

Índices, Performance e Estatísticas no IB

Introdução:

Neste artigo veremos o que é um índice, como eles podem nos ajudar na performance com o Banco de Dados InterBase (as dicas sobre índices também servem para a grande maioria dos Banco de Dados do mercado). Após, explico o que são e como entender as Estatísticas que o Banco de Dados InterBase nos mostra, veremos também algumas dicas de performance em um comando **SELECT**, uma pequena introdução a planos de Consulta **PLAN**, como usar o componente **IBStaticalService** da palheta InterBase Admin do Delphi, o uso do GSTAT para recuperar as estatísticas do InterBase via linha de comando.

Performance Lógica "Otimização":

A performance de um Banco de Dados, é muito influenciada pela estrutura e a forma no qual você projetou seu Banco de Dados. Normalização, Tamanho da Página de Dados, Índices, Stored Procedures, Trigger's, são fundamentais para a uma boa performance do seu Banco de Dados, quando bem projetados. Não existe uma regra única para análise dos "componentes" citados acima, cada caso, é um caso!

Performance Física "Otimização":

A normalização citada mais acima, também pode ser quebrada para melhorar a performance física de uma Query, devemos as vezes colocar todos os dados necessários de uma tabela em uma tabela apenas, para não haver a necessidade de pesquisas em outras tabelas (Joins).

Ok, vamos a parte técnica do artigo. Boa Leitura!

InterBase e o uso dos Índices.

A gerência de índices no InterBase é bem diferente da utilizada em outros servidores de banco de dados. O InterBase tem dois tipos de índices: Navegação e Bit Mapped.

Índices de Navegação

Um índice de navegação é percorrido de valor em valor, na ordem em que foi criado. Isso permite que uma query acesse o primeiro elemento de uma tabela muito rapidamente, caso exista um índice para uma coluna no ORDER BY. Veja a query abaixo:

```
SELECT                                emp_no,                                first_name
FROM                                  employee
ORDER BY hire_date;
```

Aqui, caso exista um índice para a coluna Hire_date, a leitura dos registros será bem mais rápida.

Índices Bit Mapped

Um índice bit mapped pode ser usado para se conseguir executar o **JOIN** ou **ORDER BY** em que cada índice é percorrido para verificar os valores antes que todos sejam "juntados" usando operações lógicas de **OR/AND**. Vamos citar o Otimizador de Query.

Otimizador de Query:

No "motor" do Banco de Dados InterBase existe um "componente" que chamamos de "Otimizador de Query's", este otimizador, tem a função de montar o melhor Plano de Consulta para a sua Query, sendo ela um simples:

SELECT Coluna FROM TABELA WHERE Coluna = :COLUNA, ou, uma Query complexa onde são usados vários Join's:

```
SELECT * FROM employee E, departament D, job J
WHERE E.Depto_no = D.Dept_no AND
E.Job_Code = J.Job_Code AND
E.Job_Country = J.Job_Country AND
E.Job_Grade = J.Job_Grade AND
E.Dept_no < 150;
```

"O SELECT montado acima é apenas um exemplo, existem Query's bem mais complexas do que a apresentada neste exemplo, onde podemos usar Agrupamentos, Union's, etc."

O Motor do Banco de Dados InterBase, avaliou a sua Query, combinando os índices, o relacionamento usado, e no final ele verificou se a Otimização está boa ou não. Caso não esteja a contento para o Banco de Dados InterBase "Estatística", o InterBase pode re-avaliar a sua Query, ou simplesmente não fazer nada, montando uma Query "Natural". E para todo este mecanismo, o InterBase usou o nosso: Índice Bit Mapped.

Índices é uma das partes mais importantes quando se fala em otimização em um banco de dados. Mas, atenção: não crie índices aleatoriamente. Cada índice, assim como cada coluna em um servidor de banco de dados SQL, precisa ser analisado antes da criação e ter verificado seu real propósito. Pois, em contra partida, o índice deixa a manutenção na tabela mais lenta.

Por default, nenhum índice é definido quando se cria uma tabela. O InterBase cria um índice automaticamente quando é definido uma Primary Key e quando é definida uma Foreign Key. No caso de uma Primary Key, é criado um índice único; já para uma Foreign Key, é criado um índice único em cima da chave primária e um índice que permite duplicações para a chave estrangeira.

Reconstrução de índices

Depois de muitas alterações na sua base de dados, o seu índice pode ficar desbalanceado. Você pode, e deve, melhorar o balanceamento de seu índice executando uma reconstrução. O procedimento para reconstrução de índices é o seguinte.

O primeiro passo é desativar o índice:

```
ALTER TABLE nome-do-índice INACTIVE;
```

O segundo passo é reativar o índice:

```
ALTER TABLE "nome-do-índice" ACTIVE;
```

Isso deixa o índice balanceado e com sua performance mais rápida.

Estatísticas

Após várias alterações do tipo inserção/atualização em uma base de dados, onde um determinado índice atua, o InterBase recalcula a seletividade do índice "Lembram da explicação do índice Bit Mapped ?!". Isto é, ele recalcula os valores estatísticos internos indicando qual o menor caminho para efetuar uma determinada query.

Para forçar o InterBase a realizar esta operação, use o seguinte comando:

```
SET STATISTICS INDEX "nome-do-índice";
```

Como o servidor vai avaliar cada índice existente para a coluna em questão, tome cuidado para não criar índices em colunas que já possuam índices. Nesse caso, ocorreria queda de performance.

Dicas para eficiência dos índices:

1. Crie índices para todas as colunas que participem de ORDER BY e JOINS.
2. Reconstrua os índices periodicamente, com ALTER INDEX.
3. Lembre-se que índices são ordenados de forma ascendente e ou descendente. Um índice só será usado apenas se a ordenação no ORDER BY for a mesma do índice.
4. Índices prejudicam a performance em operações de atualização de dados (INSERT, UPDATE, DELETE) porque são atualizados pelo banco durante essas operações. Nesse caso, você pode desativar índices antes de realizar muitas atualizações e ativá-los novamente depois*.
5. Em SELECTs, use sempre a seguinte ordenação para as colunas:
 - A. Colunas SmallInt,
 - B. Colunas Integer;
 - C. Colunas Data;
 - D. Colunas String;

Seguindo esta ordem, seu SELECT será mais rápido.

6. Faça Backups e Restores periódicos. Alguns dos benefícios são:

Reconstrução dos índices

Eliminação de versões antigas de registros

Desfragmentação das páginas de dados

Reescrita das páginas de uma tabela em uma área contínua (desfragmentação)

Proteção dos seus dados contra imprevistos

Ao executar um Backup, execute o shutdown no servidor! O Backup pode ser feito com o InterBase em uso, mas como o backup, na realidade, é uma transação à parte, ele enxergará apenas os registros como existiam no momento que o backup iniciou.

* Não é possível desativar e ou ativar um índice em Stored Procedure's e ou Trigger, pois, não podemos usar comandos DDL (Data Definition Language) em Stored Procedure's e ou Trigger's. Para usarmos este mecanismo, precisamos fazê-lo pelo Delphi.

Estatísticas e Performance

Outra dica relacionada à performance é analisar as estatísticas mantidas pelo banco. Nessa análise, podemos ver como está a média de preenchimento do banco (Average Fill), o número de páginas de dados (Data Pages) e o valor de profundidade de árvore (Depth).

Para começar, usaremos o IBConsole para obter estatísticas do InterBase, com o banco de exemplo employee.gdb. Abra o IBConsole e clique com o botão direito em cima do Alias para employee.gdb e escolha Database Statistics (veja a Figura 1).

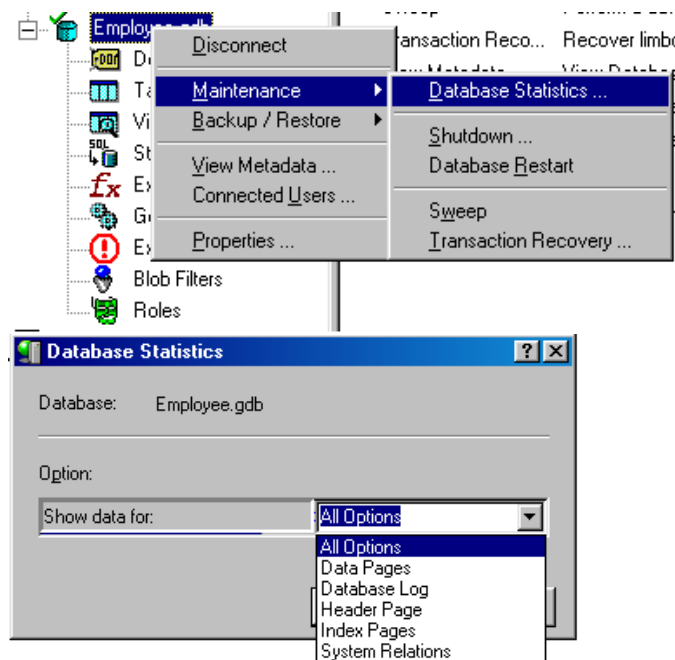


Figura 1. Acesso a estatísticas do banco no IBConsole

Na janela Database Statistics, escolha "All Options". Com isso, o InterBase mostra todas as estatísticas do banco, como a seguir:

Service started at 16/10/2001 17:43:47

Database "C:\Arquivos de programas\Firebird\Examples\v5\Employee.gdb"

```
(...)  
Analyzing          database          pages          ...  
COUNTRY          (128)  
Data    pages:    1,    data    page    slots:    1,    average    fill:    15%  
(...)  
  
Index          RDB$PRIMARY1          (0)  
Depth:        1,    leaf    buckets:    1,    nodes:    14  
(...)  
  
EMPLOYEE          (131)  
Data    pages:    2,    data    page    slots:    2,    average    fill:    42%  
(...)  
  
Index          NAMEX          (1)  
Depth:        1,    leaf    buckets:    1,    nodes:    42  
(...);
```

Informações sobre tabelas

A listagem começa com informações gerais sobre a base de dados (arquivo, data de criação, tamanho de página, Shadow's, OAT - Oldest Active Transaction, versão do ODS e outras). Logo depois, vêm as informações sobre as tabelas e índices, ordenadas alfabeticamente pelo nome da tabela.

Uma das informações sobre tabelas é o número de Data Pages e o Average Fill, importantes para as tabelas maiores ou mais acessadas na base. Se o valor de Average Fill for menor que 60%, devemos fazer um Backup e um Restore para tentar diminuir a fragmentação. Se a fragmentação não for reduzida, uma solução é aumentar o tamanho da página (a última versão do FireBird, uma versão open source do InterBase, já aceita páginas maiores que 8192).

Informações sobre índices

O InterBase utiliza uma variação de Árvore-B para índices, com um valor de Depth que normalmente deve estar abaixo de 4. Se não estiver, você deve reconstruir o índice. Novamente, se a profundidade não diminuir, você deve aumentar o tamanho da página de dados. Nodes indica o número de valores existentes no índice.

Plano de Consulta – PLAN's

Uma pequena introdução a Planos de Consulta, pois, o mesmo também está relacionado a Performance de um Banco de Dados InterBase.

PLAN é: A forma que o InterBase determina como será usado os índices para a recuperação das informações das tabelas envolvidas no SELECT. A explicação de índices Bit Mapped vai de encontro com boa parte da explicação sobre Plan´s!

Um bom artigo sobre PLAN´s é encontrado no seguinte link:

http://www.ibphoenix.com/ibp_quep.html

IBStatisticalService

O componente IBStatisticalService, na paleta Interbase Admin é responsável pelo retorno das estatísticas de bancos InterBase. Vamos dar uma olhada nas suas principais propriedades e métodos.

A propriedade Active conecta o componente à base de dados. DatabaseName define o caminho da base. A propriedade Protocol define o protocolo usado para conexão e ServerName especifica o nome ou o IP do servidor.

A propriedade Options determina quais estatísticas serão extraídas. Veja a seguir a definição do tipo da propriedade Options:

```
type TStatOption = (DataPages, DbLog, HeaderPages, IndexPages, SystemRelations);
```

Agora veja o significado de cada opção:

- DataPages: Recupera as estatísticas de dados das tabelas
- DbLog: Interrompe as estatísticas após retornar informações de Log
- HeaderPages: Interrompe as estatísticas após retornar informações de cabeçalhos das páginas de dados
- IndexPages: Recupera estatísticas de índices
- SystemRelations: Recupera estatísticas de tabelas de sistema

Um dos métodos mais importantes é GetNextLine, que retorna a linha da estatística e pula pra o próximo registro da estatística, e ServiceStart, usado para iniciar o serviço.

Aplicação

Vamos criar a nossa aplicação de Estatística via Delphi. Inicie o Delphi e abra um projeto novo. Acrescente um Button, um SpeedButton, um Edit, um OpenDialog e um IBStatisticalService. Sua tela deverá ficar parecida com a mostrada na Figura 2.

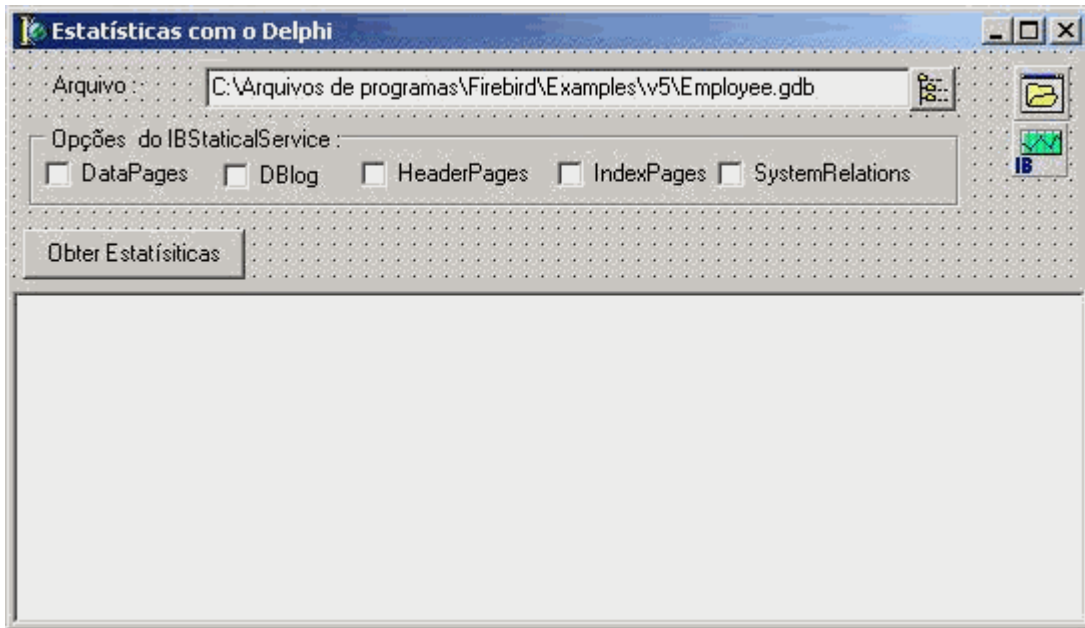


Figura 2. Exemplo para obtenção de estatísticas no Delphi

Nota: A palheta InterBase Admin do Delphi, só aparece se você tiver o InterBase instalado na sua maquina

Dê um duplo clique no componente IBStatisticalService para mostrar uma tela de configurações para o serviço. Clique em browse para escolher o banco employee.gdb e configure as opções de conexão como na Figura 3.

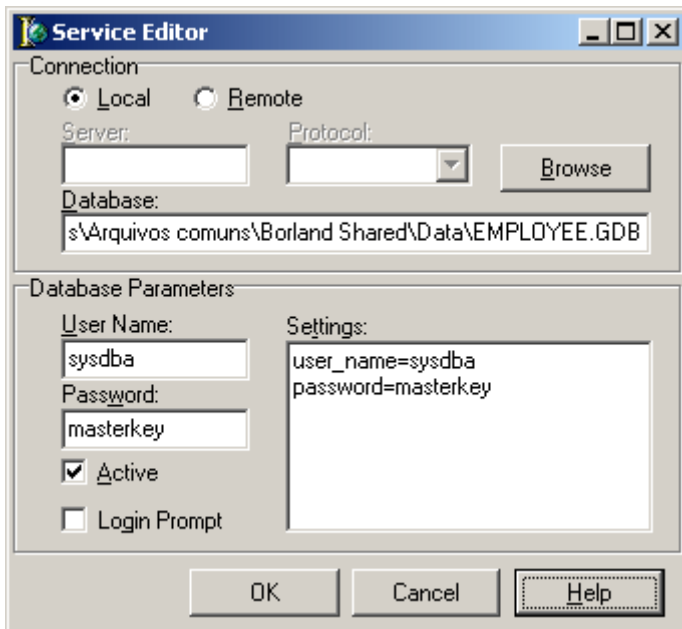


Figura 3. Configuração do componente IBStatisticalService

Para o botão "Obter Estatísticas", crie o seguinte código:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  with IBStstclSrvcExemplo do
    try
      { O DataBase escolhido. }
      DatabaseName := EdtGDB.Text;

      { Não pede Login. }
      LoginPrompt := False;
      { Um IF Simples para testarmos      qual a opção escolhida para mostrar
as
estatísticas com o componente      IBStatisticalService.
}

      if ChkBxDataPages.Checked then
        Options := [DataPages];

      if ChkBxDBLog.Checked then
        Options := [DbLog];

      if ChkBxHeaderPages.Checked then
        Options := [HeaderPages];

      if ChkBxIndexPages.Checked then
        Options := [IndexPages];

      if ChkBxSystemRelations.Checked then
        Options := [SystemRelations];

      { Inicia o Serviço. }
      ServiceStart;

      while not Eof do
        MmEstatisticas.Lines.Add(GetNextLine);
        { GetNextLine, além de      ir para próxima linha, obtem a linha
corrente
antes de saltar a mesma. Por      isso a atribuição direta no
Lines.Add()
do Memo.
}
      finally
        { Foi executado até fim      e ou por um acaso ocorreu um erro, se
desconecta
do Banco.
}
        Active := False;
      end;
end;
```

E, por último, o evento para abrir o OpenFileDialog e escolher o banco InterBase desejado.

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
```

```
if OpnDlgGDB.Execute then EdtGDB.Text := OpnDlgGDB.FileName;  
end;
```

Após o código ter sido corretamente, podemos fazer vários testes com a propriedade Options do IBStatisticalService e vermos o que as mesmas nos retornam!

Estatísticas com GSTAT

GSTAT é um utilitário que acompanha o InterBase/FireBird, usado para a obtenção de estatísticas via linha de comando. É muito útil no Linux. Veja sua sintaxe básica:

```
GSTAT [opcoes] GDB
```

As seguintes opções podem ser usadas:

- a Analise das Páginas de Dados;
- d Analise única e exclusivamente das Páginas de Dados;
- h Informações de Cabeçalho;
- i Analise de Índices;
- s Analise da System Table;
- z Informações de versão do Banco;

Veja um exemplo de uso:

```
GSTAT -h -i -a C:\Arquivos de programas\Firebird\Examples\v5\Employee.GDB
```

Conclusões

A performance de um banco de dados está ligada diretamente a uma série de fatores, que vão desde a rede física, ao tamanho correto da página de dados, normalização, Otimização e os não menos importantes índices e a forma de escrita do SELECT.

Autor :
Anderson Haertel Rodrigues (AHR) é administrador de bancos de dados, desenvolvedor de sistemas e componentes. Mora em Florianópolis/SC e pode ser contactado em anderson.hr@bol.com.br.