

Capítulo 5

Almacenamiento del motor de base de datos

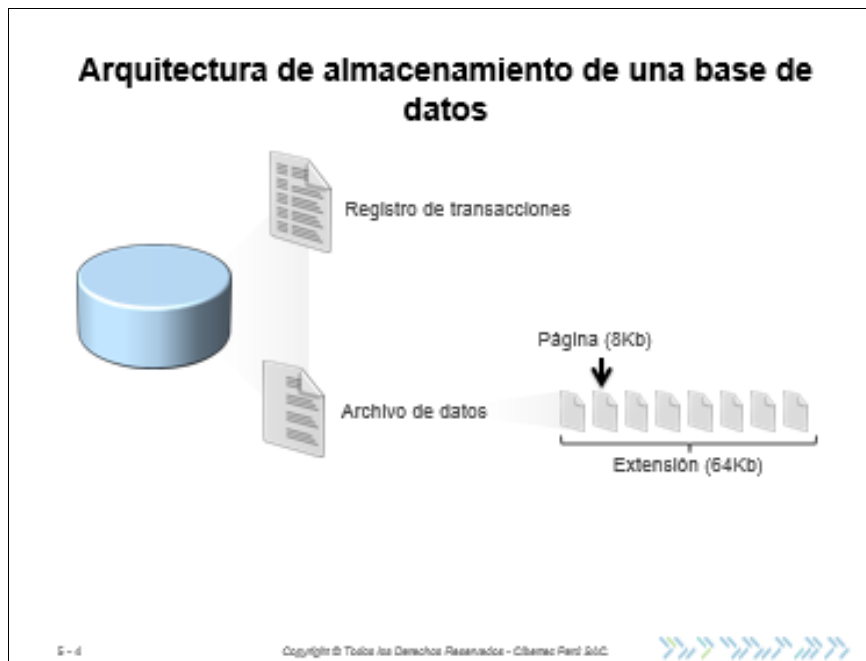
Al finalizar el capítulo, el alumno podrá:

- Explicar cómo se almacena físicamente la información de SQL Server
- Diseñar la arquitectura de almacenamiento según requerimientos del negocio
- Reconocer las diferentes tecnologías de almacenamiento disponibles en el mercado
- Monitorear el espacio utilizado por las bases de datos para conocer con anticipación las necesidades de almacenamiento

Temas

1. Arquitectura de almacenamiento de una base de datos
2. Ubicación física de los archivos de base de datos
3. Gestión del almacenamiento de la base de datos
4. Tecnologías de almacenamiento de bases de datos

1. Arquitectura de almacenamiento de una base de datos



Una parte importante del trabajo de un administrador de bases de datos consiste en planear y crear bases de datos. Reconocer cómo SQL Server almacena los datos, procesa las transacciones, y usa los archivos y grupos de archivos, permitirá tomar decisiones sobre la mejor ubicación de los archivos de base de datos para obtener el máximo rendimiento y optimizar la recuperación de desastres. El planeamiento de la capacidad permitirá predecir cuánto espacio en disco puede necesitar una base de datos.

1.1 Almacenamiento de datos

Todas las bases de datos tienen los siguientes archivos:

- Un archivo de datos principal (extensión .mdf)
- Uno o más archivos de registro de transacciones (extensión .ldf).

Una base de datos también puede tener archivos secundarios de datos (extensión .ndf).

Estos archivos físicos tienen los nombres de los archivos del sistema operativo y los nombres de los archivos lógicos que se pueden usar en las instrucciones Transact-SQL. La ubicación predeterminada para todos los archivos de datos y los registros de transacciones es la carpeta Data de la ruta predeterminada de la instancia de SQL Server.

Al crear una base de datos, una copia de la base de datos model, donde se incluyen las tablas del sistema, se copia en la base de datos y el resto de la base de datos se rellena con páginas vacías.

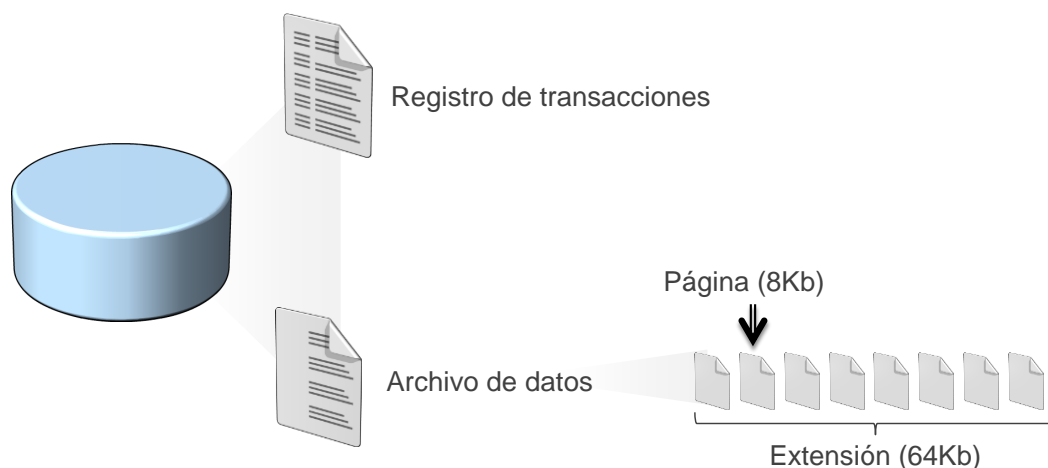
Los datos se almacenan en bloques de 8 kilobytes (Kb) de espacio en disco contiguo llamado páginas. Esto significa que una base de datos puede almacenar 128 páginas por megabyte (Mb).

Las tablas y los índices se almacenan en extensiones. Una extensión es una colección de ocho páginas físicamente contiguas, o bien 64 KB. Por consiguiente, una base de datos tiene 16 extensiones por megabyte. Las tablas pequeñas pueden compartir las extensiones con otros objetos de la base de datos.

Un registro no puede abarcar varias páginas. Por lo tanto, la cantidad máxima de datos en una fila única menos el espacio requerido para la carga de fila es de 8.060 bytes. Hay dos excepciones a esta regla:

- La característica de desbordamiento de fila en SQL Server permite que las filas contengan columnas definidas como varchar, nvarchar, varbinary y sql_variant o tipos CLR definidos por el usuario, que pueden exceder el tamaño de la página siempre y cuando cada columna se encuentre dentro del límite de 8.000 bytes.
- Las columnas definidas como texto, imagen o binario, así como, las columnas de tipo varchar, nvarchar y varbinary definidas con el especificador máximo (MAX), se almacenan como un puntero en la página de datos que hace referencia a un conjunto de páginas de desbordamiento donde se almacena el valor de columna real.

Los archivos de registro de transacciones contienen la información necesaria para la recuperación de la base de datos en caso de un error del sistema, pero no contienen las páginas de datos.



1.2 Registro de transacciones

Una transacción es un conjunto de una o más sentencias Transact-SQL tratadas como una única unidad de trabajo y recuperación. Dentro de una transacción, las instrucciones Transact-SQL se deben realizar por completo o no realizarse.

SQL Server realiza una transacción cuando cualquiera de las siguientes instrucciones Transact-SQL se ejecuta como una transacción.

- ALTER TABLE
- CREATE
- DELETE
- DROP
- FETCH
- GRANT
- INSERT
- OPEN
- REVOKE
- SELECT
- TRUNCATE TABLE
- UPDATE

De forma predeterminada, SQL Server funciona en modo de confirmación automática. Esto significa que una transacción implícita se confirma después de la ejecución sin una instrucción COMMIT TRANSACTION que finalice la transacción, SQL Server también puede funcionar en modo de transacción implícita. Cuando cualquiera de las instrucciones Transact-SQL mencionadas inicia una transacción, la transacción debe finalizar con la instrucción COMMIT TRANSACTION.

SQL Server realiza una transacción explícita cuando se definen explícitamente el principio y el fin de la transacción. Se puede definir el principio y fin de la transacción en Transact-SQL usando las instrucciones BEGIN TRANSACTION y COMMIT TRANSACTION.

Proceso de registro de transacciones

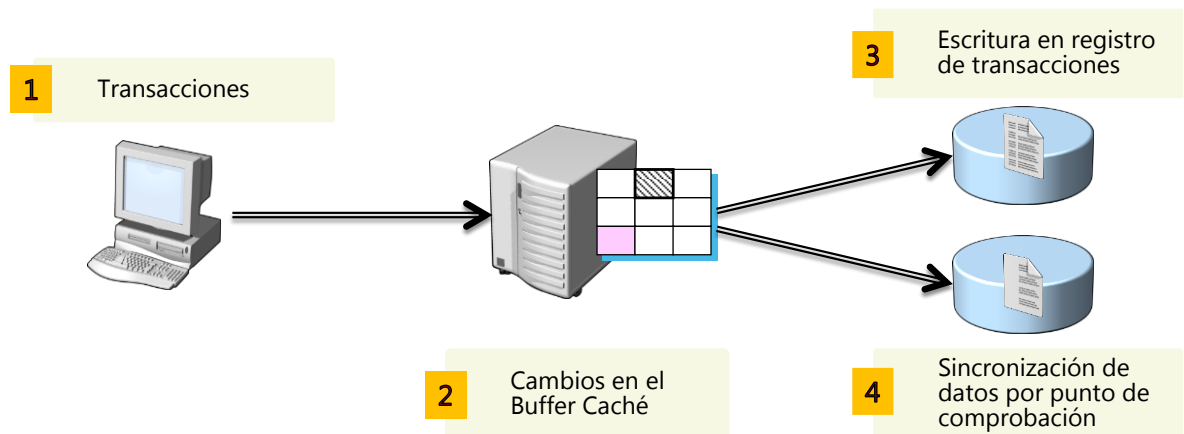
SQL Server registra todas las transacciones en un registro de transacciones para mantener la coherencia de la base de datos y contribuir en la recuperación. El registro es un área de almacenamiento que realiza automáticamente un seguimiento de los cambios efectuados en una base de datos. SQL Server registra las modificaciones en el registro del disco a medida que dichas modificaciones se ejecutan y antes de que se escriban en la base de datos.

El registro de transacciones registra las modificaciones de datos a medida que suceden. El proceso de registro sigue estos pasos:

1. Una aplicación envía una modificación de datos.
2. Cuando se ejecuta la modificación, SQL Server carga las páginas de datos afectadas del disco en la memoria, llamada buffer caché, siempre y cuando una consulta anterior no las haya puesto ya en el buffer caché.
3. SQL Server graba todas las instrucciones de modificación de datos en el registro a medida que se producen. El cambio siempre se graba en el registro y se escribe en el disco antes de aplicarse a la base de datos. Este tipo de registro se denomina registro de escritura anticipada (Write Ahead Log - WAL).
4. Periódicamente, el proceso de punto de comprobación (CHECKPOINT) escribe en el disco todas las transacciones completadas en la base de datos.

Si se produce un error del sistema, el proceso de recuperación automática usa el registro de transacciones para poner al día todas las transacciones confirmadas y revertir las transacciones incompletas.

El registro usa marcadores de transacción durante la recuperación automática para determinar los puntos inicial y final de una transacción. Se considera que una transacción está completa cuando el marcador BEGIN TRANSACTION tiene asociado un marcador COMMIT TRANSACTION. Las páginas de datos se escriben en el disco cuando se produce un punto de comprobación.

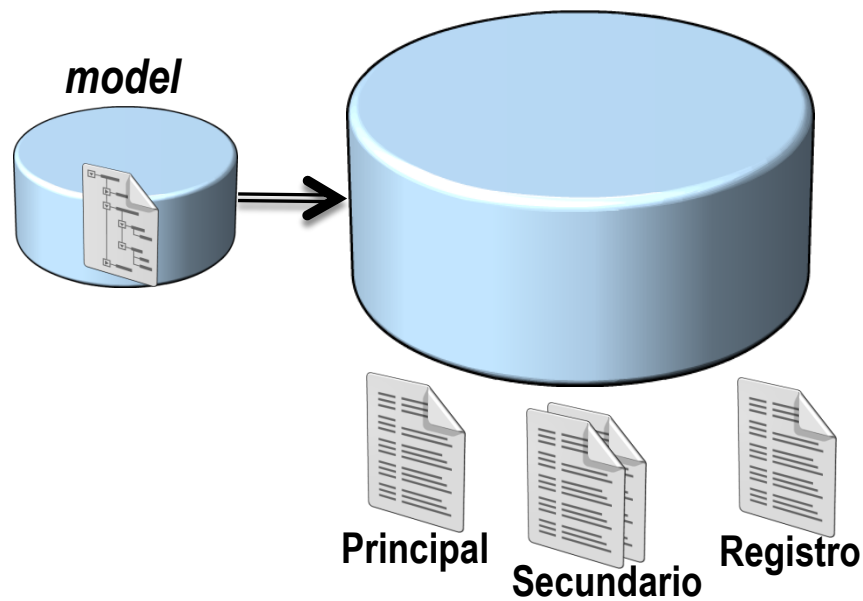


1.3 Dimensionamiento de la base de datos

Una de las funciones principales de un administrador de bases de datos es asignar, administrar y supervisar los requisitos de almacenamiento y espacio para SQL Server y sus bases de datos. Calcular el espacio que una base de datos requiere puede ayudar a planear el diseño de almacenamiento y a determinar los requisitos de hardware.

Al planear la base de datos, se prepara la estructura lógica. Bajo esa estructura, hay varios archivos físicos y objetos que ocupan espacio en disco. Esto incluye tablas de usuario, tablas del sistema, índices, registros de transacciones y objetos grandes (LOB, large object) que se almacenan en páginas de desbordamiento independientes de la propia tabla de referencia. Para calcular el tamaño de una base de datos, se deberían tener en cuenta todos estos elementos.

Al crear una base de datos, SQL Server crea copias de los objetos de la base de datos model, incluidas las tablas del sistema que contienen información sobre archivos, objetos, permisos y restricciones, así como, páginas vacías para rellenar el tamaño inicial especificado de la base de datos. Las tablas del sistema crecen en tamaño a medida que se crean objetos en la base de datos. Cada objeto creado genera una nueva fila que se inserta en una o más tablas del sistema.



Cuando calcule la cantidad de espacio que ocupará su base de datos, considere los factores siguientes:

- El tamaño de los objetos de la base de datos model y las tablas del sistema, incluido el crecimiento previsto.
- La cantidad de datos en tablas, incluido el crecimiento previsto.
- El número y tamaño de índices, en particular el tamaño del valor clave, el número de filas y la configuración del factor de relleno. La configuración del factor de relleno le permite reservar espacio dentro de una tabla para un futuro crecimiento.
- El tamaño del registro de transacciones, que se ve afectado por la cantidad y frecuencia de la actividad de modificación, el tamaño de cada transacción y con qué frecuencia se realizan copias de seguridad o se vuelca el registro.

**Nota**

Como punto de partida, debería asignar el 25% del tamaño de la base de datos al registro de transacciones para entornos de procesamiento de transacciones en línea (OLTP, online transaction processing). Puede asignar un porcentaje menor para bases de datos que se usan principalmente para consultas.

- El tamaño de las tablas del sistema, como el número de usuarios, objetos, etc., que no suele representar un porcentaje grande del tamaño de la base de datos.


2. Ubicación física de los archivos de base de datos

Ubicación física de los archivos de base de datos

Archivos físicos

Considerar tipos de archivo.

- Archivo de datos:
 - La base de datos puede contener uno o varios.
 - Principal (.mdf).
 - secundario (.ndf).
 - Escritura aleatoria.
- Registros de transacciones:
 - Típicamente la base de datos contiene uno, pero pueden ser varios.
 - Escritura secuencial.

5 - 12Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.

Al crear una base de datos, SQL Server crea un archivo de datos y un registro de transacciones para esa base de datos y permite definir la ubicación de estos archivos.

Cada base de datos debe tener un archivo principal y un archivo de registro de transacciones. También puede tener uno o más archivos secundarios.

El archivo principal contiene la información de inicio de la base de datos y apunta al resto de archivos de la base de datos. Los datos de usuario y los objetos se pueden almacenar en este archivo principal o en archivos secundarios de datos. Cada base de datos tiene un archivo principal. La extensión de nombre de archivo recomendada para los archivos principales de datos es **.mdf**.

Se pueden crear archivos secundarios para almacenar datos de usuario. Algunas bases de datos pueden ser lo bastante grandes para que requieran varios archivos secundarios de datos o también pueden usar los archivos secundarios en unidades de disco independientes para repartir los datos entre varios discos. La extensión de nombre de archivo recomendada para los archivos secundarios de datos es **.ndf**.

Cada base de datos debe tener un registro de transacciones. A menos que se especifique lo contrario, un archivo de registro de transacciones se crea automáticamente con un nombre generado por el sistema. Normalmente, el archivo de registro de transacciones es aproximadamente el 25% del tamaño de la base de datos. La extensión de nombre de archivo recomendada para los archivos de registro de transacciones es **.ldf**.

2.1 Ubicación de los archivos de base de datos

Se puede mejorar el rendimiento e implementar la tolerancia a errores si se administra la ubicación de archivos de datos y registros de transacciones en discos.

SQL Server usa llamadas de entrada/salida (E/S) de Windows Server para realizar lecturas y escrituras de disco. Además, administra cuándo y cómo se realizan las operaciones de E/S de disco, pero deja que Windows Server realice las operaciones de E/S subyacentes. El subsistema de E/S incluye el bus del sistema, las tarjetas del controlador de disco, los discos, las unidades de cinta, las unidades de CD-ROM y muchos otros dispositivos de E/S. Los discos suelen ser a menudo el mayor cuello de botella de un sistema.

En el contexto de administración del almacenamiento en disco para SQL Server se debe tener presente las definiciones siguientes:

- El rendimiento hace referencia en parte a la velocidad en las operaciones de lectura y escritura.
- La tolerancia a errores hace referencia a la capacidad del sistema para seguir funcionando sin perder datos cuando se produce un error en una parte del sistema.

Para bases de datos más grandes, se debe distribuir tantos datos por tantas unidades físicas como sea posible. De esa manera, se produce una mejora en el rendimiento puesto que se tiene acceso a los datos en paralelo, usando varios archivos. En general, se debe crear un archivo para cada disco físico y agrupar los archivos en uno o más grupos de archivos. SQL Server puede realizar los recorridos siguientes:

- Recorridos de los datos en paralelo si el equipo tiene varios procesadores y varios discos.
- Varios recorridos en paralelo para una tabla única si el grupo de archivos de la tabla contiene varios archivos.

Para distribuir de forma uniforme los datos en todos los discos, se debe usar las tecnologías de matriz redundante de discos independientes (RAID) y, si es necesario, se configurarán grupos de archivos definidos por el usuario para distribuir los datos por varios conjuntos de secciones de hardware.

**Nota**

Una técnica avanzada es separar las tablas de los índices no agrupados. Un índice no agrupado es un índice cuyo orden lógico es diferente al orden físico almacenado de las filas en disco.

Consideraciones sobre el registro de transacciones

Se debe crear el registro de transacciones en un disco independiente o usar RAID. Dado que el archivo de registro de transacciones se escribe secuencialmente, usar un disco dedicado independiente permite que los cabezales de disco permanezcan en posición correcta para la operación de escritura siguiente. Usar RAID también proporciona tolerancia a errores.

Por ejemplo, si el entorno de producción de SQL server tiene varias bases de datos en el mismo servidor, se podrían usar discos independientes para cada registro de transacciones. Esta estrategia contribuye a obtener un rendimiento óptimo.

Consideraciones sobre la base de datos tempdb

Se recomienda ubicar la base de datos **tempdb** en un subsistema de E/S rápido independiente de las bases de datos de usuario para garantizar un rendimiento óptimo. Use RAID para distribuir la base de datos tempdb por varios discos para obtener un mayor rendimiento.

Además, es recomendable crear tantos archivos de datos de la tempdb como núcleos de procesamientos se tengan asignados a la instancia, considerando un máximo de 8 archivos.

2.2 Grupos de archivos

Los grupos de archivos son colecciones de archivos con nombre y se usan para simplificar la ubicación de los datos.

Si la configuración de hardware incluye varias unidades de disco, se pueden ubicar objetos y archivos específicos en discos individuales mediante la agrupación de los archivos de base de datos en uno o más grupos de archivos. Los grupos de archivos mejoran el rendimiento al distribuir los datos por varios discos y usar subprocesos paralelos para el procesamiento de consultas. Los grupos de archivos también pueden facilitar el mantenimiento de la base de datos.

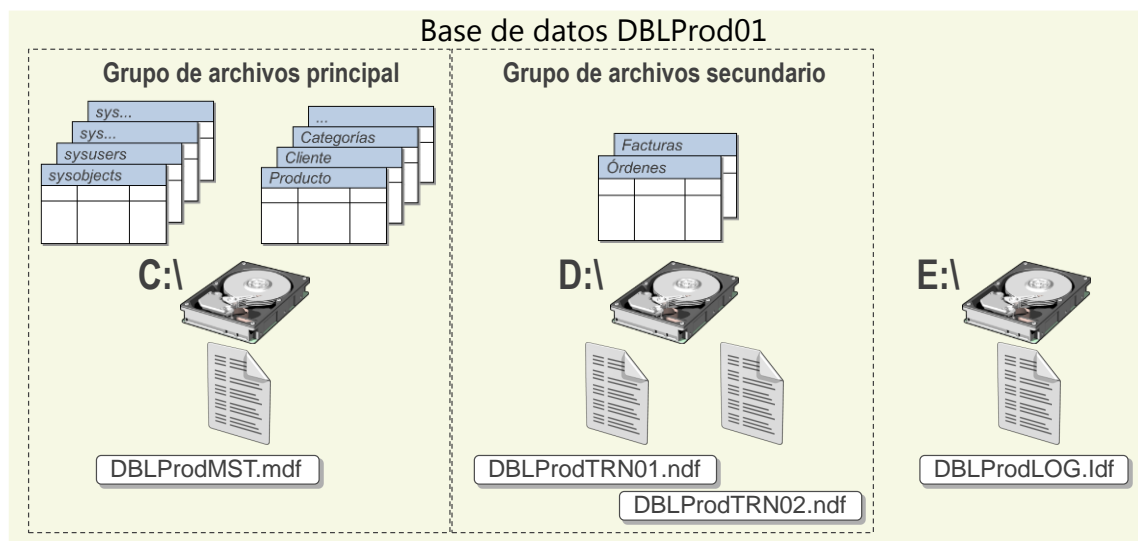
Las bases de datos de SQL Server tienen un grupo de archivos principal y también puede tener grupos de archivos definidos por el usuario. El grupo de archivos principal contiene el archivo principal de datos con las tablas del sistema. También, contiene cualquier archivo secundario definido por el usuario que no esté asignado a otros grupos de archivos.

Un grupo de archivos definido por el usuario consta de archivos de datos agrupados con fines de ubicación y administración.

Almacenamiento de archivos en grupos de archivos

Los grupos de archivos usan una estrategia de relleno proporcional en todos los archivos del grupo de archivos. Cuando los datos se escriben en el grupo de archivos, cada archivo se rellena en paralelo. Cada archivo se coloca físicamente en un disco o conjunto de discos. SQL Server mantiene una asignación de archivos que asocia cada objeto de base de datos a su ubicación en el disco.

Los archivos de registro de transacciones no forman parte de un grupo de archivos. El espacio del registro de transacciones se administra por separado del espacio de datos.



Consideraciones para la creación de grupos de archivos

Para determinar la cantidad de grupos de archivos necesaria, los administradores de bases de datos deberán tener en cuenta los requisitos de rendimiento y facilidad de administración, qué grupo de archivos se usará como grupo de archivos predeterminado y cuándo se usarán grupos de archivos de solo lectura.

Los grupos de archivos se usan principalmente para controlar la ubicación de los datos. Hay dos razones fundamentales para controlar dicha ubicación: rendimiento y facilidad de administración.

Rendimiento.- Es posible mejorar el rendimiento de las consultas si se crean varios archivos en discos físicos diferentes de un único grupo de archivos y se asignan tablas que tienen acceso con frecuencia a ese grupo de archivos. La estrategia de relleno proporcional que usa SQL Server cuando escribe los datos a los archivos de un grupo de archivos implica que los datos se distribuyen eficazmente por los discos físicos, por lo que se produce una mejora en el rendimiento de E/S.

También, se pueden usar varios grupos de archivos para separar las tablas de sus índices no agrupados, lo que en algunos casos puede optimizar el acceso a los datos.

Facilitar la administración.- Es posible usar varios grupos de archivos para separar datos según los diferentes requisitos de administración. Por ejemplo, puede separar los datos que se actualizan con frecuencia de aquellos datos que son relativamente estáticos o de los datos de solo lectura e implementar diferentes estrategias de copia de seguridad para cada grupo de archivos basadas en la volatilidad de los datos.

También, puede separar los índices susceptibles a la fragmentación de aquellos que se modifican con menor frecuencia para optimizar las tareas de desfragmentación y minimizar su impacto en el resto de la base de datos.

Grupos de archivos predeterminado

En SQL Server se puede designar un grupo de archivos como predeterminado. Este grupo se establece en el grupo de archivos principal en el momento de la creación de la base de datos, a menos que se especifique lo contrario. El grupo de archivos predeterminado contiene las páginas para todas las tablas e índices que no tienen un grupo de archivos específico cuando se crean.

Si el grupo de archivos principal sigue siendo el grupo de archivos predeterminado, se debe dimensionar apropiadamente o se debe establecer su crecimiento automático para no quedarse sin espacio. El grupo de archivos principal debe ser lo bastante grande para contener todas las tablas del sistema y cualquier tabla e índice que no se haya asignado a un grupo de archivos definido por el usuario.

Si el grupo de archivos principal se queda sin espacio, no se podrá agregar información a las tablas del sistema. Sin embargo, si un grupo de archivos definido por el usuario se queda sin espacio, solo se ven afectados los archivos de usuario que están específicamente asignados a ese grupo de archivos.

Se debe cambiar el grupo de archivos predeterminado si se usan grupos de archivos definidos por el usuario. Si la base de datos tiene varios grupos de archivos, se debería asignar como predeterminado uno de los grupos de archivos definidos por el usuario. Esto evitará que el crecimiento inesperado de la tabla provoque una restricción en las tablas del sistema del grupo de archivos principal.

Grupos de archivos de solo lectura

Cualquier grupo de archivos, a excepción del grupo de archivos principal, se puede marcar como de solo lectura. Un grupo de archivos marcado como de solo lectura no se puede modificar de ningún modo. Para evitar alteraciones accidentales en los datos, se deben colocar las tablas pertinentes en un grupo de archivos y marcar el grupo de archivos como de solo lectura.

SQL Server admite la compresión NTFS de grupos de archivos definidos por el usuario como de solo lectura y bases de datos de solo lectura. Considere comprimir los datos de solo lectura si se ha limitado el espacio en disco y se tiene un volumen grande de datos estáticos que deben estar disponibles para un acceso de solo lectura.

3. Gestión del almacenamiento de la base de datos

Gestión del almacenamiento de la base de datos

Vista del sistema sys.database_file

Contiene información de los archivos de base de datos, incluyendo el espacio utilizado.

Campos utilizados para gestión de almacenamiento:

- **size**: Tamaño actual de archivo expresado en páginas de 8kb.
- **max_size**: Tamaño máximo del archivo expresado en páginas de 8kb.

Función fileproperty: utilizada con el parámetro del nombre del archivo de datos y la propiedad 'spaceused', se obtiene el espacio utilizado.

```
SELECT FILEPROPERTY(name, 'SpaceUsed') FROM  
sys.database_files
```

5 - 18

Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.



Monitorear el espacio utilizado por las bases de datos es una tarea que deben llevar a cabo los administradores para garantizar el funcionamiento correcto de la base de datos y evitar que el espacio disponible en el subsistema de almacenamiento se agote.

Existen diferentes métodos de obtener información para el monitoreo del uso de espacio, algunas de ellas serán detalladas a continuación.

3.1 Vista del sistema sys.database_file

La vista de catálogo sys.database_file contiene información de todos los archivos físicos de la base de datos en uso. Entre los campos que contiene se encuentra información sobre:

- Códigos y nombres
- Ubicación física
- Tamaños y espacios utilizados
- Estado del archivo y propiedades
- Información de copia de seguridad

Para el monitoreo de espacio utilizado son dos campos los que se utilizan:

Campo size.- Tamaño actual de archivo expresado en páginas de 8kb.

Campo max_size.- Tamaño máximo del archivo expresado en páginas de 8kb.

Además, se puede utilizar la función FILEPROPERTY para obtener el espacio utilizado de un archivo. La siguiente consulta muestra cómo se puede obtener información del almacenamiento de una base de datos:

```
USE WideWorldImporters
GO

SELECT
    name,
    Type_desc,
    physical_name,
    size AS Actual,
    fileproperty(name, 'SpaceUsed') AS Utilizado
FROM sys.database_files
```

Name	Type_desc	physical_name	Actual	Utilizado
WWI_Primary	ROWS	D:\SQLData\SQL130\WideWorldImporters.mdf	131072	1152
WWI_Log	LOG	D:\SQLData\SQL130\WideWorldImporters.ldf	12800	10233
WWI_UserData	ROWS	D:\SQLData\SQL130\WideWorldImporters_UserData.ndf	262144	53584
WWI_InMemory_Data_1	FILESTREAM	D:\SQLData\SQL130\WideWorldImporters_InMemory_Data_1	14487	N/A



Nota

Los campos de almacenamiento son expresados en páginas de 8Kb. Si se desea obtener la información en otra medida se deben realizar los cálculos respectivos.

3.2 Procedimiento almacenado del sistema sp_helpfile

Este procedimiento muestra información de los archivos físicos de datos y registro de transacciones de la base de datos en uso. El resultado mostrado por este procedimiento almacenado es resumido comparado con la vista sys.database_files. Muestra información de identificación, ubicación física y almacenamiento.

Al ser un procedimiento almacenado se ejecuta con el comando EXECUTE y se obtienen todos los campos mostrados por el procedimiento almacenado de una sola vez:

```
USE WideWorldImporters
GO

EXECUTE sys.sp_helpfile
GO
```

Name	id	filename	filegroup	Size KB	maxsize	growth	usage
WWI_Primary	1	D:\SQLData\SQL130\WideWorldImporters.mdf	PRIMARY	1048576	Unlimited	65536	data only
WWI_Log	2	D:\SQLData\SQL130\WideWorldImporters.ldf	NULL	102400	2147483648	65536	log only
WWI_UserData	3	D:\SQLData\SQL130\WideWorldImporters_UserData.ndf	USERDATA	2097152	Unlimited	65536	data only

**Nota**

Los campos size y max_size están expresados en Kilobytes. Si se desea obtener la información en otra medida se deben realizar los cálculos respectivos.

3.3 Procedimiento almacenado del sistema sp_spaceused

Este procedimiento almacenado, a diferencia de lo visto anteriormente, no muestra la información detallada por archivo. Muestra la información resumida del espacio utilizado por toda la base de datos. El procedimiento muestra como resultado dos conjuntos de datos:

El primer conjunto de datos muestra el campo database_size que indica el espacio total utilizado por la base de datos, expresado en Mb. También se muestra el campo unallocated space que indica qué porción del total de la base de datos aún no ha sido asignado para uso de objetos.

El segundo conjunto de datos muestra la siguiente información:

Campo reserved.- Espacio asignado a objetos expresado en Kb.

Campo data.- Espacio utilizado por los datos, expresado en Kb.

Campo index_size.- Espacio utilizado por índices, expresado en Kb.

Campo unused.- Espacio libre, expresado en Kb.

```
USE WideWorldImporters
GO
```

```
EXECUTE sys.sp_spaceused
GO
```

database_name	database_size	unallocated space
WideWorldImporters	102717.19 MB	99316.61 MB

reserved	data	index_size	unused
3157392 KB	2373680 KB	775152 KB	8560 KB

Al procedimiento almacenado sp_spaceused se le puede pasar como parámetro el nombre de una tabla para conocer información detallada del almacenamiento de dicho objeto.

3.4 Comando de control de base de datos dbcc showfilestats

Los comandos de control de base de datos (DBCC – Data Base Control Command) permiten ejecutar diversas operaciones de administración en las bases de datos.

DBCC SHOWFILESTATS es un comando que muestra información muy resumida del espacio utilizado por los archivos de datos de una base de datos. No muestra información del registro de transacciones.

La información es expresada en extensiones, las cuales representan 8 páginas contiguas de 8kb cada una. Por lo tanto, representan porciones de espacio de 64Kb. Mucha de las operaciones de asignación de espacio, como el autotrecimiento de la base de datos se ejecuta en base a extensiones, por lo que este comando resulta muy apropiado para monitorear el espacio utilizado por la base de datos.

```
USE WideWorldImporters
GO
```

```
DBCC SHOWFILESTATS
```

Fileid	File Group	Total Extents	Used Extents	Name	FileName
1	1	1638400	49344	WWI_Primary	D:\SQLData\SQL130\WideWorldImporters.mdf
3	2	32768	6698	WWI_UserData	D:\SQLData\SQL130\WideWorldImporters_UserData.ndf

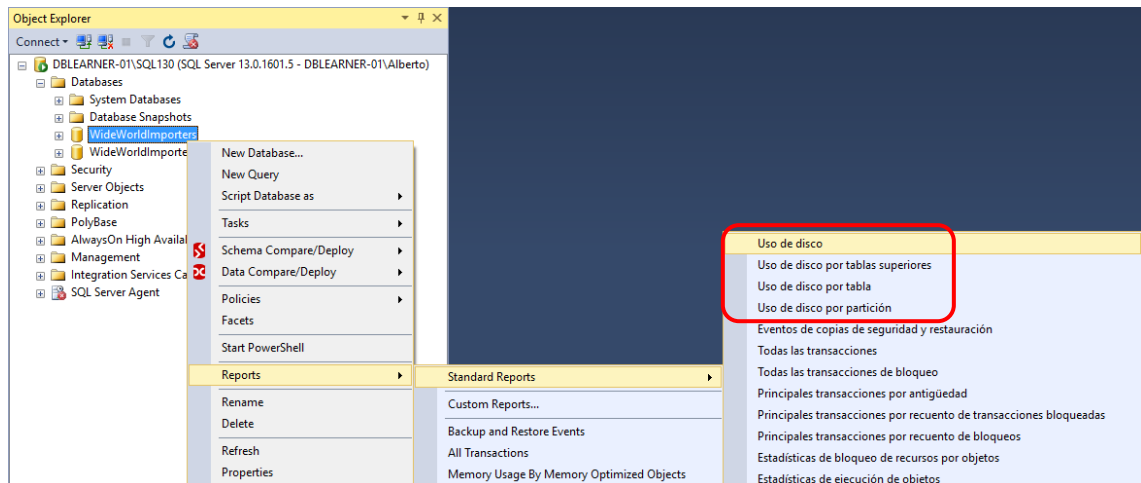
**Nota**

Los campos *TotalExtents* y *UsedExtents* están expresados en extensiones de 64Kb. Si se desea obtener la información en otra medida se deben realizar los cálculos respectivos.

3.5 Reportes del sistema

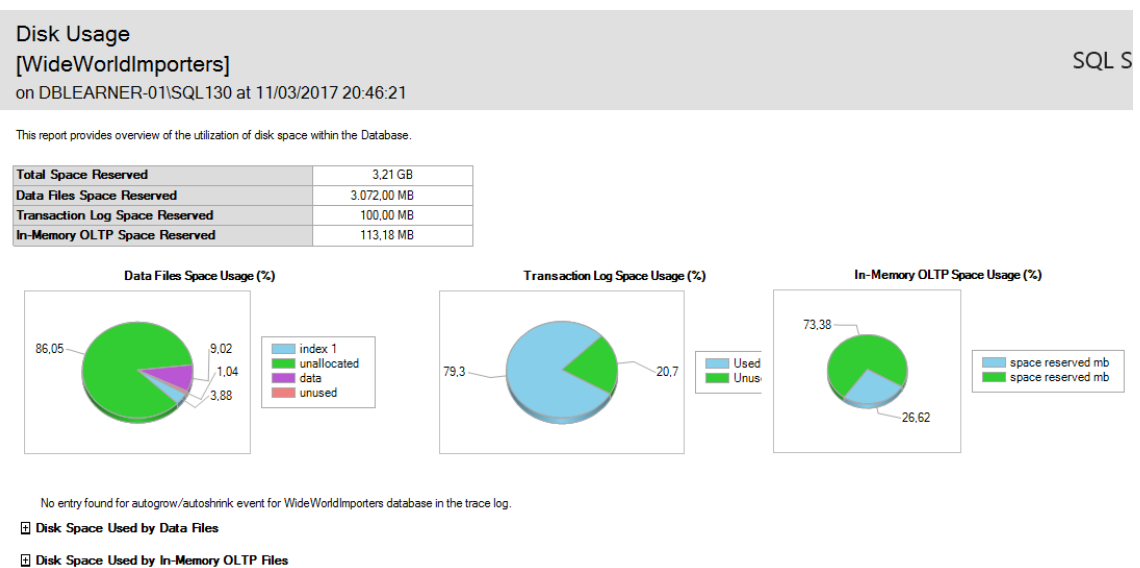
SQL server ofrece a los administradores una serie de reportes del sistema para administrar diferentes aspectos del motor de base de datos. Entre ellos se cuenta con reportes de utilización de disco, los cuales muestran información actualizada sobre diferentes aspectos al almacenamiento de la base de datos:

- Uso de disco
- Uso de disco por tablas superiores
- Uso de disco por tablas
- Uso de disco por partición



Reporte de uso de disco

Este informe proporciona un desglose numérico y gráfico del uso de espacio de los archivos del registro de transacciones y de datos. El uso de espacio de los archivos de datos se categoriza como un porcentaje asignado a las páginas de índice, el espacio sin asignar, a los datos y al espacio sin usar.



Índice. - Cantidad de espacio en disco utilizada para contener las páginas de índice.

Sin asignar. -La cantidad de espacio en disco disponible para la base de datos pero que aún no se ha asignado a ningún objeto.

Datos. -La cantidad de espacio en disco utilizada por páginas de datos.

No utilizado. -La cantidad de espacio de disco asignado a uno o más objetos, pero que aún no se ha usado.

Reporte de uso de disco por tablas superiores

Este informe proporciona un detalle numérico de la información de las primeras mil tablas de la base de datos. Además, indica el número de registros que contiene cada tabla, muestra el uso de espacio reservado, asignado, a los datos, a las páginas de índice y al espacio sin usar.

Disk Usage by Top Tables

[WideWorldImporters]

SQL Ser

on DBLEARNER-01\SQL130 at 11/03/2017 20:48:04

This report provides detailed data on the utilization of disk space by top 1000 tables within the Database. The report does not provide data for memory optimized tables.

Table Name	# Records	Reserved (KB)	Data (KB)	Indexes (KB)	Unused (KB)
Sales.Invoices	70.510	105.360	90.856	14.128	376
Warehouse.ColdRoomTemperatures_Archive	3.654.736	78.408	76.040	592	1.776
Sales.OrderLines	231.412	71.928	38.240	33.200	488
Sales.InvoiceLines	228.265	53.288	41.552	11.504	232
Warehouse.StockItemTransactions	236.667	52.672	2.720	37.648	12.304
Sales.CustomerTransactions	97.147	26.248	8.944	12.536	4.768
Sales.Orders	73.595	11.752	5.504	6.104	144
Sales.OrdersDemo	69.391	5.528	5.240	8	280
Application.Cities	37.940	4.880	3.960	896	24
Purchasing.SupplierTransactions	2.438	4.296	248	688	3.360
Purchasing.PurchaseOrderLines	8.367	2.216	1.248	832	136
Application.Countries	190	1.952	1.744	72	136
Application.People_Archive	961	1.728	232	184	1.312
Warehouse.StockItems_Archive	444	1.080	168	112	800
Application.People	1.111	1.064	624	312	128
Sales.Customers	663	976	304	344	328
Application.StateProvinces	53	680	384	64	232
Application.Countries_Archive	36	656	480	16	160
Purchasing.Suppliers	13	576	8	120	448
Warehouse.StockItems	227	496	112	128	256

Registros. - Cantidad de líneas de la tabla.

Reservado. - La suma de los tres campos siguientes.

Datos. - La cantidad de espacio en disco utilizada por páginas de datos.

Índice. - Cantidad de espacio en disco utilizada para contener las páginas de índice.

No utilizado. - La cantidad de espacio de disco asignado a uno o más objetos, pero que aún no se ha usado.

Reporte de uso de disco por tablas

Este informe es idéntico al anterior, ya que muestra el detalle numérico de la información, pero de todas las tablas de la base de datos. Además de indicar el número de registros que contiene cada tabla, muestra el uso de espacio reservado, asignado a los datos, a las páginas de índice y al espacio sin usar.

Disk Usage by Table [WideWorldImporters]

SQL Ser

on DBLEARNER-01\SQL130 at 11/03/2017 20:48:59

This report provides detailed data on the utilization of disk space by tables within the Database. The report does not provide data for memory optimized tables.

Table Name	# Records	Reserved (KB)	Data (KB)	Indexes (KB)	Unused (KB)
Application.Cities	37.940	4.880	3.960	896	24
Application.Cities_Archive	28	72	8	8	56
Application.Countries	190	1.952	1.744	72	136
Application.Countries_Archive	36	656	480	16	160
Application.DeliveryMethods	10	144	8	24	112
Application.DeliveryMethods_Archive	1	72	8	8	56
Application.PaymentMethods	4	144	8	24	112
Application.PaymentMethods_Archive	1	72	8	8	56
Application.People	1.111	1.064	624	312	128
Application.People_Archive	961	1.728	232	184	1.312
Application.StateProvinces	53	680	384	64	232
Application.StateProvinces_Archive	104	400	256	16	128
Application.SystemParameters	1	216	8	40	168
Application.TransactionTypes	13	144	8	24	112
Application.TransactionTypes_Archive	1	72	8	8	56
dbo.AppPeople	155	136	16	8	112
dbo.LogCambiosDDL	3	72	8	8	56
Purchasing.PurchaseOrderLines	8.367	2.216	1.248	832	136
Purchasing.PurchaseOrders	2.074	416	136	192	88

Registros. - Cantidad de líneas de la tabla.

Reservado. - La suma de los tres campos siguientes.

Datos. - La cantidad de espacio en disco utilizada por páginas de datos.

Índice. - Cantidad de espacio en disco utilizada para contener las páginas de índice.

No utilizado. - La cantidad de espacio de disco asignado a uno o más objetos, pero que aún no se ha usado.

Reporte de uso de disco por partición

Este informe proporciona un detalle numérico de la información de las tablas de la base de datos, pero detallando las particiones de los objetos. Indica información más resumida, detallando solo el número de registros que contiene cada partición, el espacio reservado y el espacio utilizado.

Disk Usage by Partition

[WideWorldImporters]

SQL Ser

on DBLEARNER-01\SQL130 at 11/03/2017 20:49:56

This report provides detailed data on the utilization of disk space by index and by partitions within the Database. The report does not provide data for memory optimized tables.

Table Name	# Records	Reserved (KB)	Used (KB)
[-] Application.Cities	37.940	4.880	4.856
[-] Index (PK_Application_Cities)	37.940	3.976	3.976
[-] Index (FK_Application_Cities_StateProvinceID)	37.940	904	880
[-] Application.Cities_Archive	28	72	16
[-] Index (ix_Cities_Archive)	28	72	16
[-] Application.Countries	190	1.952	1.816
[-] Index (PK_Application_Countries)	190	1.808	1.760
[-] Index (UQ_Application_Countries_FormalName)	190	72	40
[-] Index (UQ_Application_Countries_CountryName)	190	72	16
[-] Application.Countries_Archive	36	656	496
[-] Index (ix_Countries_Archive)	36	656	496
[-] Application.DeliveryMethods	10	144	32
[-] Index (PK_Application_DeliveryMethods)	10	72	16
[-] Index (UQ_Application_DeliveryMethods_DeliveryMethodName)	10	72	16
[-] Application.DeliveryMethods_Archive	1	72	16
[-] Index (ix_DeliveryMethods_Archive)	1	72	16
[-] Application.PaymentMethods	4	144	32
[-] Index (PK_Application_PaymentMethods)	4	72	16
[-] Index (UQ_Application_PaymentMethods_PaymentMethodName)	4	72	16
[-] Application.PaymentMethods_Archive	1	72	16
[-] Index (ix_PaymentMethods_Archive)	1	72	16

Registros. - Cantidad de líneas de la tabla.

Reservado. - Espacio total reservado para el objeto, incluye datos, índices y espacio no utilizado.

Utilizado. - Espacio utilizado por los datos e índices del objeto.

4. Tecnologías de almacenamiento de bases de datos

Tecnologías de almacenamiento de bases de datos

Direct Attached Storage (DAS)

- Método tradicional de almacenamiento.
- Disco duro interno.
- No está en red. Directamente conectado al servidor de base de datos.
- Uso exclusivo por parte del servidor donde está instalado.
- Pueden ser uno o varios.
- Económico, simple de instalar y administrar.
- Único punto de falla junto con el servidor.

6 - 26 Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.

Como administradores de bases de datos, una de las tareas claves que se deben atender es manejar los requerimientos de almacenamiento de las bases de datos.

Constantemente los administradores se encuentran añadiendo más espacio de disco en un servidor o tratando de aprovechar el espacio no utilizado en otros servidores. Por lo general, estas tareas siempre terminan con el deseo del administrador por encontrar una manera más eficiente de gestionar el almacenamiento. A continuación, se revisarán las diferentes tecnologías con las que se cuenta hoy en día para almacenar las bases de datos.

4.1. Direct Attached Storage (DAS)

DAS o Almacenamiento directamente conectado es la forma tradicional de almacenamiento, que se utilizó desde los inicios de la computación. Este tipo de almacenamiento no está en red, sino que está directamente conectado a un servidor. El almacenamiento puede estar conformado por uno o varios discos, los cuales están conectados a un controlador de discos que en conjunto forman parte del servidor.

La principal ventaja de DAS es la facilidad de instalación y administración, además de que son relativamente económicos. Sin embargo, al no estar el almacenamiento de tipo DAS compartido entre servidores, su uso está limitado a equipos individuales. Por lo tanto, si se requiere aumentar el almacenamiento, este se tiene que planificar individualmente por servidor y por lo general, implica instalar varios discos.

Con el almacenamiento de tipo DAS también se corre el riesgo de consumir el espacio total de disco. En estos casos, aumentar la capacidad puede ser complicado y no tan rápido como se desearía.

Con almacenamiento NAS o SAN se pueden evitar estos problemas.

4.2. Network Attached Storage (NAS)

NAS o almacenamiento adjuntado a la Red es un almacenamiento compartido conectado a la red. Los equipos de NAS son simples de instalar. Inicialmente NAS es más costoso de implementar comparado con DAS (y menos costoso que implementar una SAN). Sin embargo, el almacenamiento centralizado y compartido de NAS se convierte con el tiempo más económico y simple de administrar comparado con DAS.

NAS provee alta capacidad de almacenamiento, facilidad para compartir datos, consolidación de recursos y fácil acceso a archivos para múltiples equipos o clientes.

NAS no siempre es adecuado para implementaciones de SQL Server y siempre se deben comparar las implementaciones de DAS vs NAS.

Otra limitación de NAS es que, al ser una solución de almacenamiento de red, el desempeño de entrada y salida (I/O) está limitado por la velocidad y tráfico de la conexión de red LAN.

4.3. Storage Area Networks (SAN)

Una SAN o Red de área de almacenamiento es una red independiente, separada de la red LAN o WAN, específica para conectar dispositivos de almacenamiento. Es como tener una red adicional, en la que solo se encuentran discos para ser utilizados por los servidores.

Las redes SAN son muy confiables, escalables y tolerantes a fallas. SAN provee mayor disponibilidad que NAS debido a que todos los dispositivos de almacenamiento están disponibles para todos los servidores. Los usuarios finales se benefician de la capacidad optimizada de la red. SAN es muy eficiente cuando se requieren mover grandes bloques de datos. Son aplicables para bases de datos muy grandes (VLD – Very Large database) o para aplicaciones de misión crítica que consumen mucho ancho de banda.

Sin embargo, SAN es complejo de implementar y administrar, y su implementación inicial es muy costosa. Las primeras implementaciones de SAN requerían fibra óptica, lo cual elevaba su costo y complejidad.

iSCSI SANs

Con la llegada de la tecnología iSCSI SAN, el costo de implementar una SAN ha descendido considerablemente. Además, iSCSI SAN usa infraestructura Ethernet estándar para transmitir los datos entre los dispositivos.

Hay que considerar que, aunque iSCSI SAN ofrece gran flexibilidad en almacenamiento remoto comparado con SAN de canal de fibra, son más lentas y no son mucho más sencillas de administrar.

4.4. Solid State Disk (SSD)

SSD o discos de estado sólido es el último avance en tecnología de almacenamiento. Basada en la tecnología Flash introducida en los años 90s, debido a su alto costo solo era utilizada por organizaciones militares y aeroespaciales de las potencias mundiales. Pero con el tiempo, el descenso en los costos ha llevado esta tecnología al mundo empresarial. Sus primeras aplicaciones fueron en laptops portátiles de alta gama.

SSD utiliza una conexión Serial ATA (SATA) y usa la memoria de estado sólido para almacenar los datos. Adicionalmente a la memoria Flash, SSD puede usar RAM estática (SRAM) o DRAM. Debido a que los SSD imitan a los discos duros, pueden fácilmente reemplazarlos.

SSD provee numerosas ventajas comparado con tecnologías de almacenamiento tradicional:

- Debido a que no hay movimientos mecánicos involucrados, el acceso a datos es mucho más veloz.
- La ausencia de movimientos mecánicos también aumenta su confiabilidad debido a la menor probabilidad de falla.
- SSD usa menos corriente que otras opciones de almacenamiento.
- SSD requiere menor necesidad de enfriamiento.

Como desventajas se pueden considerar el costo y las actuales capacidades limitadas de las unidades SSD, pero estos son temas que se irán resolviendo con el avance de la tecnología.

4.5. Redundant Array of Inexpensive Discs (RAID)

RAID (matriz redundante de discos independientes) es un sistema de discos, denominado matriz, que contiene varias unidades de disco para proporcionar mayor rendimiento, confiabilidad y capacidad de almacenamiento a un costo menor. Las matrices con tolerancia a errores están divididas en seis niveles RAID: del 0 al 5.

Cada nivel utiliza un algoritmo diferente para implementar la tolerancia a errores. Si bien RAID no forma parte de SQL Server, su implementación puede afectar directamente al rendimiento de SQL Server. Con SQL Server se utilizan normalmente los niveles RAID 0, 1 y 5.

Una matriz de discos de hardware mejora el rendimiento de E/S debido a que las funciones de E/S, tales como la creación de bandas y reflejos, se controlan de forma eficaz mediante el firmware. Por el contrario, una RAID basada en el sistema operativo representa un menor costo, pero consume ciclos del procesador. Cuando el costo representa un problema y se necesita redundancia y alto rendimiento, los volúmenes RAID-5 son una buena solución.

La creación de bandas de datos (RAID 0) es la configuración RAID con mejor rendimiento, pero si un disco tiene errores, se deja de tener acceso a todos los datos del conjunto de bandas. Una técnica común de instalación de los sistemas de administración de bases de datos relacionales consiste en configurar la base de datos en una unidad RAID 0 y a continuación, colocar el registro de

transacciones en una unidad reflejada (RAID 1). Se puede obtener el mejor rendimiento de E/S de disco para la base de datos y mantener la posibilidad de recuperar los datos a través de un registro de transacciones reflejado, siempre y cuando, se realice copias de seguridad de la base de datos con regularidad.

Si es necesario recuperar rápidamente los datos, se debe tener en cuenta la posibilidad de reflejar el registro de transacciones y colocar la base de datos en un disco RAID 5, el cual proporciona redundancia de todos los datos de la matriz. Esto permite que se generen errores en un solo disco y la mayoría de las veces se reemplace sin que se produzca tiempo de inactividad del sistema. Además, RAID 5 ofrece un rendimiento menor, en comparación con RAID 0 o RAID 1, pero proporciona una confiabilidad mayor y una recuperación más rápida.