Log de transacciones SQL Server (.ldf)

Publicado por [ownerdba](https://dbasqlserver.wordpress.com/author/ownerdba/" \o "Ver todas las entradas de ownerdba) el [26 de marzo de 2012](https://dbasqlserver.wordpress.com/2012/03/26/log-de-transacciones-sql-server-ldf/) en [Errores en Base de Datos](https://dbasqlserver.wordpress.com/category/errores-en-base-de-datos/), [Performance](https://dbasqlserver.wordpress.com/category/performance-2/), [Tips y trucos SQL Server](https://dbasqlserver.wordpress.com/category/tips-y-trucos-sql-server/)

Les dejo este post que contiene todo lo que requerimos saber acerca del log de transacciones en SQL Server, que hacer si se llena?, que estados puede tener?, como respaldarlo?, compatibilidad con modelos de recuperación? y mas.

Cada base de datos contiene al menos un archivo de datos y un archivo de registro de transacciones. SQL Server almacena los datos físicamente en el archivo de datos (.mdf y .ndf). El archivo de transacciones (.ldf) almacena los detalles de todas las modificaciones que se realizan sobre la base de datos de SQL **Server**.

La escritura en el Log de transacciones es secuencial, y esta optimizado para ello. Se podría decir que (por norma general) carece de sentido crear más de un fichero de log de transacciones. Aunque el algoritmo de escritura en los .ldf es algo más complejo: si tuviéramos más de un fichero, la escritura la haría formando un bucle circular pasando por cada uno de ellos, respetando la secuencialidad en las **transacciones**. A diferencia de los ficheros de datos, donde si es posible mejorar el rendimiento de una base de datos, aumentado su número.

Es aconsejable ubicar el fichero del log de **transacciones** en diferente disco donde se encuentren los ficheros de datos.

Modelo de recuperación del log de **transacciones** en una Base de datos

El objetivo de este punto, no es explicar los procesos de backup y restore, sino hacer un resumen de los distintos estados en que se pueden configurar el log de **transacciones**.

El modelo de recuperación de una base de datos puede cambiarse en cualquier momento. No es frecuente cambiar de modelo de recuperación. Tenemos tres modos de configurar el log de transacciones: [Simple](http://adnetwork.fullservices.com.ar/Simple/b084df13eff6/), Full (completo) y, bulk-logged (recuperación optimizado para cargas masivas de registros).

***Modelo de recuperación Simple:***

Sin necesidad de hacer copias de seguridad del log de transacciones, se reduce automáticamente el espacio de registro, manteniendo al mínimo el espacio del fichero segun termina las transacciones de las consultas. De este modo no es necesario administrar el espacio del log de transacciones.

Los cambios realizados después de la copia de seguridad más reciente no están protegidos. En caso de desastre, es necesario volver a realizar dichos cambios. Sólo se puede recuperar hasta el final de una copia de seguridad.

***Modelo de recuperación Completa:***

Requiere copias de seguridad del log de transacciones. No se pierde [trabajo](http://adnetwork.fullservices.com.ar/trabajo/m08129219709035286126/) si un archivo de datos se pierde o resulta dañado. Se puede recuperar hasta cualquier momento, por ejemplo, antes del error de aplicación o usuario.

Si la base de datos resulta dañada, se deben repetir los cambios realizados desde la última copia de seguridad del log de transacciones. Se puede recuperar hasta un determinado momento, siempre que las copias de seguridad se hayan completado hasta ese momento.

***Modelo de recuperación bulk-logged:***

Requiere copias de seguridad del log de transacciones, para ir liberando espacio en el .ldf. Puede considerarse complemento del modelo de recuperación completa, pero no sustituto, ya que permite operaciones de copia masiva de alto rendimiento, (por ejemplo, operaciones realizadas con BCP.exe, Bulk Insert, etc.…), reduciendo el uso del espacio de registro.

Si la base de datos resulta dañada o se han realizado operaciones masivas desde la última copia de seguridad completa, se han de repetir los cambios desde esa última copia de seguridad. Se puede recuperar hasta el final de cualquier copia de seguridad completa. No admite recuperaciones a un momento dado.

El “set recovery…” es la opción que permite elegir el modelo de recuperación [entre](http://adnetwork.fullservices.com.ar/entre/t084dd8b8421/" \o "" \t "_blank)simple, bulk-logged y completo. Los tres permiten realizar restores, pero sólo el modelo de recuperación completo registra y mantiene en el log de transacciones todas las operaciones e implica la realización de backups del log dentro de la política de backups. Es la opción por defecto y la recomendable (salvo circunstancias excepcionales). Ello permite operaciones como estas (recuperar un backup hasta un punto en el tiempo, hasta una marca). En el modelo de recuperación simple no es así, las operaciones quedan mínimamente logadas, los backups del log no pueden realizarse (carecen de sentido).

A modo de ejemplo muestro como cambiar el modelo de configuración del log de transacciones:

*alter database bbdd set recovery simple — Pone el modelo de recuperación del log a simple*

*go*

*alter database bbdd set recovery full — Pone el modelo de recuperación del log a completa.*

*go*

Si cambias al modelo de recuperación simple, interrumpirá la cadena de copias de seguridad de registros. Por lo tanto, es muy recomendable realizar una copia de seguridad del registro inmediatamente antes de realizar el cambio. De esta manera, podrá recuperar la base de datos hasta ese momento. Tras el cambio, necesitará realizar copias de seguridad completas y periódicas para proteger sus datos y para truncar la parte inactiva del registro de transacciones.

El cambio al modelo de recuperación completa o bulk-logged sólo tiene efecto después de la primera copia de seguridad de base de datos.

*backup database TU\_bbdd to disk = ‘<Unida:\_ruta> TU\_bbdd.bak’ with init* *, NOUNLOAD , NAME = N’Copia de seguridad TU\_bbdd’, NOSKIP , STATS = 10, NOFORMAT*

Si no se quieren sobrescribir los ficheros de backup. Muestro un ejemplo, de cómo crear los backups, teniendo en cuenta el nombre de los ficheros, de esta forma mantendremos una secuencia en los nombres:

*declare @fichero varchar(250)*

*select @fichero = ‘< Unida:\_ruta>bbdd\_tlog\_’ + convert(varchar(20), getdate(),112) + left(replace(convert(varchar(10), getdate(), 114), ‘:’, ”), 4) + ‘.Bak’*

*backup database TU\_bbdd to disk = @fichero with init;*

Las copias de seguridad del log de transacciones son un aspecto fundamental de los modelos de recuperación completa o bulk-logged. Las copias de seguridad de registros permiten que se trunque el registro de transacciones. Si no realiza la copia de seguridad con la frecuencia suficiente, el registro de transacciones se puede expandir hasta quedarse sin espacio en disco. Muestro un ejemplo, de cómo crear los backups del log de transacciones, teniendo en cuenta el nombre de los ficheros, de esta forma mantendremos una secuencia en los nombres, por si fuera necesario restaurar, en un punto en el tiempo.

*declare @fichero varchar(250)*

*select @fichero = ‘< Unida:\_ruta TU\_bbdd\_tlog\_’ + convert(varchar(20), getdate(),112) + left(replace(convert(varchar(10), getdate(), 114), ‘:’, ”), 4) + ‘.trn’*

*backup log [TU\_bbdd] to disk = @fichero*

Si cambia del modelo de recuperación completa o bulk-logged al modelo de recuperación simple, interrumpirá la cadena de copias de seguridad de registros. Por lo tanto, es muy recomendable realizar una copia de seguridad del registro inmediatamente antes de realizar el cambio. De esta manera, podrá recuperar la base de datos hasta ese momento. Tras el cambio, necesitará realizar copias de seguridad periódicas para proteger sus datos y para truncar la parte inactiva del registro de transacciones.

Registro de transacciones lleno (Error 9002)

En este tema se tratan las posibles respuestas a un registro de transacciones lleno y se sugiere cómo evitar esta situación en el futuro. Cuando el registro de transacciones se llena, SQL Server Database Engine (Motor de base de datos de SQL Server) genera un error 9002. El registro se puede llenar cuando la base de datos está en línea o en recuperación. Si el registro se llena cuando la base de datos está en línea, la base de datos seguirá en conexión, pero solo se puede leer y no actualizar. Si el registro se llena durante la recuperación, Motor de base de datos marca la base de datos como RESOURCE PENDING. En ambos casos, es necesaria la intervención del usuario para proporcionar espacio de registro.

Si la base de datos estaba en recuperación cuando se produjo el error 9002, una vez resuelto el problema, recupere la base de datos mediante:

ALTERDATABASE nombreDeBaseDeDatos SETONLINE

**Las columnas log\_reuse\_wait y log\_reuse\_wait\_desc de la vista de catálogosys.database**.

La respuesta apropiada a un registro de transacciones lleno depende en parte de la condición o condiciones que han causado que el registro se llene. Para descubrir qué impide el truncamiento del registro en un caso determinado, use las columnas**log\_reuse\_wait** y **log\_reuse\_wait\_desc** de la vista de catálogo **sys.database**.

Podemos mostrar la información del estado en que se encuentra el log de transacciones con la consulta :

Select name as base\_datos, log\_reuse\_wait, log\_reuse\_wait\_desc,recovery\_model\_desc as modo\_recuperacion\_log, page\_verify\_option\_desc aspage\_verify\_bbdd, user\_access\_desc as user\_access, state\_desc as estado\_bbddfromsys.databases

Podemos cruzar el resultado de la anterior consulta, con la siguiente tabla, para entender el resultado de la consulta anterior:

|  |  |  |
| --- | --- | --- |
| valor log\_reuse\_wait | valor log\_reuse\_wait\_desc | Descripción |
| 0 | NOTHING | Hay actualmente uno o más archivos de registro virtual reutilizables. |
| 1 | CHECKPOINT | No se ha producido ningún punto de comprobación desde el[ÚLTIMO AVISO del](http://adnetwork.fullservices.com.ar/click/u08129219811067994612/)[último](http://adnetwork.fullservices.com.ar/%C3%BAltimo/u08129219811067994612/) truncamiento o el encabezado del registro no se ha movido más allá de un archivo de registro virtual (todos los modelos de recuperación). Éste es un motivo habitual para retrasar el truncamiento. |
| 2 | LOG\_BACKUP | Se necesita una copia de seguridad del registro para hacer avanzar el encabezado del registro (modelos de recuperación completos o registrados de forma masiva sólo). Cuando se completa la copia de seguridad del registro, se avanza el encabezado del registro y algún espacio del registro podría convertirse en reutilizable. |
| 3 | ACTIVE\_BACKUP\_OR\_RESTORE | Existe una recuperación o copia de seguridad de datos en curso (todos los modelos de recuperación).La copia de seguridad de datos funciona como una transacción activa y, cuando se ejecuta, la copia de seguridad impide el truncamiento. |
| 4 | ACTIVE\_TRANSACTION | Podría existir una transacción de larga duración en el inicio de la copia de seguridad del registro. En este caso, para liberar espacio se podría requerir otra copia de seguridad delregistro.Se difiere una transacción. Una*transacción diferida* es efectivamente una transacción activa cuya reversión se bloquea debido a algún recurso no disponible. |
| 5 | DATABASE\_MIRRORING | Se realiza una pausa en la creación de reflejo de la base de datos o, en el modo de alto rendimiento, la base de datos reflejada está notablemente detrás de la base de datos de la entidad de seguridad (sólo para el modelo de recuperación completa). |
| 6 | REPLICATION | Durante las replicaciones transaccionales, las transacciones relevantes para las publicaciones no se han entregado aún a la base de datos de distribución (sólo para el modelo de recuperación completa). |
| 7 | DATABASE\_SNAPSHOT\_CREATION | Se está creando una instantánea de base de datos (todos los modelos de recuperación). Éste es un motivo habitual, por lo general breve, para retrasar el truncamiento del registro. |
| 8 | LOG\_SCAN | Se está produciendo un recorrido del registro (todos los modelos de recuperación). Éste es un motivo habitual, por lo general breve, para retrasar el truncamiento del registro. |
| 9 | OTHER\_TRANSIENT | No se utiliza este valor actualmente. |

Una de las cosas que se pueden revisar, cuando nos vemos ante un registro de transacciones lleno son:

1. Realizar copias de seguridad del registro.

2. Liberar espacio en disco para que el registro pueda crecer automáticamente.

3. Mover el archivo de registro a una unidad de disco con suficiente espacio.

Mover el fichero .ldf de una base de usuario de ubicación:

Antes de nada, realiza un backup de la bbdd, por lo que pueda pasar.

Extraiga la ubicación física de sus archivos de la base de datos.

USE DataBaseName

GO

EXECsp\_helpfile

GO

Desatachamos la base de datos:

usemaster

GO

Execsp\_detach\_db ‘DataBaseName’

GO

Movemos los ficheros .mdf y .ldf a los nuevos volúmenes, en nuestro caso el .ldf:

usemaster

GO

Execsp\_attach\_db ‘DataBaseName’,'<Ruta y unidad>DataBaseName\_Data.MDF’,'<Nueva unidad>DataBaseName\_Log.LDF’

GO

Esta forma, tenderá a desaparecer en versiones futuras, por lo que os paso como se ha de realizar:

Borraremos la Base de datos, sin eliminar los ficheros de datos ni el fichero del log de transacciones. copiamos los .mdf y .ldf a sus discos definitivos. y:

USE [master]

GO

CREATEDATABASE [database\_name] ON

(FILENAME=N'<Ruta y unidad>DataBaseName\_Data.mdf’),

(FILENAME=N'<Nueva ruta y unidad>DataBaseName\_Log.ldf’)

FORATTACH;

GO

4. Aumentar el tamaño del log de transacciones.

En este ejemplo aumentamos el tamaño del fichero .ldf a 2GB:

USEmaster

GO

ALTERDATABASE database\_name

MODIFYFILE ( NAME =N'<Nombre\_logico\_ldf>’, SIZE = 2097152KB )

GO

5. Agregar un archivo adicional para el log de transacciones en un disco diferente. Para agregar un archivo de registro a la base de datos, utilice la cláusula ADD LOG FILE de la instrucción ALTER DATABASE

En este ejemplo se añaden 2 archivos adicionales para el log de transacciones:

USEmaster;

GO

ALTERDATABASE databasename

ADDLOGFILE

(

NAME = test1log2,

FILENAME='<Unidad y ruta>databasename2log.ldf’,

SIZE = 5MB,

MAXSIZE = 100MB,

FILEGROWTH = 5MB

),

(

NAME = test1log3,

FILENAME='<Unidad y ruta>databasename 3log.ldf’,

SIZE = 5MB,

MAXSIZE = 100MB,

FILEGROWTH = 5MB

);

GO

6. Terminar o eliminar una transacción de larga duración. Ve el siguiente punto, “transacción activa”.

Transacción activa

Una causa probable de que el registro esté lleno es que se haya realizado una transacción de ejecución prolongada. Una transacción de ejecución prolongada se mantiene activa en el log de transacciones e impide el truncamiento del archivo.

Una transacción de ejecución muy prolongada hará que el registro de transacciones se llene. Para buscar las transacciones de ejecución prolongada, use la vista:**sys.dm\_tran\_database\_transactions**

Select\*fromsys.dm\_tran\_database\_transactions

Esta vista de administración dinámica devuelve información sobre las transacciones en la base de datos. En una transacción de ejecución prolongada, las columnas de especial interés incluyen la hora de la primera entrada del registro (**database\_transaction\_begin\_time**), el estado actual de la transacción (**database\_transaction\_state**) y el número de secuencia (LSN) del registro inicial del log de transacciones (**database\_transaction\_begin\_lsn**).

Valores que puede adoptar el campo database\_transaction\_type:

1 = Transacciones de lectura/escritura

2 = Transacción de sólo lectura

3 = Transacción de sistema

Valores que puede adoptar el campo database\_transaction\_state:

1 = La transacción no se ha inicializado.

3 = La transacción se ha inicializado, pero no se han generado registros.

4 = La transacción ha generado registros.

5 = La transacción se ha preparado.

10 = La transacción se ha confirmado.

11 = La transacción se ha revertido.

12 = La transacción se está confirmando. En este estado el registro se está generando, pero no se ha materializado o se ha hecho permanente.

Otra forma de detectar las transacciones activas es con: **DBCC OPENTRAN:**

Esta instrucción “DBCC OPENTRAN” permite identificar el Id. de usuario del propietario de la transacción, por lo que se puede realizar un seguimiento del origen de la misma para terminarla de forma más ordenada, otro dato que nos devuelve es el SPID de la transacción.

Para detener la transacción, puede que deba utilizar la instrucción KILL. Utilice esta instrucción con sumo cuidado, especialmente cuando se estén ejecutando procesos críticos

**DBCC SQLPERF (LOGSPACE)**

Puede supervisar el uso del espacio del log de transacciones mediante el comando DBCC SQLPERF (LOGSPACE). Este comando devuelve información sobre la cantidad de espacio del registro de transacciones actualmente en uso e indica cuándo es necesario el truncamiento del registro de transacciones. Para obtener información acerca del tamaño actual de un archivo de registro, su tamaño máximo y la opción de crecimiento automático de este archivo, se pueden usar las columnas **size**, **max\_size** y**growth** de ese archivo de registro en **sys.database\_files**.

Formas para liberar espacio o reducir el tamaño del log de transacciones.

Es conveniente que antes de cualquier reducción del log, realice un backup del mismo:

declare @fichero varchar(250)

select @fichero ='< Unidad:\_ruta TU\_bbdd\_tlog\_’+convert(varchar(20),getdate(),112)+left(replace(convert(varchar(10),getdate(),114),':’,”), 4)+’.trn’

backuplog [TU\_bbdd] todisk= @fichero

Cuando hacemos backup del log y no se especifica lo contrario, marca la parte no activa del log (transacciones que ya se les hizo commit) para ser reusada con nuevas transacciones. Si la parte no activa del log de transacciones no se marca para ser reusada, con el “backup del log”, se podrá llegar al final del archivo, y si las transacciones continúan, entonces el log deberá crecer para seguir almacenando el resto de transacciones. Ojo, que el backup completo no trunca el log, así que en dependencia de la frecuencia con que hagas el backup del log, tiene sentido hacer un backup del log después del backup completo.

Casos prácticos 1.:

Backuplog“<BaseDatos\_Name>”with no\_log –>> borra la parte no activa del log, pero no reduce el log.

Backuplog<BaseDatos\_Name>with truncate\_only –>> Borra todo el log, pero no reduce el log.

Desde SQL Server 2008, ya no se puede truncar el log, realizando un backup, con las opciones no\_log y truncate\_only, no funcionan.

Luego, lo importante es saber qué impide que el log de transacciones se reduzca o se recicle, para lo cual, lo más fácil es mirarlo en sys.databases, en la columna log\_reuse\_wait\_desc. Aquí tienes su explicación: <http://msdn.microsoft.com/es-es/library/ms345414.aspx>. En función de lo que marque esa columna, el modelo de recuperación que tengas y para la base de datos que quieres reducir, tendrás que tomar una acción u otra, a continuación, como realizar una reducción con un shrink:

DBCC SHRINKFILE(<NombreLogico\_log>, 512) –>> Reduce el log a n Mb

Casos prácticos 2.:

Si no se deja comprimir o borrar el log por que no tiene sus transacciones como distribuidas habrá que ponerlas, de la forma:

EXECsp\_repldone @xactid =NULL, @xact\_segno =NULL, @numtrans = 0, @time = 0,@reset = 1

—>Ver el tamaño de los log y él % de ocupación de las bbdd de una instancia:

dbcc sqlperf (logspace)

—>”esto tarda mucho…” Mostrar información de espacio actualizada acerca de una base de datos

EXECsp\_spaceused @updateusage =N’TRUE';

GO

Casos prácticos 3.:

Recomendaron para SQL Server 2008, con problemas de reducción del log de transacciones:

Ejecuta dos o tres veces la instrucción CHECKPOINT. Esto asegurará que todas las páginas de memoria se han escrito en el fichero de datos:

CHECKPOINT

BACKUPLOG NombreBaseDeDatos TODISK='<Ruta y unidad>triton1.trn’

DBCC SHRINKFILE (N’Nombre\_logico\_Log’, 0, TRUNCATEONLY)

GO

Casos prácticos 4.:

Comprueba qué valor muestra el campo “log\_reuse\_wait” de sys.databases para esa base de datos en cuestión. Si ese campo indica efectivamente que el problema es que hay transacciones activas, puedes revisarlas con el comando DBCC OPENTRAN (<http://msdn.microsoft.com/es-es/library/ms182792.aspx>)

alterdatabase BBDD\_Name setrecoverysimple

go

checkpoint

go

checkpoint

go

checkpoint

go

checkpoint

go

alterdatabase BBDD\_Name setrecoveryfull

go

–Recomendable, para comenzar secuencia, para posterior backup del log.:

backupdatabase BBDD\_Name todisk='<Ruta y unidad>.bak’withinit

go

DBCC SHRINKFILE(pubs\_log, 2)

Go

Casos prácticos 5.:

La operación de reducción está bloqueada

Es posible bloquear las operaciones de reducción mediante una transacción que se ejecuta con un nivel de aislamiento basado en el control de versiones de filas. Por ejemplo, si se está ejecutando una operación de eliminación grande con un nivel de aislamiento basado en el control de versiones de filas cuando se ejecuta una operación DBCC SHRINK DATABASE, la operación de reducción esperará a que la operación de eliminación se haya completado antes de reducir los archivos. Cuando esto sucede, las operaciones DBCC SHRINKFILE y DBCC SHRINKDATABASE imprimen un mensaje informativo (5202 en el caso de SHRINKDATABASE y 5203 para SHRINKFILE) en el registro de errores de SQL Server cada cinco [minutos](http://adnetwork.fullservices.com.ar/minutos/q08129219771428629261/) durante la primera hora, y cada hora sucesivamente. Por ejemplo, si el registro de errores contiene el siguiente mensaje de error:

*DBCC SHRINKFILE for file ID 1 is waiting for the snapshot*

*transaction with timestamp 15 and other snapshot transactions linked to*

*timestamp 15 or with timestamps older than 109 to finish.*

Esto significa que la operación de reducción está bloqueada por transacciones de instantánea que tienen marcas de tiempo anteriores a 109, que es la última transacción que ha completado la operación de reducción. También indica que las columnas**transaction\_sequence\_num** o **first\_snapshot\_sequence\_num** de la vista de administración dinámica sys.dm\_tran\_active\_snapshot\_database\_transactions contienen el valor 15. Si la columna **transaction\_sequence\_num** o**first\_snapshot\_sequence\_num** contiene un número inferior al de la última transacción completada mediante una operación de reducción (109), la operación de reducción esperará a que las transacciones finalicen.

Para resolver el problema, puede llevar a cabo una de las tareas siguientes:

* · Finalizar la transacción que está bloqueando la operación de reducción.
* · Finalizar la operación de reducción. Si finaliza la operación de reducción, se conservará todo el trabajo completado.
* · No hacer nada y permitir que la operación de reducción espere a que finalice la transacción que la está bloqueando.

Espero les sirva,  Saludos