

Firebird 1.0

Para muitos um nome totalmente desconhecido, o Firebird é a mais nova encarnação do banco de dados InterBase.

Um pouco de história

Quando a Borland anunciou a abertura do código fonte do InterBase, a notícia foi muito bem recebida pela comunidade Open Source que estava ganhando um banco de dados mundialmente reconhecido pela sua eficiência e simplicidade de instalação/manutenção. Após alguns meses de muita ansiedade, decepções e inseguranças, o fato finalmente aconteceu (em 25 de Julho de 2000) e todos puderam ter acesso ao código do InterBase.

Naquele instante, alguns usuários do InterBase (entre eles alguns ex-funcionários da própria Borland que participaram no seu desenvolvimento) decidiram criar uma versão bifurcada do IB que tivesse o apoio e a participação ativa da comunidade no seu desenvolvimento. Nesse momento nasceu o Firebird.

De lá para cá muita coisa tem sido feita. Pessoas que antes eram apenas usuários passaram a conhecer e se aprofundar no código do IB, dedicando grande parte do seu tempo ao estudo e limpeza do código do InterBase.

Muitos bugs foram detectados e corrigidos, incluindo aí uma grande falha de segurança (backdoor) encontrada no código do IB que permitia que qualquer usuário conhecedor do login/senha do backdoor tivesse acesso completo a qualquer banco de dados em qualquer servidor InterBase.

Para a surpresa de muitos, a mesma empresa que abriu o código do IB anunciou que continuaria vendendo uma versão certificada do InterBase que, segundo a Borland, contém algumas diferenças da versão Open Source. Isso infelizmente acabou por dificultar ainda mais o entrosamento e a troca de informações entre os dois grupos, assim criando uma divisão clara dos códigos.

Mas então, se o Interbase/Firebird é tão bom, porque ele não é tão reconhecido como o Oracle, o Microsoft SQL server e outros servidores SQL ? Aparentemente, o maior problema enfrentado pelo Interbase durante todos os anos de sua existência foi a falta de marketing e divulgação adequada por parte da Borland/Inprise/ISC nos meios especializados (revistas, etc...). No entanto, após ter seu código liberado como Open Source a situação começou a mudar rapidamente pois agora as licenças de utilização e distribuição são totalmente FREE ! Isso mesmo, custo 0, de graça !!! Isso quer dizer que você não precisará mais utilizar as famosas (e já mais do que ultrapassadas) base de dados padrão xBase ou Paradox para diminuir o custo do seu cliente. Você vai poder contar com um Banco de dados poderoso, eficiente e seguro e seu cliente não vai precisar pagar nada à mais por isso !

Novos recursos

Não só da correção de bugs se ocupou a equipe de desenvolvimento do Firebird. Ao mesmo tempo em que o código ia sendo desvendado, bugs iam sendo corrigidos e alguns novos recursos começaram a ser implementados.

Alguns desses recursos faziam parte da lista de requisições dos usuários do InterBase já há algum tempo. A seguir estão listadas algumas das novas implementações foram realizadas na versão 1.0 do Firebird.

Páginas de 16K

O tamanho máximo de uma página no InterBase/Firebird era de 8K. A implementação de páginas de 16K permite uma melhor otimização do banco de dados, assim como arquivos maiores.

Porta padrão 3050

Agora a porta de conexão padrão do Firebird é a 3050, caso não seja encontrada uma outra configuração no arquivo SERVICES.

Descriptors em UDFs

Esse recurso utiliza mecanismos internos já existentes para passar estruturas de dados da engine do banco de dados para as UDFs. O impacto mais dramático disso é que agora os valores NULLs podem ser enviados e recebidos para/das UDFs.

Possibilidade de se desabilitar índices associados a uma chave estrangeira

Até hoje a criação de uma chave estrangeira em uma tabela no Firebird fazia com que um índice fosse automaticamente criado para o campo relacionado com a chave. Isso muitas vezes gerava problemas com o otimizador de queries em situações onde os índices tinham pouca seletividade. Com a possibilidade de se desabilitar tais índices, esses problemas poderão ser contornados. Esse recurso ainda está pendente até que todas as implicações sejam revistas e, portanto, pode não estar disponível no primeiro release da versão 1.0.

UDFs e Generators monitorados

Agora o banco não permite que UDFs e generators, que estão sendo usados em triggers, stored procedures ou em campos calculados, sejam apagados.

Suporte a arquivos maiores

O IB/FB anteriormente utilizava ponteiros de 32 bits para operações com arquivos, fazendo com que o tamanho máximo de um único arquivo do banco de dados ficasse limitado a 4 Gb. O Firebird 1.0 passou a usar ponteiros de 64 bits, o que permitiu um aumento considerável no tamanho de um arquivo que compõe um banco de dados (lembre-se que um banco pode ser composto de vários arquivos, sendo que o maior tamanho de um banco IB/FB reportado até hoje é de 980Gb). No entanto, o tamanho máximo de um arquivo depende também do sistema operacional e do sistema de arquivo utilizados (ex : Fat, Fat32, NTFS, HPFS, EXT/2, etc...).

GROUP BY UDF

Agora é possível utilizar UDFs em cláusulas group by em selects mas é de responsabilidade do programador se certificar que as UDFs retornam valores válidos e

não absurdos. Isso é muito útil quando se tem que agrupar o resultado de um select baseado em uma informação que não se encontra naturalmente em um campo, como por exemplo agrupar os aniversariantes de um determinado mês baseado em um campo data (que como todos devem saber, armazena dia, mês e ano).

RECREATE PROCEDURE

Esse comando permite que se recrie uma stored procedure com o mesmo nome já existente sem necessidade de "dropar" ela antes.

RECREATE TABLE

Similar ao Recreate Procedure, mas para Tables.

SELECT [FIRST (<integer expr m>)] [SKIP (<integer expr n>)]

Um recurso muito requisitado principalmente por quem usa o Firebird em aplicações Web que permite retornar um bloco de um result set :

- FIRST faz com que somente os primeiros *m* registros sejam retornados.
- SKIP faz com que os primeiros *n* registros retornados sejam desconsiderados.

SUBSTRING(<string expr> FROM <pos> [FOR <length>])

Função interna que implementa a função SUBSTRING() do ANSI SQL. Retorna o substring composto dos caracteres localizados na posição *pos* até o final do string. Se a opção *FOR* for utilizada, o substring terá o tamanho máximo determinado por *length*. Uma função simples, mas que fazia falta.

PLANONLY

Faz com que o comando SQL seja enviado ao servidor e que o PLAN adotado para a execução seja retornado, sem que o comando seja executado. Isso é útil para saber se o plano adotado pelo otimizador realmente é o mais eficiente.

CURRENT_USER e CURRENT_ROLE

Essas duas novas variáveis contextuais foram criadas para mostrar o usuário atual e o role atual (caso esteja definido) dentro do contexto atual de uma transação.

DROP GENERATOR

Comando utilizado para apagar um generator. Anteriormente era necessário manipular diretamente as tabelas do sistema no banco de dados para remover um generator.

Conversão automática de strings para BLOBs em comandos de insert/update

Uma característica simples mais muito efetiva e que realmente fazia falta. Antes era impossível atribuir um texto em um campo blob utilizando comandos SQL sem o uso de uma UDF para converter o texto (string) em Blob.

Forced Writes ON como padrão no Win32

Nunca foi documentado, mas a opção de Forced Writes foi desligada como padrão para Win32 no lançamento do IB 6.0. Isso ocasionou muita corrupção de dados especialmente quando o IB estava rodando em Win 9x sem no-breaks.

Novas funções da FUDFLIB

Algumas UDFs muito úteis foram adicionadas à biblioteca padrão que acompanha o Firebird (FUDFLib), entre elas: NVL, NULLIF, SDOW, DOW, RIGHT, StringToBlob etc.

Compatibilidade

No momento o Firebird é totalmente compatível com bancos criados no InterBase 6. Isso quer dizer que se você tem um banco que foi criado ou está sendo usado com um servidor InterBase 6.0 esse mesmo banco pode ser utilizado com um servidor Firebird 1.0. Isso se deve ao fato da ODS (On Disk Structure) usada pelo Firebird ser a mesma da utilizada pelo InterBase 6.0.

No entanto, você deve tomar cuidado com o contrário, ou seja, se você criou um banco de dados no Firebird 1.0 e quer utilizá-lo em um servidor IB, isso só vai ser possível se seu banco não estiver utilizando nenhum dos novos recursos incorporados no Firebird. Por exemplo, se seu banco está configurado com páginas de 16K, o InterBase não poderá acessá-lo pois o limite do tamanho de página no InterBase é de 8K.

É aconselhável, nos casos de mudança de servidor, que um backup seja feito do banco de dados no servidor originalmente utilizado e que um restore seja feito no novo servidor.

Porque usar o Firebird ao invés do InterBase ?

A seguir estão alguns fatos à serem considerados que ajudarão na resposta dessa pergunta. A decisão é sua.

Ao contrário da Borland, que se mantém em silêncio no que diz respeito a novas implementações no InterBase, bem como se novos recursos serão disponibilizados na versão Open Source do banco ou se somente estarão disponíveis na versão certificada (paga), o projeto Firebird é totalmente aberto à comunidade. Qualquer um pode se informar da situação atual do código, saber o que foi alterado e o que está por vir. A comunidade está aberta para a participação de todos que quiserem ajudar no desenvolvimento do código, correção de bugs, documentação, desenvolvimento de ferramentas, etc.

Não existe, no entanto, uma comunidade aberta trabalhando no código atual do InterBase da Borland, portanto mesmo que você queira trabalhar nele, não encontrará muita gente com quem conversar, trocar idéias e experiências. A Borland mantém uma equipe de desenvolvimento interna trabalhando na versão certificada do banco, mas,

como dito anteriormente, não se sabe se as novas implementações estarão disponíveis na versão Open Source do InterBase.

No Firebird não existe pressão para upgrades e as correções de bugs são disponibilizadas quase que instantaneamente. Além disso, hoje o Firebird suporta dez sistemas operacionais (Win32, Linux, Solaris (Sparc e X86), Mac OS X, FreeBSD, NetBSD, AIX, HP-UX e SCO) enquanto que o InterBase suporta apenas três (Win32, Linux e Solaris).

Onde eu posso conseguir o Firebird 1.0 ou obter mais informações?

O Firebird está disponível para download através dos sites da IBPhoenix (www.ibphoenix.com) e do Firebird (www.firebirdsql.org), tanto os binários como o código source. Os binários vêm com instaladores para a maioria das plataformas suportadas.

PLAN	O otimizador do Firebird automaticamente gera um PLANo de acesso aos dados visando a máxima eficiência da operação (como por exemplo um select). O plano define quais índices serão ou não utilizados na recuperação dos dados, joins entre tabelas, bem como a ordem que isso vai acontecer.
UDF	User Defined Function – Um dos grandes recursos do Firebird é permitir que o desenvolvedor crie qualquer tipo de função e use-a dentro do banco de dados. Para isso basta você utilizar uma linguagem que gere DLLs e seguir as regras para a definição dessas UDFs.
Forced Writes	É o modo como o Firebird grava as informações no disco. Se Forced Writes estiver ON (ligado), então toda informação gravada no banco será imediatamente gravada no HD, se estiver OFF (desligado), a informação permanecerá um certo tempo no cache de disco antes de ser definitivamente gravada. A opção OFF, por motivos óbvios, somente deve ser utilizada em servidores estáveis (de preferência rodando Linux) e que possuam NoBreaks.
Páginas	Um banco de dados Firebird é composto de diversas páginas de determinado tamanho. Essas páginas armazenam informações sobre o gerenciamento/uso do espaço dentro do banco, ponteiros, etc...
ODS	On Disc Structure – é a versão da estrutura do banco de dados utilizada pelo Firebird. Geralmente quando uma nova versão do servidor é lançada ela utiliza uma estrutura diferente da anterior, ou seja, tem um número de ODS diferente. Por isso deve-se fazer um backup antes de dar upgrade no servidor, e restaurá-lo depois do upgrade pois ele será restaurado com a nova ODS.
Services	O arquivo services é um arquivo de configuração do sistema operacional que diz que serviço atende à determinada porta do TCP/IP. O Firebird tem por padrão utilizar a porta 3050 para comunicação.
Generator	São geradores de números sequenciais internos do banco de dados. São geralmente utilizados para gerar um valor inteiro para ser armazenado em uma chave primária em alguma tabela. Eles não ficam dentro do contexto transacional, por isso garantem que os números nunca vão se repetir, mas podem deixar

"buracos" dentro da sequência caso uma transação seja desconsiderada (rolled back).

Para muitos o nome Interbase/Firebird é totalmente desconhecido, para outros é sinônimo de eficiência e fácil manutenção (fácil ? Bem, na verdade a manutenção de um banco de dados IB/FB é quase 0).

O Interbase não é um produto novo. Ele está no mercado há mais de 12 anos, e nesse tempo adquiriu respeito e admiração de muitos programadores, desenvolvedores e clientes (entre eles podemos citar a NASA, o exército americano, etc...). O Firebird é a continuação do Interbase Open Source e vem sendo desenvolvido por uma comunidade de programadores espalhados por todo o mundo.

Principais características

À seguir algumas características que fazem com que o Interbase/Firebird se equipare e também se sobressaia em muitos aspectos aos seus concorrentes.

Sistema "Multi-Geracional"

Uma das maiores vantagens do Interbase/Firebird sobre seus concorrentes é o uso de um sistema otimista de "concorrência" no acesso ao banco de dados. Seguindo uma linha totalmente diferente da maioria dos bancos de dados, onde você tem bloqueio de páginas inteiras (PAGE LOCKS), o Interbase/Firebird "guarda" várias versões dos registros mantendo assim uma visão consistente dos dados durante uma transação, independente de alguma informação ter sido alterada após a transação ter sido iniciada.

Usando um exemplo prático, geralmente você tem situações onde várias pessoas estão utilizando um mesmo banco de dados ao mesmo tempo. Alguns rodando relatórios, outros inserindo ou alterando registros. Imagine que entre o preview de um relatório na tela e a sua impressão definitiva, um outro usuário altere ou insira uma informação no banco de dados que influa no resultado desse relatório. Nos BD convencionais, com certeza você obterá um relatório impresso diferente do que você viu na tela (preview) pois os dados alterados interferiram no resultado final do relatório. No Interbase/Firebird, através dos diversos tipos de isolamento de transação, você pode ter uma mesma "imagem" dos dados inalterada pelo tempo que quiser, sem impedir que outros usuários continuem acessando ou alterando as informações.

Quando um usuário altera uma informação de um campo em um registro, o Interbase/Firebird cria um novo registro com os campos que tiveram seus dados alterados. Esse registro contém um "timestamp" que permite ao banco de dados saber qual é a informação mais atualizada. O registro contendo os dados anteriores não é descartado enquanto houver uma transação ativa "enchergando-os". Após o registro ser liberado, ele é automaticamente "marcado como lixo", removido ou tem seu espaço re-utilizado pelo banco de dados.

O mecanismo de timestamp também é utilizado para saber se mais de uma pessoa tentou alterar os mesmos dados num determinado período, o que gera automaticamente uma exceção (evento de erro) no BD conhecido como deadlock, garantindo assim a consistência das informações.

Configuração e Manutenção

Uma outra grande vantagem do Interbase/Firebird é quanto à definição das características físicas do banco de dados. Em muitos servidores SQL, o analista ou DBA deve previamente estimar, definir e preparar um espaço do disco rígido para ser usado pelo BD. Os BDs criados no Interbase/Firebird são arquivos comuns do sistema operacional (geralmente com a extensão GDB), que crescem e diminuem conforme a necessidade, sem que seja necessária a intervenção do DBA ! A manutenção de um banco de dados Interbase/Firebird é praticamente 0, assim como sua configuração.

O Interbase/Firebird automaticamente mantém sua base de dados limpa e consistente através de rotinas automáticas de manutenção (SWEEP) executadas geralmente quando o BD está em "idle" (estado de espera). Cada banco de dados Interbase/Firebird consiste de apenas 1 arquivo... todas as informações, tabelas, índices, etc... ficam armazenados dentro desse arquivo, facilitando muito as operações de administração e backup dos mesmos. Caso o arquivo ultrapasse o tamanho máximo suportado pelo sistema operacional você pode dividir o banco em múltiplos arquivos, sendo que o gerenciamento desses arquivos é totalmente transparente para o usuário.

Suporte à domínios

O Interbase/Firebird suporta o uso de domínios na definição de campos. Na verdade, todo campo criado em uma tabela no Interbase/Firebird possui um domínio próprio, criado automaticamente pelo sistema ou definido previamente pelo analista. Com essa tecnologia, fica muito fácil fazer alterações em cascata em campos do mesmo tipo.

Um exemplo : Imagine que no seu banco de dados existam 20 campos do tipo Numérico, definidos com 2 casas decimais. Você pode definir um domínio chamado DINHEIRO, especificando o tipo de informação (Numérica) e o número de casas decimais utilizadas pelo mesmo e atribuir esse domínios aos 20 campos do seu BD. Se no futuro você tiver a necessidade de alterar as casas decimais de 2 para 4, basta você alterar o domínio DINHEIRO e todas as colunas referenciadas por esse domínio serão automaticamente alteradas.

Aqui entra uma outra característica do IB. Quando você altera um domínio, os dados armazenados utilizando esse domínio não são imediatamente convertidos para o novo formato, o que ocasionaria um certo overhead no servidor. Por exemplo : Imagine que você tenha um domínio definindo o tipo VARCHAR(10). Em um dado momento, você necessita alterar o tamanho do campo, de 10 para 20. Você então altera o domínio para VARCHAR(20), mas os dados que já estão gravados só serão convertidos ao novo formato quando forem utilizados de alguma maneira. Observe no entanto, que isso fica transparente ao usuário... você enxergará todos os dados como VARCHAR(20) !!

Multi-plataforma

O Firebird atualmente roda nos sistemas operacionais Windows, Linux (i386), Solaris (Sparc), HP-UX (i386), MacOS X, FreeBSD, Solaris (i386) e AIX. O Interbase foi o primeiro banco de dados profissional a ter uma versão disponível para o Linux.

Outras características :

- Otimização de Queries. O próprio servidor otimiza uma Query de maneira a buscar os resultados da maneira mais eficiente e rápida possível. Também é possível definir um plano de otimização (PLAN) manualmente.
- Suporte à campos BLOB, que podem conter qualquer tipo de dado, desde textos até objetos gráficos ou binários.
- Suporte à diversos protocolos : Local, TCP/IP, NetBeui, IPX/SPX (Novell).
- Funções definidas pelo usuário : Esse é um recurso muito poderoso do Interbase/Firebird . Através dele, você pode utilizar dentro de seus banco de dados, funções criadas por você mesmo usando qualquer linguagem que gere DLLs.
- Suporte à múltiplas transações (multi-transacional)
- Acesso nativo através da API (sem necessidade da BDE) através de componentes como o Interbase Objects, FreeIB+, IBX (que acompanha o Delphi 5,6 e 7), ZeosLib, DBExpress, e alguns outros.
- Campos Inteiros de 64 bits : O Interbase/Firebird 6.0 (dialect 3) oferece suporte à campos numéricos inteiros de 64 bits. Ele utiliza uma técnica em que números de ponto flutuante podem ser armazenados no banco de dados como inteiros, removendo assim o risco de erros de arredondamento. Toda a conversão entre inteiros e floats é transparente ao usuário.
- Replicação : O Interbase/Firebird 6.0 oferece suporte à replicação de dados através de componentes de terceiros (que infelizmente, até o momento não são free, mas que tem um custo bem acessível). Visite a página de downloads da Interbase-BR para ver uma lista dos replicadores disponíveis.

Resumo

Como você pode ver, o Interbase/Firebird é muito mais do que um simples banco de dados. Ele oferece uma solução completa para pequenas e médias empresas e possui recursos que competem de igual para igual com outros servidores SQL mais conhecidos, com a vantagem de ser Open Source e grátis !

Links úteis

Interbase-BR- <http://www.interbase-br.com>

IBPhoenix - <http://www.ibphoenix.com>

Firebird - <http://www.firebirdsql.com>

Interbase - <http://www.borland.com/interbase>

Claudio Valderrama - <http://www.cvalde.com>

Lista de discussão em português - <http://br.groups.yahoo.com/group/interbase-br>

Para utilização do FIREBIRD é preciso instalar o Banco de dados e o console, que são arquivos distintos.

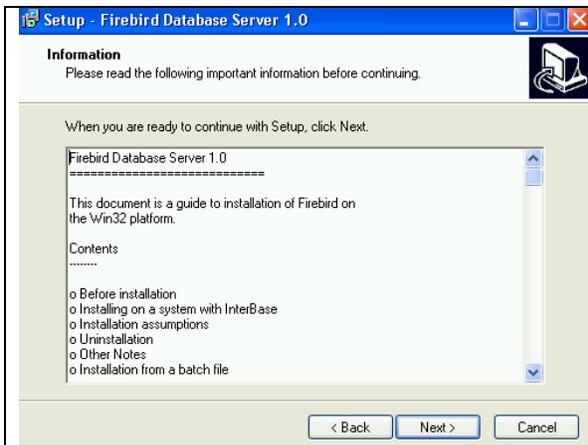
Instalação do FIREBIRD para Windows

1. Realizar o download do arquivo “Firebird-1.0.2.908-Win32.exe” no site <http://www.firebirdsql.com>;
2. clique no arquivo e será apresentada a seguinte janela:

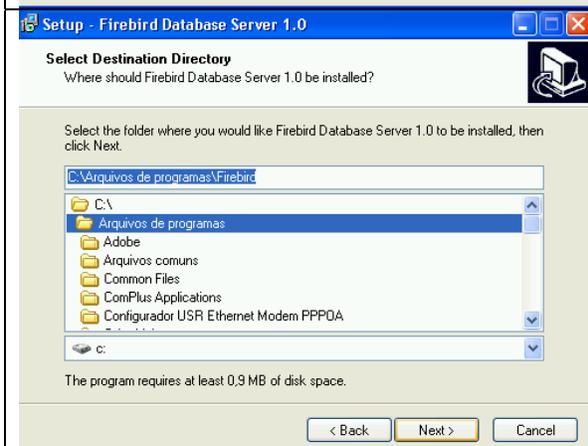


3. será clique no botão `Sim` e e será apresentada a seguinte janela:

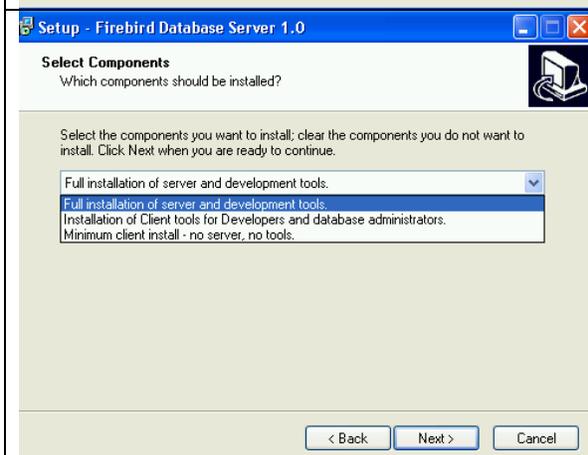
	<p>- clique em “Next”</p>
	<p>- depois de aceitar os termos de licença, clique em “next”</p>



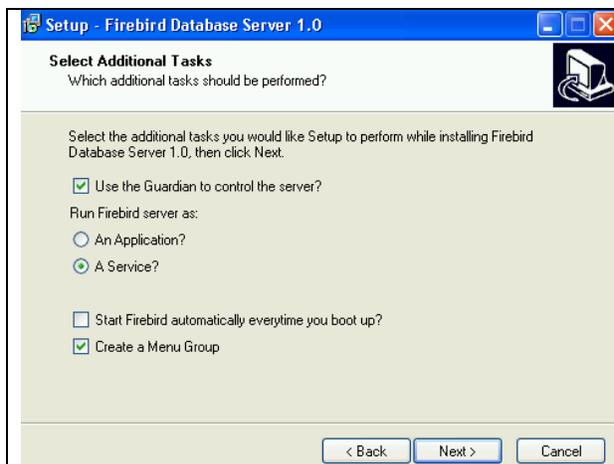
- clique em "Next"



- escolha o local para instalação do FIREBIRD e clique em "Next"



-escolha a opção do seu interesse:
- instalação full - será instalado o servidor e as ferramentas de desenvolvimento em sua máquina
-instalação client com ferramentas de desenvolvimento - será instalado o client e as ferramentas para acesso há um servidor instalado em outra maquina
- instalação client - será instalado apenas o necessário para acesso há um servidor localizado em outra máquina.
-clique em "next"



-o uso do Guardian deve ser desabilitado para Win2000/XP.

-pode definir se o servidor ira rodar como um serviço ou como uma aplicação

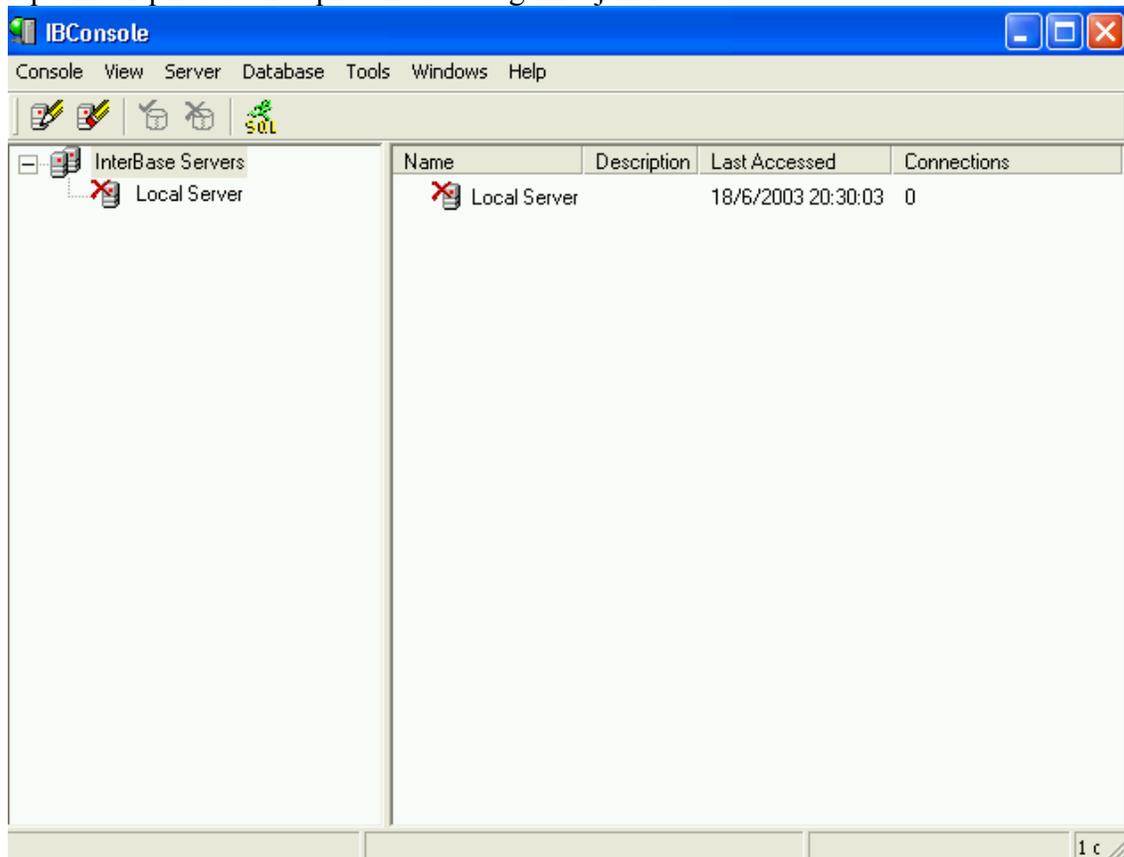


-o 1 programa a ser utilizado é o FIREBIRD MANAGER (arquivo fbmgr.exe)

-nele é possível iniciar ou parar o database e definir o seu inicio automático ou manual.

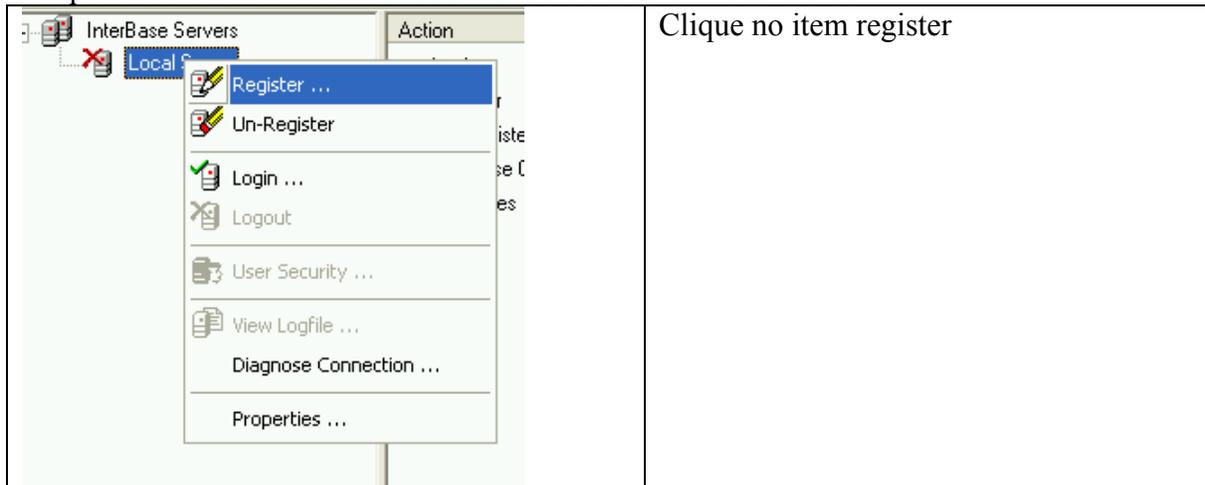
Para manusear o servidor é necessária a utilização de um console. Existem diversos tipos de consoles gratuitos a disposição para download no site www.firebase.com.br. Nesta apostila será descrito o uso do console IBConsole, que é um dos mais utilizados e que apresenta menor quantidade de bugs, em um servidor local.
Arquivo: IBConsole.exe

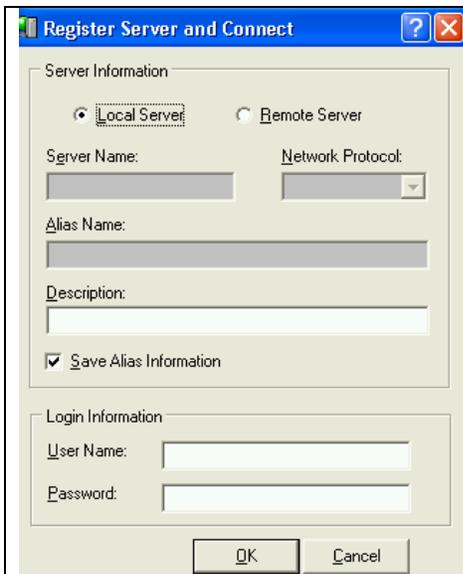
Clique no arquivo e será apresentada a seguinte janela:



O primeiro passo deve ser registrar o servidor local

Clique o botão direito do mouse com o cursor sobre "Local Server"

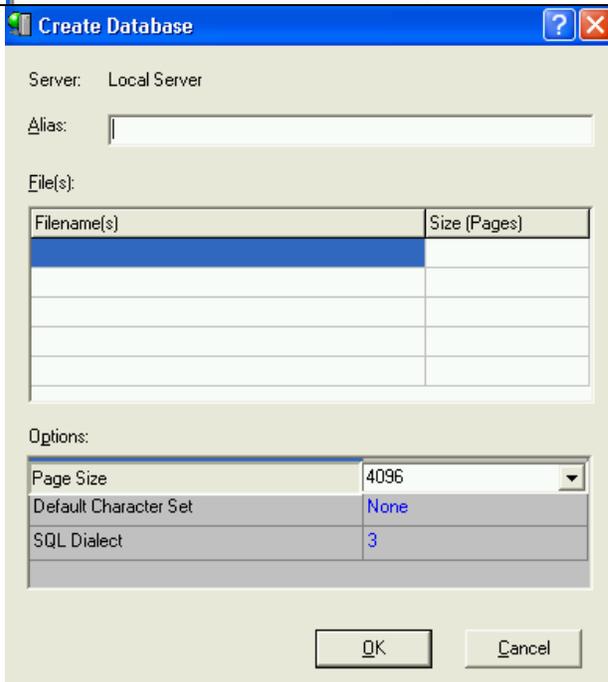




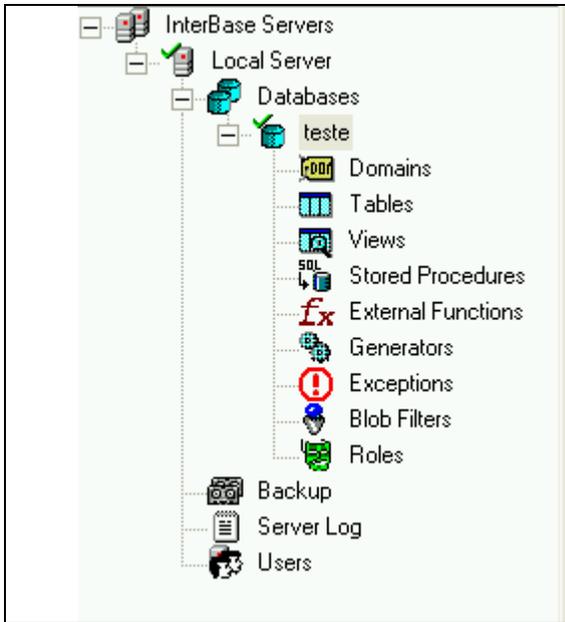
Preencha os campos User Name e Password.
 -os padrões são SYSDBA, para User Name, e masterkey(minúsculo), para password.
 - clique em OK



-pronto. O servidor já está operacional no console.
 -o próximo passo é criar ou utilizar um database já existente.



Para criar um database selecione a opção Create Database.
 - no campo Alias digite o Apelido do Database
 - no campo Filename digite o caminho e o nome do arquivo do database com a extensão GDB.
 - Ex: "c:\databases\nomedodatabse.gdb"
 -deixe o campo Size em branco.
 -não altere os valores dos demais campos
 -clique em OK

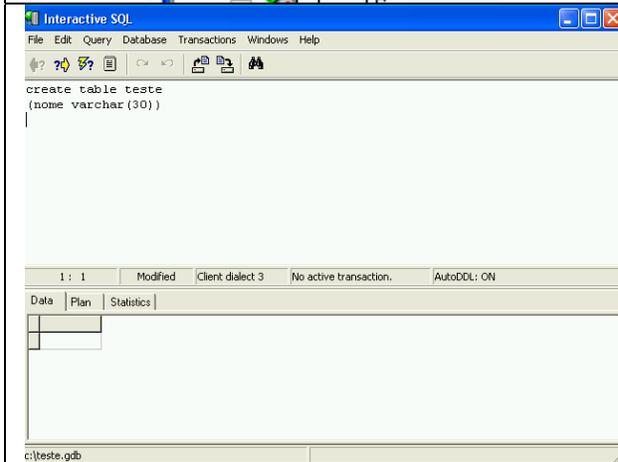


Pronto! O database está criado.

- todos os objetos do database devem ser criados através de código SQL
- Para a utilização de códigos SQL pode ser usado o SQL interativo incluído no console.

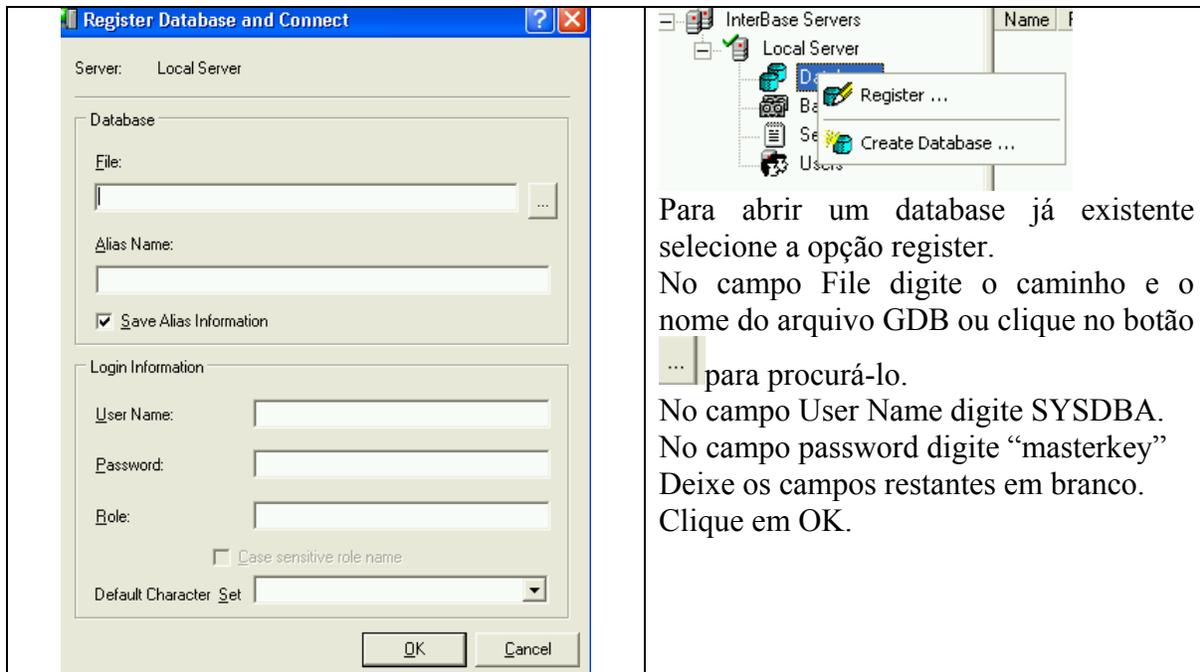


Clique no botão SQL e será aberto o console para SQL



Digite o código SQL e clique no botão  para executá-lo

Para rever os códigos já executados utilize os botões:  



Para abrir um database já existente selecione a opção register. No campo File digite o caminho e o nome do arquivo GDB ou clique no botão ... para procurá-lo.

No campo User Name digite SYSDBA. No campo password digite “masterkey” Deixe os campos restantes em branco. Clique em OK.

Guia rápido de configuração de clientes Firebird (Windows) em rede

A configuração de um micro terminal de rede para ficar habilitado à acessar um servidor Firebird é muito simples e rápida.

A primeira coisa à se fazer é instalar o cliente do banco de dados (Interbase/Firebird) na máquina terminal. Para isso utiliza-se o próprio instalador do servidor de banco de dados, escolhendo a opção de instalar somente o cliente ou então o cliente + ferramentas.

A instalação mínima do cliente copia para o terminal o arquivo responsável pela comunicação com o servidor (antigo gds32.dll, atualmente fbclient.dll), o arquivo contendo as mensagens que podem ser retornadas pelo servidor (antigo interbase.msg, atualmente firebird.msg), a biblioteca do Microsoft C necessária (msvcrt.dll), além de fazer algumas alterações no registro do Windows e dependendo do caso no arquivo SERVICES do Windows.

A instalação do cliente + Ferramentas faz tudo que a instalação mínima faz e também instala os utilitários do Firebird (gbak.exe, gsec.exe, etc...), documentação, etc...

Após a instalação do cliente, o terminal já está preparado para acessar o servidor Interbase/Firebird (não é preciso reiniciar a máquina).

O próximo passo é configurar o seu aplicativo para acessar o banco de dados na rede. Essa configuração diz respeito basicamente ao path de conexão com o banco. O formato do path de conexão depende do protocolo de rede utilizado, sendo que o **recomendado é o TCP/IP**.

Para rede rodando em TCP/IP

No caso da rede TCP/IP o path deve ser montado no seguinte formato :

*Quando o servidor Interbase/Firebird estiver rodando em uma máquina **Windows** :*

Path =
ip_ou_nome_do_servidor:path_local_no_servidor_para_o_arquivo_do_banco_de_dados

Sendo assim, se temos um servidor chamado "SERVIDOR" (com o IP 192.168.1.1) e o nosso banco de dados está gravado no arquivo c:\banco\dados.gdb, o path de conexão ficaria :

servidor:c:\banco\dados.gdb ou 192.168.1.1:c:\banco\dados.gdb

Em determinadas situações, quando se usa o nome do servidor no path de conexão deve-se alterar o arquivo HOSTS do Windows no terminal adicionando uma linha com o IP e o nome da máquina para que o windows seja capaz de identifica-lo na rede, por exemplo :

192.168.1.1 SERVIDOR

*Quando o servidor Interbase/Firebird estiver rodando em uma máquina **Linux**, o path segue a mesma estrutura, lembrando que em Linux não existe letra do drive, portanto no exemplo anterior teríamos algo do tipo :*

servidor:/banco/dados.gdb ou 192.168.1.1:/banco/dados.gdb

Assim como no exemplo anterior, em determinadas situações é necessário configurar o arquivo HOSTS.

Redes com NetBEUI

No caso do protocolo utilizado ser o NetBEUI, o path de conexão segue a seguinte estrutura :

Path =
\\nome_do_servidor\path_local_no_servidor_para_o_arquivo_do_banco_de_dados

No exemplo anterior teríamos

```
\\servidor\c:\banco\dados.gdb
```

Rede Novell com IPX/SPX

No caso do protocolo ser o IPX/SPX (Novell - descontinuado no Firebird 1.5), a estrutura é :

Path = servidor_netware@volume:/path_para_o_arquivo_do_banco_de_dados

No exemplo anterior teríamos

```
servidor@vol1:/banco/dados.gdb
```

Note que o suporte à IPX/SPX está descontinuado no Firebird portanto se você tem uma rede Novell é recomendável que você instale o servidor Firebird em um cliente da rede rodando Windows e acesse esse servidor através do TCP/IP.