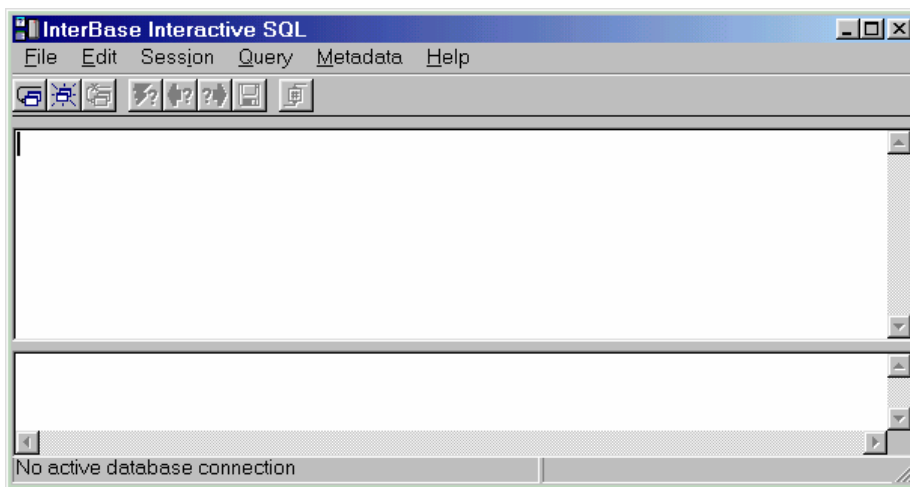
Em plataformas multi-camadas, a varredura ocorre em um processo à parte, de forma que isto não irá bloquear outros processos.

Aplicar Sweep não é o único modo de se promover uma limpeza e reorganização dos dados. Copiar um banco de dados alcança o mesmo resultado, forçando-o extrair somente os registros válidos. Assim sendo, estabelecer backup e restore periodicamente reduz a necessidade de se aplicar sweep.

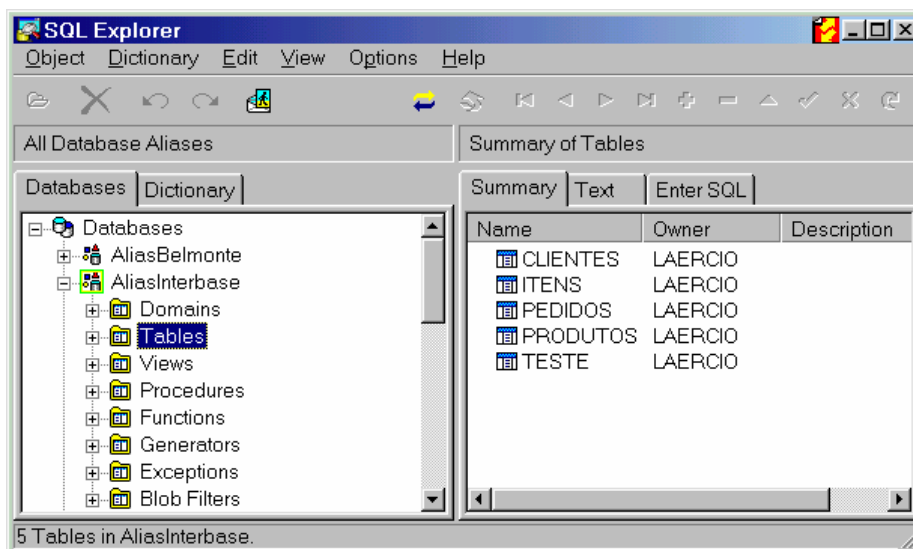
## Capítulo 2

### Interactive SQL

Interface para manipulação da estrutura de um banco de dados. Em suas opções de menu, encontraremos utilidades só encontrada nesta interface. Porém muitas tarefas, principalmente as que dizem respeito a criação dos objetos de um banco de dados, excluindo-se o próprio Interbase, é aconselhável o uso de outras ferramentas, que apresentem melhor interação visual com as partes (coleção de objetos) do banco de dados. A melhor alternativa para os que usam Delphi, seria a adoção do SQL Explorer.



(Interface do Interactive SQL)



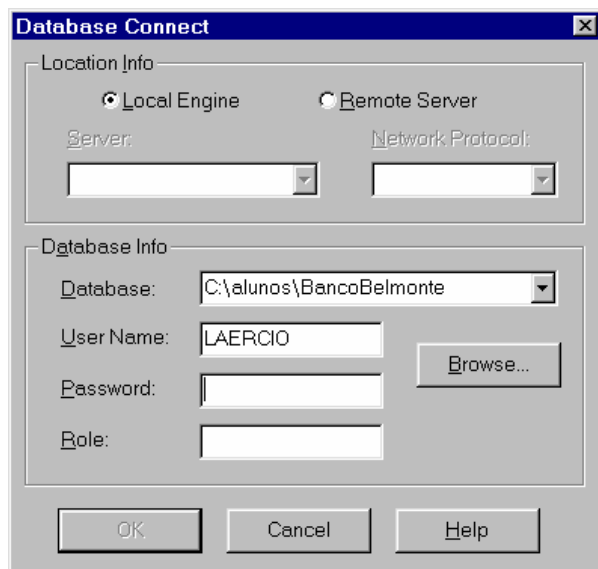
(SQL Explore)

## File | Database Connect

Este diálogo permite conectar um servidor de banco de dados local ou remoto.

No campo de texto Banco de dados, entre com o nome do banco de dados para o qual queira conectar-se (incluindo caminho completo), ou escolha na lista e selecione um banco de dados previamente acessado.

São requeridas um nome de usuário e senha. Esta diálogo informa, extraindo do arquivo INTERBAS os 10 últimos arquivo de bancos de dados para os quais houve conexão, como também o nome do usuário e protocolo de rede utilizado para cada servidor.

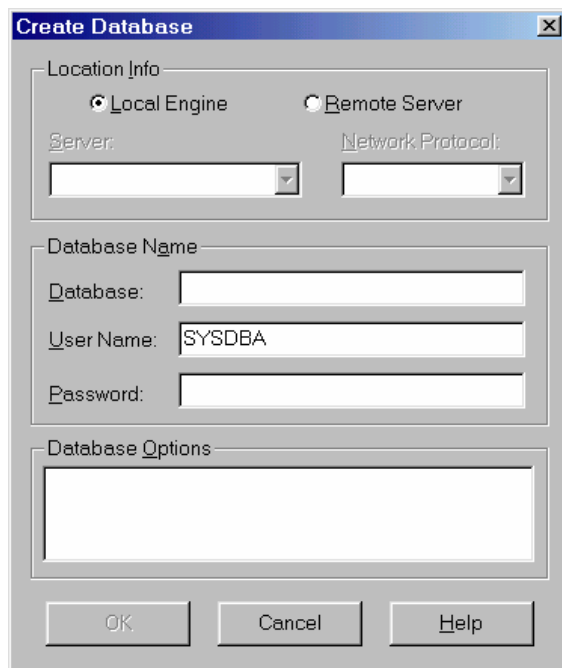


## File | Create Database

Esta caixa de diálogo permite criar um banco de dados novo em um servidor de banco de dados local ou distante.

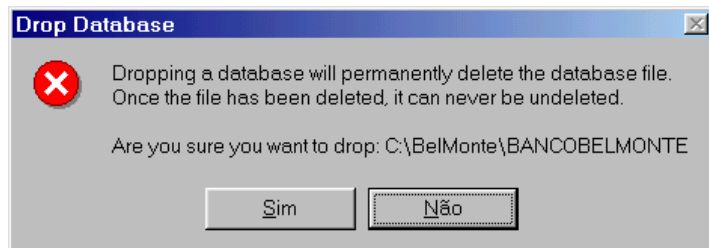
São exigidos um nome de usuário e senha para criar um banco de dados. Naturalmente, este usuário tem que ter permissão para tanto.

No campo de Banco de dados, entre com o nome do banco de dados a ser criado, inclusive o nome de arquivo, com patch, incluindo também uma extensão (opcional).



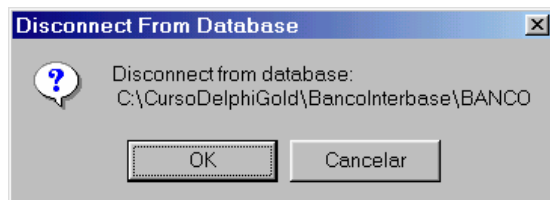
## File | Drop Database

Exibe a caixa de dialogo abaixo, onde o usuário poderá confirmar ou não a exclusão de um banco de dados.



## File | Disconnect Database

Exibe a caixa de dialogo abaixo, possibilitando desconectar um banco previamente conectado.



## **File | Run an Script**

Exibe a caixa de dialogo do Windows "Open File", para escolha de um arquivo padrão texto, contendo qualquer código SQL ou para execução.

## **File | Save Result to a File**

Abre a caixa de dialogo do Windows "Save to File", possibilitando salvar todos os comandos (script) contido na área de trabalho do Interactive SQL.

## **File | Save Session to a File**

Abre a caixa de dialogo do Windows "Save to File", possibilitando salvar todos os comandos (script) aplicados na sessão.

## **File | Commit Work**

Finaliza todas as operações já realizadas com sucesso, mas pendentes na memória.

## **File | Rollback Work**

Cancela todas as operações já realizadas com sucesso, mas pendentes na memória.

## **Edit | Undo - Copy - Past - Select All - Clear Output**

Operações básicas a serem utilizadas entre o clipboard e a área de texto do Interactive SQL.

## **Session | Basic Settings**

Esta caixa de diálogo o permite a modificar todas as configurações básicas do Windows .

### **Display Query Plan:**

Se assinalado, exibe o plano de execução escolhido pelo optimizer. Para modificar o optimizer use a opção de PLANO do SQL declaração SELETA.

### **Auto Commit DDL:**

Seu default é True, promovendo execução de commit automaticamente sobre a área do W.

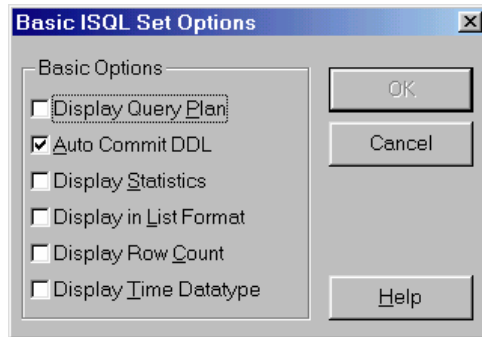


### Display Statistics:

As estatísticas de desempenho aparecem depois do resultado de cada execução:

### Display in List Format:

Exibe os dados em formato de coluna vertical, uma linha de cada vez, com títulos a esquerda e coluna a direita, quando habilitado. Se desabilitado, W exibe dados em formato tabular, que permite apresentar linhas mais longas.



## Session | Advanced Settings

Esta caixa de diálogo permite modificar as configurações avançadas do W que tenham parâmetros.

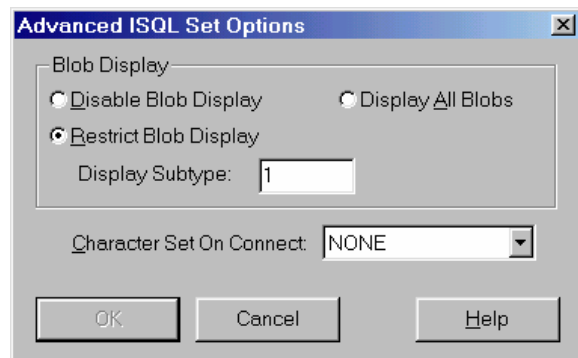
### BLOB display

Determina o modo de exibição de um subtipo BLOB. SELECT sempre exibe um ID BLOB (um identificador para o campo Blob). O default de um SELECT exibirá dados BLOB em baixo da linha associada.

Disable BLOB Display:	Não exibe conteúdos de colunas BLOB.
Display ALL BLOBs:	Exiba dados BLOB de todos os subtipos.
Restrict BLOB Display:	Só exibe conteúdos de colunas BLOB para o subtipo especificado. Use 0 para subtipo desconhecido; 1 para subtipo texto

### Character set

Determina o tipo de caracter ativo fixado para o Interbase, afetando as conexões que ocorram a partir de sua manipulação.



## Session | Display Settings

Exibe como resultado as configurações atuais, como exemplo:

```
SET
Print statistics:      OFF
List format:          OFF
Row Count:            OFF
Autocommit DDL:      ON
Access Plan:          OFF
Display BLOB type:    1
Terminator:           ;
Time:                 OFF
```

## Session | Conect Version

Exibe como resultado dados sobre versão do servidor, como exemplo:

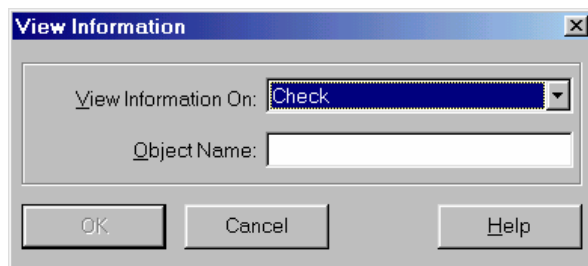
```
SHOW VERSION
Version: WI-V5.1.1.680
/x86/Windows NT (access method), version "WI-V5.1.1.680"
on disk structure version 9.0
```

## Query | Execute - Previous - Next

Opções que se referem a execução do código (instrução SQL /) contido na área de texto do Interactive SQL, possibilitando retroceder a instruções anteriores e avançar a próxima.

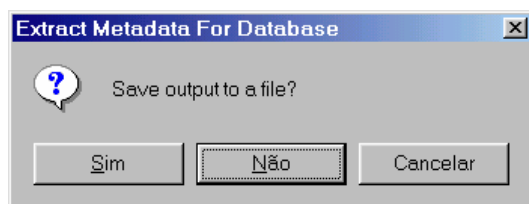
## Metadata | Show

Permite visualizar informações de Metadata exibindo nome, informações sobre todos os objetos do banco de dados, como tabelas, domain, constraints, tabelas de sistema, triggers, e stored procedures



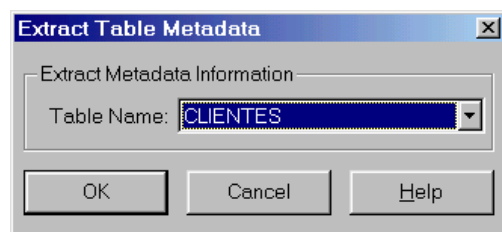
## Metadata | Extract Database

Permite conforme caixa de dialogo abaixo, salvar em arquivo script do banco de dados. Excelente para recriar todo um banco de dados, como também documentar sua estrutura.



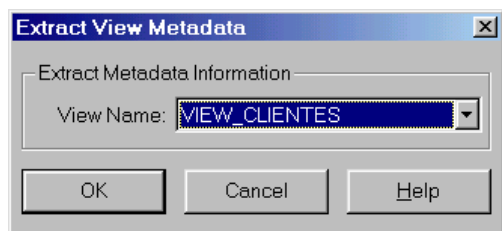
## Metadata | Extract Table

O mesmo que o anterior, mas somente a tabela escolhida.



## Metadata | Extract View

Idem, mas somente a view escolhida.



## Capítulo 3

### Linguagem Procedural

#### Tipos de Dados

Nome	Size	Faixa	Descrição
BLOB	Variável	-	Redimensionável dinamicamente, aplicado em armazenamento de tipos texto longo, imagens e arquivos de som.
CHAR(n)	n Caract.	1 a 32.767 bytes, ou 32 K	Tipo caracter de tamanho fixo
DATE	64 bits	01/01/1000 até 11/01/5941	Inclui também a hora.
DECIMAL	Variável	precisão de 1 a 15 Scale de 1 a 15	Número de ponto flutuante, exemplo: Decima(8,2) - Resultado PPPP.SSS
DOUBLE PRECISION	64 bits	$1.7 \times 10^{-308}$ até $1.7 \times 10^{308}$	Somente para plataformas de 64 bit Tipo científico de 15 dígitos de precisão

#### Funções Especiais

Função	Tipo	Descrição
AVG()	Agregação	Calcula a média dos valores de uma coluna
CAST()	Conversão	Converte uma coluna de um tipo para um outro tipo
COUNT()	Agregação	Retorna o número de linhas resultante de uma Query
GEN_ID()	Numérica	Retorna um número extraído do Interbase
MAX()	Agregação	Retorna o maior valor de uma coluna
MIN()	Agregação	Retorna o menor valor de uma coluna
SUM()	Agregação	Totaliza os valores de uma coluna numérica
UPPER()	Conversão	Converte uma string para maiúsculo

#### Criando o Banco de Dados

1) CREATE DATABASE "C:\CursoDelphiGold\Banco\BANCO" PAGE\_SIZE 1024;

## Construção de Tabelas

Construir tabelas no InterBase significa mais do que criar estruturas que acomodarão dados na forma básica de intercessão de linhas por colunas.

O InterBase é antes de tudo, um Sistema Gerenciador de Banco de Dados Relacional, que conjuga as tarefas de armazenamento de forma a possibilitar o relacionamento dos dados destas estruturas tabular. Desta forma, além dos aspectos básicos como definir quais colunas serão definidas para quais tipos, deve-se observar a análise necessária para modelar o relacionamento entre as tabelas existente.

Já na etapa de definição da estrutura, é possível agregar restrições que garantam a total integridade dos dados, definir valores padrão e status que obrigam ao preenchimento de certas colunas.

- 2) CREATE TABLE CLIENTES (ID\_CLIENTE INTEGER NOT NULL,  
NOME VARCHAR(50) NOT NULL,  
CREDITO NUMERIC(15, 2) NOT NULL,  
STATUS SMALLINT NOT NULL,  
TIPO VARCHAR(1) NOT NULL,  
SNPJ VARCHAR(14) NOT NULL,  
HISTORICO BLOB SUB\_TYPE TEXT SEGMENT SIZE 80,  
FOTO BLOB SUB\_TYPE 0 SEGMENT SIZE 80,  
DATA DATE NOT NULL,  
CONSTRAINT PK\_CLIENTES PRIMARY KEY (ID\_CLIENTE),  
CONSTRAINT UK\_SNPJ UNIQUE (SNPJ));
  
- 3) CREATE TABLE ITENS (ID\_PEDIDO INTEGER NOT NULL,  
ID\_PRODUTO INTEGER NOT NULL,  
QUANTIDADE INTEGER NOT NULL,  
PRECO NUMERIC(15, 2) NOT NULL,  
CONSTRAINT PK\_ITENS PRIMARY KEY (ID\_PEDIDO, ID\_PRODUTO));
  
- 4) CREATE TABLE PEDIDOS (ID\_PEDIDO INTEGER NOT NULL,  
DATA DATE NOT NULL,  
ID\_CLIENTE INTEGER NOT NULL,  
CONSTRAINT PK\_PEDIDOS PRIMARY KEY (ID\_PEDIDO));
  
- 5) CREATE TABLE PRODUTOS (ID\_PRODUTO INTEGER NOT NULL,  
DESCRICAO VARCHAR(25) NOT NULL,  
PRECO NUMERIC(15, 2) NOT NULL,  
SALDO INTEGER NOT NULL,  
CONSTRAINT PK\_PRODUTOS PRIMARY KEY (ID\_PRODUTO));

## Índices

Índice no InterBase corresponde a um mecanismo de simples definição, porém, de extrema eficiência no tocante à performance no acesso a dados.

Numa análise preliminar, podemos estabelecer que baseado nas possibilidades variadas pelo qual uma aplicação pode solicitar informações contidas em uma tabela, conjunto de tabelas e mesmo pela combinação de várias tabelas, a implementação de estruturas de índices determinara o diferencial destas operações no tocante ao tempo de retorno.

Os índices ficam armazenados em estruturas particulares a do bloco de dados, sendo composto por seu próprio bloco. A rapidez no acesso a dados baseado em critério imposto pelas restrições na cláusula Where ou combinação de colunas na mesma cláusula, bem como, pela referência explícita na cláusula Order By é alcançada em função do mapeamento pela leitura no bloco de dados referente a estrutura de índice, pela imediata localização na página de dados. O mecanismo InterBase para recuperação de dados ordenado ou por critérios, não obriga a existência de índices para as colunas envolvidas. Porém sua performance e até mesmo a possibilidade do InterBase atingir o objetivo, estará intimamente ligado aos recursos disponíveis para o Servidor InterBase ativo. Vale ressaltar, que a demanda de dados no servidor InterBase pode atingir a magnitude dos terabytes, inviabilizando algumas operações em detrimento da capacidade ou não do mecanismo ter ou não esta característica.

Alguns exemplos ainda neste capítulo, exemplificam, com base no script apresentado, como criar tais estruturas de dados. Um índice pode ser criado para atender a demanda de dados com base na identificação do seu conteúdo da coluna para qual foi definido, ou pela combinação de colunas. Nisto, caracteriza-se pela consequência do aspecto performance.

Mas um índice também pode ser criado com o objetivo de definir uma condição (Constraint) de unicidade (Unique) para uma coluna ou combinação de colunas, sem deixar agregar os conceitos associados a performance.

Há alguns critérios que devem ser observados, quando da criação de um índice. Algumas colunas são boas candidatas a comporem índices, ficando a definição para os aspectos relacionados a seguir:

- Colunas utilizadas com frequência em cláusulas Where e Order By.
- Colunas utilizadas na operações que envolvam Group By
- Colunas referenciadas nas funções de agregação Min e Max
- Colunas definidas com Chave Estrangeira e Chave Primária

Alguns critérios determinam o sentido contrário, ou seja, as colunas não candidatas a índice:

- Colunas em que a ocorrência de valores nulos é constante
- Colunas de tipos Booleano, Blob
- Colunas com muita ocorrência de linhas idênticas
- Colunas de tabelas com poucos registros

De forma conclusiva, ressalta-se o fato de não nos perdermos em adoção de um número excepcionalmente exagerado na criação de índices, já que isto acarretaria uma sobre carga no processo de manutenção (Update, Insert e Delete), devido ao fato das diversas estruturas de índices terem que ser atualizadas.

Na estrutura de dados InterBase, não há a indexação lógica dos dados, assim sendo, para visualizar os dados numa certa ordem, é necessário fazer uso da cláusula Order By em um comando Select.

**Sintaxe:**

- 6) CREATE DESCENDING INDEX IDX\_CREDITOCLIENTES ON CLIENTES(CREDITO);
- 7) CREATE INDEX IDX\_NOMECLIENTES ON CLIENTES(NOME);
- 8) CREATE INDEX IDX\_NOME ON TESTE(NOME);
- 9) ALTER TABLE ITENS ADD CONSTRAINT FK\_ID\_PEDIDO FOREIGN KEY (ID\_PEDIDO) REFERENCES PEDIDOS(ID\_PEDIDO);
- 10) ALTER TABLE ITENS ADD CONSTRAINT FK\_ID\_PRODUTO FOREIGN KEY (ID\_PRODUTO) REFERENCES PRODUTOS(ID\_PRODUTO);
- 11) ALTER TABLE CLIENTES ADD CONSTRAINT CC\_STATUS CHECK (STATUS IN (0, 1));
- 12) ALTER TABLE CLIENTES ADD CONSTRAINT CC\_TIPO CHECK ((Tipo = 'J' ) or (Tipo = 'F'));
- 13) ALTER TABLE PRODUTOS ADD CONSTRAINT CC\_SALDO CHECK (Saldo >= 0);

## Generator

Objeto que funciona como um contador, fornecendo um número seqüencial. É amplamente utilizado em triggers, que visam atribuir um valor para campos chave.

- 14) CREATE GENERATOR GEN\_CLIENTES;
- 15) CREATE GENERATOR GEN\_PEDIDOS;
- 16) CREATE GENERATOR GEN\_PRODUTOS;

## Exception

A linguagem procedural ISQL, contém um mecanismo próprio para lidar com as ocorrências de erros (Exceptions).

Um erro é caracterizado como sendo a ocorrência de uma exception, que dentro do contexto da ISQL pode ser identificado e tratada.

Quando ocorre um erro, o InterBase levanta uma exceção, que informa a conexão responsável em três parâmetros:

SQLCODE	Mensagem.	Descrição
< 0	SQLERROR	Ocorreu erro, não sendo executado a instrução ISQL.
0	SUCCESS	Sucesso na execução dos comandos ISQL.
+1 até 99	SQLWARNING	System warning or informational message
+100	NOT FOUND	Registros não encontrados, ou atingido o último registro.

**Nota:** Consultar referências para ver tabela de erros possíveis.

Baseado nestes parâmetros pode-se agregar a lógica de um bloco ISQL, procedimentos específicos para detectar a ocorrência da exception, bem como, um tratamento específico, inclusive com tomada da decisão.

O conceito de exception no InterBase, não significa necessariamente que um erro tenha realmente ocorrido. Uma simples declaração do comando Exception indicando uma exceção previamente criada, levanta uma exceção que caracteriza de forma forçosa uma situação de erro.

### Criando Exceptions:

```
20) CREATE EXCEPTION EXC_REGEXISTENTE  
"INI Este registro já existe. FIM";
```

```
21) CREATE EXCEPTION EXC_USUARIOSEMPERMISS  
"INI Esta conta de usuário não tem permissão para esta operação. FIM";
```

```
22) CREATE EXCEPTION EXC_SALDO  
"Não há saldo para este produto";
```

### Utilizando Exceptions:

```
IF (SaldoProduto < New.Quantidade) THEN  
EXCEPTION EXC_SALDO;
```



## O Comando Select

O comando Select, como sugere o nome, é aplicado nas operações que objetivam acessar dados nas estruturas de esquema, para interação e visualização.

Agregado às cláusulas From (de uso obrigatório), Where, Order By, Group By e Having (uso opcional), compõe um mecanismo eficiente e flexível para conjugar uma necessidade de visão de dados.

Comparado com qualquer dialeto das diversas ferramentas existentes, apresenta uma facilidade com redução substancial da escrita do código.

Na seqüência, de forma gradativa, apresentaremos o comando Select com seus agregados, direcionados para as tabelas de exemplo do tópico Tables.

### Comando Select

SELECT	Lista de colunas requeridas
INTO	Lista de variáveis que receberão os valores (colunas) retornadas de SELECT
FROM	Lista de tabelas dos campos referidos em SELECT e considerados em WHERE
WHERE	Especifica as restrições que afetam as linhas a serem selecionadas, bem como, define a condição de junção entre as tabelas envolvidas.
GROUP BY	Define as colunas que fazem parte da condição de agrupamento. Todas as colunas requeridas em SELECT, exceto as que representam funções, devem ser indicadas.
HAVING	Restringe linhas alcançadas pela condição de agrupamento.
UNION	Combina o resultado de dois ou mais SELECT produzindo uma única tabela.
ORDER BY	Define quais colunas farão parte da condição de ordenação. A nomenclatura ASC (default) define ordenação ascendente ou DESC para descendente.
PLAN	Specifies the query plan that should be used by the query optimizer instead of one it would normally choose
FOR UPDATE	Specifies columns listed after the SELECT clause of a DECLARE CURSOR statement that can be updated using a WHERE CURRENT OF clause

Select simples  
Select \* From Produtos

Retorna:

ID_PRODUTO	DESCRICAO	SALDO	PRECO
1	Teclado	1000	12
2	Drive 1.44	1000	25
3	Mouse	1000	5
4	Monitor	1000	280
5	Gabinete	1000	40

5 rows selected.

Onde:

Select - Comando \* Indica todas as colunas  
From - Cláusula que define quais as tabelas envolvidas  
Clientes Nome da Tabela

```
Select Clientes.Nome, Clientes.CNPJ  
From Clientes
```

Retorna:

NOME	CNPJ
Lydia Guerço Rodrigues	4235445324534534
Etienne Malman	5353325364569000
Cristine M L Branco G. Rodrigues	3543545544554888
Evelyn Branco Malman	5242455254545480
Raphael Malman Costa	5454544545457877
Laercio Guerço Rodrigues	3543535534535454
Horacio Rodrigues	7557765765776766
Dinorah Lima Branco	6677777777675678
Renata Rozendo Rodrigues	5675675666565656
Pedro Henrique Rozendo Rodrigues	5454673214324233

11 rows selected.

Neste exemplo, de forma explícita, é feito referência às colunas que se deseja como retorno. Este critério resulta numa carga menor no tocante ao processamento e tráfego de dados, devido a diminuição horizontal da estrutura.

```
Select Clientes.Nome, Clientes.CNPJ  
From Clientes  
Where Clientes.Datacad >= ' 01/OCT/99 '
```

A cláusula Where acrescentada ao procedimento determina uma diminuição no sentido vertical, pela restrição aplicada às linhas (registros) que não atendam ao critério.

```
Select Clientes.Nome, Clientes.CNPJ  
From Clientes  
Where Clientes.Datacad >= ' 01/01/1999 '  
Order By Clientes.Nome
```

Retorna:

NOME	DATA CAD
Cristine M L Branco G. Rodrigues	09-DEC-99
Dinorah Lima Branco	08-DEC-99
Pedro Henrique Rozendo Rodrigues	08-DEC-99
Raphael Malman Costa	08-DEC-99
Victoria Emanuelle Alves	08-DEC-99

Com este último adendo (Order By), vemos o retorno de dados apresentado em uma ordem que não a natural. Vale ressaltar, que não é necessário a existência de índices para que haja sucesso nesta operação. Porém, conforme já abordado no tópico índice, todo o benefício inerente a existência de tais índices, se aplica aqui.

Select com Join

```
Select Clientes.Nome, Pedidos.DataPed, Pedidos.ID_Cliente
From Clientes, Pedidos
Where Clientes.ID_Cliente = Pedidos.ID_Cliente
Order By Pedidos.DataPed
```

Aqui é apresentado uma instrução que promove um Join (Junção) entre duas tabelas. Na clausula Select pode ser observado a referência a campos de duas tabelas, que obriga referencia-las na clausula From. Na prática, o que temos aqui, é uma representação de uma teoria de conjuntos no tocante a comparação das ocorrências pela igualdade. Isto fica explícito pela seqüência da clausula Where (Clientes =Pedidos.ID\_Cliente), onde haverá a seleção de dados baseado nas múltiplas ocorrências em Pedido.ID\_Cliente para cada ocorrência Clientes.ID\_Cliente. Normalmente estas referências às colunas de múltiplas tabelas (o Intercepta até 255 para comprar um Join) se dão entre as chaves primárias e chaves secundárias, sem que isto seja uma regra.

Usasndo Group By e Having

```
Select Clientes.Nmoe,Sun(Itens.Quantidade*Itens.PrecoVenda).o total
From Clientes, Pedidos, Itens
Where Clientes.ID_Cliente= Pedidos.ID_Cliente
      AND Pedidos.ID_Pedido =Iten.ID_Pedido
Grop By Clientes.Nome
Having Sum(Itens Quantidade*Itens.PrecoVenda) >=1 000,00
Order By Clientes.Nome DESC
```

A clausula Group By promove a agregação dos elementos baseado na lista de colunas indicada. Todas as colunas relacionadas na clausula Select devem ser referenciados em Select, as colunas retornadas pelas funções SUM, MIN, MAX e AU5. Na prática, a execução de uma instrução que contenha em Group By necessita de pelo menos uma função na clausula Select.

A clausula Having, que é opcional, funciona como restrição para o resultado alcançado pelo agrupamento de Group By.

## Select Into

Select Into é uma variação do comando Select dentro do PL/SQL, que objetiva a atribuição de valores retornados pela instrução para as variáveis rejeitadas na cláusula Select. É importante ressaltar, que o uso é exclusivo de bloco PL/SQL.

```
Declare Variable Var_Navre Char(50);
Declare Variable Var_Data Date;

Begin
    Select Clientes.Nome, Clientes.Data
    Into Var_Nome, Var_Data
    Where Clientes.ID_Clientes = 10
End
```

## View

O objeto View é criado com o objetivo de se dimensionar o acesso a uma tabela ou combinação de tabelas, de forma a definir a estrutura no sentido horizontal (pela referência criteriosa na cláusula Select de algumas colunas) e no sentido vertical (pela aplicação de restrições na cláusula Where)

Uma View é um objeto composto por uma instrução Select, podendo ser aplicado todo o seu composto (conforme abordado no tópico referente ao comando Select), ou seja Select, From, Where, Group By, Having e Order By.

Sua aplicação prática, se estende desde aspectos administrativos, como por exemplo, o fornecer acesso limitado a uma tabela para usuários, como também, elaborar instruções que combinem tabelas (Join), para atender demanda de solicitação de dados numa dimensão não encontrada pelas estruturas lógicas definidas. Como exemplo para esta segunda hipótese, citamos a necessidade de um acesso a dados que tivesse como retorno, uma lista de clientes com compras num certo período e que apresentasse a descrição dos produtos envolvidos.

Todas estas possibilidades se aplicam a criação de View que façam referência a outras View combinadas com Tabelas.

**17) CREATE VIEW VIEW\_CLIENTES (ID\_CLIENTE, NOME, CREDITO, STATUS) AS**

```
SELECT      CLIENTES.ID_CLIENTE, CLIENTES.NOME,
            CLIENTES.CREDITO, CLIENTES.STATUS
FROM        CLIENTES;
```

**18) CREATE VIEW VIEW\_FULLSELECT (DESCRICAÇÃO, TOTAL) AS**

```
SELECT      PRODUTOS.DESCRICAÇÃO,
            SUM(ITENS.QUANTIDADE * ITENS.PREÇO) AS TOTAL
FROM        PRODUTOS, ITENS, PEDIDOS
WHERE       ITENS.ID_PRODUTO = PRODUTOS.ID_PRODUTO AND
            ITENS.ID_PEDIDO = PEDIDOS.ID_PEDIDO
GROUP BY   PRODUTOS.DESCRICAÇÃO
HAVING     SUM(ITENS.QUANTIDADE * ITENS.PREÇO) > 1000;
```

**19) CREATE VIEW VIEW\_PRODITENSPEDCLI**

```
(DESCRICAÇÃO, QUANTIDADE, PREÇO, DATA, NOME) AS
SELECT      PRODUTOS.DESCRICAÇÃO, ITENS.QUANTIDADE,
            ITENS.PREÇO, PEDIDOS.DATA, CLIENTES.NOME
FROM        PRODUTOS, ITENS, PEDIDOS, CLIENTES
WHERE       PRODUTOS.ID_PRODUTO = ITENS.ID_PRODUTO AND
            PEDIDOS.ID_PEDIDO = ITENS.ID_PEDIDO AND
            CLIENTES.ID_CLIENTE = PEDIDOS.ID_CLIENTE;
```

## Procedures

Stored Procedures é um objeto de banco de dados que contém comandos que se resumem em linhas de comando que processam tarefas, ou fazem chamadas a outras procedures.

Uma Stored Procedure (procedimento armazenado), além de executar instruções, possibilita a interação e aplicação que a invoca e o seu corpo de código por intermédio dos parâmetros definidos no seu cabeçalho.

A execução de uma Stored Procedure sempre ocorre de forma explícita, ou de forma interativa, como por exemplo pela utilização do SQL\* Plus ou sendo referenciada no corpo do código de uma aplicação Cliente.

Neste último exemplo um procedimento armazenado, se revela um excelente núcleo para manter código que processa tarefa envolvendo a estrutura de dados. Manter o código de processamento armazenado no banco de dados, traz de imediato a facilidade na manutenção em um único local, refletindo na aplicação que faz referência a esta procedure.

Se elevarmos a análise para um ambiente de desenvolvimento corporativo onde se caracteriza a heterogeneidade pela adoção de linguagens de programação múltiplas, o uso de Stored Procedure alcança importância sem igual, já que seu uso conforme abordado no parágrafo anterior, irá atingir uma maior magnitude.

Diferente de uma Trigger, que tem o seu processamento disparado pela ocorrência de comandos para a tabela associada, uma Stored Procedure ocorre quando explicitamente solicitada.

Na sequência, teremos alguns exemplos práticos do cotidiano, que visam explorar a plenitude do uso de procedures no contexto do desenvolvimento de sistemas comerciais.

```
23) CREATE PROCEDURE PROC_GENCLIENTES RETURNS (NUMERO INTEGER) AS  
begin  
    NUMERO = GEN_ID(Gen_clientes, 1);  
    Suspend;  
end
```

```
24) CREATE PROCEDURE PROC_GENPRODUTOS RETURNS (NUMERO INTEGER) AS  
begin  
    NUMERO = GEN_ID(Gen_Produtos, 1);  
    Suspend;  
end
```

```
25) CREATE PROCEDURE PROC_GENPEDIDOS RETURNS (NUMERO INTEGER) AS  
begin  
    NUMERO = GEN_ID(Gen_pedidos, 1);  
    Suspend;  
end
```

```
29) CREATE PROCEDURE PROC_SELECT RETURNS (VARNOME CHAR(60)) AS  
BEGIN  
    FOR  
        SELECT NOME FROM CLIENTES INTO :VARNOME  
    DO  
        SUSPEND;  
END
```

```

26) CREATE PROCEDURE PROC_INCPRODUTOS (PARDESCRICA0 VARCHAR(25),
      PARPRECO NUMERIC(15, 2),
      PARSALDO INTEGER) AS
  DECLARE VARIABLE
  VarReg INTEGER;
begin
  BEGIN
    VarReg = GEN_ID(Gen_Produtos, 1);
    Insert Into Produtos Values( :VarReg,
      :ParDESCRICA0,
      :ParPRECO,
      :ParSaldo);
    WHEN SQLCODE -530 DO
      EXCEPTION EXC_REGEXISTENTE;
  END
  SUSPEND;
END

```

```

27) CREATE PROCEDURE PROC_ALTPRODUTOS (PARID_PRODUTO INTEGER,
      PARDESCRICA0 VARCHAR(25),
      PARPRECO NUMERIC(15, 2),
      PARSALDO INTEGER) AS
begin
  Update Produtos Set
    DESCRICA0 = :PARDESCRICA0,
    PRECO = :PARPRECO,
    Saldo = :PARSALDO
  Where ID_Produto = :PARID_PRODUTO;
END

```

```

28) CREATE PROCEDURE PROC_EXCPRODUTOS (PARID_PRODUTO INTEGER) AS
begin
  Delete From Produtos
  Where ID_Produto = :PARID_PRODUTO;
END

```

```

29) CREATE PROCEDURE PROC_SELECT RETURNS (VARNOME CHAR(60)) AS
BEGIN
  FOR
    SELECT NOME FROM CLIENTES INTO :VARNOME
  DO
  SUSPEND;
END

```

```

30) CREATE PROCEDURE PROC_ESTATVENDA (PARID_CLIENTE INTEGER,
      PARDATAINI DATE,
      PARDATAFIM DATE)
      RETURNS ( TOT_VENDA NUMERIC(15, 2),
              AVG_VENDA NUMERIC(15, 2),
              MIN_VENDA NUMERIC(15, 2),
              MAX_VENDA NUMERIC(15, 2)) AS
BEGIN
  SELECT      SUM(ITENS.QUANTIDADE * ITENS.PRECO),
              AVG(ITENS.QUANTIDADE * ITENS.PRECO),
              MIN(ITENS.QUANTIDADE * ITENS.PRECO),
              MAX(ITENS.QUANTIDADE * ITENS.PRECO)
  FROM        CLIENTES, PEDIDOS, ITENS
  WHERE       CLIENTES.ID_CLIENTE = PEDIDOS.ID_CLIENTE      AND
              PEDIDOS.ID_PEDIDO = ITENS.ID_PEDIDO          AND
              CLIENTES.ID_CLIENTE = :PARID_CLIENTE         AND
              PEDIDOS.DATA BETWEEN
                  :PARDATAINI AND :PARDATAFIM
  INTO        :TOT_VENDA,
              :AVG_VENDA,
              :MIN_VENDA,
              :MAX_VENDA;
  SUSPEND;
END

```



## Triggers

Triggers são blocos ISQL que contém códigos que implementam instruções que podem afetar dados contidos em tabelas, validos dados baseadas em certas condições, acionar procedimentos armazenados entre outros objetos.

Uma Trigger, que significa gatilho, é definida (criada) e associada a um evento (ocorrência) conseqüente aos comandos Insert, Delete ou Update.

Ao contrário de uma Store Procedure (Procedimento Armazenad), uma Trigger não é invocada nem referenciada em código Cliente nem no servidor InterBase . Conforme sugere seu nome, uma Trigger é disparada ( executada )pela manipulação da tabela na qual foi criada.

Conforme conceitualmente já abordado no tópico integridade, uma Trigger é um excelente mecanismo para se controlar as restrições, tendo em vista a possibilidade de interferir nos valores que estão sendo manipulados, pode-se ainda estabelecer uma comunicação clara com a aplicação Front-End.

Como exemplos de situações em que a adoção de Triggers se aplica, relacionamos o seguinte:

- Fazer manutenção (atualização) em outras tabelas quando ocorrerem operações de inclusão, alteração ou exclusão na tabela corrente.
- Estabelecer validação para colunas.
- Alimentar tabelas de controle de Log.
- Sinalizar aplicação Front-End de alguma ocorrência específica.

```
30) CREATE TRIGGER TRIG_TESTACREDITO_INC FOR CLIENTES
      ACTIVE BEFORE INSERT POSITION 0 AS
BEGIN
      IF (NEW.CREDITO > 20000) THEN
          BEGIN
              NEW.CREDITO = 20000;
          END
      END
END
```

```
31) CREATE TRIGGER TRIG_TESTACREDITO_ALT FOR CLIENTES
      ACTIVE BEFORE UPDATE POSITION 0 AS
BEGIN
      IF (NEW.CREDITO > 20000) THEN
          BEGIN
              NEW.CREDITO = 20000;
          END
      END
END
```

```
32) CREATE TRIGGER TRIG_ID_CLIENTE FOR CLIENTES
      ACTIVE BEFORE INSERT POSITION 0 AS
BEGIN
      NEW.ID_Cliente = GEN_ID(Gen_clientes, 1);
END
```

```

33) CREATE TRIGGER TRIG_ATUALIZASALDOPRODUTO_ALT FOR ITENS
    ACTIVE BEFORE UPDATE POSITION 0 AS
        DECLARE VARIABLE VARSALDO INTEGER;
BEGIN
    UPDATE     PRODUTOS SET
    PRODUTOS.SALDO = PRODUTOS.SALDO + OLD.QUANTIDADE
    WHERE     PRODUTOS.ID_PRODUTO = OLD.ID_PRODUTO;
    SELECT     PRODUTOS.SALDO
    FROM       PRODUTOS
    WHERE     PRODUTOS.ID_PRODUTO = NEW.ID_PRODUTO
    INTO       :VARSALDO;
    IF (:VARSALDO >= NEW.QUANTIDADE) THEN
        BEGIN
            UPDATE PRODUTOS SET
            PRODUTOS.SALDO = PRODUTOS.SALDO - NEW.QUANTIDADE
            WHERE PRODUTOS.ID_PRODUTO = NEW.ID_PRODUTO;
        END
    ELSE
        BEGIN
            EXCEPTION EXC_SALDO;
        END
END

34) CREATE TRIGGER TRIG_ATUALIZASALDOPRODUTO_EXC FOR ITENS
    ACTIVE BEFORE DELETE POSITION 0 AS
BEGIN
    UPDATE PRODUTOS SET
    PRODUTOS.SALDO = PRODUTOS.SALDO + OLD.QUANTIDADE
    WHERE PRODUTOS.ID_PRODUTO = OLD.ID_PRODUTO;
END

35) CREATE TRIGGER TRIG_ATUALIZASALDOPRODUTO_INC FOR ITENS
    ACTIVE BEFORE INSERT POSITION 0 AS
        DECLARE VARIABLE VARSALDO INTEGER;
BEGIN
    SELECT     PRODUTOS.SALDO
    FROM       PRODUTOS
    WHERE     PRODUTOS.ID_PRODUTO = NEW.ID_PRODUTO
    INTO       :VARSALDO;

    IF (:VARSALDO >= NEW.QUANTIDADE) THEN
        BEGIN
            UPDATE PRODUTOS SET
            PRODUTOS.SALDO = PRODUTOS.SALDO - NEW.QUANTIDADE
            WHERE PRODUTOS.ID_PRODUTO = NEW.ID_PRODUTO;
        END
    ELSE
        BEGIN
            EXCEPTION EXC_SALDO;
        END
END

```

## **Outros Objetos**

### **Role**

#### **Sintáxe:**

```
CREATE ROLE Nome_da_Role;
```

#### **Exemplo:**

```
Create Role Administrador;
```

### **Grant**

#### **Sintáxe:**

```
GRANT UPDATE ON Clientes TO Administrador;
```

```
GRANT Administrador TO user1, user2, user3;
```

```
GRANT SELECT, DELETE ON Clientes TO Administrador WITH GRANT OPTION;
```

### **Revoke**

#### **Sintáxe:**

```
REVOKE SELECT ON Clientes FROM Administrador;
```