

Capítulo II

Bases de Datos de Microsoft SQL Server 2012

En este capítulo conoceremos qué es una base de datos, qué objetos forman parte de ella, qué son las bases de datos de sistema y una introducción al papel que cumplen en SQL Server, cómo se crea una base de datos SQL Server y qué tipos de archivos la conforman, así como la función de cada tipo de archivo.

1. ¿QUÉ ES UNA BASE DE DATOS?

Una base de datos es un conjunto de datos organizados de modo tal que facilita las operaciones de almacenamiento, manipulación y consulta de los datos. La estructura de las bases de datos de Microsoft SQL Server está basada en el modelo entidad-relación y podemos describirla como un conjunto de tablas con datos relacionados, aunque la base de datos no solo almacena los datos sino también su estructura, y otros objetos auxiliares como índices, vistas, procedimientos almacenados, además de elementos de seguridad como los usuarios y permisos.

1.1. Objetos de una base de datos Microsoft SQL Server

Una base de datos relacional está formada por objetos de diversos tipos, cada uno de los cuales cumple un papel específico dentro de la estructura de la base de datos.

Tablas (tables), son los objetos en los que se almacenan los datos. Una tabla tiene un formato fila-columna similar a las tablas ó listas de Microsoft Excel. Cada fila es un registro único en la tabla (no existe una fila igual a otra en la misma tabla), y cada columna almacena los valores de propiedades específicas del registro. Por ejemplo, si tenemos una tabla EMPLEADO, cada fila almacena los datos de un único empleado, y en cada columna podemos almacenar su apellido paterno, su apellido materno, su nombre, su fecha de nacimiento, su estado civil, número de hijos, etc.

Columnas (columns), son las partes de la tabla que almacenan los datos. A cada columna debe asignársele un tipo de dato y un nombre o identificador único. Por ejemplo, en la tabla EMPLEADO podemos definir las columnas: *paterno* de tipo cadena, *materno* de tipo cadena, *nombre* de tipo cadena, *fecha_nac* de tipo fecha, *estado_civil* de tipo cadena, *hijos* de tipo numérico, etc.

Tipos de datos (datatypes), en una base de datos podemos almacenar distintos tipos de datos tales como cadenas, fechas y números; incluso podemos almacenar datos binarios. Cada columna de una tabla debe contener solo un tipo de datos.

Claves principales (primary keys), éstas son un elemento muy importante en las bases de datos. Una clave principal es un valor único que permite identificar a un registro de la tabla, diferenciándolo de otros registros. La clave principal de una tabla se almacena en una columna de la misma. Por costumbre, suele ser la primera columna de la tabla.

Claves foráneas (foreign keys), son columnas cuyos valores hacen referencia a las claves principales de otras tablas o de la misma tabla. Las bases de datos relacionales utilizan las claves principales y las claves foráneas para relacionar los datos almacenados en tablas distintas cuando se hacen consultas.

(1)

CATEGORIA		
IdCategoria	Nombre	Descripcion
1	GOLOSINAS	GALLETAS,CHOCOLATES,CARAMELOS,TOFFES
2	EMBUTIDOS	JAMONADAS,JAMONES,SALCHICHAS,CHORIZOS
3	HIGIENE PERSONAL	JABONES,P.DENTALES,SHAMPOOS,P.H.
4	LACTEOS	LECHES,YOGURES,QUESOS,MANTEQUILLAS
5	LICORES Y GASEOSAS	VINOS,WHISKIES,BEBIDAS GASIFICADAS
6	LIMPIEZA	DETERGENTES,DESINFECTANTES,ACCESORIOS

(3)

(1)

PRODUCTO			
IdProducto	IdCategoria	Nombre	
		UnidadMedida	
11	1	CHOCOLATE BARRA REGULAR	BARRA 2 ONZAS
65	3	JABON ROSAS Y LIMON BLANCO	UNIDAD X 105 GR
66	3	JABON ROSAS Y LIMON ROSADO	UNIDAD X 105 GR
101	4	MANTEQUILLA LAIVE C/SAL	PAQUETE 0.5 LB
102	4	MANTEQUILLA LAIVE	PAQUETE 114 GR
103	4	MANTEQUILLA FERM C/SAL	PAQUETE 227 GR
105	4	MARGARINA ASTRA	PAQUETE 230 GR
106	4	DORINA CLASICA	POTE 225 GR
10	1	WAFER CHOCOLATE FIELD	PAQUETE X 9 UNIDADES
13	1	CHOCOLATE BARRA MILKY WAY	BARRA 2.15 ONZAS
76	3	P.H. BLANCO ROLL KLEENEX	PAQUETE X 2 UNIDADES
18	1	GALLETAS CHIPS AHOY	PAQUETE X 6 UNIDADES
2	1	CARAMELOS SURTIDO DE FRUTAS	PAQUETE 450 GR
74	3	P.H. BLANCO SUAVE (ROJA)	PAQUETE X 4 UNIDADES
130	6	DETERGENTE LIMON INVICTO	BOLSA 225 GR

(1) Identificador de tabla.

(2) Identificador de columna.

(3) Clave primaria de la tabla CATEGORIA.

(4) Clave foránea de la tabla PRODUCTO que relaciona a cada producto con los datos de su categoría.

Restricciones (constraints), son elementos de la base de datos que definen mecanismos, con base en el servidor, para controlar la integridad de los datos. Las restricciones que podemos definir en una tabla de la base de datos son: clave principal (primary key), claves foráneas (foreign keys), valores únicos (unique), valores predeterminados (defaults), y reglas de validación (checks).

Procedimientos almacenados (stored procedures), son pequeños programas que se escriben en código Transact-SQL (el lenguaje de programación de Microsoft SQL Server) y se almacenan en el servidor bajo un nombre. Al ejecutar el procedimiento almacenado, se ejecuta el código Transact-SQL contenido en él. Podemos escribir código Transact-SQL que ejecuta un informe semanal, guardarlo en la base de datos como un procedimiento almacenado, y ejecutar el procedimiento almacenado cada vez que se necesita generar informes. También pueden emplearse como mecanismos de seguridad.

Desencadenantes ó disparadores (triggers), son una forma especial de los procedimientos almacenados que se ejecutan automáticamente cuando se agrega (INSERT), se modifica (UPDATE) o se elimina (DELETE) datos de una base de datos. También puede definirse triggers que se ejecutan cuando se modifica la definición de un objeto de la base de datos. La integridad de datos y las reglas de negocios se suelen definir mediante triggers. Por ejemplo, un trigger puede asegurar que todo producto en un almacén tenga una descripción, además del nombre.

Funciones definidas por el usuario (user-defined functions), adicionalmente a las funciones predefinidas proporcionadas por Transact-SQL, podemos crear nuestras propias funciones para ampliar la capacidad del lenguaje, y utilizarlas en nuestras soluciones de bases de datos.

Índices (indexes), son un elemento de la base de datos que cumple una función similar al índice alfabético ó temático de un libro. Los índices pueden ayudarle a organizar los datos a efecto de que las consultas se ejecuten con mayor rapidez.

Estadísticas (statistics), representan la distribución estadística de los elementos de una columna ó conjunto de columnas. Esta distribución es utilizada para establecer la selectividad de los datos en la columna ó conjunto de columnas, o en otras palabras, la eficiencia de un índice en la recuperación de datos.

Vistas (views), son consultas almacenadas en la base de datos que pueden hacer referencia a una o varias tablas. Pueden considerarse una forma especial de procedimiento almacenado que ejecuta una sola instrucción Transact-SQL, una sentencia SELECT. Puede crearlas y guardarlas a fin de utilizarlas con facilidad en el futuro. Se usan para ocultar a los usuarios columnas sensibles (columnas a las que no deben tener acceso); por lo regular las vistas excluyen ciertas columnas de una tabla, o bien, vinculan varias tablas.

Sinónimos (synonyms), permiten que un objeto pueda ser referenciado utilizando un nombre alternativo. A veces, para referirse a un objeto es necesario calificarlo completamente, lo que genera una referencia larga y compleja para el objeto. En este caso, podemos crear un sinónimo que permita establecer la referencia al objeto utilizando un nombre más corto y sencillo.

Tipos de datos definidos por el usuario (user-defined data types), son tipos de datos con nombre propio creados en base a los tipos de datos estándar de SQL Server. Se usan para estandarizar el proceso de creación de tablas de una base de datos.

Usuarios (users), son elementos de la base de datos que registran los permisos de quienes acceden a una base de datos. Una cuenta de usuario que accede a una base de datos obtiene sus permisos a través de un usuario de dicha base de datos.

2. BASES DE DATOS DEL SISTEMA

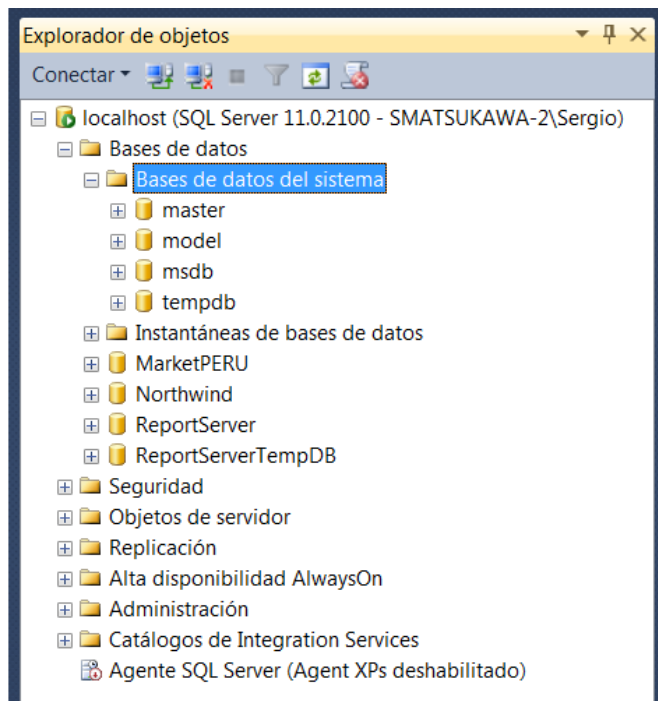
El proceso de instalación de Microsoft SQL Server crea en el servidor 4 bases de datos de sistema: master, model, msdb y tempdb.

Una base de datos de sistema contiene tablas de sistema; éstas contienen la metadata del sistema, es decir, los datos que permiten operar y administrar el sistema.

Las bases de datos que nosotros creamos se conocen como bases de datos de usuario, y contienen tablas de usuario (las tablas que creamos); estas tablas son las que almacenan nuestros datos. Una base de datos de usuario también contiene tablas de sistema que almacenan la metadata de la base de datos, es decir, los datos que permiten operar y administrar la base de datos.

Adicionalmente, el proceso de instalación crea dos bases de datos del Report Server. Estas datos como definición de reportes, reportes almacenados en caché, y la metadata de los reportes, que permiten operar y administrar los reportes.

La siguiente es una descripción breve del papel que cumple en SQL Server cada una de las bases de datos de sistema.



master: Es la principal base de datos del sistema. Controla los usuarios y las operaciones sobre el servidor manteniendo datos como cuentas de usuario, variables de entorno, mensajes de error del sistema, etc.

model: Proporciona una plantilla o modelo para cualquier base de datos nueva. Cuando se crea una base de datos, todo el contenido de la model se copia en la nueva base de datos. Permite personalizar el proceso de creación de bases de datos.

msdb: Almacena toda la data que utiliza el SQL Server Agent para programar alertas y trabajos, y para registrar operadores.

tempdb: Para almacenamiento de tablas temporales.

3. CREACIÓN DE BASES DE DATOS

Tal como hemos mencionado arriba, una base de datos es una colección de tablas que contienen datos, y además otros objetos como vistas, índices, procedimientos almacenados, desencadenantes, y usuarios, 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, el registro del personal de una organización, ó el control académico en una institución educativa.

3.1. Archivos que conforman una base de datos

Los datos y todos los objetos de la base de datos se almacenan en archivos de bases de datos. Una base de datos de Microsoft SQL Server está formada por mínimo 2 archivos: un archivo primario y un archivo de registro de transacciones. La base de datos puede tener además varios archivos secundarios y archivos de registro de transacciones adicionales.

Archivo primario

Es el archivo principal de una base de datos. Almacena las tablas de sistema, 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, y se utiliza **.mdf** como la extensión en el identificador del archivo.

Archivos secundarios

Estos archivos se usan para almacenar las tablas de usuario y demás objetos de la base de datos. Cuando la base de datos es muy grande, se puede utilizar archivos secundarios para distribuir los objetos de modo que puedan administrarse mejor. Los archivos secundarios pueden estar almacenados en discos diferentes, lo que hace que el rendimiento entrada/salida de la base de datos mejore. De ser necesario, pueden estar incluso en computadoras distintas. La extensión de los archivos secundarios es **.ndf**.

Archivos de registro de transacciones (transaction log)

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. Una base de datos debe tener por lo menos un archivo de registro de transacciones. La extensión del archivo es **.ldf**.

3.2. Definición de transacción

Una transacción es un conjunto de operaciones de modificación de datos que se ejecuta sobre una base de datos y que debe ser procesado como una unidad. Si una transacción está formada por ejemplo, por 20 operaciones, para que la transacción se considere válida se deben ejecutar las 20 operaciones; si una de las operaciones falla, la transacción se considera inválida, y por lo tanto, debe anularse.

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 de base de datos 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**.

¿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?

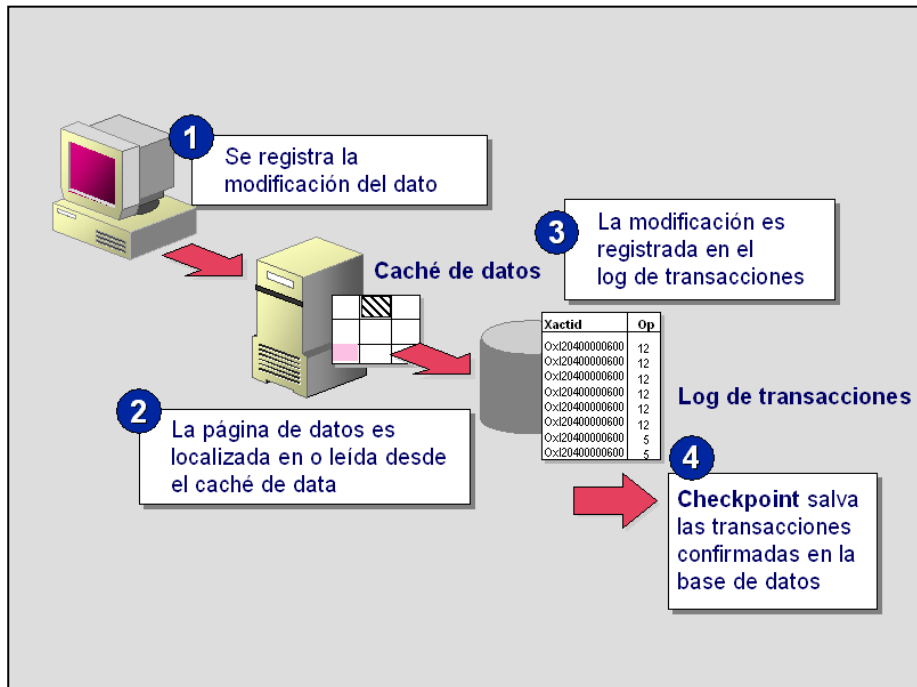
Si no se diseña un mecanismo para corregir el error, la base de datos perdería consistencia. Según las tablas **Factura** y **Detalle_factura** se habrían vendido 10 unidades del producto ABC, por lo que en la tabla **Producto** deberíamos tener 10 unidades menos. Esto último no es cierto al no haberse 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 para garantizar la consistencia de los datos, o en caso contrario, la transacción debe ser anulada.

3.3. El Registro de Transacciones como mecanismo para garantizar la consistencia de los datos

El archivo de registro de transacciones registra todas las operaciones llevadas a cabo por las sentencias INSERT, UPDATE y DELETE, y otras que producen modificaciones en los datos de la base de datos. El proceso se describe a continuación.



- (1) Una aplicación envía una modificación de datos al servidor SQL.
- (2) 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.
- (3) 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.
- (4) 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.

3.4. Creación de una base de datos – La instrucción CREATE DATABASE

Debemos tener en cuenta lo siguiente antes de crear una base de datos:

- Solo pueden crear bases de datos las cuentas de inicio de sesión que pertenecen a los roles fijos de servidor sysadmin ó **dbcreator**, ó que tienen asignado el permiso para ejecutar el comando CREATE DATABASE.
- 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 estime el tamaño de la base de datos, y las propiedades de cada uno de los archivos en los que residirá la base de datos.

La siguiente es la sintáxis de la forma básica del comando CREATE DATABASE.

Sintáxis

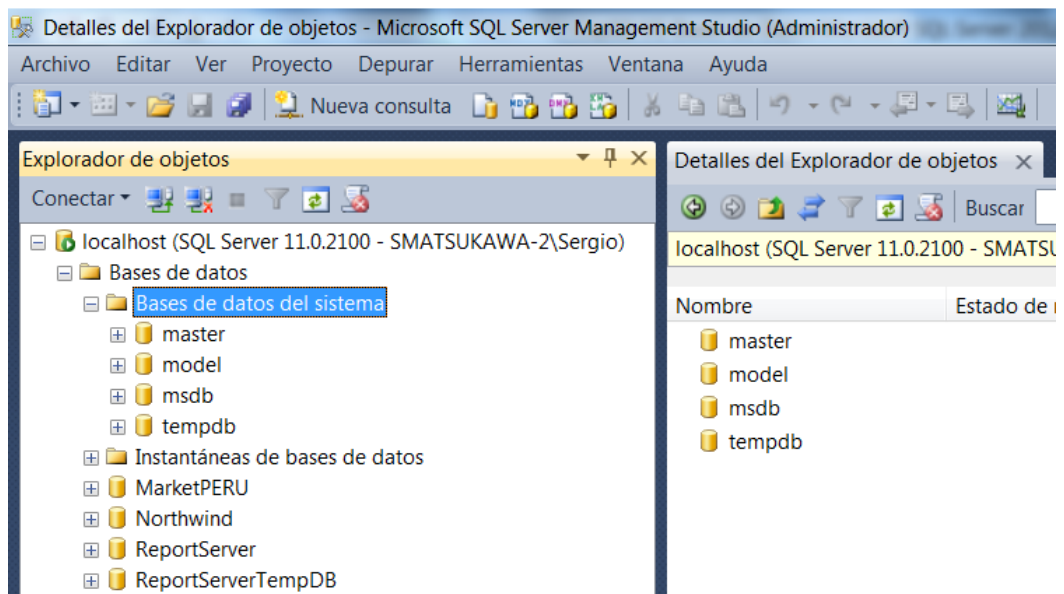
```
CREATE DATABASE nombre_basedatos
ON [PRIMARY] (
    NAME = nombre_lógico_data ,
    FILENAME = 'ubicación_y_nombre_archivo_data' ,
    SIZE = tamaño [KB|MB|GB|TB] ,
    MAXSIZE =
        tamaño_máximo [KB|MB|GB|TB|UNLIMITED] ,
    FILEGROWTH = incremento_crecimiento [KB|MB|%] )
LOG ON (
    NAME = 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. Cada una de las propiedades va separada de la anterior por una coma.
- 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 inicial del archivo.

- **tamaño_máximo** es el máximo tamaño que puede alcanzar el archivo si se necesitara 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 2.1: Creación de una base de datos con las propiedades predeterminadas

1. En **SQL Server Management Studio** haga clic en el botón **Nueva consulta** de la barra de herramientas.



2. Se abre el panel de Código.

Nota: a partir de este punto, todos los ejercicios referidos a instrucciones Transact-SQL los ejecutaremos en el panel de Código.

3. En el panel de Código escriba las siguientes instrucciones:

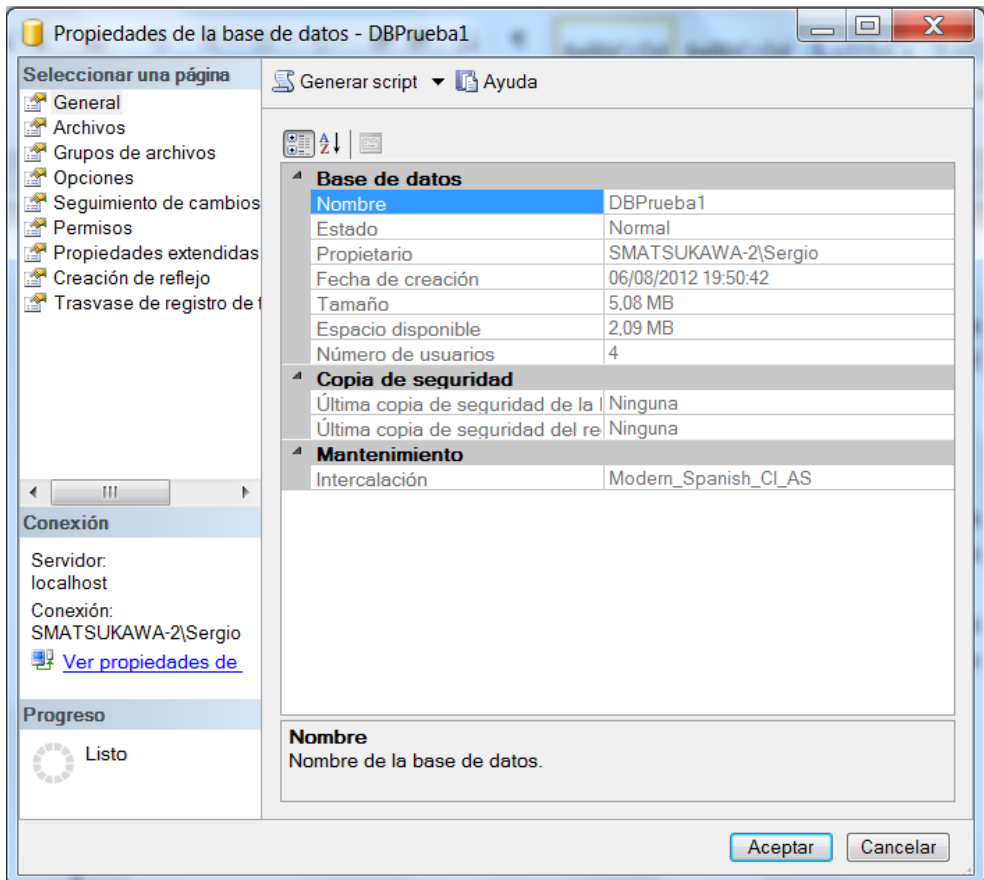
```
USE master
go
```

```
CREATE DATABASE DBPrueba1
go
```

- Haga clic en el botón **Ejecutar** de la barra de herramientas, ó pulse la tecla [F5]. Si las instrucciones se ejecutan sin error se muestra el siguiente mensaje:

Comandos completados correctamente.

- En el **Explorador de objetos** haga un clic secundario sobre **Bases de datos**, y luego clic en **Actualizar**. Se muestra la base de datos **DBPrueba1** recién creada.
- Haga un clic secundario sobre el nombre de la base de datos **DBPrueba1**, y luego clic en **Propiedades**. Se muestra la ventana de propiedades de la base de datos.



- Todas las propiedades de la base de datos **DBPrueba1** se han copiado de las propiedades de la base de datos de sistema **model**. Seleccione la página **Archivos**. Observe que la ubicación predeterminada de los archivos de la base de datos es la carpeta **Data** en la carpeta de instalación de la instancia de SQL Server.

8. Cierre la ventana de propiedades de la base de datos.

Ejercicio 2.2: Creación de una base de datos especificando las propiedades de los archivos de la base de datos

1. En una nueva instancia del panel de Código digite y ejecute las siguientes instrucciones Transact-SQL:

```
USE master  
go
```

```
CREATE DATABASE DBPrueba2  
ON PRIMARY (  
    NAME = DBPrueba2_data ,  
    FILENAME =  
    'D:\SQLServer2012\Data\DBPrueba2.mdf' ,  
    SIZE = 8MB ,  
    MAXSIZE = 15MB,  
    FILEGROWTH = 1MB )  
LOG ON (  
    NAME = DBPrueba2_log ,  
    FILENAME =  
    'D:\SQLServer2012\Data\DBPrueba2_log.ldf' ,  
    SIZE = 3MB ,  
    MAXSIZE = 8MB,  
    FILEGROWTH = 10% )  
go
```

Nota: Se asume que la ruta **D:\SQLServer2012\Data** 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**. Puede revisar la tabla **sysdatabases** ejecutando la siguiente consulta (para ejecutar solo esta consulta, selecciónela con el mouse, y luego haga clic en **Ejecutar**):

```
SELECT name FROM sysdatabases
go
```

	name
1	master
2	tempdb
3	model
4	msdb
5	ReportServer
6	ReportServerTempDB
7	Northwind
8	MarketPERU
9	DBPrueba1
10	DBPrueba2

4. PROPIEDADES Y OPCIONES DE CONFIGURACIÓN DE UNA BASE DE DATOS

Cuando creamos una base de datos, la mayoría de sus propiedades y opciones de configuración son tomadas de la base de datos de sistema **model**. Podemos revisar la configuración de la base de datos usando los procedimientos que mostraremos a continuación.

4.1. El procedimiento de sistema **sp_helpdb**

Este procedimiento entrega información acerca de todas las bases de datos en el servidor o de una base de datos específica.

Sintaxis

```
sp_helpdb [nombre_basedatos]
```

Si el procedimiento se ejecuta sin el argumento, entrega información de todas las bases de datos en la instancia de SQL Server.

Ejercicio 2.3: Propiedades de todas las bases de datos del servidor

1. En el panel de Código escriba y ejecute lo siguiente:

```
EXEC sp_helpdb
go
```

2. Se muestra la lista de bases de datos en el servidor mostrando sus propiedades básicas.

	name	db_size	owner	d...	created	status
1	DBPrueba1	5.08 MB	SMATSUKAWA-2\Sergio	9	Ago 6 2012	Status=ONLINE, Updateability=READ_WRITE,
2	DBPrueba2	11.00 MB	SMATSUKAWA-2\Sergio	10	Ago 6 2012	Status=ONLINE, Updateability=READ_WRITE,
3	MarketPERU	5.08 MB	SMATSUKAWA-2\Sergio	8	Jul 19 2012	Status=ONLINE, Updateability=READ_WRITE,
4	master	6.63 MB	sa	1	Abr 8 2003	Status=ONLINE, Updateability=READ_WRITE,
5	model	5.06 MB	sa	3	Abr 8 2003	Status=ONLINE, Updateability=READ_WRITE,
6	msdb	21.25 MB	sa	4	Feb 10 2012	Status=ONLINE, Updateability=READ_WRITE,
7	Northwind	5.08 MB	SMATSUKAWA-2\Sergio	7	Jul 7 2012	Status=ONLINE, Updateability=READ_WRITE,
8	ReportServer	11.94 MB	SMATSUKAWA-2\Sergio	5	Jul 7 2012	Status=ONLINE, Updateability=READ_WRITE,
9	ReportServerTempDB	5.08 MB	SMATSUKAWA-2\Sergio	6	Jul 7 2012	Status=ONLINE, Updateability=READ_WRITE,
10	tempdb	8.75 MB	sa	2	Jul 23 2012	Status=ONLINE, Updateability=READ_WRITE,

Ejercicio 2.4: Propiedades de una base de datos específica

1. En el panel de Código escriba y ejecute lo siguiente:

```
EXEC sp_helpdb DBPrueba1
go
```

2. Ahora el panel de resultados muestra dos cuadrículas: una con las propiedades básicas de la base de datos **DBPrueba1**, y otra con las propiedades de sus archivos.

	name	db_size	owner	d...	created	status	com
1	DBPrueba1	5.08 MB	SMATSUKAWA-2\Sergio	9	Ago 6 2012	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	110

	name	fileid	filename	filegroup	size	maxsize	growth
1	DBPrueba1	1	D:\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQ...	PRIMARY	4160 KB	Unlimited	1024 KB
2	DBPrueba1_log	2	D:\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQ...	NULL	1040 KB	2147483648 KB	10%

4.2. Configuración de una base de datos – La instrucción ALTER DATABASE

Salvo que se especifique explícitamente al momento de crear la base de datos, ésta toma su configuración de la establecida en forma predeterminada en la base de datos de sistema **model**. El siguiente cuadro muestra algunas de las opciones de configuración de una base de datos.

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.

Ejercicio 2.5: Configuración de una base de datos

En este ejercicio haremos uso de la instrucción ALTER DATABASE para configurar la base de datos **DBPrueba1** como monousuaria.

1. En el panel de Código escriba y ejecute lo siguiente:

```
ALTER DATABASE DBPrueba1
SET single_user
go
```


2. Para verificar la base de datos usaremos la función **DatabasePropertyEx**.

```
SELECT DatabasePropertyEx(  
    'DBPrueba1', 'UserAccess')  
go
```

La función devuelve como resultado el valor SINGLE_USER.

4.3. La función DatabasePropertyEx

Esta función devuelve el valor actual de una opción ó propiedad de la base de datos especificada.

Sintaxis

```
DatabasePropertyEx( 'nombre_basedatos', 'propiedad' )
```

- **propiedad** es el nombre de la opción ó propiedad de base de datos cuyo valor se desea conocer.

5. EJERCICIOS PROPUESTOS

Estos ejercicios le permitirán conocer su nivel de asimilación de los temas presentados en este capítulo.

1. Escriba una sentencia Transact-SQL que le permita crear la base de datos de nombre **Reto1** conformada por:
 - un archivo primario de 10 MB, de 20 MB de tamaño máximo, y con un factor de crecimiento de 10%.
 - un archivo de registro de transacciones de 5 MB, de 8 MB de tamaño máximo, y con un factor de crecimiento de 1 MB.

Ambos archivos estarán ubicados en la carpeta de datos predeterminada de su instancia de SQL Server.

2. Usando la interfaz gráfica de SQL Server Management Studio debe crear una base de datos de nombre **Reto2** conformada por:
 - un archivo primario de 10 MB, sin límite de crecimiento, y con un factor de crecimiento de 10%.
 - un archivo de registro de transacciones de 5 MB, de 8 MB de tamaño máximo, y con un factor de crecimiento de 10%.

Ambos archivos estarán ubicados en la carpeta **C:\SQLServer2012\Data** de su disco.