



Creación de bases de datos SQL Server 2005

En este capítulo veremos cómo crear una base de datos SQL Server 2005, cómo administrar sus archivos, y qué son las instantáneas de bases de datos.

Esta página se ha dejado en blanco intencionalmente.

Capítulo 3

Creación de bases de datos SQL Server 2005

Contenido

- ❑ *Archivos de una base de datos*
 - ✓ *¿Qué es una transacción?*
 - ✓ *¿Cómo trabaja el registro de transacciones?*
- ❑ *Creación de una base de datos*
 - ✓ *La instrucción CREATE DATABASE*
 - ✓ **Ejercicio 5:** *Creación de una base de datos con las propiedades predeterminadas*
 - ✓ **Ejercicio 6:** *Creación de una base de datos indicando las propiedades de los archivos de bases de datos*
 - ✓ *El procedimiento sp_helpdb*
 - ✓ *Configuración de una base de datos*
 - ✓ *El procedimiento sp_dboption*
 - ✓ **Ejercicio 7:** *Configuración de una base de datos (sp_dboption)*
 - ✓ **Ejercicio 8:** *Configuración de una base de datos (ALTER DATABASE)*
 - ✓ *La función DatabaseProperty*
 - ✓ *Creación de una base de datos removable*
 - ✓ *El procedimiento sp_create_removable*
 - ✓ *El procedimiento sp_certify_removable*
 - ✓ **Ejercicio 9:** *Creación de una base de datos removable*

- ❑ *Administración de los archivos de una base de datos*
 - ✓ *La instrucción ALTER DATABASE*
 - ✓ **Ejercicio 10:** *Administración de los archivos de una base de datos*
 - *Incremento del tamaño de un archivo de base de datos*
 - *Adición de un archivo a una base de datos*
 - *Eliminación de un archivo de una base de datos*
- ❑ *Reducción del tamaño de una base de datos*
 - ✓ *La instrucción DBCC SHRINKFILE*
 - ✓ *La instrucción DBCC SHRINKDATABASE*
 - ✓ **Ejercicio 11:** *Reducción del tamaño de una base de datos*
- ❑ *Eliminación de una base de datos*
 - ✓ *La instrucción DROP DATABASE*
 - ✓ **Ejercicio 12:** *Eliminación de bases de datos*
- ❑ *Instantáneas de una base de datos*
 - ✓ *Creación de una instantánea de base de datos*
 - ✓ **Ejercicio 13:** *Creación de una instantánea de base de datos*

Creación de bases de datos SQL Server 2005

Una base de datos es una colección de tablas que contienen datos, y además otros objetos como vistas, índices, procedimientos almacenados, funciones definidas por el usuario, y desencadenantes, definidos para soportar las operaciones a ejecutar con los datos. La data almacenada en una base de datos está generalmente relacionada con un objetivo ó proceso en particular, tales como control del inventario en un almacén, ó registro del personal de una organización.

Archivos de una base de datos

Una base de datos SQL Server 2005 está formada por tres tipos de archivos:

Archivo primario

Es el archivo de inicio de una base de datos. Almacena las tablas de sistema de la base de datos, y puede también almacenar las tablas de usuario y otros objetos de la base de datos. Cada base de datos tiene solo un archivo primario, siendo **.mdf** la extensión del archivo.

Archivos secundarios

Estos archivos se usan para almacenar las tablas de usuario y demás objetos de la base de datos. Si el archivo primario puede almacenar todos los objetos, entonces el uso de los archivos secundarios no es necesario. Sin embargo en ocasiones, la base de datos puede ser muy grande siendo necesario distribuir los objetos en múltiples archivos secundarios, pudiendo estar éstos distribuidos en diferentes discos. La extensión de los archivos secundarios es **.ndf**.

Archivos de registro de transacciones (transaction log)

Como su nombre indica, este archivo registra todas las transacciones a ejecutar sobre la base de datos, y es un mecanismo de seguridad para recuperar la base de datos ante la eventualidad de una falla en el sistema. Toda base de datos debe tener por lo menos un archivo de registro de transacciones. La extensión del archivo es **.ldf**.

¿Qué es una transacción?

Una **transacción** es un conjunto de modificaciones de datos (operaciones) que debe ser procesado como una unidad.

Por ejemplo, se tiene las tablas **factura**, **detalle_factura** y **producto** de una base de datos de **ventas**. La tabla **producto** registra el nivel de inventario (stock) de cada producto.

Supongamos que se tiene que registrar la venta de 10 unidades del producto ABC. Para ello, la aplicación procede de la siguiente manera:

Operación 1 Registra en la tabla **factura**, los datos de la cabecera de la factura.

Operación 2 Registra en la tabla **detalle_factura**, los detalles de la venta.

Operación 3 Actualiza el nivel de inventario del producto ABC en la tabla **producto**.

Pregunta:

¿Qué ocurriría si durante la ejecución de la aplicación se completan las operaciones 1 y 2, y por una falla del sistema, la operación 3 no se lleva a cabo?

Respuesta:

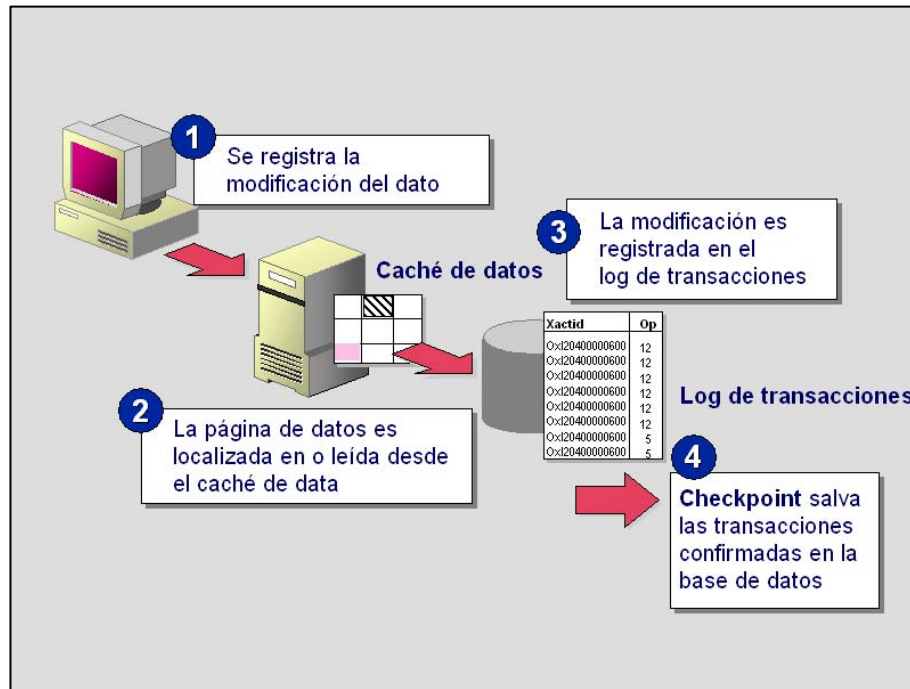
Si no se diseña un mecanismo para corregir el error, la base de datos perdería consistencia. Según las tablas **factura** y **detalle_factura** se han vendido 10 unidades del producto ABC, por lo que en la tabla **producto** deberíamos tener 10 unidades menos. Esto último no es cierto al no haberse completado la operación 3.

En otras palabras, las operaciones 1, 2 y 3 forman una transacción, y si ésta no se completa, el sistema debe deshacer todas las operaciones.

Una transacción asegura que las operaciones se llevarán a cabo completas, o en caso contrario, la transacción será anulada para garantizar la consistencia de los datos.

¿Cómo trabaja el registro de transacciones?

El archivo de registro de transacciones almacena todas las transacciones llevadas a cabo con las sentencias INSERT, UPDATE y DELETE. El proceso se lleva a cabo de la siguiente manera:



- Una aplicación envía una modificación de datos al servidor SQL.
- Las páginas de datos afectadas por la modificación son localizadas en el caché, o leídas desde el disco al caché, si no se encuentran en ella.
- Cada sentencia de modificación de datos es registrada en el transaction log antes que el cambio se lleve a cabo en la base de datos.
- El punto de control (**checkpoint**) escribe las transacciones confirmadas (committed) en la base de datos.
- Si el sistema falla, el proceso de recuperación automático usa el registro de de transacciones para recuperar todas las transacciones confirmadas y anula todas las transacciones incompletas.

Creación de una base de datos

Antes de crear una base de datos tenga en cuenta lo siguiente:

- Solo pueden crear bases de datos las cuentas de inicio de sesión que pertenecen a los roles fijos de servidor **sysadmin** y **dbcreator**, aunque el permiso se puede conceder a otras cuentas.
- La cuenta de inicio de sesión que crea la base de datos se convierte en el usuario dueño (**dbo**) de la base de datos.
- Defina el nombre y tamaño de la base de datos, los archivos (y sus propiedades) en los que residirá la base de datos.

La instrucción CREATE DATABASE

Sintaxis

```
CREATE DATABASE nombre_basedatos
ON [ PRIMARY ] (
    NAME = nombre_lógico_data ,
    FILENAME = 'ubicación_y_nombre_archivo_data' ,
    SIZE = tamaño [ KB|MB|GB|TB ] ,
    MAXSIZE = tamaño_máximo [ KB|MB|GB|TB|UNLIMITED ] ,
    FILEGROWTH = incremento_crecimiento [ KB|MB|% ] )
LOG ON (
    NAME = nombre_lógico_log ,
    FILENAME = 'ubicación_y_nombre_archivo_log' ,
    SIZE = tamaño [ KB|MB|GB|TB ] ,
    MAXSIZE = tamaño_máximo [ KB|MB|GB|TB|UNLIMITED ] ,
    FILEGROWTH = incremento_crecimiento [ KB|MB|% ] )
```

- La cláusula **ON PRIMARY** define las propiedades del archivo primario.
- La cláusula **LOG ON** define las propiedades del archivo de registro de transacciones.
- **nombre_lógico_data**, **nombre_lógico_log** es el nombre a utilizar cuando en una sentencia SQL se tiene que hacer referencia al archivo de datos o al archivo de log respectivamente.

- **ubicación_y_nombre_archivo** es una cadena que incluye la ruta y el nombre del archivo. La ruta debe especificar una carpeta existente en el servidor en el que está instalado SQL.
- **tamaño** especifica el tamaño del archivo.
- **tamaño_máximo** es el máximo tamaño que puede alcanzar el archivo si se requiriera de espacio adicional.
- **incremento_crecimiento** es la cantidad de espacio que se añade al archivo cada vez que se necesita espacio adicional. Se puede especificar como una magnitud constante en KB ó MB, ó como una tasa de crecimiento (%).

Ejercicio 5: Creación de una base de datos con las propiedades predeterminadas

1. En **SQL Server Management Studio**, en el menú **File** ejecute **New, New SQL Server Query**. Luego, seleccione el servidor SQL al que desea conectarse.

Nota: A partir de este punto, todos los ejercicios acerca de las instrucciones Transact-SQL los ejecutaremos en la ventana del Code Editor .
--

2. En el **Code Editor**, digite las siguientes instrucciones:

```
USE master  
go
```

```
CREATE DATABASE Prueba1  
go
```

3. Ejecute las instrucciones. Si no hay errores recibimos el mensaje

```
Command(s) completed successfully.
```

4. En el **Object Explorer**, expanda la carpeta **Databases**. Se debe mostrar la base de datos recién creada **Prueba1**.

Nota: Si en el **Object Explorer** no se mostrara la base de datos creada, haga un clic secundario sobre la carpeta **Databases**, y luego ejecute **Refresh**.

5. Para revisar las propiedades de la base de datos, haga un clic secundario sobre el nombre de la base de datos, y luego ejecute **Properties**.
6. Observe en el diálogo **Database Properties** que la ubicación predeterminada es la carpeta **Data** dentro de la carpeta de instalación de la instancia de SQL Server. Además, la base de datos ha asumido las propiedades de la base de datos de sistema **model**.
7. Cierre el diálogo **Database Properties**.

Ejercicio 6: Creación de una base de datos indicando las propiedades de los archivos de bases de datos

1. En el **Code Editor**, digite y ejecute las siguientes instrucciones SQL:

```
USE master
go

CREATE DATABASE MarketPERU
ON PRIMARY(
    NAME = MarketPERU_data ,
    FILENAME = 'D:\SQLServer2005\Data\MarketPERU.mdf' ,
    SIZE = 8MB ,
    MAXSIZE = 15MB,
    FILEGROWTH = 1MB )
LOG ON(
    NAME = MarketPERU_log ,
    FILENAME = 'D:\SQLServer2005\Data\MarketPERU_log.ldf' ,
    SIZE = 3MB ,
    MAXSIZE = 8MB,
    FILEGROWTH = 10% )
go
```

Notas:

1. Se asume que la ruta **D:\SQLServer2005\Data** ya existe en el disco del servidor.
 2. Cuando se crea una base de datos se añade una entrada con el nombre de la base de datos en la tabla de sistema **sysdatabases** de la base de datos de sistema **master**.
-
2. Para revisar la tabla **sysdatabases**, digite y ejecute el siguiente procedimiento almacenado de sistema:

```
EXEC sp_helpdb  
go
```

	name	db_size	owner	dbid	created	status
1	AdventureWorks	134.25 MB	DESARROLLO\Administrator	8	May 28 2005	Status=ONLINE,...
2	AdventureWorksDw	67.13 MB	DESARROLLO\Administrator	7	May 28 2005	Status=ONLINE,...
3	MarketPERU	11.00 MB	DESARROLLO\Administrator	10	Jun 7 2005	Status=ONLINE,...
4	master	6.31 MB	sa	1	Apr 8 2003	Status=ONLINE,...
5	model	1.63 MB	sa	3	Apr 8 2003	Status=ONLINE,...
6	msdb	4.94 MB	sa	4	Jul 20 2004	Status=ONLINE,...
7	Prueba1	1.62 MB	DESARROLLO\Administrator	9	Jun 7 2005	Status=ONLINE,...
8	ReportServer\$PERUDev2005	3.88 MB	DESARROLLO\Administrator	5	May 28 2005	Status=ONLINE,...
9	ReportServer\$PERUDev2005TempDB	2.88 MB	DESARROLLO\Administrator	6	May 28 2005	Status=ONLINE,...
10	tempdb	8.75 MB	sa	2	Jun 7 2005	Status=ONLINE,...

El procedimiento sp_helpdb

Entrega información acerca de todas las bases de datos del servidor, ó de una base de datos específica.

Sintáxis

```
sp_helpdb [ nombre_basedatos ]
```

sp_help ejecutado sin el argumento entrega información acerca de todas las bases de datos del servidor.

Configuración de una base de datos

En algunas ocasiones necesitará cambiar la configuración de la base de datos para poder realizar alguna tarea especial. Si desea conocer qué opciones puede configurar para su base de datos, ejecute el procedimiento almacenado de sistema **sp_dboption**.

Algunas de las opciones de bases de datos más utilizadas se muestran en el siguiente cuadro:

Opción de base de datos	Descripción
autoclose	Cuando está en true , la base de datos es cerrada, y sus recursos liberados, después que el último usuario se desconecta.
autoshrink	Cuando está en true , los archivos de la base de datos reducen su tamaño automáticamente en forma periódica.
concat null yields null	Cuando está en true , si algunos de los operandos de una concatenación es NULL, el resultado es NULL.
dbo use only	Cuando está en true , solo el dueño de la base de datos la puede utilizar.
offline	Cuando está en true (on) , la base de datos está fuera de línea y no puede ser utilizada. Cuando está en false (off) , la base de datos está en línea.
read only	Cuando está en true , la base de datos es de solo lectura y no se permite ningún cambio en ella.
single user	Cuando está en true , solo un usuario a la vez puede acceder a la base de datos.

Más adelante veremos otras opciones de bases de datos a medida que vayamos avanzando con los temas del presente libro.

El procedimiento sp_dboption

Permite revisar y cambiar las opciones de configuración de una base de datos.

Sintaxis

```
sp_dboption [ 'nombre_basedatos'  
            [ , 'opción_basedatos', true|false ] ]
```

- **opción_basedatos** es el nombre de la opción de base de datos cuyo valor se desea configurar a **true** ó a **false**.

Notas:

El procedimiento **sp_dboption** será eliminado de las versiones futuras de SQL Server.

Para configurar las opciones de una base de datos utilice la instrucción ALTER DATABASE, que ahora incluye la cláusula SET para modificar las opciones de una base de datos.

Ejercicio 7: Configuración de una base de datos (sp_dboption)

En este ejercicio vamos a revisar y cambiar la configuración de la base de datos **Prueba1** creada en el Ejercicio 4.

Para ver la lista de opciones configurables de una base de datos

En el **Code Editor** ejecute la siguiente instrucción:

```
EXEC sp_dboption  
go
```

La ejecución del procedimiento **sp_dboption** sin parámetros solo muestra la lista de opciones configurables para una base de datos sin especificar cuál es el estado de cada opción.

Para ver las opciones configuradas a true en una base de datos específica

A continuación vamos a ver la configuración de la base de datos **Prueba1**.

```
EXEC sp_dboption Prueba1  
go
```

El resultado es el siguiente:

The following options are set:	
1	auto create statistics
2	auto update statistics

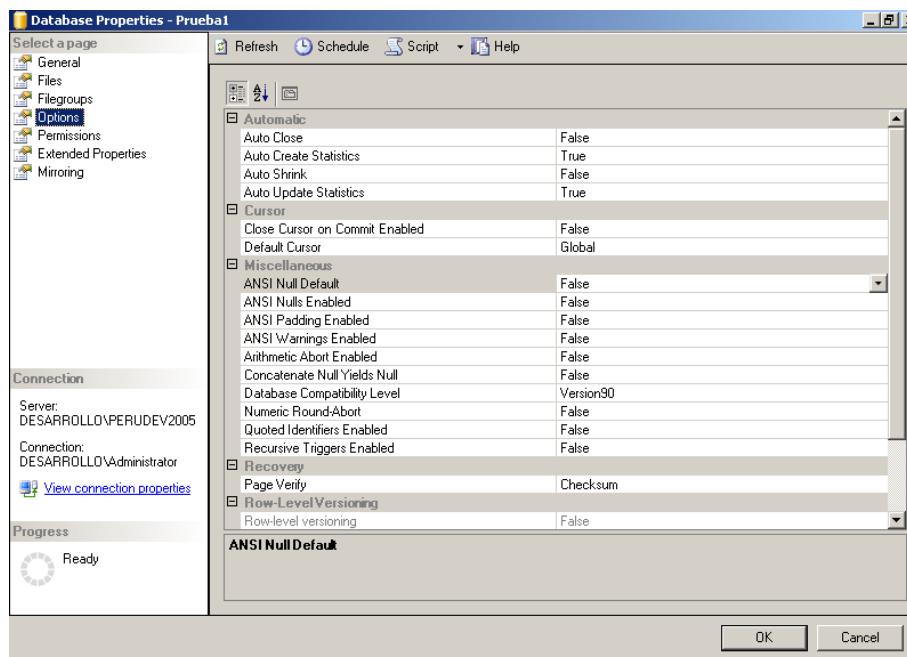
Para cambiar el estado de una opción de base de datos

Ahora, vamos a establecer la base de datos **Prueba1** como de solo lectura.

```
EXEC sp_dboption Prueba1, 'read only', true  
EXEC sp_dboption Prueba1  
go
```

The following options are set:	
1	read only
2	auto create statistics
3	auto update statistics

También podemos revisar y cambiar las opciones de configuración de una base de datos desde la página **Options** del dialogo **Database Properties** de la base de datos.



Ejercicio 8: Configuración de una base de datos (ALTER DATABASE)

En el ejercicio anterior, configuramos la base de datos **Prueba1** como de solo lectura. En este ejercicio haremos uso de ALTER DATABASE para configurar la base de datos como de lectura-escritura.

1. En el **Code Editor** ejecute la siguiente instrucción:

```
ALTER DATABASE Prueba1
    SET Read_Write
go
```


2. Para verificar la base de datos haremos uso de la función **DatabaseProperty**:

```
SELECT DatabaseProperty('Prueba1', 'isReadOnly')  
go
```

El resultado de la función es 0 (cero), lo que indica que **isReadOnly** está en **False**.

La función DatabaseProperty

Retorna el valor de la opción de configuración de la base de datos especificada.

Sintaxis

```
DatabaseProperty( 'nombre_basedatos' , 'opción_basedatos' )
```

- **opción_basedatos** es el nombre de la opción de base de datos cuyo valor se desea obtener.

Creación de una base de datos removable

SQL Server 2005 permite crear una base de datos que se puede distribuir a otros usuarios a través de medios removibles tales como los CD-ROM.

Para crear una base de datos removable utilice el procedimiento almacenado de sistema **sp_create_removable**. Este procedimiento crea tres ó más archivos para la base de datos:

- un archivo para las tablas de sistema,
- un archivo para el registro de transacciones, y
- uno ó más archivos secundarios para los datos

El procedimiento `sp_create_removable`

Sintaxis

```
sp_create_removable 'nombre_basedatos' ,  
    'nombre_lógico_sistema',  
    'ubicación_y_nombre_archivo_sistema', tamaño_MB,  
    'nombre_lógico_log',  
    'ubicación_y_nombre_archivo_log', tamaño_MB,  
    'nombre_lógico_data',  
    'ubicación_y_nombre_archivo_data', tamaño_MB
```

- **nombre_lógico_sistema**, **nombre_lógico_data**, **nombre_lógico_log** es el nombre a utilizar cuando en una sentencia SQL se tiene que hacer referencia al archivo de tablas de sistema, al archivo de log ó al archivo de datos respectivamente.
- **ubicación_y_nombre_archivo** es una cadena que incluye la ruta y el nombre del archivo. La ruta debe especificar una carpeta existente en el servidor en el que está instalado SQL.
- **tamaño_MB** especifica el tamaño en MB del archivo.

Para verificar si una base de datos removable está configurada adecuadamente para ser copiada a un medio removable, ejecute el procedimiento **sp_certify_removable**.

El procedimiento `sp_certify_removable`

Sintaxis

```
sp_certify_removable 'nombre_basedatos' [ , AUTO ]
```

- Si se especifica **AUTO**, se le entrega la propiedad de la base de datos y de todos los objetos de la base de datos al administrador del sistema, eliminando cualquier usuario de base de datos que haya sido creado, y eliminando los permisos que no son predeterminados.

Ejercicio 9: Creación de una base de datos removable

1. En el **Code Editor** ejecute las siguientes instrucciones:

```
sp_create_removable 'RemovableDB',  
    'Rem_sys', 'D:\SQLServer2005\Data\Rem_sys.mdf', 3,  
    'Rem_log', 'D:\SQLServer2005\Data\Rem_log.ldf', 3,  
    'Rem_dat', 'D:\SQLServer2005\Data\Rem_dat.ndf', 7  
go  
  
sp_helpdb  
go
```

2. Para verificar que la base de datos se puede distribuir en un medio removable, ejecute:

```
EXEC sp_certify_removable 'RemovableDB'  
go
```

3. Trate de copiar los archivos de la base de datos **RemovableDB** a otra carpeta en su disco duro.
4. Para poder distribuir la base de datos en un medio removable, previamente debe ser puesta fuera de línea para que ningún usuario pueda acceder a ella.

```
ALTER DATABASE RemovableDB  
    SET offline  
go
```

5. Ahora, trate nuevamente de copiar los archivos a otra carpeta en su disco duro.

Administración de los archivos de una base de datos

La instrucción ALTER DATABASE

Permite añadir ó eliminar archivos de una base de datos. Se puede utilizar también para modificar los atributos de los archivos, tales como el nombre ó el tamaño de un archivo. También se puede usar para cambiar el nombre de una base de datos, y para configurar las opciones de configuración de una base de datos.

Para ejecutar ALTER DATABASE debemos tener el permiso ALTER sobre la base de datos. Este permiso se concede de modo predeterminado al rol fijo de servidor **sysadmin**, y a los roles fijos de bases de datos **db_ddladmin** y **db_owner**.

Sintaxis

```
ALTER DATABASE nombre_basedatos
    ADD FILE especificación_archivo , ...
    | ADD LOG FILE especificación_archivo
    | REMOVE FILE nombre_lógico_archivo
    | MODIFY FILE especificación_archivo
    | MODIFY NAME nuevo_nombre_basedatos
    | SET opción_configuración_basedatos , ...
```

- **especificación_archivo** son los atributos del archivo de base de datos tal como se definieron en el comando CREATE DATABASE.
- **opción_configuración_basedatos** es una de las opciones de configuración de base de datos definidas en el procedimiento **sp_dboption**.

Nota: Las opciones de configuración de una base de datos se pueden establecer también con la instrucción ALTER DATABASE. Esta versión aún soporta el procedimiento **sp_dboption**, pero no es seguro que versiones posteriores la soporten.

Ejercicio 10: Administración de los archivos de una base de datos

Incremento del tamaño de un archivo de base de datos

1. Primero averiguamos que tamaño tienen los archivos de la base de datos **MarketPERU**.

```
USE Master
go

sp_helpdb MarketPERU
go
```

	name	db_size	owner	dbid	created
1	MarketPERU	11.00 MB	DE SARROLLO\...	10	Jun 7 2005

	name	fileid	filename	filegroup	size
1	MarketPERU_data	1	D:\SQLServer20...	PRIMARY	8192 KB
2	MarketPERU_log	2	D:\SQLServer20...	NULL	3072 KB

2. Ahora, cambiamos el tamaño del archivo primario incrementándolo a 15 MB.

```
ALTER DATABASE MarketPERU
    MODIFY FILE(
        NAME = MarketPERU_data ,
        SIZE = 15MB )
go

sp_helpdb MarketPERU
go
```

	name	db_size	owner	dbid	created
1	MarketPERU	18.00 MB	DE SARROLLO\...	10	Jun 7 2005

	name	fileid	filename	filegroup	size
1	MarketPERU_data	1	D:\SQLServer20...	PRIMARY	15360 KB
2	MarketPERU_log	2	D:\SQLServer20...	NULL	3072 KB

Adición de un archivo a una base de datos

1. Para añadir un archivo secundario a la base de datos **MarketPERU**:

```
ALTER DATABASE MarketPERU
ADD FILE(
    NAME = MarketPERU_sec1 ,
    FILENAME = 'D:\SQLServer2005\Data\MarketPERU_sec1.ndf' ,
    SIZE = 5MB ,
    MAXSIZE = 10MB ,
    FILEGROWTH = 1MB )
go
```

Eliminación de un archivo de una base de datos

1. Para eliminar un archivo de la base de datos **MarketPERU**:

```
ALTER DATABASE MarketPERU
REMOVE FILE MarketPERU_sec1
go
```

La eliminación de un archivo de base de datos solo se puede ejecutar cuando el archivo está vacío.

Reducción del tamaño de una base de datos

La instrucción DBCC SHRINKFILE

Reduce el tamaño de un archivo de base de datos para la base de datos activa.

Sintáxis

```
DBCC SHRINKFILE( nombre_lógico_archivo, tamaño_final_enMB )
```

Solo pueden ejecutar esta instrucción los miembros del rol fijo de servidor **sysadmin** y del rol fijo de base de datos **db_owner**.

La instrucción DBCC SHRINKDATABASE

Reduce el tamaño de los archivos de datos de la base de datos especificada.

Sintáxis

```
DBCC SHRINKDATABASE( nombre_basedatos ,  
                     porcentaje_espacio_libre_después_reducción )
```

Solo pueden ejecutar esta instrucción los miembros del rol fijo de servidor **sysadmin** y del rol fijo de base de datos **db_owner**.

Ejercicio 11: Reducción del tamaño de una base de datos

1. Para reducir el tamaño del archivo primario de la base de datos **MarketPERU** hasta 10 MB:

```
USE MarketPERU
go

DBCC SHRINKFILE( MarketPERU_data, 10 )
go
```

2. Para reducir toda la base de datos **MarketPERU** hasta que quede solo 50% de espacio libre:

```
DBCC SHRINKDATABASE( MarketPERU, 50 )
go
```

Eliminación de una base de datos

Cuando eliminamos una base de datos, se eliminan los archivos que conforman la base de datos. Si no se dispone de copia de seguridad de estos archivos, la base de datos se pierde definitivamente.

La instrucción DROP DATABASE

Esta instrucción elimina una ó más bases de datos de un servidor SQL. La operación elimina los archivos que conforman la base de datos. Sin embargo, si la base de datos está OFFLINE al momento de su eliminación, los archivos que la conforman no son eliminados. Estos archivos se pueden eliminar posteriormente desde el Explorador de Windows.

Sintaxis

```
DROP DATABASE nombre_basedatos , ...
```

Para ejecutar el comando DROP DATABASE debemos tener por lo menos el permiso CONTROL sobre la base de datos.

Ejercicio 12: Eliminación de bases de datos

1. La siguiente instrucción elimina las bases de datos: **MarketPERU**, **Prueba1** (una base de datos Read-Only), y **RemovableDB** (una base de datos que está OFFLINE).

```
DROP DATABASE MarketPERU, Prueba1, RemovableDB  
go
```

Instantáneas de una base de datos

Una instantánea de base de datos es una vista de la base de datos, estática y de solo lectura, hasta un instante de tiempo determinado. Se pueden crear varias instantáneas de una base de datos, y cada una de ellas es consistente hasta el momento en que se creó la instantánea.

Como una instantánea de base de datos contiene la data hasta el momento de creación de la instantánea, se puede utilizar para la creación de reportes con data hasta un momento específico en el tiempo.

Creación de una instantánea de base de datos

Para crear una instantánea de base de datos hacemos uso de la instrucción CREATE DATABASE con la siguiente sintaxis.

Sintaxis

```
CREATE DATABASE nombre_instantánea_basedatos
ON (
    NAME = nombre_lógico_archivo_basedatos_origen ,
    FILENAME = nombre_físico_archivo_instantánea )
AS SNAPSHOT OF nombre_basedatos_origen
```

Ejercicio 13: Creación de una instantánea de base de datos

El siguiente ejemplo crea una instantánea de la base de datos **AdventureWorks**. Observe que el nombre se ha especificado de modo que indica que representa una instantánea, e informa acerca del momento en que se creó la instantánea.

```
CREATE DATABASE AdventureWorks_instante_24Oct2005_1500
ON ( NAME = AdventureWorks_data,
    FILENAME = 'F:\Program Files\
    Microsoft SQL Server\MSSQL.1\MSSQL\Data\
    AdventureWorks_instante_01.ss')
AS SNAPSHOT OF AdventureWorks
go
```