

# BACKUPS COM INTERBASE 2007 – MUITO MAIS QUE SEGURANÇA

Olá pessoal,

Backups! Todos nós sabemos a importância deles. Mas será que estamos dando a devida atenção para nossos processos de backup? Podemos garantir para nossos clientes que seus backups estão íntegros? Estamos utilizando as ferramentas certas? Estamos explorando todas as possibilidades? Para quem utiliza bancos de dados InterBase 2007, realizar backups pode ser muito mais do que garantir uma recuperação de perda de dados.

Nesse artigo quero mostrar detalhadamente os recursos de backup do InterBase 2007 e como eles podem nos ajudar não somente com cópias, mas também na melhoria de performance e disponibilidade de informações. Tudo isso utilizando tanto as ferramentas gráficas, quanto linhas de comando. A ideia é rever nossos conceitos de backup e melhorar nossas rotinas, garantindo ao cliente um ambiente o mais seguro possível.

## TIPOS DE BACKUP

O InterBase 2007 disponibiliza dois tipos de backup: *lógicos e físicos*. Os backups lógicos são importantes principalmente em processos de manutenção do banco de dados. Os backups físicos são os mais adequados para cópias de segurança.

O importante é: sempre executar nossos processos de backup utilizando um dos dois tipos de backup disponíveis. Não podemos jamais executar uma cópia simples (copiar / colar) do banco de dados. O copiar / colar pode facilmente gerar corrupções no banco de dados de produção que, com o tempo, podem causar um erro mais sério.

Vamos detalhar cada um dos tipos de backup.

## BACKUPS LÓGICOS

Quando executamos um processo de backup completo, utilizando o conhecido comando *gbak*, estamos realizando um backup lógico. Esse backup extrai cada registro do banco de dados e armazena em um formato diferente. Assim, o destino do backup não é uma cópia do banco de dados original. Apesar de ser muito utilizado como o único processo de backup, os backups lógicos são mais úteis para manutenção do banco de dados. Com backups lógicos podemos:

Realizar uma limpeza de registros obsoletos do arquivo de banco de dados;

- Reorganizar a estrutura de índices;
- Desfragmentar os blocos de dados;
- Reiniciar os controles de sistema;
- Extrair estrutura de dados e dados dos bancos de dados;
- Alterar a estrutura de paginação do banco de dados;
- Dividir o banco de dados em múltiplos arquivos;
- Transportar o banco de dados entre plataformas distintas;
- Melhor a performance geral do banco de dados.

Para realizarmos backups lógicos, podemos utilizar o utilitário de linha de comando *gbak* ou mesmo o console gráfico *IBConsole*. Por linha de comando podemos preparar rotinas que podem ser agendadas pelos gerenciadores de tarefas dos sistemas operacionais, automatizando o processo. Abaixo um exemplo de execução do comando *gbak* para backup com vários parâmetros:

```
c:\databases> gbak -b -g -ig -t -y backup_log.txt -user sysdba -pass masterkey banco.ib banco.ibk
```

Onde:

*-b*: Indica que estamos iniciando um processo de backup do banco de dados para um arquivo destino.

*-g*: Indica que durante o processo de backup, o sistema de garbage collector não será realizado. Assim, a execução do backup é feita mais rapidamente.

`-ig`: Indica que qualquer erro de checksum será ignorado durante o processo de backup. Essa flag é muito útil para garantirmos que o backup será realizado sem erro. Se durante o backup aparecer alguma mensagem de erro devido à uma corrupção de checksum, realize o backup utilizando esse flag para que o `gbak` ignore o erro e possa corrigi-lo durante o processo de restauração backup.

`-t`: Indica que o backup será gravado em um formato *transportável*. Essa flag é fundamental quando precisamos mudar de plataforma. Se nosso banco de dados está em uma máquina com sistema operacional Windows e queremos levá-la para um servidor InterBase 2007 rodando em uma máquina Linux por exemplo, precisamos realizar o processo de backup e restauração do banco de dados utilizando a flag `-t`.

`-y`: Indica um arquivo destino para o log da execução do backup. Essa flag é muito importante quando criamos rotinas de backups.

`-user e -pass`: Indica o usuário e a senha para realizar o backup.

O comando `gbak` pode ser executado mesmo com conexões ativas no banco de dados. Todos os dados consistentes no início do processo serão copiados. O arquivo resultante (`banco.ibk`) serve como nossa cópia de segurança. Mas, como já dito, o backup lógico é ideal para manutenção do banco de dados. Assim, devemos programar restaurações periódicas de nossos backups para garantir um arquivo mais limpo, sem fragmentação, com estruturas balanceadas e com tamanho menor. Abaixo um exemplo de comando `gbak` para restauração do banco de dados:

```
c:\databases> gbak -c -p 8192 -pr 62500 -r -y restore_log.txt -user sysdba -pass masterkey banco.ibk banco.ib
```

Onde:

`-c`: Indica que estamos iniciando o processo de restauração do backup do banco de dados.

`-p 8192`: Indica o tamanho de página do banco de dados (*Database Page Size*). O InterBase 2007 suporta os seguintes tamanhos: 1024, 2048, 4096, 8192, 16384, expressos em bytes. Páginas maiores permitem que mais informações sejam transferidas para memória cache no momento da leitura de dados. Hoje em dia, com o acesso cada vez mais fácil a recursos de hardware, principalmente memória RAM, podemos utilizar páginas de 8 Kb como padrão em nossos bancos de dados. Vale a pena testar caso a caso e perceber a diferença de performance com a mudança. Páginas de 1 e 2 Kb não fazem mais sentido, pois teríamos páginas menores do que as do próprio sistema operacional, o que poderia causar uma queda na performance.

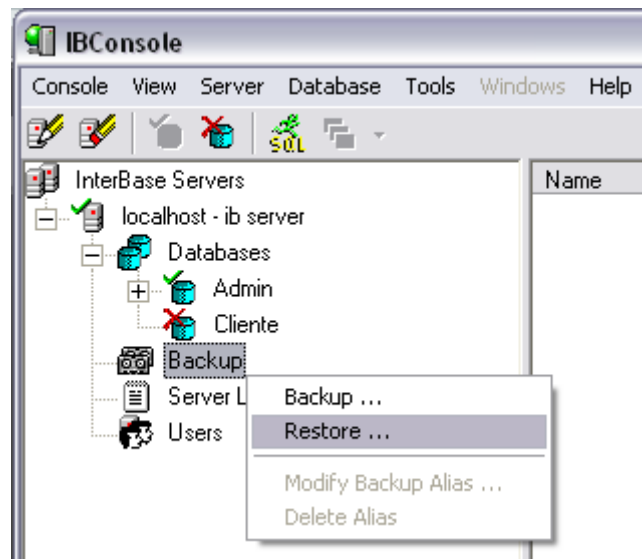
`-pr 62500`: Essa é uma nova flag que permite uma pré alocação do arquivo de banco de dados. Indica dizer, nesse exemplo, que o arquivo restaurado já terá 62500 páginas alocadas (500 Mb se utilizarmos páginas de 8Kb). Ou seja, mesmo que o banco de dados seja menor que 500 Mb, o arquivo já será criado com esse tamanho. Esse recurso é interessante, pois podemos dimensionar o tamanho de nossos bancos de dados, conforme uma previsão de crescimento dos mesmos. É um parâmetro opcional. Se não for informado, o arquivo será criado até o tamanho real do mesmo.

`-r`: Indica que o arquivo destino (`banco.ib`) será substituído durante a restauração do backup. Não podemos ter nenhuma conexão ativa no momento dessa restauração.

`-y`: Indica um arquivo destino para o log da execução do backup. Essa flag é muito importante quando criamos rotinas de backups.

`-user e -pass`: Indica o usuário e a senha para realizar o backup.

Podemos utilizar o console *IBConsole* para executar os processos de backups lógicos também, conforme figura abaixo:



## BACKUPS FÍSICOS

O backup físico do InterBase 2007, também conhecido como *Online Dump*, copia o banco de dados a nível de página e salva as informações em um arquivo no formato original. Significa que teremos uma réplica exata do banco de dados original ao final do processo. Essa réplica é, por padrão, criada no formato *read-only*. Os backups físicos são os ideais para cópias de segurança.

Os backups físicos podem ser criados não apenas nos discos do próprio servidor InterBase 2007, mas também em qualquer unidade de rede mapeada no servidor. Com isso podemos preparar rotinas para gerar backups automaticamente em diversos computadores de nossa estrutura, até mesmo remotamente através de uma conexão de rede segura.

Uma das grandes vantagens dos backups físicos é a velocidade do processo. O backup físico analisa os dados consistentes no início do processo e dispara uma ação de cópia página a página do banco de dados original. Esse processo entra na fila de transações do servidor InterBase 2007, que consegue administrar de forma que o *Online Dump* não penalize a performance dos outros processos concorrentes. Enquanto um processo de backup lógico (*gbak -b*) demora cerca de 30 minutos para ser executado em um banco de dados de 2 gigabytes, o backup físico não demora mais do que 5 minutos concluir sua cópia.

Com os backups físicos podemos:

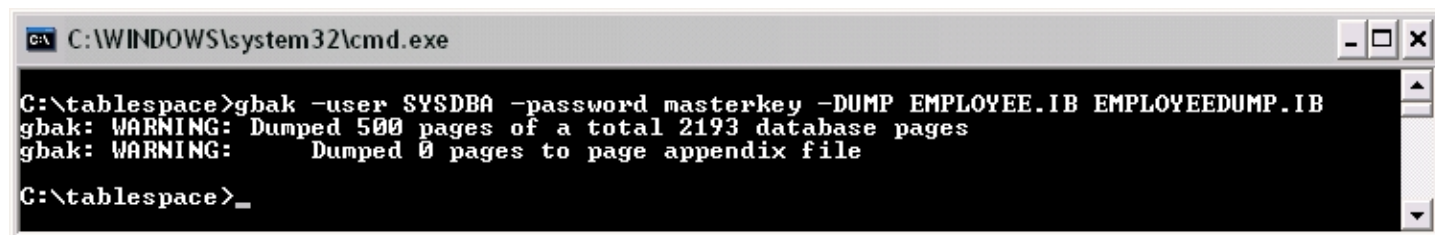
- Copiar rapidamente nossos bancos de dados em produção;
- Utilizar as cópias para validar os bancos de dados de produção sem a necessidade de retirar os usuários dos sistemas;
- Distribuir nossos bancos de dados remotamente;
- Alterar o caminho em bancos de dados multi-files de uma forma mais rápida do que pelo processo de backup lógico;
- Disparar vários backups ao longo do dia sem penalizar a performance;
- Atualizar nossos backups com o recurso de *backups incrementais*.

Da mesma forma que com os backups lógicos, utilizamos o utilitário de linha de comando *gbak* para realizarmos os backups físicos, ou mesmo podemos usar o console gráfico *IBConsole*. Vejamos como fica o comando *gbak*:

```
gbak -d -user sysdba -pass masterkey banco.ib banco_dump.ib
```

Em uma sintaxe mais simples, apenas alteramos a flag para *-d(ump)*. Assim o comando *gbak* irá *dumpear* o banco de dados *banco.ib* para a réplica *banco\_dump.ib*. Reparem que a extensão propositadamente permanece a mesma (.ib), para mostrar que o banco destino está no mesmo formato do banco original. Assim, o processo de restauração do backup praticamente não existe. O arquivo *banco\_dump.ib* já pode ser lido normalmente. Lembrando apenas que esse arquivo está no formato *read only*.

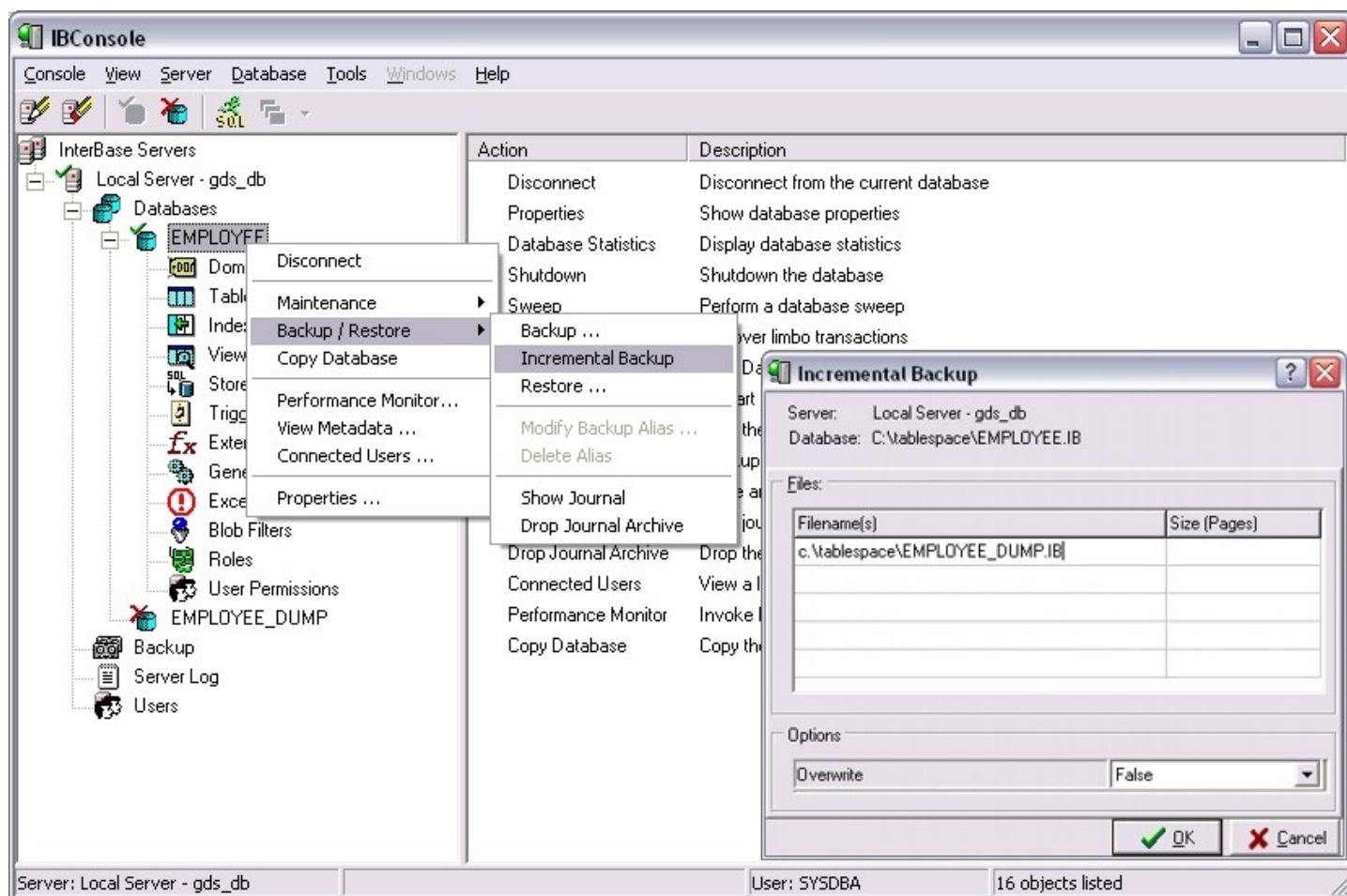
O formato *read only* é essencial para que o backup incremental possa ser executado. Isso por que se executarmos novamente o mesmo comando, o *backup físico* irá copiar apenas as páginas de dados alteradas desde o último backup. Para isso, o banco de dados de origem é totalmente *scaneado (full scan)* para identificar todas as páginas alteradas. A figura abaixo mostra que apenas um total de 500 páginas foram copiadas de um total de 2193 páginas. Essas 500 páginas foram as que sofreram alteração desde o último backup. Se o arquivo destino não estiver no formato *read only*, não haveria como garantir uma cópia consistente dos dados.



```
C:\WINDOWS\system32\cmd.exe
C:\tablespace>gbak -user SYSDBA -password masterkey -DUMP EMPLOYEE.IB EMPLOYEE_DUMP.IB
gbak: WARNING: Dumped 500 pages of a total 2193 database pages
gbak: WARNING: Dumped 0 pages to page appendix file
C:\tablespace>_
```

Esse processo torna o backup físico ainda mais rápido. Imaginem preparar um backup de um banco de dados muito grande. Copiar ele inteiro a cada backup tornaria o backup muito lento. Dessa forma, com o backup incremental, a atualização da primeira cópia garante uma performance melhor. Significa dizer que podemos, por exemplo, gerar um backup de nossos bancos de dados a cada uma hora, ao invés de depender apenas de um único backup noturno.

A imagem abaixo mostra o caminho para executarmos o processo de *backup físico* utilizando a ferramenta *IBConsole*:



Caso desejarmos substituir o arquivo destino do backup físico, podemos acrescentar a flag *-ov (overwrite)*

```
gbak -d -ov -user sysdba -pass masterkey banco.ib banco_dump.ib
```

É possível preparar e rodar vários backups físicos de uma mesma origem simultaneamente, mas a performance ficará comprometida. Muito importante notar também que, por ser uma cópia fiel do banco de dados de origem, o arquivo *dumpeado* não é transportável. Ou seja, não podemos utilizar a cópia para levar o banco de dados para outro servidor InterBase 2007 com outro sistema operacional. Se desejarmos fazer esse processo, devemos utilizar o *backup lógico* com o parâmetro *-t* ativado.

A restauração de um backup físico é praticamente instantânea. Precisamos apenas alterar o formato para *read write*. Para isso utilizamos a sintaxe abaixo:

```
gbak -mo read_write -user sysdba -pass masterkey banco_dump.ib
```

Uma vez transformado para o formato *read write*, o arquivos *banco\_dump.ib* não poderá mais receber informações de um *backup incremental*. Precisaremos criar outra cópia *full* para iniciamos o processo novamente.

*Backups lógicos* ou *backups físicos*, cada um com sua função e facilidade. O importante é sempre criarmos as rotinas em nossos clientes, evitando qualquer eventual falha que possa comprometer nosso trabalho. Lembrando que com o InterBase 2007 temos: *backup lógico* = manutenção periódica e melhoria de performance; *backups físicos* = cópias de segurança a qualquer momento. Assim são as ferramentas de backup do InterBase 2007: muito mais do que segurança.

Referência:

**InterBase 2007 Operations Guide – cap. 8.**

Felipe Santos  
felipe.santos@presence.com.br

Felipe Santos é especialista em InterBase e redes. Trabalha com o InterBase há mais de 7 anos e em ambientes Linux há mais de 9 anos. Administrador de sistemas ERP de grande porte, com soluções InterBase rodando em Linux em mais de 100 empresas. Especialista em ambientes críticos. Atuou em um dos maiores casos do InterBase no Brasil no ano de 2005. Participante ativo na comunidade, com artigos publicados. Palestrante em eventos como IB Tour e Borcon Conference. Instrutor em treinamentos avançados de InterBase. Atualmente trabalhando na área comercial / técnica do InterBase na Presence Tecnologia - distribuidora do produto no Brasil.