

Reparación y mantenimiento de la base de datos

Esta nota técnica intenta explicar el proceso de reparación de una base de datos Firebird, como también algunos puntos a tener en cuenta para mantener la base y prevenir posibles roturas.

Importante: Los temas tocados en este documento son muy sensibles. Ante un problema en la base de datos se recomienda comunicarse de forma urgente con nuestro soporte técnico.

Los procesos de reparación se realizan con comandos básicos de Firebird. Estos están disponibles tanto en Windows como en Linux. El proceso de reparación se debe hacer a nivel local (en el mismo servidor).

Si se rompen los índices de la base, Firebird sigue trabajando sin ellos (lo que hace extremadamente lento el acceso). Una lentitud notoria en el acceso a los datos puede ser un indicio de problemas en la base de datos.

Por otro lado, durante el proceso de reparación se vuelcan todos los datos a un archivo de backup desde el cual luego se regenera, por lo que el proceso también puede servir para realizar un mantenimiento de la base de datos (eliminación de datos temporales conocidos como **garbage**, regeneración de los índices, etc.).

Como primer punto, siempre antes de comenzar a trabajar sobre la base se debe asegurar que no está en uso, cerrando el sistema LabWin en todas las terminales y ejecutando en el servidor el comando **gfix** del siguiente modo:

gfix -user SYSDBA -password masterkey -shut -force <archivo_FDB>

Nota: Cabe aclarar que los comandos **gbak** y **gfix** deben invocarse desde la carpeta c:\archivos de programa\firebird\firebird_2_1\bin (ó ...\\firebird_2_5\bin, dependiendo de la versión de Firebird instalada).

Ante una base de datos rota (*internal gds consistency error*) o duda de rotura (lentitud extrema) lo ideal es intentar realizar un *dump* de los datos (volcarlos hacia un archivo de backup) mediante el comando **gbak**:

gbak -user SYSDBA -password masterkey -mode read_only -b <archivo_FDB> <archivo_FBK>

Suponiendo que el sistema se halle en la ruta C:\Labwin3 y hallamos creado una carpeta SOS en la raíz del disco para alojar el backup, el comando sería:

gbak -user SYSDBA -password masterkey -mode read_only -b c:\labwin3\basedat\basedat.fdb c:\sos\basedat.fbk

Si recibimos algún error que impida que se realice el proceso debemos proceder a ejecutar el comando **gfix**. Este prepara (soluciona los problemas que impiden el paso anterior) el archivo FDB para luego poder realizar el proceso de volcado.

gfix -user SYSDBA -password masterkey -mend -full -ignore <archivo_FDB>

Con las rutas expresadas en el ejemplo anterior sería:

gfix -user SYSDBA -password masterkey -mend -full -ignore c:\labwin3\basedat\basedat.fdb

Una vez finalizado este proceso se deberá ejecutar el comando **gbak** comentado anteriormente.

Una vez finalizado el volcado de datos, debemos volver a regenerar la base (archivo FDB) a partir del backup realizado (archivo FBK), para ello también se utiliza el comando **gbak** con los siguientes parámetros:

```
gbak -user SYSDBA -password masterkey -c -k <archivo_FBK> <archivo_FDB>
```

Lo que podría ser, siguiendo los ejemplos anteriores:

```
gbak -user SYSDBA -password masterkey -c -k c:\sos\basedat.fbk c:\sos\basedat.fdb
```

Una vez realizado este proceso obtendríamos un nuevo archivo FDB en la carpeta C:\SOS. Lo único que restaría es reemplazar el archivo original roto (situado en c:\labwin3\basedat\) por el nuevo (se recomienda no eliminar el archivo original, sino renombrarlo y moverlo a otra carpeta).

Como punto adicional, en este último proceso se podría personalizar el tamaño de pagina y buffers. Esto es recomendable en bases de datos grandes (laboratorios con mucho volumen de trabajo diario y muchas terminales conectadas al servidor). Esto se hace con los parámetros **-page_size <tamaño>** y **-buffers <tamaño>**. Por ej.:

```
gbak -user SYSDBA -password masterkey -page_size 8192 -buffers 8192 -c -k c:\sos\basedat.fbk c:\sos\basedat.fdb
```

Cabe aclarar que durante este proceso se regeneran los índices de la base y descarta toda la “basura” que puede generarse por transacciones incompletas o canceladas, por lo que más allá de posibles roturas, siempre viene bien dado que realiza una optimización que se verá reflejada en la velocidad de acceso a los datos y tamaño del archivo FDB (no se asuste si ve que el tamaño del FDB generado durante el proceso es más chico que el del original, esto es algo normal y esperable). Durante el proceso de recolección de basura *runtime* se elimina el *garbage* pero dicho espacio quedará en blanco, sin alterar el tamaño de la base (archivo .FDB).

Garbage collection, Housekeeping y el comando gstat

En toda base de datos, ante un proceso de transacción se generan registros “copia” de los afectados por dicha transacción. Algunos (transacciones terminadas con **commit**) luego se eliminan, pero otros (transacciones terminadas con **rollback**) quedan eternamente hasta que se realiza un proceso denominado **garbage collection** (recolección de basura) que recorre la base eliminando dichos registros. El detonante de este proceso (es decir, lo que hace que el proceso se corra) se denomina **sweep interval** y deriva de la diferencia entre la última transacción válida y la próxima transacción. Para consultar dicho valor puede ejecutar el comando **gstat** del siguiente modo:

```
gstat -user SYSDBA -password masterkey -h <archivo_FDB>
```

Ver el siguiente ejemplo, que produce la salida de la imagen de abajo:

```
gstat -user SYSDBA -password masterkey -h c:\labwin3\basedat\basedat.fdb
```

Aquí el valor a analizar es el resultado de **Next transaction – Oldest transaction**. En este caso sería:
 $9.139.866 - 9.139.864 = 2$.

Ver también que en este caso el **Sweep interval** se halla en 10.000, lo que quiere decir que cuando dicha diferencia alcance o supere el valor 10.000 se disparará la recolección de basura.

Cabe aclarar que mientras se realiza el proceso de recolección de basura la performance de la base decae. Por otro lado, si se desactiva el proceso (estableciendo un **Sweep interval** de valor 0) tarde o temprano se debe realizar un proceso de dump para regenerar la base y durante dicho proceso eliminar la basura (dado que el proceso de recolección nunca se ejecutará). Lo ideal es encontrar el valor apropiado para el volumen de trabajo del laboratorio específico, logrando el balance justo para que no se acumule mucha basura ni se ejecute la recolección a cada rato. El valor por defecto es 10.000 pero en laboratorios de gran volumen de trabajo se puede establecer entre 30.000 y 50.000 transacciones.

El ajuste de dicho valor se realiza mediante el comando **gfix**, ejecutado con los siguientes parámetros:

gfix -user SYSDBA -password masterkey -h <valor> <archivo_FDB>

por ejemplo:

gfix -user SYSDBA -password masterkey -h 30000 c:\labwin3\basedat\basedat.fdb

También con el comando **gfix** podemos forzar que se lance el proceso de recolección de basura, ejecutando:

gfix -user SYSDBA -password masterkey -s <archivo_FDB>

Escritura forzada

Hay un parámetro que nos permite seleccionar entre dos posibles modos de trabajo: escritura sincrónica y asincrónica (cache en memoria RAM).

En el primer caso, todo lo que va a la base de datos se graba inmediatamente en el archivo FDB, lo que hace que haya muchas más interacciones I/O con el disco, influyendo directamente sobre la performance general.

En el segundo caso los datos “calientes” se hallan en un espacio intermedio dentro de la memoria RAM (cache) logrando así menos interacciones I/O con el disco y mejorando notablemente la performance.

La ventaja del primer caso es que agrega seguridad a los datos mientras que el segundo ante cualquier anomalía extraña (muy poco probable) podría llegar a perder los datos almacenados en la memoria sin llegar a volcarlos al archivo.

Para cambiar este modo de trabajo se utiliza el comando **gfix** del siguiente modo:

gfix -user SYSDBA -password masterkey -write <sync/async> <archivo_FDB>

Como se imaginarán **-write sync** habilita la escritura forzada mientras que **-write async** la deshabilita.

Si el volumen de trabajo es alto la escritura forzada habilitada puede derivar en un serio problema de velocidad de acceso a la base, dejando el sistema prácticamente inutilizable.

Si utiliza Windows no corre grandes riesgos al utilizar la escritura asincrónica. Ante cortes de luz o bajones de tensión la base suele romperse, pero al regenerarla no hay pérdida de datos. En Linux, Firebird puede mantener el trabajo de medio día en cache. Ante un corte de luz la base no suele romperse, pero si no llega a volcarse la cache al archivo los datos que estaban en RAM se pierden.

Recomendaciones importantes

A continuación se brindan una serie de consejos importantes para prevenir roturas de la base como así también para evitar trastornos a la hora de recuperar los datos.

- Realizar backup (copia de seguridad) del sistema y pacientes periódicamente. Si el volumen de trabajo es alto se recomienda dos veces al día. En ciertos casos (poco frecuentes) la rotura de la base es tan profunda que los procesos de reparación no dan resultados. En varios casos se tarda menos tiempo cargando los últimos pacientes sobre el backup que intentando reparar la base de datos rota. Tómese un tiempo para leer nuestra nota técnica sobre respaldo de los datos y diseñar una buena política de respaldos. Al final de cuentas los datos de sus pacientes es lo único que no posee precio dentro del laboratorio.
- Si su versión de LabWin es anterior a 3.1.392 tenga en cuenta que el backup del día sobrescribe el del día anterior. La mayoría de las veces la base de datos se corrompe ante cortes de luz. Ante un evento de este tipo trate de cambiar el nombre a la carpeta de backup existente. De este modo, si la base sufrió daño (muchas veces los síntomas salen a la vista al otro día) no corre riesgo de pisar el backup sano con uno roto. Una buena política es realizar el backup en distintas terminales según el día de la semana. A partir de la versión 3.1.392 el sistema sobrescribe el backup recién a la semana (una carpeta para cada día de la semana).
- Tratar de evitar manipular el archivo FDB con los métodos de Windows (cortar, copiar y pegar), especialmente cuando la base de datos se encuentra en uso. Si por algún motivo debe moverlo de carpeta cierre el sistema en todas las terminales y ejecute en el servidor el comando **gfix** con los parámetros **-shut -force** para forzar el cierre de la base.
- Conecte el servidor del sistema a una UPS. Ante una falla en el suministro eléctrico le dará un tiempo para apagar el servidor, previniendo así que se corrompa la base de datos. Si el servidor es Linux (hay una nota técnica que complementa este documento para tales casos) tenga en cuenta que ante un corte de luz, los datos que se encontraban en la memoria intermedia (cache) puede que no lleguen a volcarse al archivo FDB, por lo que se perderían. Puede que en estos casos se tiene de utilizar escritura forzada (**-write sync**) para evitar posibles pérdidas pero tenga en cuenta todas las altas y modificaciones se realizaran directamente sobre el disco, lo que hará que el rendimiento gral. decaiga mucho. Hágalo solo si el servidor posee un disco de estado sólido SSD.