

IBO

**Traga a luz em seus projetos em
Delphi e Firebird** 

**Entrevista com o time de
desenvolvedores do PostgreSQL** 

**Utilização básica do
INSERT/UPDATE/DELETE**



Editorial

Chegamos a terceira edição da DB FreeMagazine. Para um projeto desse tipo, cada nova edição é uma conquista e uma vitória para seus idealizadores!

Grande parte do sucesso depende de vocês, leitores! Continuem baixando e divulgando a revista na sua comunidade, entre seus amigos, professores, listas de discussão, etc. Além disso, você pode **colaborar enviando artigos** para serem publicados. Para isso, entre em contato pelo email cantu@dbfreemagazine.com.br e nos diga sobre o que gostaria de estar escrevendo! **Sua participação é extremamente importante** para a continuidade do projeto!

Vale lembrar que apesar da revista ter um carinho todo especial para os bancos open source, ela não se limita aos mesmo. Em breve teremos artigos de outros SGBDs, sejam eles open source ou não.

No dia 16 de Julho, estaremos presentes no **2º Firebird Developers Day**, em Piracicaba - SP, fazendo a cobertura completa do evento para a próxima edição da revista! Se você usa ou pretende usar o Firebird, esse evento é para você! Esse ano ele está ainda melhor, pois terá a presença de Jim Starkey e Ann Harrison, ícones da comunidade internacional!

Espero encontrar muitos de vocês lá!

Up the Irons!

Carlos H. Cantu

Editor DB FreeMagazine

Informações

DB FreeMagazine nº 003 - Ano I

Junho/2005

Contato geral:

webmaster@dbfreemagazine.com.br

Equipe editorial

Carlos H. Cantu

(cantu@dbfreemagazine.com.br)

Luiz Paulo de Oliveira Santos

(lpaulo@dbfreemagazine.com.br)

Contribuíram nessa edição

Luiz Paulo de Oliveira Santos

Carlos H. Cantu

Team PostgreSQL

É proibida a reprodução de qualquer parte do conteúdo dessa publicação sem autorização prévia por escrito.

ANUNCIE NA DB FreeMagazine

Valorize seu produto ou serviço!

anuncios@dbfreemagazine.com.br

Dica para melhor visualização:

Utilize a resolução 1024x768 pixels e configure o *Acrobat Reader* para Zoom de 100%. Feche todas as abas laterais e esconda as barras de ferramentas, liberando o máximo de área útil na tela, ou simplesmente rode a revista em modo *fullscreen*.

Saindo do Forno...

Oracle 10g release 2

Enquanto a Oracle prepara seu novo release, você já pode ir dando uma olhada nas novidades que o acompanharão, entre elas: DataBase Upgrade Assistant mais inteligente, Oracle Automatic Storage Management melhorado, Oracle Secure Backup, Asynchronous commits e muitos mais! A primeira versão sairá para Linux.

Fonte: http://www.oracle.com/technology/pub/articles/betaweek_r2.html

Nova versão do MS SQLServer

A previsão de lançamento do SQL Server 2005 é Novembro/2005, cerca de 2 anos de atraso se comparado com a data inicial estimada.

Recentemente a Microsoft disponibilizou a primeira versão de testes pública, chamada de Community Technology Preview. Segundo a MS, oito clientes incluindo a Xerox já estão utilizando o novo banco de dados.

Fonte: <http://www.computerworld.com/databasetopics/data/software/story/0,10801,102431,00.html>



Seja um AUTOR da Editora Ciência Moderna!

Se você possui algum manuscrito, original ou livro pronto, submeta à nossa apreciação enviando um resumo do seu trabalho para o e-mail: lcm@lcm.com.br. Teremos o maior prazer em avaliá-lo e, se aprovado, publicá-lo em forma de livro.

InterBase SP1

A Borland lançou recentemente o Service Pack 1 para o InterBase 7.5. O download é gratuito para os usuários registrados.

Fonte: http://www.borland.com/downloads/registered/download_interbase.html

Slony-I 1.1.0

Acaba de ser lançada a versão 1.1.0 do Slony-I, replicador para PostgreSQL. Entre as novidades está uma documentação expandida e melhorada, correções de bugs, etc.

Fonte: <http://www.postgresql.org/about/news.350>

Conferência internacional de Firebird

Inscrições abertas para a conferência internacional de Firebird, que será realizada em Praga, na República Checa, de 13 a 15 de Novembro, no Hotel Olsanka.

Fonte: http://www.ibphoenix.com/main.nfs?a=ibphoenix&s=1119890371:344250&page=fb_conf_attend

Desconto para os leitores da DB FreeMagazine no 2º FDD

Você que é usuário do Firebird e leitor da DB FreeMagazine, e ainda não fez sua inscrição para o **2º Firebird Developers Day**, poderá fazê-la pagando apenas R\$ 33,00 até o dia 3 de Julho. O evento será realizado no dia 16 de Julho, e para ter direito ao desconto você deve se inscrever clicando aqui.

Errata da Edição #002

No artigo "Desvendando o Select", na página 18, o select

```
SELECT NOME, COUNT(AC.QTDE)
FROM PESSOAS PE, ACESSOS AC
WHERE PE.UNAME = AC.PESSOA
GROUP BY (AC.PESSOA)
```

deveria ser

```
SELECT NOME, COUNT(AC.QTDE)
FROM PESSOAS PE, ACESSOS AC
WHERE PE.UNAME = AC.PESSOA
GROUP BY NOME
```

Obrigado a Misako Kadowaki por nos alertar sobre o problema.

IBO (InterBase Objects) - parte 1

Carlos Henrique Cantu

O IBO é, sem dúvida, o pacote mais completo de componentes de acesso ao Firebird/InterBase para Delphi/CBuilder! Ele oferece dezenas de componentes para os mais variados fins, desde a conexão com o banco de dados, execução de queries, execução de scripts, manipulação de cursores, até componentes mais sofisticados para elaboração de pesquisa textual e replicação simples.

Apresentarei, em uma série de três artigos, os componentes mais importantes do IBO, mostrando alguns exemplos de utilização e configuração das suas propriedades. No primeiro artigo da série, trataremos exclusivamente da instalação da versão de avaliação do IBO, e faremos uma apresentação dos principais componentes disponíveis.

A imensa variedade de componentes e propriedades que o pacote oferece, contribui para que muitos usuários que instalam o IBO pela primeira vez se assustem, e até mesmo desistam de utilizá-lo, pois ficam com a impressão que os componentes são muito complexos, mas isso não é verdade!

Para entendermos o porquê do grande número de componentes no pacote, precisamos voltar um pouco no tempo e conhecer a história do IBO.

“Há muito tempo, em uma galáxia(ops!)... aqui mesmo no planeta terra...”

O IBO foi criado por Jason Wharton – há muitos anos atrás, antes do Delphi 3 ser lançado – devido à sua frustração em relação aos recursos da maioria dos componentes de acesso ao BD. Naquela época, a BDE ainda reinava quase que absoluta entre os usuários das linguagens de programação da Borland. O problema é que para oferecer facilidade e simplicidade aos programadores, a BDE deixava de aproveitar diversos recursos específicos do InterBase (até então o Firebird ainda não existia). Como exemplo, podemos citar o fato de que a BDE não permitia ter mais de uma transação aberta em uma mesma conexão com o BD. Além disso, os controles dataware (TDBEdit, TDBLookupCombo, etc.) disponíveis no Delphi não eram otimizados para bancos de dados Cliente/Servidor, pois eram (e continuam sendo) compatíveis com qualquer tipo de SGBD, seja ele relacional ou não.

Sendo assim, Jason resolveu criar seus próprios componentes de acesso ao InterBase, de forma a aproveitar todos os recursos desse SGBD. O problema é que, naquela época, a classe TDataset continha código específico para a BDE, o que impossibilitava a criação de classes derivadas e independentes dela. Isso obrigou o Jason a derivar seus

componentes de classes mais básicas, como a TComponent. A maior implicação disso é que os componentes de acesso do IBO não são compatíveis com os componentes dataware do Delphi, obrigando-o a criar versões personalizadas dos principais componentes de edição de dados, ou seja, o IBO tem seu próprio DBEdit (TIB_Edit), DBGrid (TIB_Grid), etc., o que explica a enorme quantidade de componentes instalados pelo pacote. Apesar do grande trabalho em recriar todos esses componentes, houve a vantagem de codificá-los de forma otimizada para o InterBase, ou seja, um TIB_Grid é mais eficiente do que um DBGrid quando utilizado com o InterBase/Firebird!

Você deve estar se perguntando: “Mas então eu não posso usar o IBO com o Infopower, RxLib, ReportBuilder, etc?”. A resposta é: **pode!** Quando a Borland isolou o código da BDE da classe TDataset (no Delphi 3), Jason criou versões de seus componentes de acesso compatíveis com essa classe. No IBO, esses componentes são identificados pelo nome da classe começando por TIBO. Ou seja, os componentes cujos nomes começam com IB_ são puros e não descendem da TDataset, os componentes que começam com



IBO são descendentes do TDataSet e portanto compatíveis com qualquer componente dataware de terceiros ou do próprio Delphi!

Licenciamento

Ops! É isso mesmo que você está pensando! O IBO não é um componente gratuito! No entanto, o

autor disponibiliza o pacote sob a licença Trustware, baseada na confiança. Através dessa licença, os indivíduos que querem utilizar o IBO em produção, mas ainda não obtiveram recursos financeiros suficientes através dos softwares que fazem uso do IBO, recebem uma licença temporária, para que possam desenvolver e distribuir as aplicações sem limitação, devendo fazer o registro assim que tiverem condições para isso.

Além disso, através de uma parceria exclusiva entre a FireBase e o Jason, os brasileiros podem adquirir o IBO por um **preço bastante reduzido!** Basta fazer a compra através do link disponível no endereço **www.firebase.com.br/fb/IBO.html**

Características marcantes

O IBO possui inúmeras características que o diferenciam da maioria dos outros componentes de acesso para InterBase/Firebird. A mais notável é a inteligência embutida nos componentes no que se refere ao cache de informações em memória e controle implícito de transações.

Ele pode controlar automaticamente a abertura e encerramento das transações, sem no entanto impedir que o desenvolvedor faça o controle explícito quando necessário. Quando transações implícitas são utilizadas, o

IBO sempre que possível toma todo o cuidado necessário para impedir que elas fiquem abertas por muito tempo, e impeçam o avanço da OAT e conseqüentemente prejudiquem o processo de *garbage collection*.

Os componentes de recuperação de dados (dataset) são inteligentes, podendo recuperar as informações (fetch) conforme a necessidade, evitando portanto um *FetchAll* no dataset. Há também a possibilidade de configurar os componentes para recuperar as informações nos períodos em que a aplicação está ociosa (em idle).

Devido à bufferização de dados inteligente, com o IBO você pode até mesmo encerrar (commit/rollback) uma transação sem que no entanto os datasets associados a ela sejam fechados, como acontece em outros componentes (IBX, etc).

Para aqueles que desejam utilizar os componentes em conexões de baixa velocidade (internet), o IBO oferece o cache local da estrutura do banco de dados, evitando que a aplicação cliente fique constantemente recuperando essas informações do servidor.

A utilização de generators

para geração de números seqüenciais é bastante simplificada através da propriedade *GeneratorLinks* dos datasets, que fazem o incremento e recuperação de um generator, associando o valor obtido diretamente ao campo especificado.

Existem também componentes exclusivos para criação de aplicações de busca textual e de replicação simples.

O autor garante **total compatibilidade** do IBO tanto com o Firebird quanto com o InterBase.

Instalando o IBO

Até pouco tempo atrás, a aventura de utilizar o IBO já começava com a sua instalação, que era totalmente manual, e considerada complicada e trabalhosa pelos mais leigos. Felizmente hoje temos um instalador disponível, que faz quase todo o trabalho de



ClubeDelphi
Complementar

Firebird
ESSENCIAL

CARLOS H. CANTU

Firebird Essencial

Primeiro livro brasileiro que trata especificamente dos recursos do SGBD Firebird (versões 1.0 e 1.5). O autor reuniu no livro todo o material produzido por ele para as revistas ClubeDelphi e SQLMagazine. Os artigos foram revisados, atualizados e muitos deles complementados, de forma a proporcionar ao leitor uma fonte de informação rica, atualizada e confiável. Um capítulo inédito sobre a criação de UDFs foi escrito exclusivamente para o livro.

Você aprenderá a instalar o SGBD, criar procedures, catálogos em CDROM, criar backups, gerenciar usuários, utilizar campos BLOB de forma adequada, identificar os tipos de dados disponíveis no Firebird, e muito mais!

Verifique o sumário do livro em
www.firebase.com.br/fb/livro/fbessencial

www.firebase.com.br

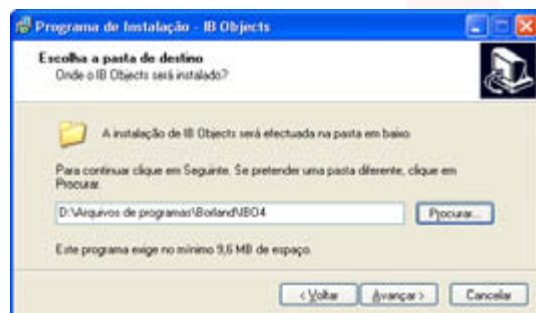
instalação automaticamente, poupando-nos trabalho.

Para baixar a versão de avaliação do IBO, vá até o site www.ibobjects.com e clique em "Download & evaluate IBO" e baixe o arquivo **IBO4_5.zip**, com cerca de 1.5MB.

O IBO é compatível com o Delphi 3,4,5,6,7 e 2005 (apenas para Win32) e com o CBuilder 4,5 e 6. Há também uma versão beta para o "quase falecido" Kylix.

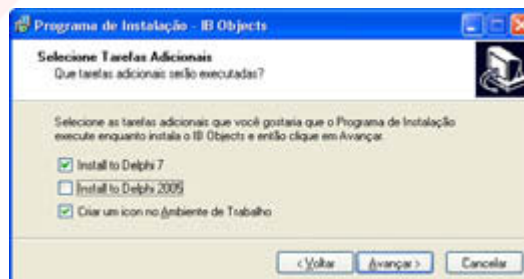
Ao descompactar o arquivo baixado, um novo arquivo com o nome IBO4_5.exe será criado no seu HD. Feche todas as IDEs do Delphi que possam estar abertas e rode o instalador.

Na primeira tela apresentada, escolha o idioma que você quer utilizar durante o processo de instalação. No artigo usaremos o "português". A próxima tela é a de boas vindas, clique em "Avançar" para exibir a tela de seleção do diretório para a instalação do IBO, conforme figura abaixo:

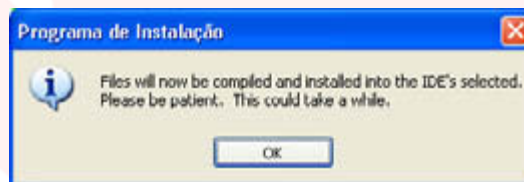


Após selecionar o diretório de instalação, clique em avançar e na nova tela apresentada mantenha selecionada a opção "**IBO Full installation**", clicando em avançar novamente. Agora você terá a oportunidade de indicar em

quais IDEs do Delphi (caso tenha mais de uma versão instalada) deseja que o IBO seja instalado. *Note que o instalador atual tem um problema com o Delphi 2005, fazendo com que as BPLs sejam geradas no lugar errado, e gerando um erro quando você entra no Delphi.* O instalador também pode criar um ícone na área de trabalho para a ferramenta IB_SQL.

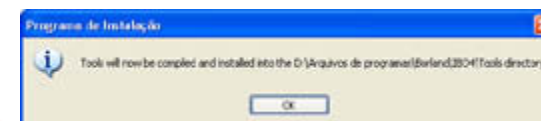


Clique em "Avançar" e depois em "Instalar". O processo de instalação iniciará a cópia dos arquivos do IBO para o diretório previamente selecionado. Em seguida, uma tela de aviso será exibida, indicando que o instalador irá compilar e instalar os componentes nas versões do Delphi selecionadas:



Clique em OK. Em seguida, uma outra tela de aviso será apresentada, informando que

as ferramentas do IBO serão compiladas:



Clique em "OK" e depois em "Concluir".

A próxima vez que o Delphi for aberto, a palheta de componentes exibirá onze novas tabs (veja **figura 1**).

Note que, por padrão, o instalador disponibiliza todos os componentes do IBO. Você pode desabilitar os pacotes que não serão utilizados através da opção "Install Packages" do menu "Component" do Delphi.

Suporte

O IBO possui uma grande comunidade de usuários que participam da lista de discussão oficial encontrada em <http://groups.yahoo.com/group/IBObjects/>. No Brasil, vários usuários do IBO participam da lista de discussão da FireBase (www.firebase.com.br). Um outro site brasileiro que contém diversas informações e exemplos para o IBO é o site do RedDevil, em <http://www.reddevil.eti.br/ibob.htm>.

O site oficial também possui inúmeros documentos que focam áreas e componentes específicos, sendo uma ótima fonte de informação para qualquer usuário. Esses documentos podem ser acessados em <http://www.ibobjects.com/TechInfo.html>.



Figura 1. Palhetas de componentes do IBO

Componente	Descrição	BDE
TIB_Session	Classe base para os componentes do IBO. Geralmente não é necessária, a não ser que você esteja programando em MultiThread.	TSession
TIB_Connection	Responsável pela conexão com o Banco de Dados	TDatabase
TIB_Transaction	Permite configurar uma transação associada a uma conexão.	-
TIB_Cursor	Dataset unidirecional e não bufferizado, por isso é muito eficiente.	-
TIB_DSQL	Utilizado para executar comandos de DML ou DDL que não retornem múltiplas linhas de resultado.	-

Tabela 1. iboCore

Componente	Descrição	BDE
TIB_DataSource	Componente que serve de fonte de dados (ponte) entre um dataset nativo do IBO e um componente de edição nativo.	TDataSource
TIB_Query	Componente nativo para execução de queries	TQuery
TIB_Database	Componente para manter a compatibilidade com o TDataBase da BDE, especialmente útil para quem está migrando aplicações da BDE para o IBO.	TDatabase
TIB_StoredProc	Utilizado para executar Stored Procedures que retornem apenas uma linha de resultado. Caso a procedure retorne várias linhas, o componente age como um TIB_Cursor, permitindo a recuperação de todas as linhas.	TStoredProc

Tabela 2. iboAccess

Componente	Descrição	BDE
TIB_Script	Responsável pela execução de scripts	-
TIB_Events	Permite que a aplicação receba a notificação de eventos enviada pelo servidor	-
TIB_Export	Permite a exportação dos dados de uma query em arquivos TXT, DBF ou formato definido pelo usuário.	-
TIB_Import	Permite a importação de dados de um arquivo ASCII (texto) para uma tabela ou através do uso de uma stored procedure.	-
TIB_DataPump	Facilita o pump (movimentação) de dados entre uma fonte e o destino. As informações lidas na fonte de dados são atribuídas automaticamente ao comando responsável pela inserção no destino, podendo ser um insert ou uma procedure.	-
TIB_Monitor	Permite monitorar todos os comandos que estão sendo enviados pela aplicação para o servidor firebird.	-
TIB_Profiler	Dá acesso às informações de utilização do servidor firebird (memória utilizada, qtd de escritas e leituras, etc), retornadas pela função isc_database_info da API.	-

Tabela 3. iboTools

Componente	Descrição	BDE
TIB_SearchPanel	Monta uma painel exibindo os campos de uma query, permitindo além da edição dos dados, a busca de informações através da entrada dos valores de filtragem nos edits referentes aos campos apropriados, quando o dataset está em modo de busca.	-
TIB_SearchEdit	Permite a busca rápida dos registros que satisfazem a informação de filtragem especificada. Deve ser utilizado em conjunto com a propriedade SearchingLinks do dataset.	-
TIB_IncSearch	Permite a procura incremental de registros, com opção de busca por proximidade, etc.	-
TIB_ParamEdit	Permite a alteração direta do valor de um parâmetro de uma Query	-
TIB_LocateEdit	Faz uma busca no dataset pela informação digitada, utilizando o método locate.	-

Tabela 4. iboSearch

Para quem desejar instalar o HELP na IDE do Delphi, basta baixar o arquivo disponível em http://www.ibobjects.com/IBO45_Help.zip e seguir as instruções nele contidas.

Por último, existe também um guia para iniciantes que pode ser adquirido na loja online do IBO, pelo endereço http://community.ibobjects.com/?WEB_RE_ID=10008.

Componentes

A seguir analisaremos os principais componentes das diversas palhetas de componentes instaladas, fornecendo uma pequena descrição de sua função e indicando quando houver um componente da BDE com função semelhante.

As palhetas:

iboCore (ver tabela 1): Possui os componentes mais básicos do IBO, responsáveis por tarefas elementares, como conexão com o banco, configuração de transações, etc.

iboAccess (ver tabela 2): Contém os componentes de acesso à dados nativos do IBO.

iboTools (ver tabela 3): Componentes para execução de tarefas, como rodar scripts, exportar/importar dados, busca textual, replicação, etc.

iboSearch (ver tabela 4): O IBO possui um modo especial de “edição de dados” chamado modo Search, que permite utilizar os controles de edição de dados para criar filtros de pesquisa no dataset associado.

iboComponents (ver tabela 1): São os componentes de edição de dados nativos do IBO. Apesar das muitas semelhanças com

Componente	Descrição	BDE
TIB_Text	Apresenta o conteúdo de um campo na forma de texto	TDBText
TIB_Label	Apresenta o label associado a um campo de um dataset	-
TIB_Edit	Utilizado para a edição de dados. Possui várias melhorias se comparado com o TDBEdit padrão do Delphi, como possibilidade de apresentar um botão junto com o edit e de ter um “label” automaticamente associado com o edit.	TDBEdit
TIB_DateTimePicker	Componente para a edição de valores de data e hora.	TDBDateTimePicker
TIB_Date	Utilizado para a edição de datas	-
TIB_Currency	Componente específico para a edição de valores monetários, com possibilidade de apresentar uma calculadora.	-
TIB_Memo	Responsável pela edição de blobs textuais sem formatação (texto puro)	TDBMemo
TIB_RichEdit	Responsável pela edição de blobs contendo texto no padrão RTF (RichEdit)	TDBRichEdit
TIB_Image	Apresenta imagens do tipo BMP armazenadas em blobs	TDBImage
TIB_ListBox	Permite a seleção do valor de um campo através de uma lista de valores	TDBListBox
TIB_ComboBox	Permite a seleção do valor de um campo através de uma lista corrida de valores	TDBComboBox
TIB_CheckBox	Permite selecionar ou não um valor associado a um campo. Usado geralmente para representar campos lógicos (true/false – ligado/desligado)	TDBCheckBox
TIB_RadioGroup	Permite agrupar uma série de radiobuttons representando valores que podem ser selecionado e armazenados no campo associado.	TDBRadioGroup
TIB_LookupList	Semelhante ao TIB_ListBox, mas a lista de valores é preenchida automaticamente de uma fonte de dados (query) associada.	TDBLookupList
TIB_LookupCombo	Semelhante ao TIB_ComboBox, mas a lista de valores é preenchida automaticamente de uma fonte de dados (query) associada.	TDBLookupCombo
TIB_TrackBar	Exibe uma barra com marcações (como uma régua) permitindo a navegação por um dataset.	-
TIB_JPEGImage	Componente para apresentar imagens JPEG armazenadas em blobs.	-

Tabela 5. iboControls

Componente	Descrição	BDE
TIB_Grid	Grade para a representação dos registros retornados por uma query. Possui inúmeros recursos não encontrados no DBGrid padrão do Delphi, como utilização de cores para indicar diferentes estados do dataset, ordenação, apresentação de glifos nos títulos das colunas, etc.	TDBGrid
TIB_ControlGrid	Permite a exibição dos registros de um dataset com formatação livre	TDBCtrlGrid
TIB_Ledger	Grade especial onde um mesmo registro pode exibir dados em mais de uma linha. Você pode definir quais colunas serão apresentadas em quais linhas.	-
TIB_ArrayGrid	Possibilita a edição de dados armazenados em campos do tipo ARRAY	-
TIB_CursorGrid	Permite a bufferização e exibição na forma de grade dos resultados de uma query executada por um TIB_Query ou TIB_Cursor.	-
TIB_StatementGrid	Monta uma grade onde cada linha pode representar os campos (e seus valores) de um dataset, ou os parâmetros utilizados nesse dataset.	-
TIB_TreeView	Exibe uma lista de valores recuperados de um dataset em forma de árvore.	-

Tabela 6. iboGrid

Componente	Descrição	BDE
TIB_ConnectionBar	Botões para conectar e desconectar um BD	-
TIB_TransactionBar	Botões para controlar transações (commit, rollback, etc.)	-
TIB_StatementBar	Botões para preparar, despreparar e executar comandos SQL	-
TIB_DatasetBar	Botões para abrir, fechar, preparar e despreparar um dataset	-
TIB_SearchBar	Permite controlar o mecanismo interno do IBO para realizar consultas (filtros) nos datasets	-
TIB_NavigationBar	Botões de navegação pelo dataset (próximo, anterior, último, primeiro registro, etc.)	TDBNavigator
TIB_UpdateBar	Botões para controlar a edição de dados (insert, edit, update, post, delete...)	TDBNavigator
TIB_UtilityBar	Botões para a execução de tarefas como exportação dos dados, execução de scripts, etc.	-

Tabela 7. iboBars

Componente	Descrição
TIB_DSQDialog	Janela para execução de comandos de Dynamic SQL.
TIB_LookupDialog	Janela para pesquisas do tipo lookup.
TIB_BrowseDialog	Exibe uma janela com todos os objetos do banco de dados (tabelas, procedures, domains, etc), permitindo a navegação e edição dos dados, execução de procedimentos, etc.
TIB_ScriptDialog	Janela para edição e execução de scripts SQL.
TIB_DDL_ExtractDialog	Janela para extração da estrutura do banco de dados em forma de script SQL.
TIB_MetadataDialog	Permite a configuração de parâmetros e extração da metadata de um BD na forma de script SQL.
TIB_EventsDialog	Permite o registro e a remoção da notificação de eventos do banco de dados.
TIB_ExportDialog	Janela para exportação dos dados retornados pelo select na forma de arquivos texto ou DBF.
TIB_DataDumpDialog	Permite o DUMP da estrutura das tabelas e das informações dos campos selecionados na forma de arquivos texto delimitados.
TIB_DataPumpDialog	Janela para configuração dos comandos de PUMP de dados entre tabelas do mesmo banco ou de um banco diferente.
TIB_MonitorDialog	Exibe todos os comandos e instruções que estão sendo enviadas para o servidor.
TIB_ProfilerDialog	Permite a monitoração do tempo de execução dos comandos enviados ao servidor
TIB_StatusDialog	Exibe o status completo da aplicação, incluindo forms, conexões, datasets, transações, etc.

Tabela 8. iboDialogs

Componente	Descrição	BDE
TIBOTable	Emula um TTable da BDE	TTable
TIBOQuery	Compatível com a TQuery da BDE. Internamente utiliza um TIB_Query para recuperação dos dados.	TQuery
TIBOStoredProc	Executa Stored Procedures	TStoredProc
TIBOUpdateSQL	Permite associar comandos de inserção, edição e remoção de dados à um TIBOQuery. Existe apenas para facilitar a migração dos sistemas já existentes que utilizam a BDE, visto que o componente TIBOQuery já possui internamente propriedades para essa função.	TUpdateSQL
TIBODatabase	Realiza a conexão com um BD emulando um TDatabase. Internamente utiliza um TIB_Connection.	TDatabase
TIBOTransaction	Permite a configuração das características de uma transação.	-

Tabela 9. iboTDataSet

Componente	Descrição	Componente similar
TIB_ChunkProducer	Producer nativo do IBO para ser utilizado com aplicações webbroker	-
TIB_ChunkProvider	Provider nativo do IBO para ser utilizado com aplicações webbroker	-
TIBODatasetTableProducer	TableProducer nativo do IBO para ser utilizado com aplicações webbroker	TDataSetTableProducer

Tabela 10. iboInternet

Componente	Descrição	Componente similar
TIB_EndUserSessionAdapter	Emula um TEndUserSessionAdapter	TEndUserSessionAdapter
TIB_LoginFormAdapter	Emula um TLoginFormAdapter	TLoginFormAdapter
TIB_AdapterPageProducer	Emula um TAdapterPageProducer	TAdapterPageProducer

Tabela 11. iboWebsnap

os componentes padrão do Delphi, esses controles são otimizados para trabalhar com o IB/FB, e portanto fornecem uma melhor performance se comparados com os componentes “semelhantes”.

iboGrid (ver tabela 6): O IBO oferece vários tipos de grid para apresentação e edição de dados. Diferente do que acontece com o DBGrid padrão do Delphi, grande parte da configuração dos grids do IBO é feita diretamente nas propriedades do dataset, e não no grid propriamente dito.

iboBars (ver tabela 7): Consiste em diversas barras associadas a diversas tarefas, como a edição de dados, controle de transações, navegação pelos dados, etc.

iboDialogs (ver tabela 8): Caixas de diálogo pré-definidas e associadas com diferentes funções. Não existe componentes de diálogo similares no Delphi padrão.

iboTDataSet (ver tabela 9): Esses são os componentes de acesso a dados do IBO derivados derivados do TDataSet e portanto compatíveis com qualquer componente dataware do Delphi ou de terceiros. Internamente, esses componentes utilizam componentes nativos do próprio IBO.

iboInternet (ver tabela 10) e **iboWebsnap** (ver tabela 11): São componentes nativos do IBO para se trabalhar com aplicações web no Delphi.

No próximo artigo...

A segunda parte do artigo tratará dos componentes do IBO compatíveis com o TDataSet. Enquanto isso, você pode ir

estudando os exemplos e tutoriais que acompanham o IBO, mas não se desespere! Você pode estranhar como é feita a configuração dos controles nativos, onde diferente do que acontece com os componentes padrão do Delphi, boa parte da configuração visual é feita diretamente no componente de acesso a dados (TIB_Query...) e não no controle de edição de dados. Tudo isso será explicado na terceira parte do artigo, quando veremos os componentes e controles nativos do IBO (que não são compatíveis com o TDDataset).

Até mais!

Avalie esse artigo

Autor:

Carlos Henrique Cantu

Mini-curriculo

Bacharel em Ciência da Computação, micreiro desde os tempos de Apple II, trabalha com desenvolvimento de sistemas há 15 anos. É consultor especializado em bancos de dados Firebird e InterBase 6.0. Mantenedor do site FireBase (www.firebase.com.br) e autor do livro Firebird Essencial e presidente do DUG-BR (Delphi Users Group Brasil), tendo ministrado palestras sobre Firebird para milhares de pessoas em diversas cidade do Brasil.



Precisando de cursos ou treinamentos de Firebird?

A FireBase oferece cursos/treinamentos de Firebird ministrados dentro da sua empresa. Os cursos são ministrados por **Carlos H. Cantu**, um dos maiores evangelistas do Firebird no Brasil, autor do livro Firebird Essencial. Mais informações pelo email cursos@firebase.com.br

Entrevista com membros da equipe do PostgreSQL

A entrevista foi realizada pela DB FreeMagazine em Junho/2005 com alguns dos membros da equipe de desenvolvimento do PostgreSQL, onde falam sobre suas experiências e ligações com o projeto.

1) Fale-nos um pouco sobre cada um de vocês.

ANDREW DUNSTAN: Tenho sido desenvolvedor, administrador, gerente, instrutor e consultor na área da tecnologia da informação nos Estados Unidos e Austrália por mais de 20 anos. Minha experiência com banco de dados inclui Ingres, Oracle, Sybase, Informix, MS-SQL, MySQL e obviamente PostgreSQL. Atualmente trabalho na HCS Systems em Garner, na Carolina do Norte, como consultor, desenvolvedor Senior e fundador da Database Practice.

ROBERT TREAT: Comecei a usar Open Source no final de 1996. Meu primeiro sistema gerenciador de banco de dados foi o PostgreSQL, e desde então experimentei alguns outros bancos de dados, mas sempre retornando ao PostgreSQL. Estou atualmente envolvido no desenvolvimento do PostgreSQL

com aplicações voltadas para a indústria médica.

OLEG BARTUNOV: Astrônomo profissional, atualmente trabalho no Instituto Astronômico Sternberg, na Universidade de Moscow. Sou casado e tenho três filhos. Sou o fundador do instituto de computação em rede e participo de diversos projetos na internet, como o desenvolvimento do maior portal russo na web (Rambler) e diversos portais do governo federal, como o do ministério da educação, obviamente utilizando PostgreSQL.

JOSH BERKUS: Trabalho com banco de dados desde 1994, incluindo Paradox, Rememdy, dBase, Clipper, MS-Access, MS-SQL e PostgreSQL. Fui escultor, padeiro, organizador de laboratório, captador de recursos, sendo praticamente meu envolvimento com a área de relações públicas de projetos Open Source, primeiramente no Open Office e agora no PostgreSQL.

CLAUDIO FILHO: Tenho Oleg Bartunov uma consultoria de negócios para migração de soluções proprietárias para software livre, trabalhando em ambos os lados (cliente e servidor). Resido atualmente no estado do Mato Grosso – Brasil, e também trabalho em outros projetos Open Source, como OOoBr, GimpBrasil e Mozilla/Brasil.



2) Como foi seu envolvimento com o PostgreSQL?

BARTUNOV: Minha primeira experiência foi com o Postgres (4.1) em 1993, quando comprei um CD com Linux (kernel 0.99pl13) de Yggdrasil. Em 1996, participava no projeto Redline, onde o banco de dados utilizado necessitava trabalhar corretamente com textos em russo. Tive que descobrir como fazer com que o Postgres95 trabalhasse adequadamente com o idioma local. Após duas semanas, submeti um patch introduzindo o suporte local (russo) ao Postgresql.

DUNSTAN: A companhia que eu estava trabalhando em 2003 procurava um banco de dados para distribuir com seus produtos. O PostgreSQL foi visto como uma boa escolha, mas havia o problema de não rodar no Windows. Durante minha pesquisa sobre o assunto, acabei freqüentando uma lista hacker, quando me ofereci para fazer uma pequena melhoria no projeto, e a partir daí não parei mais.

TREAT: Meu chefe no momento estava usando PostgreSQL para desenvolvimento web, por isso comecei com o ele. A partir disso eu me inscrevi na lista do PostgreSQL, e tive algumas largas participações em algumas discussões. Foi então que me ofereci para fazer alguns trabalhos para os web sites do

PostgreSQL. A partir daí a bola de neve só foi crescendo.

BERKUS: Em 1998, estava procurando por um banco de dados confiável, que rodasse em Linux e tivesse as características para substituir o MS-SQL Server para meus clientes. Foi então que instalei uma cópia do PostgreSQL 7.1, que por sinal está rodando até hoje!

FILHO: Eu montei uma página pessoal sobre meu BD favorito e algumas pessoas me escreveram, enviando dicas e artigos. Em alguns meses nós formamos a comunidade Brasileira. Mostrei meu trabalho ao Justin Clift, um velho amigo, que ficou impressionado e colocou um link no site postgresql.org.



Cláudio Filho

3) O que você faz no projeto PostgreSQL agora? Em que projeto você está trabalhando?

DUNSTAN: Mais recentemente, fiz algumas correções e melhorias no nosso utilitário de importação/exportação do CSV, desenvolvido originalmente por mim para o PostgreSQL 8.0. Dediquei boa parte do meu tempo nos últimos seis meses a criar, manter e melhorar o *buildfarm*, um ambiente automatizado de builds para o PostgreSQL – veja mais em www.pgbuildfarm.org. Sou um dos administradores

da comunidade de desenvolvimento *pgfoundry*. No momento estou trabalhando em algumas melhorias no suporte pl/perl, que espero finalizar para o release 8.1.

BARTUNOV: Mantenho diversos módulos para o PostgreSQL, além do dispositivo de busca PostgreSQL - www.pgsql.ru. Estou buscando suporte para melhorias do GiST e no sistema de busca textual, além de estar envolvido no desenvolvimento de um efetivo esquema de indexação para os gigantescos catálogos astronômicos, e defendendo a adoção de PostgreSQL pela comunidade astronômica.

TREAT: Meu principal objetivo está na construção da infra-estrutura web para o PostgreSQL, incluindo e integrando novos conteúdos dentro do site, junto com o resto do time de desenvolvedores web. Aproveito para contribuir com correções sempre que eu encontro alguma coisa que precise de reparos, como correções na documentação.

BERKUS: Participo do grupo principal de desenvolvimento, atualmente composto por seis pessoas. Geralmente cuido das “Press Releases” de cada versão, me relacionando com empresas que usam e dão suporte ao PostgreSQL. Também sou o presidente da Fundação PostgreSQL, que por enquanto

existe só no papel. Estou envolvido nos testes de performance e trabalho no projeto para expandir a escalabilidade do PostgreSQL.

FILHO: Bem, no Brasil eu criei o site PostgreSQL-Br. Com a ajuda de alguns amigos, como o Diogo Biazus, o site é hoje referência do PostgreSQL em Português.

4) História do PostgreSQL

BERKUS: Nossa história está disponível em www.postgresql.org/about/history. Começamos em 1986, como um projeto acadêmico na Universidade da Califórnia em Berkeley, liderado por Michael Stonebraker. Naquela época usávamos o nome de POSTGRES, porque foi o sucessor do banco de dados INGRES.

Em 1995, o Postgres foi colocado na Internet sob o licenciamento BSD. Uma comunidade de usuários o adotou e migrou seu parser de QUEL para SQL, renomeando o projeto para Postgres95 e posteriormente para PostgreSQL. Em 1997, tivemos a produção de um release estável e em 1998 começamos a ver a adoção comercial do banco. Desde então nossa comunidade tem crescido, de uma dúzia de usuários para dezenas de milhares de comunidades ativas, com suporte de diversas companhias.

5) Quantos membros têm o grupo de desenvolvimento?

BERKUS: É difícil precisar números. Nós temos cerca de duzentos contribuintes para cada release do PostgreSQL, onde 24

peças desse grupo contribuem com grande quantidade de código. Mais informações sobre eles podem ser encontradas em www.postgresql.org/developer/bios

6) O MySQL diz ter milhões de usuários. Qual é o número estimado de usuários de PostgreSQL no mundo?

TREAT: Como temos muitos e diferentes canais de distribuição (mais de 60 sites espelhos, incluindo as maiores distribuições), especificar números é difícil. Mas eu diria que não estamos longe de ter milhões de usuários. Não estou dizendo que temos mais usuários que o MySQL, pois acredito que isso ainda não aconteça, mas assim como a relação entre Windows e Linux, nossos usuários são passionais e estão espalhados por todo o mundo.

DUNSTAN: Ninguém sabe ao certo. Geralmente as pessoas se frustram com os limites do MySQL e procuram algo melhor. Outras pessoas procuram não se guiar somente pela popularidade, e fazem testes com diversos bancos de dados antes de escolher o que irá utilizar. Geralmente o PostgreSQL se sai muito bem nesses testes, causando até surpresa em alguns dos avaliadores.

Minha opinião é que boas aplicações deveriam concentrar o acesso ao banco de dados em units específicas, e não espalhar SQL por todo o código do programa. Isso tornaria mais fácil a mudança de um banco para outro. Parece que agora algumas pessoas estão entendendo isso. O projeto Bugzilla acaba

de incluir suporte ao PostgreSQL, que estará disponível na próxima versão. Eu enviei a eles algum código de abstração para a criação do esquema de banco de dados. Acredito que nossa nova versão Windows teve centenas de milhares de downloads em apenas alguns meses.

BARTUNOV: Não tenho idéia, mas o PostgreSQL tem boa reputação na Rússia e todos os meus projetos são baseados nele. Meu review do PostgreSQL apresentado na conferência anual "Corporate Database 2005" teve mais de 10.000 downloads em duas semanas, e as críticas foram muito positivas. O suporte nativo ao Windows poderá atrair novos usuários.

7) O que podemos esperar no PostgreSQL 8.1?

BARTUNOV: Alguns dos novos recursos já estão disponíveis no CVS: índices por bitmaps, IN OUT e INOUT para PL/PGSQL, aumento no número de argumentos de funções (100 por default), otimização de MIN e MAX, melhorias no Pitr.

DUNSTAN: Como sempre muitas novidades. Aparentemente o suporte ao commit em duas fases estará disponível, e talvez alguma forma de particionar tabelas. Acredito que esses novos recursos irão aumentar bastante nossa credibilidade.

TREAT: O PostgreSQL 8.1 trará ótimas novidades. As principais, na minha opinião, são: *in-memory bit-mapped indexes* (permite

JobVox
Sistemas de automação de voz

Vox on Demand

Automação de discagem e emissão de recados	URA - Unidade Remota de Atendimento
--	---

Gravação de
ligações telefônicas

Site: <http://www.jobvox.com.br>
E-Mail: SAC@JOBVOX.COM.BR

Praciosa São Paulo

que se utilize mais de um índice para uma mesma tabela em uma query) e *shared row locks* (que será usado internamente para auxiliar a eliminar os dead-locks na checagem de foreign keys, e consequentemente aumentar a performance). Um outro recurso no qual estive pessoalmente envolvido foi uma melhoria para o PSQL usar automaticamente savepoints para capturar erros, e não ter sua transação desfeita (por um rollback).

8) Quais são as mais populares linguagens utilizadas com PostgreSQL?

DUNSTAN: Assumindo que você esteja falando de linguagens para o desenvolvimento dos softwares cliente, as mais populares são provavelmente PHP, Java e Perl. Eu desenvolvo todos os meus clientes em Java e Perl. De qualquer forma, temos muitos usuários que utilizam C, C++, C#, Ruby, Python e outras mais.

TREAT: O PostgreSQL tem boas interfaces para qualquer linguagem, mas a Perl é vista como a mais popular. A base de usuários

de PHP está crescendo cada vez mais, e ganhamos atenção de algumas tecnologias novas como MONO e Ruby.

BERKUS: Baseado no suporte requerido e no envolvimento corporativo, eu poderia dizer que Java é provavelmente a mais popular linguagem para usar com PostgreSQL, mas eu penso já ter visto de tudo, inclusive .NET.

9) Quais são as maiores limitações e “gotchas” do PostgreSQL?

TREAT: Eu acho que o maior problema é na pronuncia correta do nome do postgresQL :-). Não é Post-gree ou pahst-grey, mas Post-Gres-Q-L (ou post-gres para encurtar). Na questão técnica, penso que o principal problema é a atualização de BDs já existentes para novas versões do PostgreSQL, que geralmente exige um DUMP e um RESTORE de todos os dados do banco, o que pode ser um problema para grandes instalações. Existem soluções intermediárias, como usar Slony para replicar dados entre diferentes versões do banco de dados, mas nem sempre funciona.

DUNSTAN: Algumas dicas:

- Devemos tomar cuidado ao usar herança de tabelas – geralmente precisa-se de um pouco de intuição. Pessoalmente eu não uso esse recurso.
- Analise frequentemente o Vacuum.
- O ajuste fino para uma boa performance não é fácil, e pode exigir um pouco de “magia negra”, mas se bem feito, não haverá

arrependimentos.

- Tenha certeza de usar *connection pooling* caso haja diversas conexões de vida curta (exemplo: aplicações web).

BARTUNOV: Atualmente o PostgreSQL não tem controle de concorrência e suporte a recuperação para o GiST, que já existe em muitas outras extensões e poderia certamente melhorar nesse sentido, bem como um melhor integrador de planos.

Aplicações Web requerem um efetivo suporte a queries do tipo “Top-K”. Isso poderia ser implementado usando LIMIT e OFFSET, mas atualmente não é muito eficaz. A performance de um count(*) também é um problema inerente ao MVCC do PostgreSQL e provavelmente será melhorado no futuro.

10) Você poderia me listar alguns usuários proeminentes do PostgreSQL?

FILHO: Aqui no Brasil, alguns bons exemplos são a DATAPrev (a companhia de processamento de dados da previdência social) e o Metrô de São Paulo. Há outros, mas sem números.

DELEITE: Bem, companhias como a Apple e Red Hat comumente fazem uso do PostgreSQL para suas próprias aplicações. A Fujitsu, Pervasive e SRA também estão utilizando PostgreSQL em seus negócios.

DUSTAN: É comum as companhias não divulgarem se estão usando PostgreSQL em seus negócios. Não é raro você encontrar o

PostgreSQL sendo utilizado em lugares que nem imaginava. Isso é normal, pois para usar o PostgreSQL não é necessário assinar um cheque.

BARUNTOV: Eu posso mencionar o Rambler (www.rambler.ru).

BERKUS: Adicionando aos já citados, temos : Afiliadas, Cisco, NTT, Hyperic, GreenPlum, Netezza, Sun, VA Software, TravelPost, National Weather Service e numerosas agências governamentais que não estou autorizado a mencionar publicamente. Atualmente tenho falado com o pessoal de TI de uma grande corporação que está utilizando PostgreSQL em diversos locais.

11) Que tipo de replicação e clusterização o PostgreSQL suporta?

TREAT: O queridinho da comunidade atualmente é o *Slony*, que é um ambiente de replicação multi-plataforma que está se preparando para a versão 1.1. Mais informações podem ser obtidas em slony.info. Há também uma solução de replicação multi-master chamada *pgcluster*, que está ganhando atenção entre os usuários (pgfoundry.org/projects/pgcluster).

DUNSTAN: O autor do Slony já está trabalhando com algumas outras pessoas no *Slony-II*, que será um sistema de replicação síncrono multi-master.

BERKUS: Existem outras soluções para replicação e clustering, tanto Open Source

como comerciais, para uma variedade de usos especializados, incluindo o DBmirror, C-JDBC, pgReplicator, Mammoth Replicator, ExtenDB, e DeepGreen MPP. Nós mantivemos a replicação e o clustering fora da distribuição oficial, pois nenhum produto atualmente cobre todos os casos possíveis de utilização.

12) Alguns programadores não entendem porque o PostgreSQL para Linux é gratuito e para Windows não. Você poderia explicar?

BERKUS: Essa informação não é verdadeira. O PostgreSQL 8.0 está disponível gratuitamente para qualquer sistema operacional. Houve um tempo onde alguma companhias tinham versões do PostgreSQL para Windows que não eram gratuitas, mas eu acho que isso não acontece mais.

13) Deseja enviar alguma mensagem para os brasileiros?

FILHO: Se você usa o PostgreSQL em português, visite nossa comunidade em www.postgresql.org.br

DUNSTAN: A comunidade PostgreSQL é uma das comunidades genuinamente abertas que existem. O suporte que você pode obter é excelente. Nós temos um canal IRC para os brasileiros – junte-se a nós, você não se arrepende!

TREAT: A adoção em massa do PostgreSQL já começou, e agora é a hora de subir no barco. O PostgreSQL oferece a tecnologia e

os recursos que você precisa, e um processo de desenvolvimento onde qualquer um pode participar. Os brasileiros já têm uma comunidade, não deixe de conferir: <http://www.postgresql.org.br/index.php?corpo=links>.

Avalie esse artigo

SQL> SELECT * FROM HOSPEDAGEM WHERE QUALIDADE="INSUPERAVEL";

+-----+
| WWW.BAVS.COM.BR |
+-----+

Cada vez mais empresas desenvolvem sistemas multi-usuários que necessitam que seus dados sejam disponibilizados na Internet, em tempo real, para atender às necessidades de seus clientes. Oferecemos as melhores soluções em hospedagens de bancos de dados MySQL e Firebird (1.0 e 1.5), com acesso direto e sem restrições de conexão.

PLANOS DE HOSPEDAGEM		
PRO I	PRO III	SEMI D. II
Principais Características: ✓ 100 Mb espaço em disco ✓ Firebird 1.0 ✓ MySQL 3.23 ✓ PHP 4 ✓ Perl 5 ✓ CGI ✓ Diretório SSL Gratuito ✓ Configuração Gratuita ✓ Mensalidade: R\$ 29,90	Principais Características: ✓ 300 Mb espaço em disco ✓ Firebird 1.5 ✓ MySQL 4 ✓ PHP 4 / Perl 5 / CGI ✓ JSP (Tomcat) ✓ Servlet ✓ ASP .NET (Mono/C#) ✓ Configuração Gratuita ✓ Mensalidade: R\$ 69,90	Principais Características: ✓ 1 Gb espaço em disco ✓ Firebird 1.5 ✓ MySQL 4 ✓ PHP 4 / Perl 5 / CGI ✓ JSP (Tomcat) ✓ Servlet ✓ ASP .NET (Mono/C#) ✓ Configuração Gratuita ✓ Mensalidade: R\$ 145,00

+ PLANOS
acesse:
WWW.BAVS.COM.BR
+ INFORMAÇÕES

clientes.com.satisfação
Email: info@bavs.com.br
Atendimento Eletrônico 24hs:
(19) 3423-0251
Vendas On-line (ambiente seguro):
<http://www.bavs.com.br>

WWW.BAVS.COM.BR

Utilização básica do INSERT, UPDATE e DELETE

Luiz Paulo de O. Santos

Dando seqüência ao artigo publicado na segunda edição da DB FreeMagazine, estaremos agora estudando outras instruções fundamentais do SQL, são elas: INSERT (inserir registros em uma tabela), UPDATE (alterar o conteúdo de um ou mais campos/registros) e o DELETE (remover registros de uma tabela).

Utilizaremos como base para os exemplos, as mesmas tabelas usadas no artigo "Desvendando o Select", publicado na segunda edição.

INSERT

Sintaxe:

```
INSERT [ INTO ]
{ tabela | view }
[ (lista de colunas) ]
{ [DEFAULT] VALUES | VALUES ( lista
de conteúdo dos campos ) ] |
SELECT ... }
```

Obviamente, só podemos inserir registros em tabelas previamente criadas no BD através do comando CREATE TABLE. Costumo dizer que

os bancos de dados devem ser gerados para armazenar informações, portanto um banco cheio de NULLs indica provavelmente alguma falha na modelagem.

***Nota:** Ao inserirmos linhas (registros) em uma tabela ou view omitindo algumas de suas colunas (campos), o servidor automaticamente atribuirá a elas os valores DEFAULT, caso tenham sido declarados durante a criação das mesmas.*

Basicamente o INSERT é utilizado de duas maneiras:

Na sua forma mais comum, indicamos através da cláusula VALUES os valores que serão atribuídos às colunas, na seqüência em que foram discriminadas na instrução de insert. Desta forma, a linha terá um código claro, legível e de fácil entendimento. O exemplo abaixo irá inserir na tabela PESSOAS um novo registro com os valores LARA para o campo NOME, 24 para IDADE e 55 para PESO:

```
INSERT INTO PESSOAS (NOME, IDADE,
PESO) VALUES ("LARA", 24, 55);
```

Na instrução a seguir, não incluímos o campo PESO na instrução insert. Assim, caso haja um valor *default* associado ao campo PESO, o servidor irá atribuí-lo automaticamente, caso contrário o campo PESO ficará **nulo** (null).

```
INSERT INTO PESSOAS (NOME, IDADE) VA-
LUES ("LUIZ PAULO", 32);
```

Note que não haverá erro durante a inserção de registros omitindo alguns campos, mesmo

que esses campos estejam definidos como NOT NULL, desde que tenham valores **default** associados..

Outra forma de realizarmos inserções de registros é utilizar um select como fonte de dados para o insert, veja:

```
INSERT INTO PESSOAS (NOME, IDADE)
VALUES (SELECT FUNNOME, FUNIDA
FROM FUNCIONARIOS
WHERE OK= 'SIM');
```

A instrução acima irá inserir na tabela PESSOAS, registros baseados no retorno da consulta realizada na tabela funcionários, filtrando somente os registros onde o campo OK seja igual a 'SIM'.

UPDATE

Sintaxe:

```
UPDATE {tabela | view}
SET {coluna | variavel} = {DEFAULT |
expressão} [,...]
WHERE condição
```

Update é a instrução SQL que fará alterações nos registros já existentes no banco de dados.

Basicamente devemos informar a tabela ou view que desejamos atualizar, a clausula SET, o campo e o valor que será atribuído a ele, podendo ser uma constante ou resultado de uma expressão, ou até mesmo o retorno de uma função/UDF.

Exemplos:


```
UPDATE PESSOAS SET IDADE = IDADE + 1;
```

No exemplo acima, acrescentamos 1 para a idade de todos os registros cadastrados na tabela de PESSOAS.

```
UPDATE ACESSOS  
SET QTDE = QTDE + 1  
WHERE PESSOA IN  
(SELECT UNAME FROM PESSOAS WHERE  
IDADE > 1);
```

O comando acima somará 1 ao campo QTDE dos registros da tabela ACESSOS que possuem uma pessoa (indicado pelo campo PESSOA da tabela ACESSOS) relacionada na tabela PESSOA, e cuja idade seja maior que 1.

Nota: Devemos lembrar que para um usuário logado poder fazer um update em uma tabela, ele deve ter permissão para isso. As permissões são atribuídas pelo comando GRANT do SQL.

Devemos ter cuidado ao executar um UPDATE, pois com a simples troca do conteúdo de um registro podemos comprometer a integridade relacional do banco, “bagunçar” de maneira irreversível o conteúdo, ou até mesmo comprometer todas as bases (se estiver efetuando UPDATE em, tabelas do sistema). “Se beber, não use update” ☺

DELETE

Sintaxe:

```
DELETE [FROM] tabela [WHERE condi-  
ção];
```

A instrução DELETE remove registros de uma tabela, geralmente baseada em uma condição especificada na cláusula WHERE.

Exemplos:

```
DELETE FROM ACESSOS WHERE (QTDE < 1);
```

O comando acima remove todos os registros da tabela ACESSOS onde o campo QTDE seja menor que 1.

```
DELETE FROM ACESSOS WHERE (PESSOA IS  
NULL);
```

O comando acima remove os registros da tabela ACESSOS cujo conteúdo do campo PESSOA esteja NULL.

```
DELETE FROM PESSOAS;
```

A linha acima irá apagar todos os registros da tabela.

Atenção! Lembre-se que as instruções que você executa na sua aplicação cliente podem estar associadas a uma transação no BD. Nesse caso, devemos lembrar de dar um commit para confirmar as operações, ou um rollback para desfazê-las.

Avalie esse artigo

Autor:

Luiz Paulo de Oliveira Santos

Mini-curriculo

Luiz Paulo de Oliveira Santos é formado em Tecnologia de Processamento de Dados, especialista em Análise de Sistemas e Redes de Computadores. É analista de suporte de redes na Universidade Metodista de Piracicaba e diretor da JobVox Sistemas Informatizados. lpaulo@jobvox.com.br

Calendário de Eventos

Data	Evento	Site
13 a 15 de julho de 2005	III Encontro técnico do Software Livre Campo Grande -MS	http://www.sgi.ms.gov.br/Sgi/SoftwareLivre/III+Encontro+T%C3%A9cnico+de+Software+Livre.htm
16/07/2005	2º FireBird Developers Day Local: Piracicaba - SP - Brasil	http://www.FirebirdDevelopersDay.com.br
Até 31/07	Inscrição para Workshop: Mulheres e software livre Recife - Olinda - PE	http://mulheres.softwarelivre.org/news/4290
13/08/2005	BSDDAY São Paulo - SP	http://www.bsdday.com.br/
01 à 05 de Agosto	Encontro Nacional dos Estudantes de Computação Bonito - MS	http://www.enec.org.br/
28/Setembro à 01/Outubro	II Conferência Latino-americana e do Caribe sobre Desenvolvimento e Uso do Software Livre Recife - Olinda - PE	http://twiki.im.ufba.br/bin/view/LacFREE/WebHome
03 à 07 de Outubro	20. Simpósio Brasileiro de Banco de Dados Uberlândia - MG	http://www.sbbd-sbes2005.ufu.br/imagens/ChamadaSBBD.pdf
3,4,5 de Novembro	Conisli – Congresso Internacional de Software Livre São Paulo - SP	http://www.eventosucesusp.org.br/conisli/



Se você sabe de algum evento focando em **Banco de Dados** ou em **Software Livre** que não esteja listado aqui, envie-nos um email com os dados do evento para que possamos incluí-lo.