

IES CHAN DO MONTE

C.S. de Desarrollo de Aplicaciones Multiplataforma

Módulo Base de datos

UNIDAD 4: Introducción al SGBD SQL SERVER 2008

Índice

1.	¿Qué es SQL Server?	2
2.	Componentes de SQL Server	2
2.1	Motor de base de datos (SQL Server)	2
3.	Herramientas y utilidades de SQL Server	2
3.1	Los Libros en pantalla	4
3.1.1	Acceso a los Libros en pantalla de SQL Server	4
3.1.2	Localización de los temas de ayuda	5
3.2	Tutoriales	5
4.	Uso de SQL Server Management Studio	6
4.1	Arrancar SQL Server Management Studio	6
4.2	Usar el Explorador de objetos	7
4.2.1	Ver el Explorador de objetos	8
4.2.2	La ventana de Explorador de objetos	8
5.	Visión general de seguridad SQL Server	10
5.1	Autenticación de inicio de sesión	10
5.1.1	Autenticación de Windows	11
5.1.2	Modo mixto (autenticación de Windows y autenticación de SQL Server)	11
5.2	Inicios de sesión de servidor	12
5.3	Permisos	13
5.3.1	Permisos de objetos	13
5.3.2	Permisos de instrucciones	14
5.4	Funciones (roles) de base de datos	14
5.4.1	Funciones fijas del servidor	15
5.4.2	Funciones fijas de base de datos	15
5.4.3	Funciones de aplicación	16
6.	Arquitectura de SQL Server	17
6.1	Bases de datos de SQL Server	17
6.1.1	Bases de datos de sistema	17
6.1.2	Bases de datos de usuario	18
6.2	Tablas del sistema	19
6.3	Procedimientos almacenados del sistema	20
6.4	Funciones del sistema	21
6.5	Vistas de esquema de información	21
6.6	Objetos de bases de datos	22
6.6.1	Nombres de objetos en SQL Server	23
6.6.2	Pautas para nombrar objetos de bases de datos	24
6.6.3	Referencia a objetos de SQL Server	24
6.6.4	Esquemas	25
6.6.5	Tipos de datos	25
7.	Cómo conectarse a SQL Server	27
7.1	Utilizando SQL Server Management Studio	27
7.2	Ejecutando el comando SQLCMD	29
8.	Arrancar y detener el motor de base de datos	32
8.1	Herramientas Administrativas -> Servicios	33
8.2	Desde la línea de comandos	36
8.3	Utilizar el Administrador de configuración de SQL Server	37

1. ¿Qué es SQL Server?

SQL Server es un SGBD **RELACIONAL cliente/servidor** que utiliza **Transact-SQL** para enviar peticiones entre un cliente y SQL Server.

SQL Server utiliza Transact-SQL, una versión del lenguaje de consulta estructurado (SQL), como su consulta de base de datos y su lenguaje de programación.

El Instituto nacional americano de normalización (ANSI) y la Organización internacional de normalización (ISO) han definido estándares para SQL. Transact-SQL soporta los últimos estándares de ANSI SQL además de muchas extensiones para proporcionar un funcionamiento aumentado.

2. Componentes de SQL Server

Motor de base de datos

Reporting Services

Analysis Services

Notification Services

Integration Services

Búsqueda de texto

Réplica

Service_Broker

En el diagrama se ilustran las relaciones existentes entre los componentes de SQL Server 2008 y se indica la interoperabilidad entre ellos.



2.1 Motor de base de datos (SQL Server)

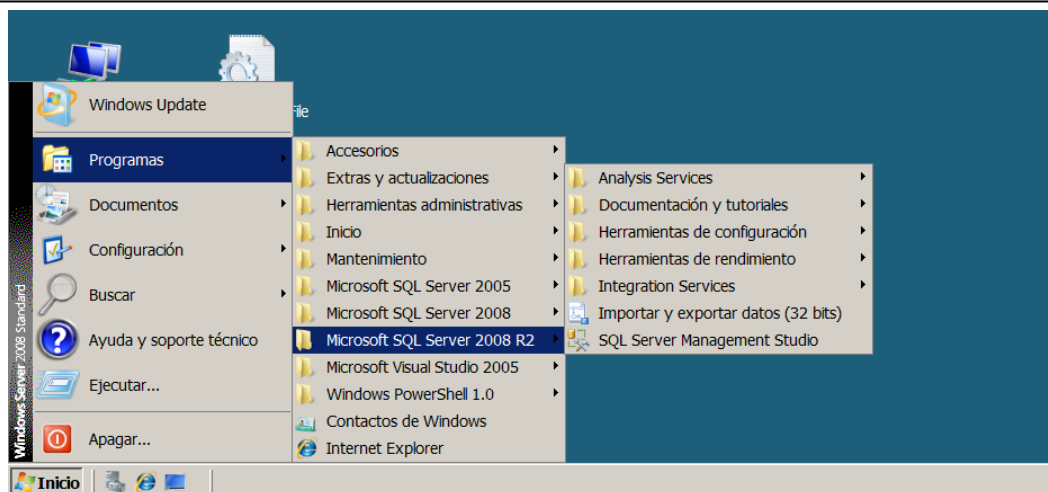
El Database Engine (Motor de base de datos) es el servicio principal para almacenar, procesar, proteger los datos y asignar los recursos del sistema. Se pueden crear tablas para almacenar datos y objetos de base de datos como índices, vistas y procedimientos almacenados para ver, administrar y proteger los datos.

SQL Server 2008 integra .NET Framework que permite poner soluciones de base de datos en ejecución usando el código escrito en lenguaje .NET por ejemplo Microsoft Visual C#® .NET o Microsoft Visual Basic® .NET. Esto significa, que ahora podemos escribir procedimientos almacenados, desencadenadores, funciones definidas por el usuario y tipos definidos por el usuario en diversos lenguajes como C# o Visual Basic.

3. Herramientas y utilidades de SQL Server

Cuando se instala SQL Server, hay una serie de herramientas e utilidades disponibles.

Para conocer los programas que se instalan, se mira en el menú Inicio, en el grupo de programas de Microsoft Server 2008.



Con una instalación completa e instalación de características compartidas, el Menú Microsoft Server 2008 contará entre otras las siguientes opciones y subopciones:

- **SQL Server Management Studio**: Es un entorno integrado para acceder, configurar, administrar y manejar todos los componentes de SQL Server 2008 R2
- **Importar y exportar datos**: Es el Asistente para importación y exportación de SQL Server para copia de datos de un origen en un destino.
- **Herramientas de configuración**
 - Administrador de configuración de Reporting Services: La herramienta de configuración de Reporting Services se utiliza para configurar un servidor de informes instalado
 - **Administrador de configuración de SQL Server**: El Administrador de configuración de SQL Server es una herramienta para: Administrar los servicios asociados a SQL Server, Configurar los protocolos de red utilizados por SQL Server Administrar la configuración de conectividad de red de los equipos cliente de SQL Server.
 - **Centro de instalación de SQL Server**
 - **Informes de uso y errores de SQL Server**: Se puede habilitar determinadas características que permitirán informar de errores y enviar información de manera automática acerca de cómo se utiliza el software y los servicios de SQL Server 2008 a Microsoft. Microsoft utiliza esta información para mejorar SQL Server, y trata toda la información recibida de forma confidencial.
- **Documentación y tutoriales**
 - Introducción a los ejemplos de Microsoft SQL Server
 - Libros en pantalla de SQL Server
 - Tutoriales de SQL Server
- **Integration Services**: entorno de integración de datos de diferentes orígenes
- **Herramientas de rendimiento**
 - **Asistente para la optimización del motor de base de datos**: El Asistente para la optimización de motor de base de datos analiza la forma en que se procesan las consultas en las bases de datos especificadas por el usuario y, a continuación, recomienda la forma en que se puede mejorar el rendimiento del procesamiento modificando las estructuras de diseño físico tales como índices, vistas indizadas y particiones.
 - **SQL Server Profiler**: Una vez que se han diseñado e implementado de forma correcta la base de datos, y los usuarios acceden a ella con regularidad para insertar, actualizar e eliminar datos, tienen que supervisar el servidor de que se ejecuta como se supone que debe hacer, por ejemplo, que rápido se ejecuta el servidor, a qué tipos de datos acceden los usuarios o si hay alguien que intente acceder sin permiso al servidor.

3.1 Los Libros en pantalla

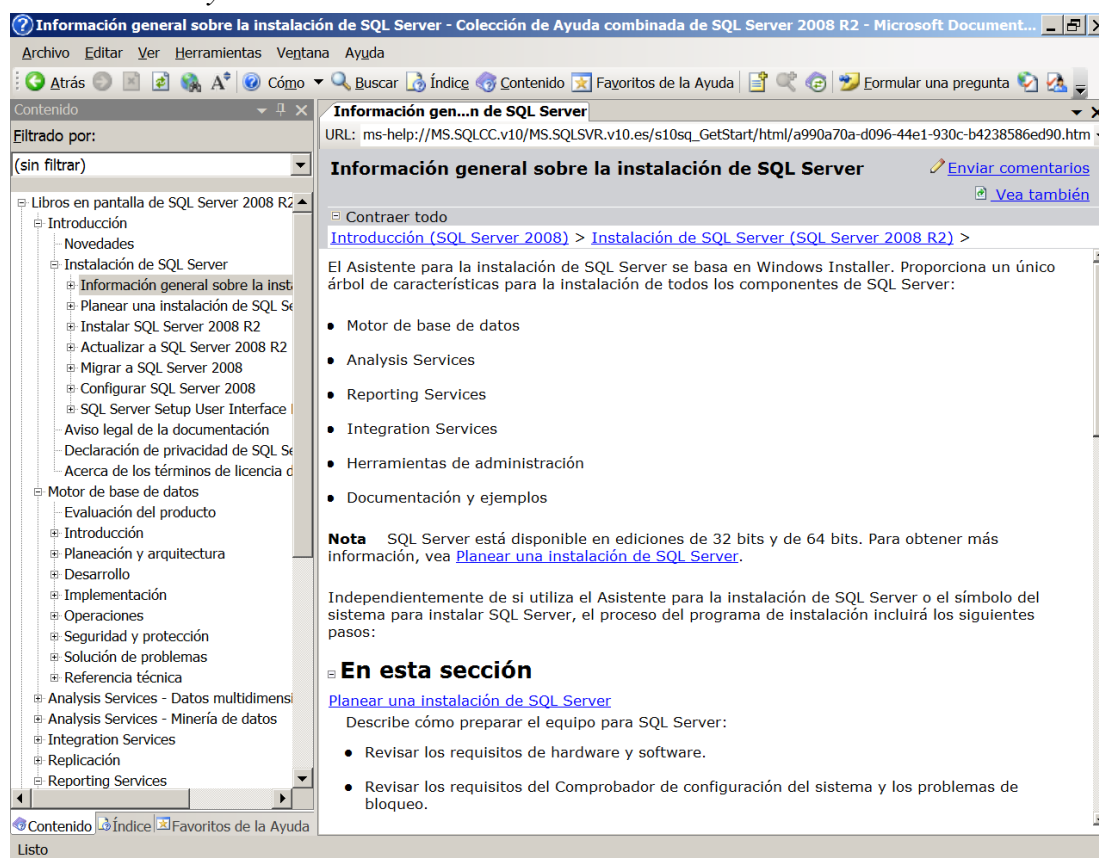
Los Libros en pantalla de SQL Server son **la fuente principal de Ayuda** para Microsoft SQL Server.

Los temas de Ayuda se almacenan en colecciones de Ayuda (las colecciones disponibles dependen del software y de las colecciones de Ayuda asociadas disponibles para su equipo durante la instalación del software).

Para ver cualquiera de las colecciones de Ayuda disponibles o buscar información se utiliza la interfaz gráfica Microsoft Document Explorer.

Con Microsoft Document Explorer, se puede:

- **Explorar títulos de temas** en la *ventana Contenido*.
- Buscar temas por palabra clave en la *ventana Índice*.
- **Buscar el texto completo** de temas en la *página Buscar*.
- Explorar los temas por categoría con la *opción Cómo*.
- **Marcar un tema útil o guardar una consulta** de búsqueda complicada en la *ventana Favoritos de la Ayuda*.



3.1.1 Acceso a los Libros en pantalla de SQL Server

Puede obtener acceso a los Libros en pantalla de SQL Server de varias formas:

- **Desde el menú Inicio:** se selecciona Todos los programas, Microsoft SQL Server, Documentación y tutoriales y, a continuación, se hace clic en Libros en pantalla de SQL Server.
- **Desde SQL Server Management Studio:** en el menú *Ayuda*, se hace clic en *Cómo*, *Buscar*, *Contenido*, *Índice* o *Favoritos de la Ayuda*.

- **Desde F1 o los botones Ayuda** de la interfaz de usuario: Para **obtener información contextual**, se presiona F1 en el teclado o se hace clic en Ayuda en el cuadro de diálogo de la interfaz de usuario.
- **Desde la ventana Ayuda dinámica**. La ventana Ayuda dinámica muestra automáticamente vínculos a los temas de los Libros en pantalla relacionados con la tarea que se esté realizando. Para iniciar la Ayuda dinámica, se hace clic en *Ayuda dinámica* en el *menú Ayuda* de SQL Server Management Studio.

3.1.2 Localización de los temas de ayuda

En los Libros en pantalla de SQL Server, se puede buscar orígenes de la Ayuda tanto locales como en pantalla.

Hay cuatro recursos que proporcionan los resultados en la página de búsqueda:

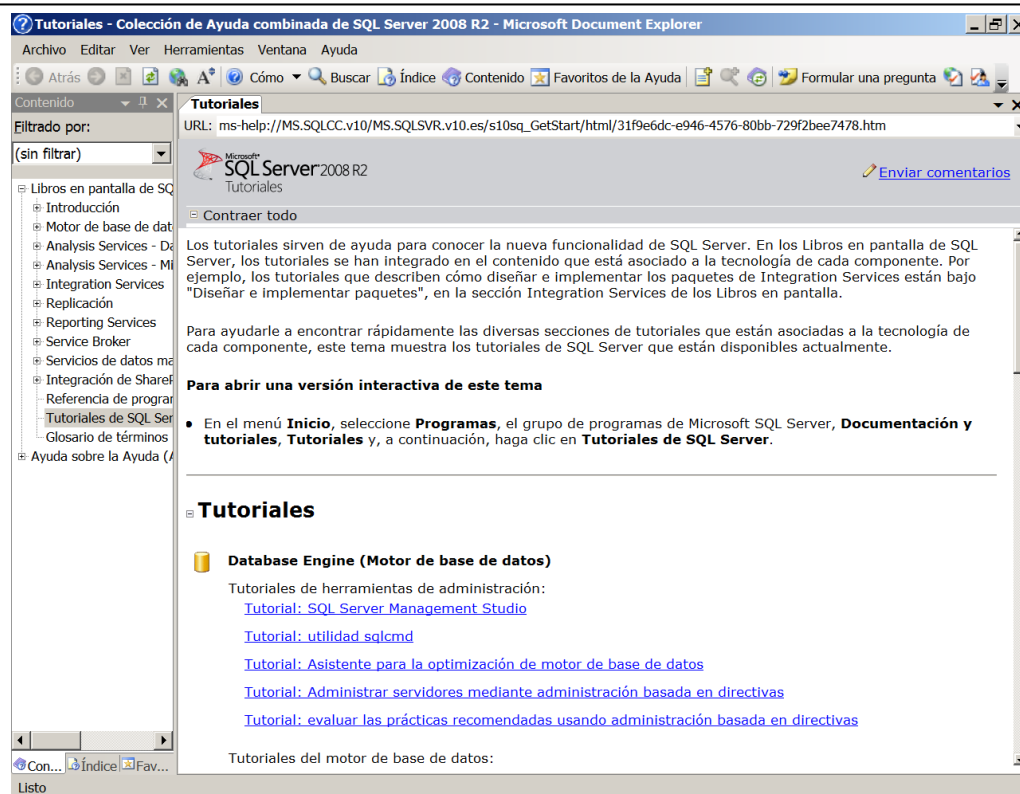
- Ayuda local
 - MSDN Online,
 - Comunidad de Codezone
 - preguntas.
- **Ayuda local.** La Ayuda local es el conjunto de temas que se instala con los Libros en pantalla de SQL Server desde el disco de instalación de SQL Server.
 - **MSDN Online.** Microsoft Developer Network (MSDN) es un conjunto de servicios diseñado para ayudar a los desarrolladores de software a escribir aplicaciones utilizando los productos y las tecnologías de Microsoft.
 - MSDN proporciona temas de los Libros en pantalla, artículos, notas del producto, evaluaciones y código de ejemplo para programadores de software que utilizan productos de Microsoft.
 - **Comunidad de Codezone.** El sitio de la comunidad de Codezone es un recurso con sugerencias de terceros, ejemplos de código, consejos y noticias de expertos independientes sobre las tecnologías de SQL Server, Microsoft .NET Framework y Microsoft Visual Studio .NET.
 - **Preguntas.** La selección de Preguntas le permite realizar búsquedas en el contenido de los foros de MSDN.

3.2 Tutoriales

Los tutoriales sirven de ayuda para **conocer la funcionalidad de cada componente** de SQL Server.

Están integrados en los Libros en pantalla de SQL Server.

Por ejemplo, los tutoriales que describen cómo utilizar la herramienta SQL Management Studio están bajo "Tutorial: SQL Server Management Studio", en la sección Motor de la base de datos de los Libros en pantalla.



4. Uso de SQL Server Management Studio

SQL Server Management Studio es un entorno integrado para obtener acceso a todos los componentes de SQL Server, así como para configurarlos y administrarlos.

SQL Server Management Studio combina un amplio grupo de herramientas gráficas con un editor de texto enriquecido para ofrecer acceso a SQL Server a los programadores y administradores, sin importar su nivel de especialización.

De forma predeterminada, SQL Server Management Studio se instala en **C:\Program Files\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE**.

Nota:

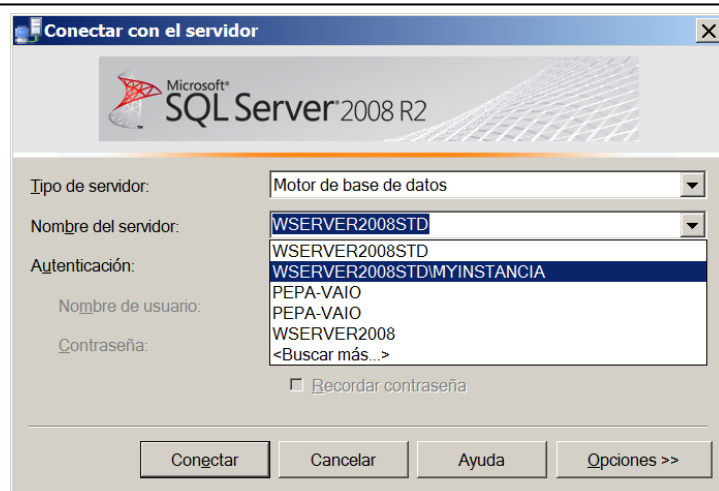
SQL Server Management Studio no se instala de manera predeterminada.

Si Management Studio no está disponible, hay que instalarlo ejecutando la instalación y en características a instalar, hay que seleccionarlo.

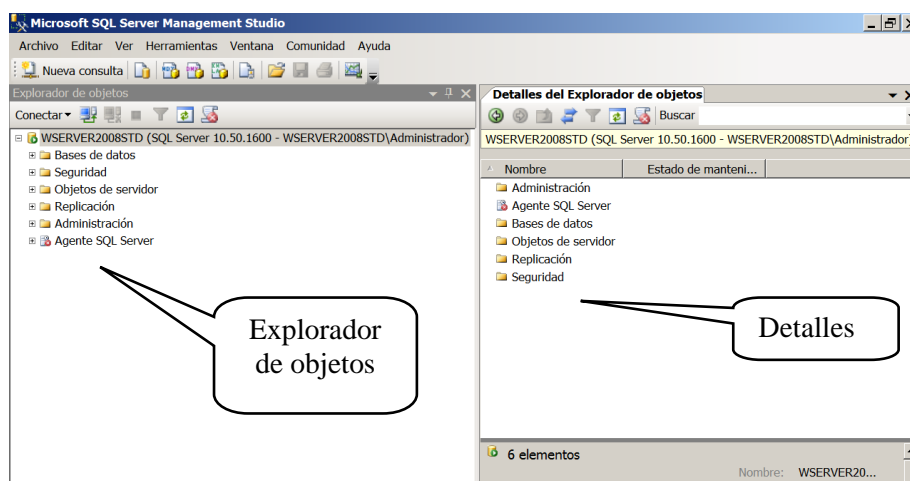
4.1 Arrancar SQL Server Management Studio

Para obtener acceso a SQL Server Management Studio, se puede hacer de dos maneras:

- En la barra de tareas, se hace clic en **Inicio**, se selecciona **Todos los programas, Microsoft SQL Server 2008 R2** y, a continuación, se hace clic en **SQL Server Management Studio**.
- Desde el indicador de comandos se escribe **SSMS**
Aparece el cuadro de diálogo **conectar al servidor**:



- El siguiente paso es **conectarnos con un servidor de base de datos**, proporcionando el nombre del servidor al que nos queremos conectar y el tipo de servidor (motor de base de datos). En la conexión podemos optar :
 - Por la **autenticación de Windows**.
 - O bien por la **autenticación con usuario de la base de datos**, siempre y cuando hayamos seleccionado la autenticación mixta en el instalación de SQL Server 2008 R2.
- Al conectarnos se nos presenta la primera pantalla de nuestra herramienta, y observamos que muestra la pantalla del **Explorador de objetos**.
- Para mostrar la ventana **Detalles del Explorador de objetos** para navegar por las carpetas y los objetos de una instancia de SQL Server, en el menú Ver, se hace clic en Detalles del Explorador de objetos o presione **F7**.



4.2 Usar el Explorador de objetos

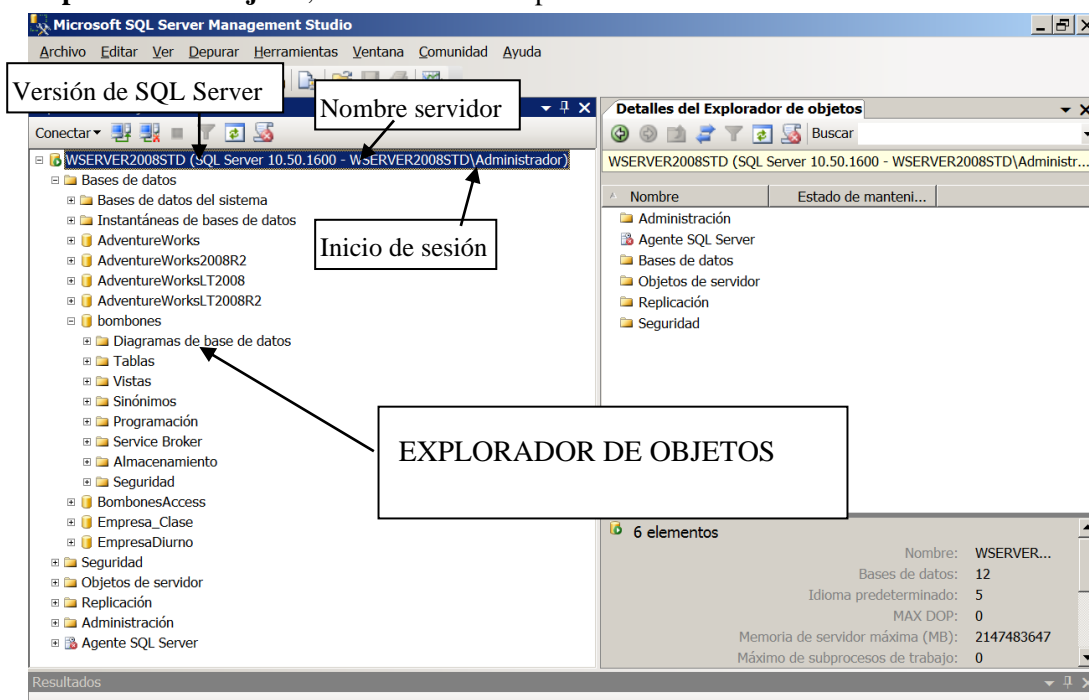
El Explorador de objetos, un componente de SQL Server Management Studio, se conecta a las instancias de Database Engine (Motor de base de datos), Analysis Services, Integration Services, Reporting Services.

Proporciona una vista de todos los objetos del servidor y presenta una interfaz de usuario para administrarlos.

Las funciones del Explorador de objetos varían ligeramente según el tipo de servidor, aunque, por lo general, incluyen características de desarrollo de bases de datos y características de administración para todo tipo de servidores.

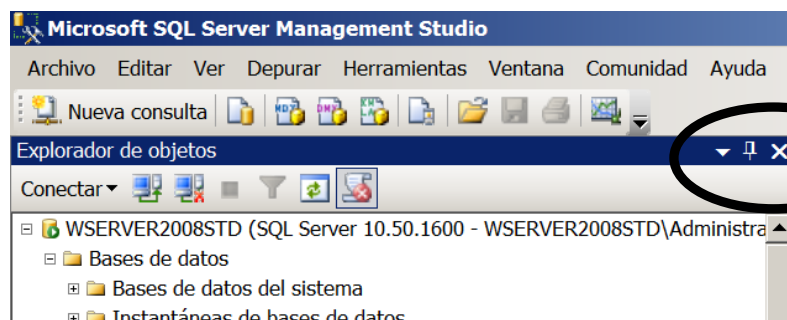
4.2.1 Ver el Explorador de objetos

El Explorador de objetos es visible de forma predeterminada en Management Studio. Si no se ve, se hace clic en **Explorador de objetos**, en el **menú Ver** o pulsando **F8**.

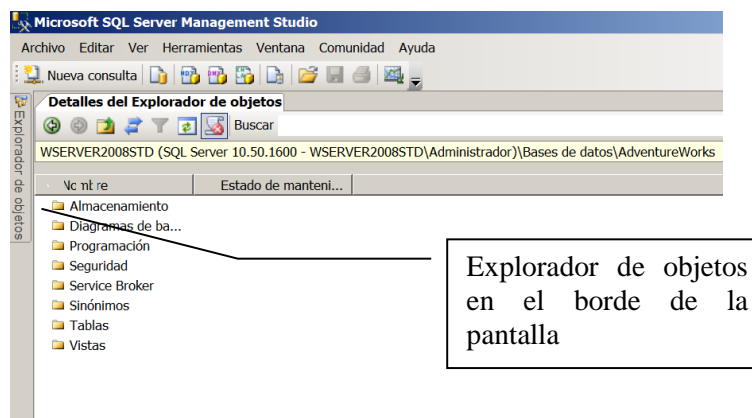


4.2.2 La ventana de Explorador de objetos



La ventana de Explorador de objetos, tiene tres iconos: Posición de la ventana (↕), Ocultar automáticamente (⌵) y cerrar (X).



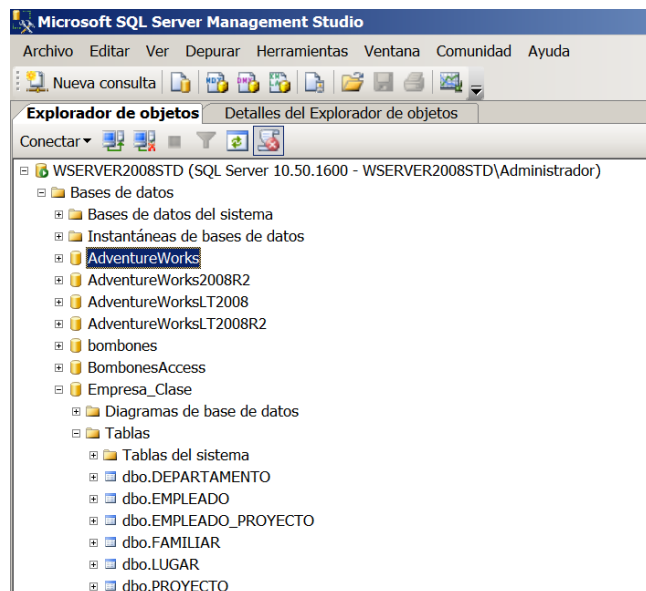
Si ocultamos automáticamente la ventana del Explorador de Windows, se convierte en pestaña al borde la pantalla principal:



Para activar y desactivar la posibilidad de ocultar automáticamente tenemos el botón chincheta:

-  Activada la opción "Ocultar Automáticamente".
-  La ventana permanece fija, tenemos desactivada la opción "Ocultar automáticamente".
- También podemos *Organizar la ventana* del explorador por **fichas, flotantes** (que se puedan mover), **acoplables** y que se puedan ocultar automáticamente. Se utiliza el icono del triangulito

Ejemplo: Vista de las ventanas de SQL Server Managenet Studio de forma en fichas:









- En el **menú ventana** hay una opción para **reestablecer el diseño de la ventana a la configuración de instalación**.

El explorador de objetos se **encuentra conectado a los diferentes tipos de servidores que tenemos**.

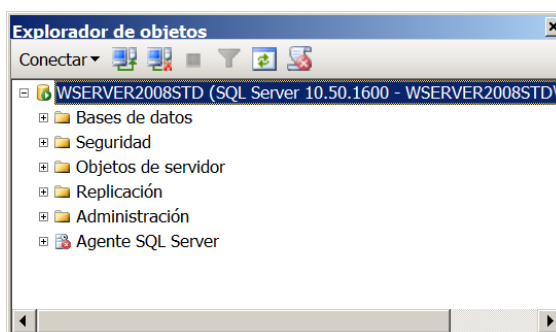
Los elementos que nos ofrece SQL Server 2008 R2 varían en función del tipo de servidor, pero hay características de desarrollo y herramientas de administración comunes para todos.



En esta ventana podemos ver una barra de herramientas que nos permite realizar unas determinadas tareas. En la siguiente tabla mostramos estos botones con la tarea que tienen asignada:

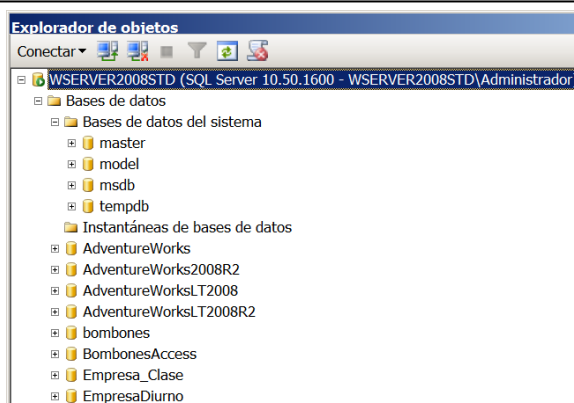
	Realiza una conexión con el servidor
	Realiza una desconexión del servidor.
	Detiene un proceso, sólo estará disponible durante la ejecución de uno
	Actualiza la lista de elementos de la carpeta que tengamos seleccionada
	Realiza un filtro o selección de objetos, permanece en gris cuando no es posible filtrar.
	Restaura el árbol de nodos del explorador, visualizando solo los nodos sin desplegar sus componentes.

El explorador de objetos usa una estructura de árbol para agrupar la información en carpetas para cada servidor.



La carpeta que vamos a utilizar es la siguiente:

Bases de datos: Administra todas las bases del sistema. Contiene la carpeta **Base de datos del sistema**, la carpeta **instantánea de base de datos** y las **bases de datos de los usuarios**.



Las bases de los usuarios **tienen una carpeta para cada tipo de objeto que contienen.**

Seguridad: Controla la configuración que afecta al servidor. Administra los inicios de sesión de SQL Server, roles de servidor, servidores vinculados

5. Visión general de seguridad SQL Server

La protección de la información, es decir, controlar el acceso a los datos de una organización, se parece mucho a la protección de una estructura física. Por ejemplo, en un negocio que está en un edificio, nos queremos que todo el mundo acceda al edificio, por ejemplo, solo los empleados. Sin embargo, también necesitaremos restricciones para las zonas a las que los empleados accedan, por ejemplo, solo los contables deberán acceder al departamento de contabilidad y casi nadie debería tener acceso a su despacho. Debemos tener instalados diversos sistemas de seguridad.

La protección del SQL Server (“El edificio”) se basa en este concepto: nadie puede entrar al menos que se le conceda acceso y, una vez que los usuarios estén dentro, los diferentes sistemas de seguridad mantienen las áreas confidenciales a salvo de accesos indebidos.

El acceso a las bases de datos se controla mediante los componentes del modelo de seguridad de SQL Server. Estos componentes son:

- Autenticación de inicio de sesión
- Inicios de sesión de servidor
- Permisos.
- Funciones.

5.1 Autenticación de inicio de sesión

Siguiendo con la anterior comparación, para que los empleados puedan entrar al edificio, necesitan algún tipo de llave, ya sea metálica o una tarjeta electrónica de acceso. Para que los usuarios de SQL Server accedan también tenemos que darles una llave. El tipo de llave que se les proporcione depende en gran parte del tipo de cerradura (Modos de autenticación) que se utilice.

El modo de autenticación es la manera en que SQL Server procesa los nombres de usuario y las contraseñas.

Un usuario debe tener una cuenta de inicio de sesión para conectarse con SQL Server.

SQL Server reconoce dos mecanismos de autenticación de inicio de sesión:

- **La Autenticación de Windows**
- **Modo Mixto** (Autenticación Windows y la Autenticación SQL Server)

5.1.1 Autenticación de Windows

Con la autenticación Windows, pueden ser usadas para propósitos de seguridad, las **cuentas de usuario y de grupo disponibles en el dominio de Windows o en el equipo local**. Esto permite a los usuarios del dominio acceder a las base de datos sin necesidad de que el administrador de la base de datos les proporcione un identificador de inicio de sesión y una contraseña separados para el SQL Server (ventaja porque los usuarios del dominio no tienen que recordar múltiples contraseñas)

Cuando se usa la autenticación de Windows para conectarse a SQL Server, Microsoft **Windows es el único responsable de autenticar al cliente**.

Cuando un usuario se conecta mediante una cuenta de usuario de Microsoft, SQL Server, valida el nombre de usuario y la contraseña usando la información del sistema operativo Windows.

La autenticación de Windows **usa el protocolo de seguridad de Kerberos**, proporciona la aplicación de directivas de contraseñas en cuanto a la validación de la complejidad de las contraseñas seguras, ofrece compatibilidad para el bloqueo de cuentas (tras un determinado número de intentos fallidos) y admite la caducidad de las contraseñas. Por esto, la autenticación Windows es el modo de autenticación predeterminado y es mucho más seguro que el modo mixto.

➤ La ventaja de utilizar autenticación Windows

- No **se pasan credenciales por la red durante la autenticación**. Los nombres de usuario y contraseñas no se incrustan en la cadena de conexión a bases de datos. Es decir, los usuarios malintencionados o atacantes no pueden obtener credenciales mediante la supervisión de la red o viendo las cadenas de conexión dentro de los archivos de configuración.
- Los usuarios están **sujetos a la administración de cuentas centralizadas**. Este modo también proporciona un control más fuerte de la seguridad, porque podemos aplicar las **directivas de contraseñas de Windows**, con las cuales las contraseñas caducan, se requiere una longitud mínima para las contraseñas, se guarda un historial de contraseñas, bloqueo de cuenta después de varias solicitudes de inicio de sesión no válidas, etc...
- Los usuarios **no tienen que recordar varios nombres de usuario y contraseña**, lo cual aumenta infinitamente la seguridad, ya que existe menos riesgo de que los usuarios anoten sus contraseñas y las guarden en un lugar poco seguro.

➤ Un inconveniente es que **sólo las cuentas de Windows pueden abrir una conexión de confianza SQL server**, lo que significa, por ejemplo un cliente que utilice Novell, o usuarios de Macintosh no pueden utilizar el modo autenticación de Windows porque no tendrían una cuenta de Windows.

5.1.2 Modo mixto (autenticación de Windows y autenticación de SQL Server)

Permite a los usuarios **conectarse usando la autenticación de Windows y la autenticación de SQL Server**.

Si se elige la autenticación de modo mixto, se deben cumplir los requisitos de las directivas de contraseñas seguras para todos los inicios de sesión de SQL Server. Esto es especialmente importante para **sa** y para los inicios de sesión que son miembros de la función fija de servidor **sysadmin**.

La autenticación SQL Server actúa de la siguiente manera:

- El usuario se conecta a su red, Windows o cualquier otra.
- El usuario abre una conexión que no es de confianza con SQL Server con un nombre de usuario y una contraseña diferente de las que se ha empleado para tener acceso a la red. **Esta conexión no es de confianza** porque SQL Server no encomienda al sistema operativo que verifique la contraseña del usuario.
- SQL Server comprueba el nombre de usuario y la contraseña introducida por el usuario con respecto a una entrada en la tabla **Syslogins**.

Nota: La autenticación de SQL Server sólo se proporciona por motivos de compatibilidad con versiones anteriores. Siempre que sea posible, utilice la autenticación de Windows.

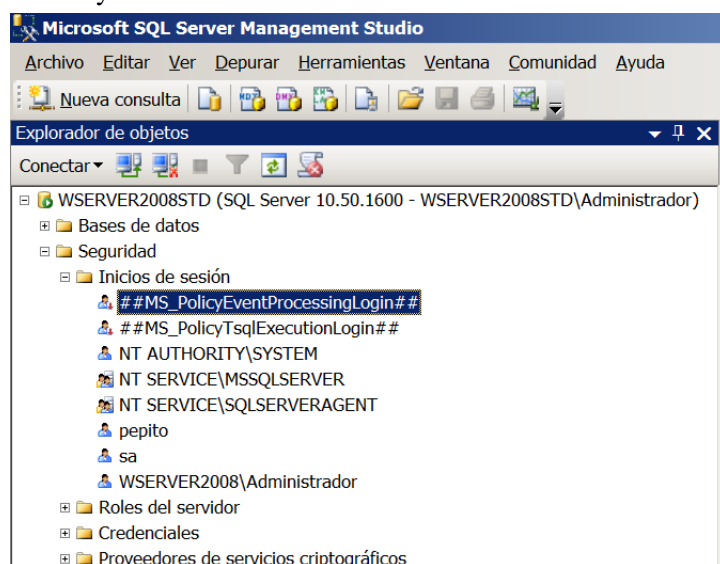
5.2 Inicios de sesión de servidor

Una vez que hemos decidido qué tipo de cerradura (**modo de autenticación**) que se va a utilizar para entrar, se puede entregar las llaves para que los empleados puedan acceder a edificio. Una llave proporciona a los empleados el acceso al edificio en general, pero no a los recursos que contiene (ejemplo, a los muebles archivadores). Del mismo modo, una llave de SQL Server (**inicio de sesión**) **proporciona acceso a SQL Server en general**, pero no a los recursos que contiene (las bases de datos). Se pueden crear dos inicios de sesión: Inicios de sesión de Windows (como las tarjetas electrónicas, más modernas) y los inicios de sesión de SQL Server (como las llaves metálicas).

- El **acceso a SQL Server** se configura usando **cuentas de inicio de sesión** de servidor.
- Los **inicios de sesión de usuarios** permiten **proteger los datos contra modificaciones no autorizadas**, sean intencionadas o no.
- Cada **inicio de sesión de usuario** tiene asignado un **nombre único y una contraseña**. Cada usuario tiene asignado su propia cuenta de inicio de sesión. Sin los inicios de sesión, todas las conexiones utilizarían el mismo el mismo identificador, lo que sería imposible crear diferentes niveles de seguridad dependiendo quien acceda a la base de datos.
- Con los **inicios de sesión** podemos crear varios niveles de seguridad permitiendo permisos diferentes a distintas cuentas de inicio de sesión con el fin de acceder a objetos y realizar funciones. Al igual que existen dos modos de autenticación, existen también dos tipos de inicios de sesión de servidor:

- **Inicios de sesión de dominio:** se crean utilizando cuentas de dominio, que pueden ser cuentas de dominio o local, cuentas de grupo local o cuentas de grupo de dominio.
- **Inicios de sesión SQL Server:** se crean especificando una contraseña y un identificador de inicio de sesión únicos.

Hay varias cuentas de inicio de sesión de servidores configurados de manera predeterminada, entre las que se incluyen:



Los inicios de sesión de servidor incluidos entre signos de número dobles (##) son exclusivamente para uso interno del sistema y se crean cuando se instala SQL Server y no deben eliminarse.

Cuenta de Inicio sa:

Todos los servidores SQL Server tienen instaladas la cuenta de **inicio de sesión sa integrada**. La cuenta *sa* es la cuenta de administrador en el inicio de sesión SQL Server.

Con el nuevo modelo de seguridad integrado con Windows, esta cuenta no es necesaria, y se proporciona principalmente por motivos de compatibilidad hacia atrás con versiones anteriores de SQL Server.

Al igual que con las cuentas de inicio de sesión de administrador, **sa tiene otorgado el rol de servidor sysadmin**, como opción predeterminada.

Con el fin de evitar acceso no autorizado al servidor, se debe configurar una contraseña fuerte para esta cuenta y también se debe cambiar la contraseña de manera periódica, como se haría con las contraseñas de Windows.

Nota: Debido a que la cuenta de inicio sa es ampliamente conocida por los usuarios malintencionados, en la medida de lo posible, sería mejor deshabilitar o eliminar esta cuenta y en lugar de utilizar esta cuenta, y hacer que los administradores de Windows sean parte del rol del servidor sysadmin.

Cualquier persona que tenga el rol sysadmin podrá iniciar sesión y administrar el servidor.

5.3 Permisos

Los permisos **determinan las acciones que los usuarios pueden realizar en SQL Server o en una base de datos.**

Los permisos se otorgan de acuerdo con el ID de inicio de sesión, las pertenencias a grupos y a roles.

Los usuarios deben tener los permisos adecuados antes de que puedan realizar cualquier acción que pueda cambiar las definiciones de la base de datos o acceda a los datos.

En SQL se usan los tipos de permisos siguientes:

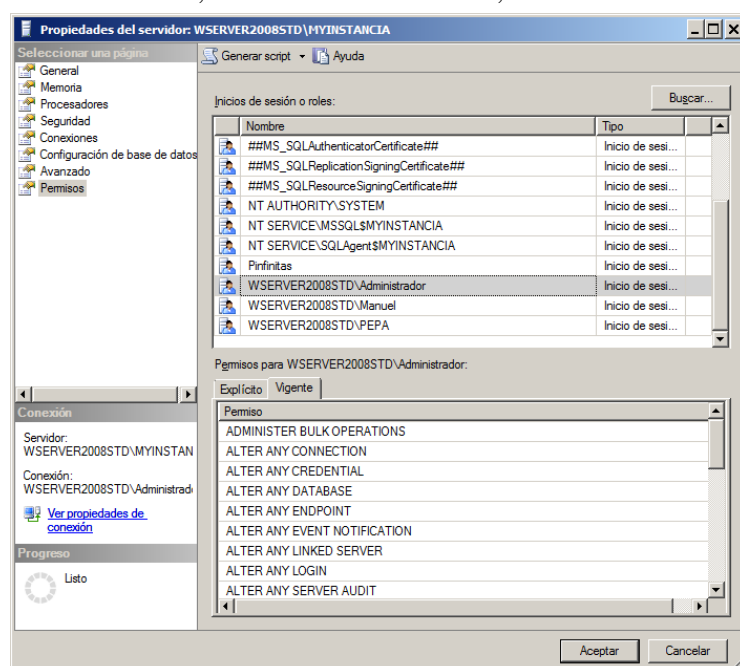
- Permisos de objetos
- Permisos de instrucciones

5.3.1 Permisos de objetos

Los objetos para los que se podrá administrar permisos dependen del alcance.

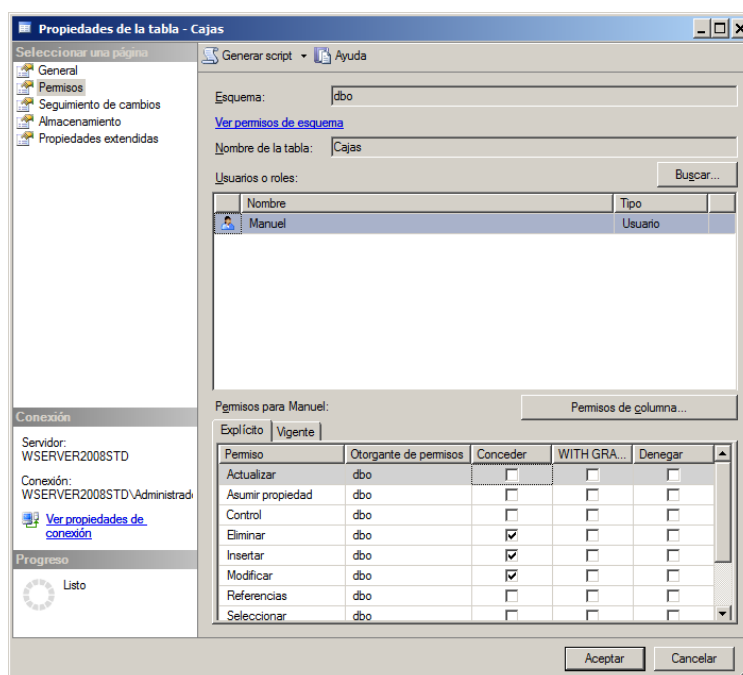
En la documentación de SQL Server, los objetos a los cuales el sistema de autorización de SQL Server controla el acceso, los denomina segurables.

En el nivel del servidor: los permisos de servidor se asignan a los administradores de base de datos para permitirles realizar tareas administrativas. Se pueden otorgar permisos para servidores, cuentas de inicio de sesión, roles del servidor y accesos al servidor. Por ejemplo, los permisos del servidor comprenden los permisos SHUTDOWN, CREATE DATA BASE, BACKUP DATABASE etc...



En el nivel de base de datos se puede administrar permisos **que se conceden para permitir el acceso a objetos de base de datos**, como permisos para roles de aplicación, tabla, vistas, tipos de datos definidos por el usuario, esquemas, procedimientos almacenados, usuarios, vistas, colecciones de esquema XML, etc...

- Por ejemplo, se puede conceder el permiso de seleccionar información de una tabla (hacer SELECT) pero denegar el permiso de insertar, actualizar o borrar (INSERT, UPDATE o DELETE) información de la tabla.



5.3.2 Permisos de instrucciones

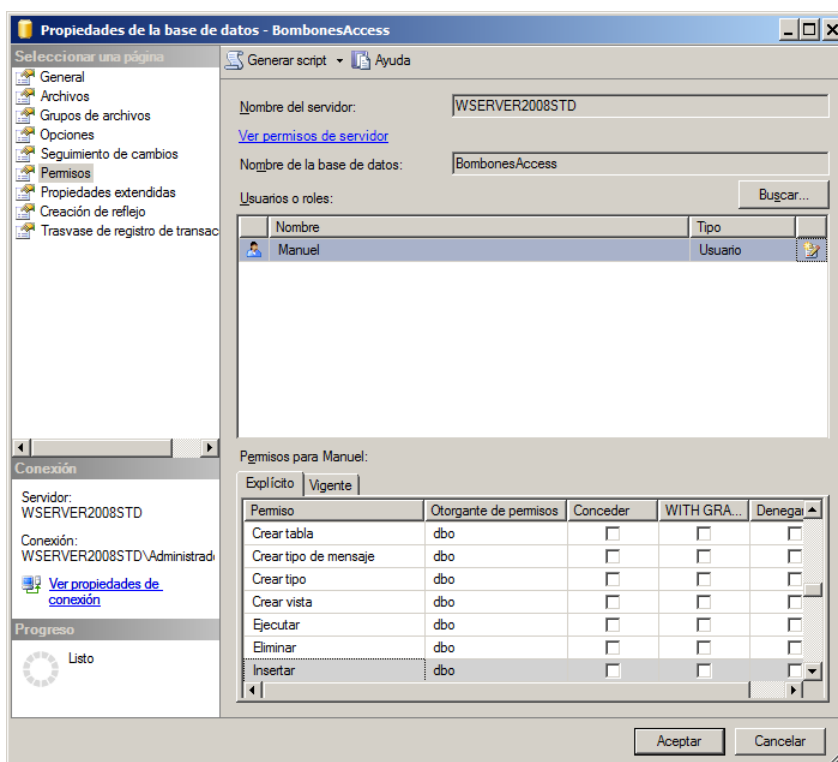
Controlan las acciones de administración, como la creación de una base de datos o añadir objetos a una base de datos.

Solo los integrantes del rol **sysadmin** y los **propietarios de la base de datos** pueden asignar permisos de instrucción.

De manera predeterminada, a los inicios de sesiones normales, no se le conceden permisos sobre las instrucciones, por lo que es necesario concederlos explícitamente a los inicios de sesión que no sean administradores.

Por ejemplo, si un usuario necesita crear una tabla, o una vista en una base de datos, se le deberá asignar el permiso de CREATE TABLE o CREATE VIEW.

Por ejemplo, los permisos sobre instrucciones que se pueden conceder o negar son: CREATE DATABASE (permiso de crear base de datos, solo los que son administradores o estar en la base de datos master), CREATE TABLE (permiso para crear tablas), CREATE VIEW(permiso para crear vistas), CREATE PROCEDURE(Permiso para crear un procedimiento almacenado), CREATE RULE (permite crear una regla de columna), CREATE FUNCTION (permiso para crear funciones definidas por el usuario), etc...

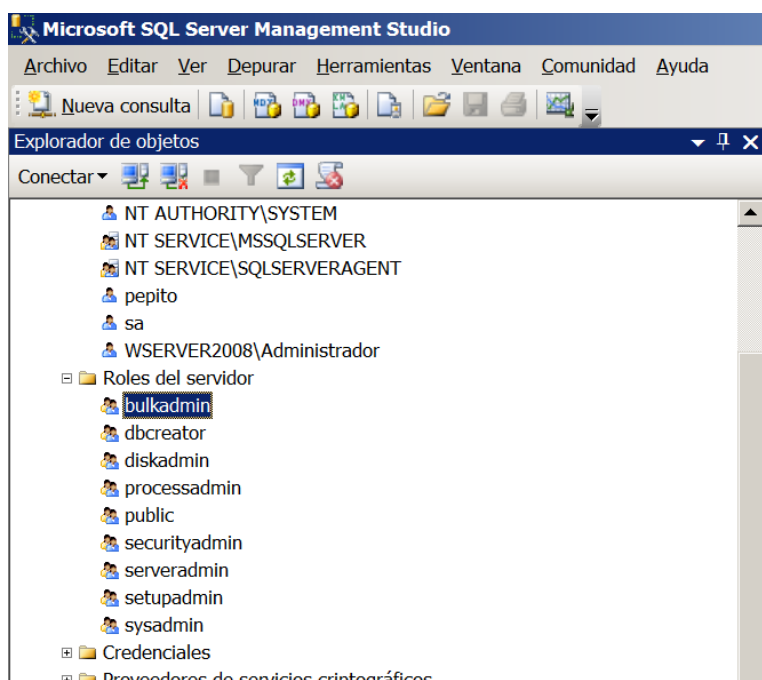


5.4 Funciones (roles) de base de datos

Las **funciones** se parecen bastante a los grupos de Windows, en el sentido de que permiten reunir **usuarios en una única unidad a la que puede aplicar permisos**.

Existen tanto las funciones al nivel de servidor como las funciones al nivel de base de datos. Un usuario puede ser miembro de más de una función.

SQL Server proporciona funciones de servidor y de base de datos **predefinidas para funciones administrativas comunes** para que pueda asignar una selección de permisos administrativos a un usuario en concreto, simplemente haciéndole miembro de las funciones apropiadas. También puede crear sus propias funciones de base de datos definidas por el usuario.



5.4.1 Funciones fijas del servidor

Las **funciones fijas** del servidor proporcionan **agrupaciones de privilegios** administrativos al nivel de servidor.

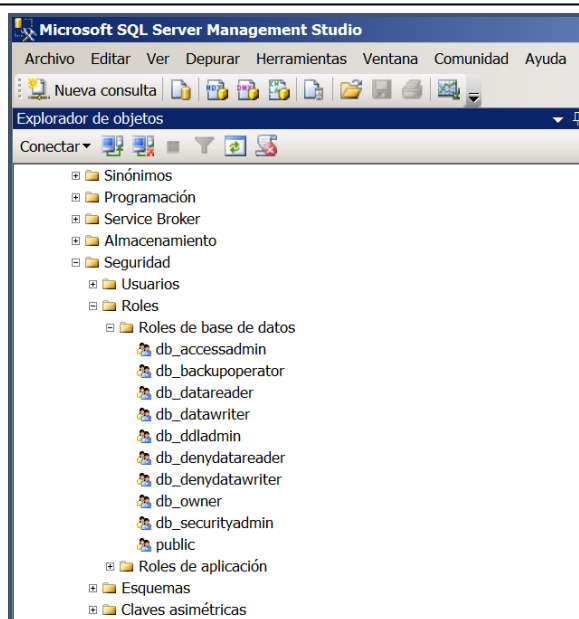
Se administran independientemente de las bases de datos. Algunos ejemplos de funciones fijas del servidor son las funciones para los administradores de sistema, creadores de bases de datos y administradores de seguridad.

Funciones fijas de servidor	Descripción
sysadmin	Está por encima de las restantes funciones y tiene la autoridad para realizar cualquier actividad en SQL Server.
setupadmin	Da a un usuario capacidad sobre la administración de servidores vinculados y los procedimientos almacenados de inicio que se instalen.
serveradmin	Se usa para conceder a un usuario la autoridad para realizar cambios en la configuración de SQL Server y apagarlo.
securityadmin	Proporciona a un usuario la capacidad de administrar los accesos, crear permisos de base de datos y leer registros de errores.
processadmin	Permite a un usuario gestionar los procesos que se ejecutan bajo SQL Server.
diskadmin	Otorga la capacidad para administrar archivos de disco
dbcreator	Proporciona a un usuario la capacidad de crear y modificar, eliminar y restaurar bases de datos en SQL Server.
bulkadmin	Otorga la capacidad de realizar insercciones masivas de datos en una base de datos.

5.4.2 Funciones fijas de base de datos

Las funciones fijas de bases de datos proporcionan agrupaciones de privilegios administrativos al nivel de base de datos.

Algunos ejemplos de funciones fijas de bases de datos son funciones para hacer copias de seguridad y restaurar una base de datos, para leer y modificar datos.



Funciones fijas de base de datos	Descripción
db_accessadmin	Proporciona a un usuario la capacidad de añadir o eliminar accesos a una base de datos.
db_backupoperator	Permite la realización de copias de seguridad.
db_datareader	Permite al usuario visualizar datos en cualquier tabla de la base de datos.
db_datawriter	Permite al usuario insertar, actualizar o borrar datos en cualquier tabla de la base de datos.
db_ddladmin	Permite crear, modificar y eliminar objetos de la base de datos.
db_securityadmin	Permite a sus miembros crear y mantener las funciones de base de datos y sus permisos, así como gestionar los permisos dentro de la base de datos.
db_denydatareader.	Diseñada para restringir el acceso a los datos de una base de datos. Los miembros de esta función no pueden leer los datos de las tablas dentro de la base de datos.
db_denydatawriter	Deniega la capacidad de escribir, modificar, actualizar o borrar datos.
db_owner.	Función que se asigna a un usuario que es propietario de una base de datos. Pueden asignar permisos, modificar opciones de la base de datos, realizar labores de mantenimiento y realizar cualquier tarea de administración sobre la base de datos, incluso su eliminación.
Public	Función por defecto para todos los usuarios de la base de datos. Los usuarios heredan los privilegios y permisos de la función public. Estos cuentan con los privilegios mínimos.

5.4.3 Funciones de aplicación

Permiten crear roles protegidos contra contraseña para **aplicaciones específicas**. Por ejemplo, supongamos que el departamento de recursos humano utiliza un programa personalizado para acceder a la base de datos y no deseamos que usen ningún programa por miedo a que dañen los datos. Podemos establecer este nivel de seguridad mediante una función de aplicación. Con esta función especial, los usuarios no pueden acceder a los datos usando únicamente el inicio de sesión de SQL server y la cuenta de base de datos, además deben utilizar la aplicación.

6. Arquitectura de SQL Server

A partir de SQL Server 2005, se introduce un nuevo modelo para administrar las unidades fundamentales de datos dentro de la BD.

Todos los datos están dentro de un objeto base de datos. Cada objeto BD contiene objetos de Esquema y todo ellos contienen las tablas, los índices, las vistas y otros objetos que integran la BD. Tenemos:

- **Base de datos.** Incluye todos los objetos dentro de una base y su propietario es un usuario específico.
- **Esquema.** Incluye todos los objetos definidos dentro.
Son **contenedores de objetos** que se usan para definir los **espacios de nombres de objetos** dentro de la base de datos.
Simplifican la administración porque crean subconjuntos de objetos que se pueden administrar de forma colectiva.
- **Objeto contenido en el esquema.** Cualquier tabla, vista u otro objeto individual que está definido y es propiedad de un esquema específico.

6.1 Bases de datos de SQL Server

Cada SQL Server tiene **dos tipos de bases de datos**:

■ Bases de datos de sistema

■ Bases de datos de usuario.

Estructuralmente no hay ninguna diferencia entre las bases de datos de sistema y de usuario.

- SQL Server reconoce y requiere bases de datos de sistema para su propia utilización. Las bases de datos de sistema almacenan información sobre SQL Server como un conjunto. SQL Server utiliza estas bases de datos para operar y administrar el sistema.
- Las bases de datos de usuario son bases de datos creadas por los usuarios.

En la versión Microsoft SQL Server 2008 Enterprise Edition, hay disponible las instantáneas de base de datos: Una imagen de una base de datos en un punto determinado de tiempo

Cuando se instala SQL Server, la instalación de SQL Server crea cuatro bases de datos de sistema y podemos también instalar dos ejemplos de bases de datos de usuario.

6.1.1 Bases de datos de sistema

La tabla siguiente describe cada base de datos de sistema:

Base de datos	Descripción
Master	<p><u>Registra toda la información de sistema</u> de SQL Server para una instancia de SQL Server.</p> <p>Dentro de esta información se incluyen <u>los metadatos de todas las instancias</u>, como las cuentas de inicio de sesión, los extremos, los servidores vinculados y la configuración del sistema. Asimismo, maestra es la base de datos que registra la existencia de las demás bases de datos, la ubicación de los archivos de las bases de datos y la información de inicialización de SQL Server.</p> <p>Por lo tanto, SQL Server no puede iniciarse si la base de datos master no está disponible.</p> <p>En SQL Server, los <u>objetos de sistema</u> (base de datos, funciones, etc.) no se almacenan en la base de datos master, sino en la base de datos de resource</p> <p>Esta base de datos <u>se modifica cada vez que se crea base de datos</u>, se administra una</p>

	<p>cuenta o se cambia parámetros de configuración.</p> <p>Como es una base de datos del sistema muy importante, hay que crear copia de seguridad de master de manera regular.</p> <p>Los nombres de los archivos físicos de la base de datos Master son master.mdf y mastlog.ldf. De forma predeterminada, estos archivos se encuentran en <unidadC:\Program Files\Microsoft SQL Server\MSSQL10_50.Instancia\MSSQL\DATA.</p>
Model	<p>Se utiliza como plantilla para todas las bases de datos creadas en la instancia de SQL Server.</p> <p>Las modificaciones hechas a la base de datos model, como el tamaño de la base de datos, la intercalación, el modelo de recuperación, permisos y otras opciones de base de datos, se aplicarán a las bases de datos que se creen con posterioridad.</p> <p>Uso de model: Cuando se emite una instrucción CREATE DATABASE, la primera parte de la base de datos se crea mediante la copia del contenido de la base de datos model. El resto de la nueva base de datos se llena a continuación con páginas vacías.</p> <p>Si se modifica la base de datos model, todas las bases de datos creadas posteriormente heredan los cambios. Por ejemplo, se podrían establecer permisos u opciones de base de datos o agregar objetos, como tablas, funciones o procedimientos almacenados.</p> <p>Los nombres de los archivos físicos de la base de datos Model son model.mdf y modellog.ldf. De forma predeterminada, estos archivos se encuentran en <unidadC:\Program Files\Microsoft SQL Server\MSSQL10_50.Instancia\MSSQL\DATA.</p>
Tempdb	<p>Proporciona áreas de almacenamiento para tablas temporales y otras necesidades temporales de almacenamiento de trabajo, como el procesamiento de consultas.</p> <p>Tempdb se crea de nuevo cada vez que se inicia SQL Server, de forma que el sistema siempre se inicia con una copia limpia de la base de datos.</p> <p>Las tablas y los procedimientos almacenados temporales se quitan automáticamente en la desconexión y ninguna conexión permanece activa cuando se cierra el sistema. Por tanto, en la base de datos tempdb no hay nada que deba guardarse de una a otra sesión de SQL Server.</p> <p>No se permite realizar operaciones de copia de seguridad y restauración en tempdb.</p> <p>Los nombres de los archivos físicos de la base de datos Tempdb son tempdb.mdf y templog.ldf. De forma predeterminada, estos archivos se encuentran en <unidadC:\Program Files\Microsoft SQL Server\MSSQL10_50.Instancia\MSSQL\DATA.</p>
Resource	<p>Base de datos de sólo lectura que contiene los objetos del sistema (tablas del sistema, vistas del sistemas, funciones del sistemas, procedimientos del sistema, etc..) que se incluyen con SQL Server.</p> <p>La base de datos Resource no está visible, los usuarios no pueden utilizarla ni conectarse a ella, es decir, los objetos del sistema persisten físicamente en la base de datos Resource, pero aparecen lógicamente en el esquema sys de cada base de datos.</p> <p>Los nombres de los archivos físicos de la base de datos Resource son Mssqlsystemresource.mdf y mssqlsystemresource.ldf. Estos archivos se encuentran en <unidad>:\Archivos de programa\Microsoft SQL Server\MSSQL10_50.<nombre_de_instancia>\MSSQL\Binn\.</p> <p>Cada instancia de SQL Server tiene un solo archivo mssqlsystemresource.mdf asociado y las instancias no lo comparten.</p>
MsdB	<p>La usa el servicio Agente SQL Server cuando realiza manejo de alertas, notificaciones y tareas programadas</p> <p>Los nombres de los archivos físicos de la base de datos MsdB son MSDBData.mdf y MSDBLog.ldf. De forma predeterminada, estos archivos se encuentran en <unidad>:\Archivos de programa\Microsoft SQL Server\MSSQL.1\MSSQL\Data.</p>

6.1.2 Bases de datos de usuario

Base de datos de los usuarios.

Sql Sever 2008 permite instalar una base de ejemplo AdventureWorks. Ésta base proporcionan ejemplos útiles que usted puede utilizar mientras aprende cómo trabajar con SQL Server. No se requieren

para que SQL Server opere correctamente.

6.2 Tablas del sistema

Las **tablas de sistema** almacenan información, denominada **metadatos**, sobre el sistema y los objetos en las bases de datos. Los metadatos son información sobre datos.

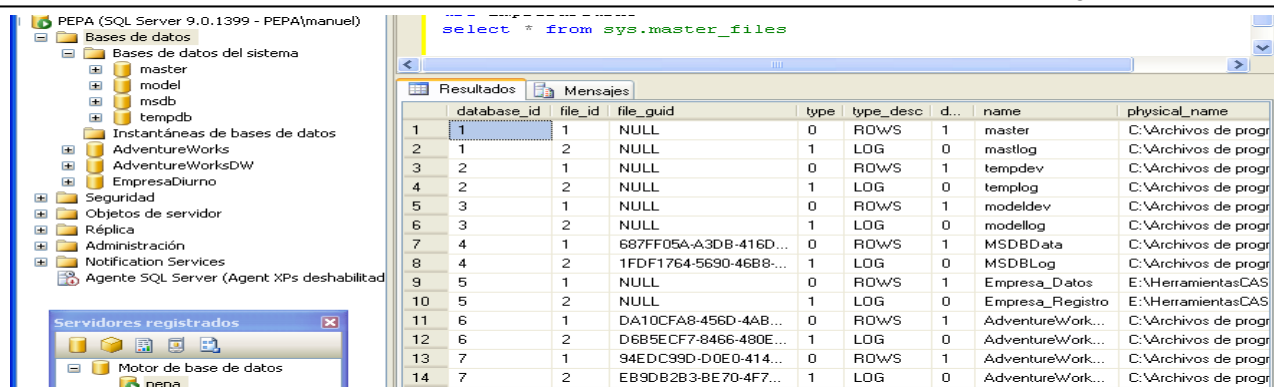
- **Catálogo de la base de datos:** Cada base de datos (incluida la master) contiene un grupo de tablas de sistema que almacenan metadatos sobre esa base de datos en concreto. Este grupo de tablas de sistema se denomina catálogo de la base de datos.
- **Catálogo del sistema:** El catálogo de sistema, encontrado sólo en la base de datos master, es un grupo de tablas de sistema que almacenan metadatos sobre el sistema completo y sobre todas las otras bases de datos.

Los metadatos de las **tablas del sistema** están disponibles a través de **vistas del catálogo** del sistema y estas vistas empiezan todas con **el prefijo sys**. La tabla siguiente identifica algunas vistas del catálogo del sistema comúnmente utilizadas.

Tabla de sistema	Base de datos	Función
sys.server_principals	Master	Contiene una fila por cada cuenta de conexión que se puede conectar a SQL Server.
sys.master_file	Master	Contiene una fila por cada archivo de cada una de base de datos almacenadas en SQL Server.
sys.messages	Master	Contiene una fila por cada error de advertencia del sistema que puede devolver SQL Server.
sys.databases	Master	Contiene una fila por cada base de datos en SQL Server.
sys.database_principals	Todas	Contiene una fila por cada usuario de Windows, grupo de Windows, usuario de SQL Server o función de SQL Server en una base de datos.
sys.database_file	Todas	Contiene una fila por cada archivo de una base de datos como se almacena en la propia base de datos. Es una vista por base de datos.
sys.procedures	Todas	Contiene una fila para cada objeto que es un procedimiento
sys.foreign_keys	Todas	Contiene una fila por objeto que constituye una restricción FOREIGN KEY
sys.indexes	Todas	Contiene una fila por índice de un objeto tabular, como una tabla, una vista o una función con valores de tabla

Se puede consultar estas vistas del catálogo del sistema, tal como se haría con cualquier otra tabla para recuperar información acerca del sistema. Ejemplo:

sys.master_file



Cuando se escriben aplicaciones que **recuperen metadatos de tablas de sistema**, deberían utilizarse **procedimientos almacenados del sistema**, **funciones de sistema** o **vistas de esquema de información** proporcionadas por el sistema.

6.3 Procedimientos almacenados del sistema

Para hacer que le sea más fácil reunir información sobre el estado del servidor y los objetos de bases de datos, SQL Server proporciona un grupo de consultas escritas denominadas procedimientos almacenados del sistema.

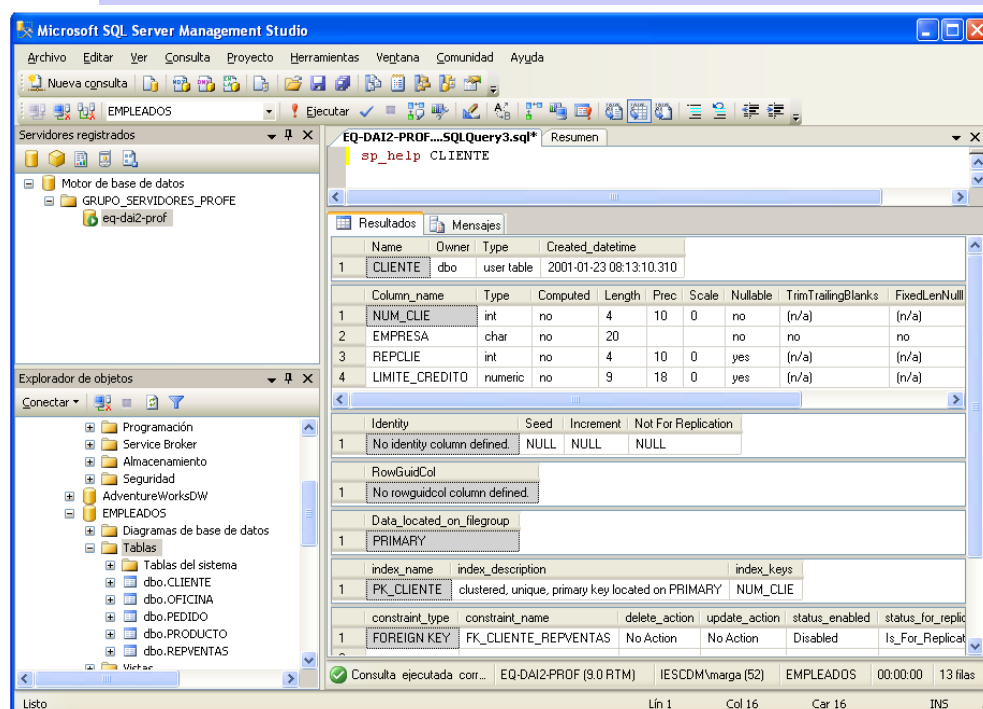
La **mayoría de los nombres de los procedimientos almacenados del sistema empiezan** con el **prefijo sp**.

Ejemplos de procedimientos almacenados son:

- **sp_help [nombre_objeto]** . Proporciona información acerca de objetos especificados de base de datos
- **sp_helpdb [nombre_basededatos]**. Proporciona información acerca de la base de datos especificada.

El siguiente ejemplo ejecuta un procedimiento almacenado del sistema para obtener información sobre la tabla CLIENTE.

```
/* Información de la tabla CLIENTE */
sp_help CLIENTE
```



Muchos **procedimientos almacenados** del sistema se utilizan para **crear o modificar información**

del sistema u objetos de bases de datos mediante la modificación de las tablas del sistema. Por ejemplo, el procedimiento almacenado del sistema **sp_addlogin** crea una nueva cuenta de conexión en la tabla del sistema master syslogins.

Ej: En el siguiente ejemplo se crea un inicio de sesión de SQL Server para el usuario pepito, con una contraseña abc123., sin especificar una base de datos predeterminada.

```
EXEC sp_addlogin 'pepito', 'abc123.';
```

Ej: Se crea un inicio de sesión de SQL Server que tiene una base de datos predeterminada

```
EXEC sp_addlogin 'pepito', 'abc123.', 'clientes';
```

6.4 Funciones del sistema

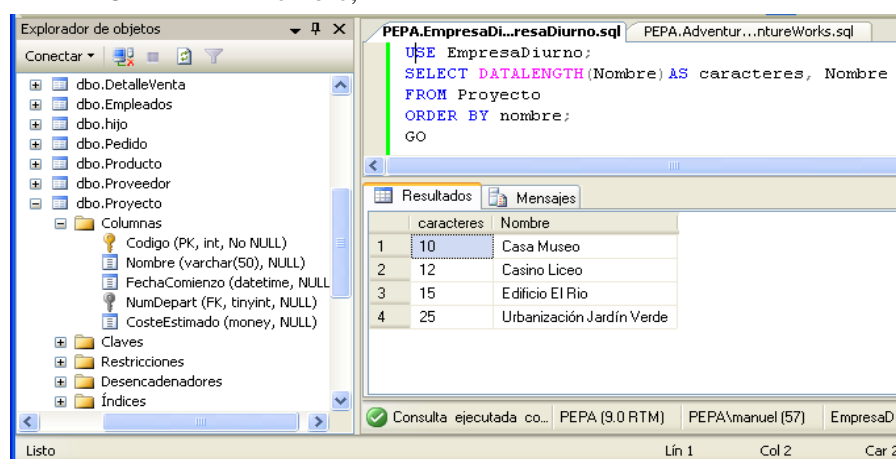
Las funciones del sistema proporcionan un método para consultar tablas del sistema desde dentro de las instrucciones Transact-SQL.

Las funciones del sistema devuelven valores únicos específicos. La siguiente tabla describe algunas de las funciones del sistema más utilizadas:

Función del sistema	Parámetros pasados	Resultados
DB_ID	Nombre de la base de datos	Devuelve el ID de la base de datos.
USER_NAME	ID del usuario	Devuelve el nombre del usuario.
COL_LENGTH	Nombres de tabla y columna	Devuelve el ancho de la columna.
DATALENGTH	Expresión	Devuelve la longitud actual del valor de una expresión de cualquier tipo de datos.

Ejemplo: Para ver la longitud de cada fila del campo nombre en tabla proyecto de una base de datos llamada EmpresaDiurno ordenada por el campo nombre:

```
USE EmpresaDiurno;
SELECT DATALENGTH(Nombre) AS caracteres, Nombre
FROM Proyecto
ORDER BY nombre;
```



6.5 Vistas de esquema de información

Una vista de esquema de información es otro de los diversos métodos que proporciona SQL Server 2008 para obtener metadatos.

Las vistas de esquema de información proporcionan una visión interna independiente de tablas del

sistema de los metadatos de SQL Server. Estas vistas son de conformidad con la definición del estándar ANSI SQL para el esquema de información. Las vistas de esquema de información permiten que las aplicaciones funcionen correctamente, aunque las versiones futuras del producto cambien las tablas del sistema significativamente.

Las vistas de esquema de información se definen en un esquema especial llamado **INFORMATION_SCHEMA**. Este **esquema se incluye en cada base de datos**. Cada vista de esquema de información contiene metadatos para todos los objetos de datos almacenados en esa base de datos en concreto.

Las vistas de esquemas de información proporcionan una manera de recuperar información valiosa de las tablas de sistema sin que se tenga que escribir una consulta para estas tablas.

La tabla siguiente muestra ejemplos de vistas de esquema de información:

Vista de esquema de información	Resultados
information_schema.schemata	Devuelve una fila por cada esquema de la base de datos actual.
information_schema.tables	Lista de tablas en la base de datos.
information_schema.columns	Información acerca de columnas definidas en la base de datos.
information_schema.tables_privileges	Información de seguridad para tablas en la base de datos.

- El siguiente ejemplo consulta una vista de esquema de información para obtener una lista de tablas en una base de datos.

Use EmpresaDiurno

SELECT * FROM information_schema.tables

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	EmpresaDiurno	dbo	hijo	BASE TABLE
2	EmpresaDiurno	dbo	beca	BASE TABLE
3	EmpresaDiurno	dbo	concede	BASE TABLE
4	EmpresaDiurno	dbo	sysdiagrams	BASE TABLE
5	EmpresaDiurno	dbo	Asigna	BASE TABLE
6	EmpresaDiurno	dbo	Ventas	BASE TABLE
7	EmpresaDiurno	dbo	DetalleVenta	BASE TABLE
8	EmpresaDiurno	dbo	Producto	BASE TABLE
9	EmpresaDiurno	dbo	Departamento	BASE TABLE
10	EmpresaDiurno	dbo	Empleados	BASE TABLE
11	EmpresaDiurno	dbo	Proyecto	BASE TABLE
12	EmpresaDiurno	dbo	Proveedor	BASE TABLE
13	EmpresaDiurno	dbo	Pedido	BASE TABLE

6.6 Objetos de bases de datos

Una base de datos es una colección de datos almacenados en tablas y los objetos que soportan el almacenaje, recuperación, seguridad e integridad de estos datos.

La tabla siguiente resume los objetos de bases de datos de SQL Server.:

Objeto de bases de datos	Descripción
Tabla	Almacena los datos como una colección de filas y columnas.
Tipo de datos	Define los tipos de valores de datos permitidos para una columna variable. SQL Server proporciona tipos de datos proporcionados por el sistema. Los usuarios pueden crear tipos de datos definidos por el usuario.
Restricción	Se utiliza para definir reglas de integridad para una columna o conjunto de columnas en una tabla y es el mecanismo estándar para aplicar la integridad de datos.

Valores predeterminados	Define un valor que se almacena en una columna si no se proporciona ningún otro valor.
Regla	Define una expresión que se utiliza para comprobar la validez de los valores que están almacenados en una columna o tipo de datos.
Índice	Es una estructura de almacenaje que proporciona orden y acceso rápido para recuperación de datos y puede aplicar la unicidad de datos.
Vista	Proporciona una manera de ver los datos de una o más tablas y otras vistas en una base de datos.
Procedimiento almacenado	Es una colección con nombre de instrucciones Transact-SQL o secuencias que se ejecutan juntas.
Desencadenador	Es una forma especial de procedimiento almacenado que se ejecuta automáticamente cuando el usuario modifica datos en una tabla.

6.6.1 Nombres de objetos en SQL Server

SQL Server proporciona **una serie de reglas estándar para identificadores de objetos** y un método para usar delimitadores de los identificadores que no son estándar. Es recomendable que los nombres de objetos usen caracteres identificadores estándar cuando sea posible.

Identificadores estándar

Los identificadores estándar tienen una longitud de 1 a 128 caracteres. No están permitidos espacios intermedios en los identificadores estándar. Las reglas para utilizar identificadores incluyen:

- El primer carácter debe ser un carácter alfabético de la a-z o A-Z o uno de los símbolos @, #, _ o \$.
- Después del primer carácter, los identificadores pueden incluir letras, números o los símbolos @, #, _ o \$.
- Los nombres de identificadores que comienzan por un símbolo tienen usos especiales:
 - Un identificador que comienza por @ indica una variable local o un parámetro.
 - Algunos identificadores de funciones SQL Server comienzan por @@, así que se debe evitar crear identificadores que comiencen por @@
 - Un identificador que comience por # indica una tabla temporal o procedimiento.
 - Un identificador que empiece por ## indica un objeto temporal global.
 - Los nombres para objetos temporales no deben tener más de 29 caracteres, incluyendo los símbolos # o ## porque SQL Server les asigna internamente un sufijo numérico.

Identificadores delimitados

Si un identificador no cumple con alguna de las reglas que acabamos de ver para los identificadores estándar **debemos delimitarlo**.

Los identificadores delimitados se usan principalmente cuando los nombres contienen espacios en blanco u otros caracteres no permitidos en los identificadores estándar o cuando se utilizan palabras reservadas para nombres de objetos o parte de los nombres de objetos.

Los **identificadores delimitados** se escriben entre **corchetes o dobles comillas** cuando se utilizan en instrucciones Transact-SQL.

Ejemplo

```
CREATE TABLE Empleado
(
  [ID Empleado] int,
  Nombre varchar(15),
  Apellidos varchar(40),
  Telefono char(9)
)
```

Los identificadores entrecomillados pueden ser usados sólo si la opción SET

QUOTED_IDENTIFIER está seleccionada en ON. Delimitadores con corchetes pueden ser usados siempre, a pesar del estado de la opción SET QUOTED_IDENTIFIER.

6.6.2 Pautas para nombrar objetos de bases de datos

Cuando se nombran objetos de bases de datos se deben:

- (1) **Asignar nombres cortos.**
- (2) Utilizar **nombres de fácil comprensión.**
- (3) Usar un sistema de **nomenclatura claro y simple.** Decidir qué es lo que funciona mejor para cada situación y ser consecuente con ello.
- (4) Usar un identificador que **distinga el tipo de objeto**, especialmente para vistas y procedimientos; por ejemplo, comience todos los nombres de vistas con las letras vw. Los administradores de sistemas a veces confunden vistas con tablas, un descuido que puede ocasionar problemas inesperados.
- (5) Mantener **nombres de objetos y de usuarios únicos**; por ejemplo, evite crear un nombre de tabla ventas y un nombre de usuario ventas en la misma base de datos.
- (6) Los nombres no distinguen entre mayúsculas y minúsculas, pero se mantienen tal y como se escriben. Las instrucciones Transact-SQL no distinguen entre mayúsculas y minúsculas, pero generalmente se introducen en mayúsculas.

6.6.3 Referencia a objetos de SQL Server

Para referenciar un objeto SQL Server se puede especificar el nombre completo del objeto o especificar sólo una parte del nombre del objeto y dejar que SQL Server determine el resto del nombre según el contexto en el que se está trabajando. **El nombre completo de un objeto SQL Server incluye cuatro identificadores.** Nombre del servidor, nombre de la base de datos, nombre del esquema y nombre del objeto según el siguiente formato:

servidor.base_datos.esquema.objeto

Un nombre que especifica las cuatro partes es conocido como nombre **completo cualificado**.

Cada objeto creado en SQL Server debe tener un único nombre completo cualificado.

Por ejemplo, dos tablas pueden ser nombradas PERSONA en la misma base de datos siempre que pertenezcan a dos esquemas diferentes. Además, los nombres de columnas deben ser únicos dentro de una tabla o vista.

Cuando se hace referencia a un objeto no se tiene por qué especificar siempre el servidor, la base de datos y el esquema; se pueden omitir los identificadores principales. Se pueden omitir incluso los identificadores intermedios si su posición es indicada por puntos. Los formatos válidos de los nombres de objetos son:

servidor.base_datos.esquema.objeto
base_datos.esquema.objeto
base_datos..objeto
esquema.objeto
objeto

Cuando se crea un objeto, SQL Server usa por defecto los siguientes parámetros cuando no especificamos el nombre completo.

- El servidor por defecto es el **servidor local**
- La base por defecto es la **actual**

- El esquema por defecto es el esquema **dbo**. El esquema ha sustituido al propietario utilizado hasta la versión SQL Server 2000.

6.6.4 Esquemas

Los esquemas (schemas) son objetos de base de datos que facilitan la resolución de nombres, es decir, los esquemas facilitan un espacio de nombres (namespace).

Todo objeto de base de datos (ej: tabla, vista, procedimiento almacenado, etc.) tiene asociado uno y sólo un esquema.

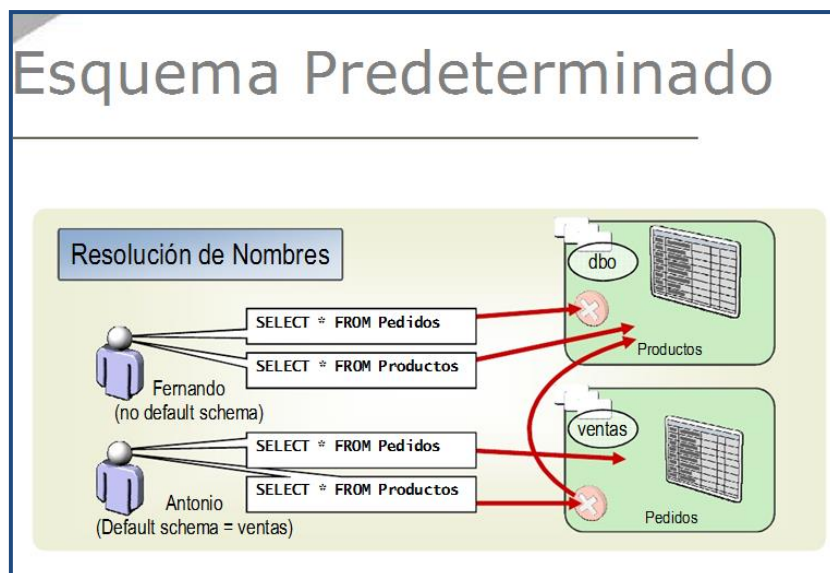
Además, todo usuario tiene asignado un esquema por defecto (default schema), en su defecto, el **esquema dbo**

- Cada objeto pertenece a un esquema de base de datos. Si al crearlo no se especifica un esquema, por defecto se le asigna dbo
- Un **esquema** se puede considerar como un contenedor de objetos.
- Los **esquemas** se pueden crear y modificar en una base de datos, y a los **usuarios** se les puede conceder acceso a un esquema.
- Es posible mover objetos entre esquemas.

Aparece el siguiente concepto: “**esquema por defecto**”. Esta propiedad del usuario se usa para determinar la tabla a la que el usuario quiere acceder para el caso en que se haga referencia a un objeto por su nombre sin incluir el esquema como parte del nombre.

En el siguiente ejemplo se intenta explicar el funcionamiento de esta propiedad:

Supongamos que en una base se tiene el usuario “Antonio” que tiene como default_schema “ventas”. Al intentar acceder a la tabla “Productos”, SQL Server 2008 intenta primero ubicarla en el esquema “Ventas”. Si la tabla no existe en ese esquema, SQL Server 2008 intenta como alternativa ubicar la tabla dentro del esquema “dbo”.



6.6.5 Tipos de datos

Todos los objetos con datos tienen asociado un tipo de datos en Transact-SQL que define la clase de los datos del objeto (carácter, entero, binario, etc). Cuando se asigna un tipo de datos a un objeto se definen cuatro atributos del objeto:

- El **tipo de datos** que contiene el objeto (carácter, entero o binario).
- La **longitud** del valor almacenado o su tamaño (la longitud de un tipo de datos image, binary o varbinary se define en bytes, la longitud de cualquier tipo de datos numérico es el número de bytes necesarios para contener el número de dígitos permitido para ese tipo de datos, y la longitud de los tipos de datos Unicode y de cadena de caracteres se define en caracteres)
- La **precisión del número** para tipos de datos numéricos (la precisión es el número de dígitos totales que puede contener el número).

- La **escala del número** para tipos de datos numéricos (la escala es el máximo número de dígitos que se puede almacenar a la derecha del separador decimal)

Por ejemplo: Si un objeto se define como money, puede tener un máximo de 19 dígitos y 4 de ellos pueden estar a la derecha del decimal. El objeto usa 8 bytes para almacenar los datos. Por tanto, el tipo de datos money tiene una precisión de 19, una escala de 4 y una longitud de 8.

Los tipos de datos en SQL Server son los siguientes:

Tipo de datos	Clase	Tamaño en bytes	Naturaleza de los datos
Bit	Entero	1	En las columnas bit sólo se pueden almacenar 0 y 1. se usa para representar valores TRUE o FALSE, o YES o NO.
Bigint	Entero	8	Números enteros entre -2^{63} y $2^{63}-1$
Int	Entero	4	
SmallInt	Entero	2	
TinyInt	Entero	1	
Decimal o Numeric	Decimal/Numérico o	Varia	Los dos nombres son sinónimos. Va desde -10^{38} y $10^{38}-1$. se especifica la precisión (o el número de dígitos que puede contener el objeto) y la escala (número de dígitos que puede colocarse a la derecha del separador decimal). Se utiliza el tipo de datos decimal para almacenar números con decimales cuando los valores de datos se deban almacenar exactamente como se especifican.
Money	Dinero	8	Va de -2^{63} y 2^{63} más una precisión de cuatro posiciones decimales
SmallMoney	Moneda	4	precisión de cuatro posiciones decimales
Float o Real	Numéricos aproximados	Varía	Debido a la naturaleza aproximada de los tipos de datos float y real, no se utilizan cuando se necesita un comportamiento numérico exacto, como, por ejemplo, en aplicaciones financieras, en operaciones que conlleven un redondeo o en comprobaciones de igualdad. En su lugar, se usan los tipos de datos enteros, decimal, money o smallmoney. También es conveniente evitar usar columnas float o real en las condiciones de búsqueda de la cláusula WHERE, especialmente los operadores = y <>. Es mejor limitar las columnas float y real a las comparaciones > o <.
DateTime	Fecha/Hora	8	Datos de fecha y hora desde el 1 de enero de 1753 hasta el 31 de diciembre de 9999 con una precisión de 3 décimas de segundo
SmallDateTime	Fecha/Hora	4	Datos de fecha y hora desde el 1 de enero de 1900 hasta el 6 de junio de 2079 con una precisión de un minuto
Timestamp/rowversion	Numérico especial	8	Es un número único para toda la base de datos que se actualiza cada vez que se actualiza una fila El tipo de datos timestamp de SQL Server no tiene nada que ver con horas o fechas. Los valores de tipo timestamp de SQL Server son números binarios que indican la secuencia relativa en la que se realizaron las modificaciones en una base de datos. El tipo de datos timestamp se desarrolló originalmente para admitir los algoritmos de recuperación de SQL Server. No se deben usar nunca columnas timestamp en claves, especialmente claves principales, porque el valor timestamp cambia cada vez que se modifica la fila. Para registrar en qué momento se producen modificaciones en una tabla, utilice un tipo de datos datetime o smalldatetime para registrar los sucesos y los desencadenadores con el fin de que se actualicen los datos automáticamente cuando se produzca alguna modificación.
UniqueIdentifier	Numérico especial	16	Es un identificador exclusivo global (GUID). Está garantizado que va a ser único en el espacio y en el tiempo.
Char	Numérico especial	Varía	Datos de caracteres de longitud fija. Los valores más cortos que la longitud establecida se rellenan con espacios hasta dicha longitud. char(n)

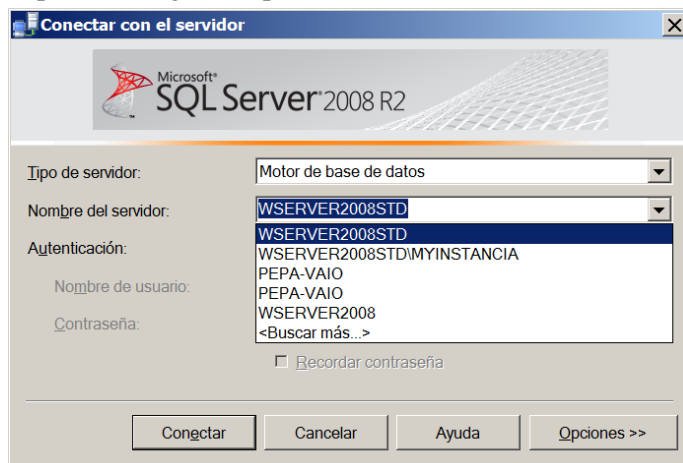
Tipo de datos	Clase	Tamaño en bytes	Naturaleza de los datos
VarChar	Carácter	Varía	Carácter de longitud variable. Los valores no se rellenan con espacios.
Text	Carácter	Varía	cadenas de caracteres con más de 8000 caracteres
NChar	Unicode	Varía	Los tipos de datos nchar, nvarchar, nvarchar y ntext se utilizan de la misma forma que los tipos char, varchar y text respectivamente, excepto que: Unicode admite un intervalo más amplio de caracteres Se necesita más espacio para almacenar caracteres Unicode. El tamaño máximo de las columnas nchar y nvarchar es 4000 caracteres, no 8000 caracteres como en las columnas char y varchar. Las constantes Unicode se especifican con una N a la izquierda: N'Una cadena Unicode'
Nvarchar	Unicode	Varía	
Ntext	Unicode	Varía	
Binary	Binario	Varia	Datos binarios de longitud fija con una longitud máxima de 8000 bytes
VarBinary	Binario	Varia	Datos binarios de longitud variable con una longitud máxima de 8000 bytes
Image	Binario	Varía	Se mantiene por herencia pero debe usarse varBinary(max)
Table	Otros	Especial	Se utiliza para almacenar un conjunto de resultados para un proceso posterior (es como las tablas temporales). Normalmente se usa para pasar uno de una función definida por el usuario. No se usa como tipo de datos dentro de una definición de tabla.
Sql_variant	Otros	Especial	permite que una única columna, parámetro o variable almacene valores de datos de distintos tipos. Cada instancia de una columna sql_variant registra el valor de los datos y los metadatos que describen el valor: su tipo de datos base, tamaño máximo, escala, precisión e intercalación.
XML	Carácter	Varía	Define un campo de caracteres para datos XML. Proporciona la validación de los datos frente a un esquema XML y el uso de funciones especiales orientadas a XML

7. Cómo conectarse a SQL Server

Puede conectarse a SQL Server utilizando la herramienta de administración gráfica SQL Server Management Studio de SQL Server o desde el símbolo del sistema.

7.1 Utilizando SQL Server Management Studio

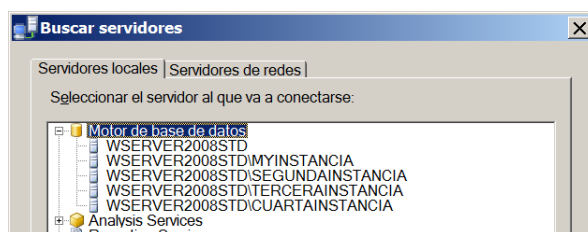
Podemos conectarnos a SQL Server, iniciando SQL Server Management Studio y conectarnos a un servidor de base de datos. Nos aparece la siguiente pantalla:



Tipo de servidor: selecciona el tipo de servidor al que se conectará: Database Engine (Motor de base de datos). El resto del cuadro de diálogo muestra simplemente las opciones que se aplican al tipo de servidor seleccionado. Cuando se registra un servidor desde Servidores registrados, el cuadro Tipo de servidor es de sólo lectura y coincide con el tipo de servidor que se muestra en el componente Servidores registrados.



Seleccionamos el **nombre del servidor**. En la lista desplegable podemos buscar todos los servidores de base de datos disponibles:



A continuación elegimos el **modo de autenticación**: Autenticación Windows o Autenticación SQL Server.



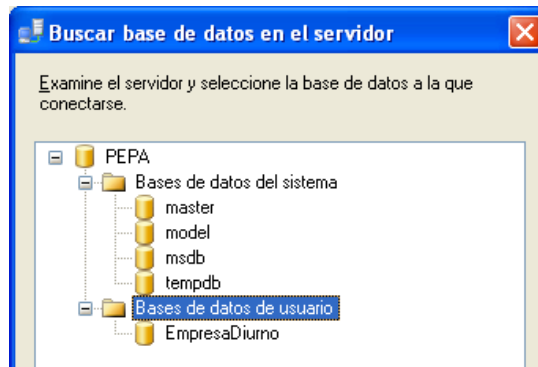
En la siguiente figura, se muestra un ejemplo de autenticación SQL Server utilizando la cuenta de inicio de sesión de administrador **sa**, en el cual hay que introducir la contraseña para este inicio.



Cuando nos conectamos a SQL Server, se debe utilizar la cuenta de usuario de Windows o una cuenta de login de SQL Server.

La **primera vez que se conecte a SQL Server**, se debe utilizar una **cuenta de administrador de Windows** (Se habrá introducido en la instalación) o utilizar una cuenta de **login sa** de SQL Server (También en la instalación, permite la posibilidad de crear esta cuenta).

En Opciones podemos conectarnos con una base de datos particular:



Si se conecta a SQL Server con éxito, entonces podrá configurar y utilizar SQL Server. Aparecerá la ventana de SQL Server Management Studio:

7.2 Ejecutando el comando SQLCMD

SQLCMD es una utilidad para el manejo de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL **mediante la línea de comandos**.

SQLCMD se comunica con SQL Server a través de la conexión OLE DB para acceso a las Base de datos **y consume muy pocos recursos**, lo que la convierte en una buena opción para aquellos sistemas en los que los recursos suponen una limitación

Utilizando la línea de comandos sqlcmd, se puede:

- Mandar instrucciones T-SQL a la base de datos SQL Server.
- Crear scripts y procedimientos.

- **Sqlcmd -U [login_id] -P [password] -S [nombreservidor]**
- **Sqlcmd ?** : se obtiene ayuda del comando

Abrimos una consola del sistema operativo:

```

C:\Archivos de programa\Microsoft SQL Server\MSSQL.1\MSSQL\Binn>sqlcmd -U sa -P
123
1> use EmpresaDiurno
2> go
Se cambió el contexto de la base de datos a 'EmpresaDiurno'.
1> select * from beca
2> go
codigo  descripción          cuantía
-----
1  INFANTIL              200.0000
2  PRIMARIA              300.0000
3  ESO                   500.0000
4  BACH/CICLOS           700.0000
5  UNIVERSIDAD          1200.0000

<5 filas afectadas>
1> exit
C:\Archivos de programa\Microsoft SQL Server\MSSQL.1\MSSQL\Binn>

```

-S [nombreservidor] si se omite, se abre una sesión en la **instancia predeterminada**

- **Opciones relacionadas con el inicio de sesión:**

-U login_id: Es el identificador de inicio de sesión del usuario.

-P password: Es una contraseña especificada por el usuario. En las contraseñas se distingue entre mayúsculas y minúsculas.

- Si se **utiliza la opción -U y no la opción -P** : además no se ha establecido la variable de entorno **SQLCMDPASSWORD**, sqlcmd solicita al usuario una contraseña.
- Si se utiliza la opción -P al final del símbolo del sistema sin una contraseña, sqlcmd utiliza la contraseña predeterminada (NULL).
- Si **no se especifica la opción -U ni la opción -P**, sqlcmd intenta **conectarse mediante el modo de autenticación de Microsoft Windows**. La autenticación se basa en la cuenta de Windows del usuario que está ejecutando sqlcmd.
- Si no se especifica -P: El mensaje de contraseña se muestra en la consola de la siguiente manera: Password: para que introduzcamos la contraseña.
- **-E trusted connection:** Este parámetro especifica una conexión de confianza, en la que SQL Server confía en que Windows verifique el nombre de usuario y la contraseña. No se introduce ningún nombre de usuario ni contraseña, por que SQLCMD detecta el nombre de usuario y contraseña que se uso para iniciar sesión en el equipo y utiliza esa información para iniciar sesión en SQL Server.
 - Si la opción -P se utiliza junto con la opción -E, se genera un mensaje de error.
 - Si se utiliza la opción -U junto con la opción -E, se genera un mensaje de error.
- **-z new password:** Cambiar contraseña:
sqlcmd -U someuser -P somepassword -z a_new_password
- Z new password and exit:** Cambiar contraseña y salir:
sqlcmd -U someuser -P somepassword -Z a_new_password
- **-S server_name [\instance_name]:** Especifica la instancia de SQL Server a la que hay que conectarse. Si no se especifica ningún equipo servidor, **sqlcmd se conecta a la instancia predeterminada de SQL Server en el equipo local**.

Esta opción es necesaria si se ejecuta sqlcmd desde un equipo remoto conectado a la red.
- **L[c]:** Si no nos acordamos del nombre del servidor al que queremos conectarnos, **-L detecta todos los servidores SQL Server de la red y muestra una lista**. El parámetro c opcional especifica una salida limpia (clean), lo que quiere decir que no aparece el encabezado servers: y se elimina los espacios que preceden a los nombres de los servidores.
- **-d db_name:** Emite una instrucción **USE db_name** cuando se inicia sqlcmd.. Especifica la base de datos inicial. El valor predeterminado es la propiedad de base de datos predeterminada del inicio de sesión. Si la base de datos no existe, se genera un mensaje de error y sqlcmd se cierra.
- **-A dedicated admin connection:** Inicia sesión en SQL Server con una **Conexión de administrador dedicada (DAC)**. Este tipo de conexión se utiliza para solucionar problemas de un servidor.

Sólo funcionará con equipos servidores que admitan DAC. Si DAC no está disponible, sqlcmd genera un mensaje de error y después se cierra.

Opciones de entrada o salida

- **-i input_file[,input_file2...]** :Identifica el archivo que **contiene un lote de instrucciones SQL o procedimientos almacenados**. Esto significa que podemos introducir varias

sentencia SQL en un fichero de texto y, en lugar de introducir la información en la línea de comandos, se puede especificar el archivo de entrada.

- Se pueden especificar varios archivos que se leerán y se procesarán en orden. No hay que utilizar espacios en blanco entre los nombres de archivo.
- sqlcmd comprobará primero si todos los archivos especificados existen. Si uno o más archivos no existen, sqlcmd se cerrará. Las opciones -i y -Q/-q se excluyen mutuamente.

Ejemplos de rutas de acceso:

- i C:\<nombreDeArchivo>
- i \\<Servidor>\<recursoCompartido\$>\<nombreDeArchivo>
- i "C:\Carpeta\<nombreDeArchivo>"

Las rutas de acceso que contienen espacios deben incluirse entre comillas.

- **-o output_file**: Identifica el archivo que recibe la salida de sqlcmd.
 - Si se especifica **-u**, el archivo output_file se almacena en formato **Unicode**.
 - Si el nombre de archivo no es válido, se genera un mensaje de error y sqlcmd se cierra.
 - SQLCMD no admite la escritura simultánea de varios procesos de sqlcmd en el mismo archivo.
 - El archivo se creará si no existe. Se sobrescribirá cualquier archivo con el mismo nombre que pertenezca a una sesión de sqlcmd anterior.

Ejemplos de rutas de acceso:

- o C:\< nombreDeArchivo>
- o \\<Servidor>\<recursoCompatido\$>\<nombreDeArchivo>
- o "C:\Carpeta\<nombreDeArchivo>"

Las rutas de acceso que contienen espacios deben incluirse entre comillas.

Opciones de ejecución de consultas

- **-q "cmdline query "**: **Ejecuta una consulta cuando se inicia sqlcmd**, pero no cierra sqlcmd cuando finaliza la consulta. Se pueden ejecutar varias consultas delimitadas por punto y coma. Utilice las comillas alrededor de la consulta, como se muestra en el siguiente ejemplo.

En el símbolo del sistema, escriba:

sqlcmd	-d	AdventureWorks	-q	"SELECT	FirstName,	LastName	FROM	Person.Contact	WHERE	LastName	LIKE	'Whi%';"			
sqlcmd	-d	AdventureWorks	-q	"SELECT	TOP	5	FirstName	FROM	Person.Contact;	SELECT	TOP	5	LastName	FROM	Person.Contact;"

Importante: No utilice el terminador GO en la consulta.

- **-Q"cmdline query " and exit**: **Ejecuta una consulta cuando se inicia sqlcmd e inmediatamente después cierra sqlcmd**. Se pueden ejecutar varias consultas delimitadas por punto y coma.
- **-e echo input**: este parámetro repite (hace eco) los comandos que se escriban. Si se introduce una consulta, por ejemplo, se repite en la primera línea del conjunto de resultados.

Comandos de sqlcmd

Además de las instrucciones Transact-SQL de sqlcmd, también están disponibles comandos. Hay que tener en cuenta que:

- Los comandos de sqlcmd se **reconocen sólo si aparecen al principio de una línea**.
- Todos los comandos de sqlcmd no distinguen entre mayúsculas y minúsculas.
- Cada comando **debe estar en una línea separada**.

- Un comando **no puede ir seguido de una instrucción de Transact-SQL o de otro comando.**
- **[:] ED:** Inicia el editor de texto. Este editor se puede utilizar para editar el lote actual de Transact-SQL o el último lote ejecutado. Para editar el último lote ejecutado, el comando ED debe escribirse inmediatamente después de que se complete la ejecución del último lote.
- **[:] RESET:** Borra la caché de instrucciones.
- **:List:** Imprime el contenido de la caché de instrucciones.
- **:Error <filename>|STDERR|STDOUT:** Redirige toda la salida de error al archivo especificado por file name, a stderr o a stdout. El comando Error puede aparecer varias veces en una secuencia de comandos. De forma predeterminada, la salida de error se envía a stderr.
 - filename: Crea y abre un archivo que recibirá la salida. Si el archivo ya existe, se truncará en cero bytes. Si el archivo no está disponible a causa de los permisos u otros motivos, la salida no se cambiará y se enviará al último destino especificado o al predeterminado.
 - STDERR: Cambia la salida del error al flujo stderr. Si se ha redirigido, el destino al cual se redirige el flujo recibirá la salida del error.
 - STDOUT: Cambia la salida del error al flujo stdout. Si se ha redirigido, el destino al cual se redirige el flujo recibirá la salida del error.
- **:Out <filename>| STDERR| STDOUT:** Crea y redirige todos los resultados de consulta al archivo especificado por file name, a stderr o a stdout. De forma predeterminada, la salida se envía a stdout.
- **[:] QUIT:** Hace que sqlcmd se cierre.
- **:EXIT :** sale de forma inmediata.
- **GO :GO** marca tanto el final de un lote como la ejecución de cualquier instrucción de Transact-SQL almacenada en caché. Cuando especifique un valor para count, las instrucciones almacenadas en caché se ejecutarán tantas veces como indique count, como un solo lote.
- **:Serverlist:** Enumera los servidores configurados localmente y los nombres de los servidores que difunden en la red.
- **:Connect server_name[instance_name] [-U user_name [-P password]] :** Conecta con una instancia de SQL Server. También cierra la conexión actual.
- **[:] !!< command>:** Ejecuta comandos del sistema operativo. Para ejecutar un comando del sistema operativo, inicie una línea con dos signos de exclamación (!!) seguidos por el comando del sistema operativo. Por ejemplo:
 - :!! Dir
 - :Help: Muestra los comandos de sqlcmd junto con una breve descripción de cada comando.

8. Arrancar y detener el motor de base de datos

Para poder conectarnos e interactuar con el motor de base de datos, hay que asegurarse de que los servicios están instalados en el equipo y que se están ejecutando.

Como mínimo, los dos servicios MSSQLSERVER y SQL Server Agent deben estar **iniciados**.

- El servicio **MSSQLSERVER** representa el **motor de la base de datos** y mientras este servicio no **esté iniciado**, es imposible conectarnos al servidor y trabajar con los datos que contiene.
- El **servicio Agent** toma a su cargo, entre otras tareas, la ejecución y la gestión de todas las tareas planificadas (ejecuta trabajos, supervisa SQL Server, activa alertas y permite la automatización de algunas tareas administrativas). Otra de las cosas a tener en cuenta para permitir las conexiones debe estar activado el Agente de SQL Server.

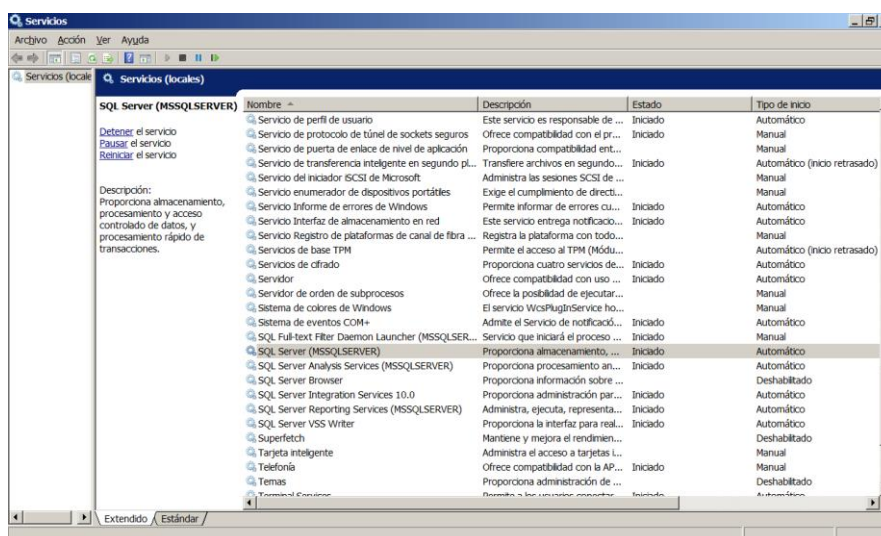
Para gestionar los servicios, se puede hacer utilizando las herramientas que proporciona el sistema operativo Windows o las herramientas propias de SQL Server:

- **Herramientas Administrativas -> Servicios**
- **Utilizar el Administrador de configuración de SQL Server.**
- **Utilizar SQL Server Management Studio.**

8.1 Herramientas Administrativas -> Servicios

Dado que cada componente en una instancia de Base de Datos SQL Server es un servicio en Windows, la forma de comprobar, iniciar o parar un componente será utilizar el cuadro de diálogo **Servicios de Windows**. Para abrir la ventana Servicios, podemos proceder:

- **Inicio -> Panel de Control -> Herramientas Administrativas -> Servicios.**
- **Inicio, Ejecutar, escribimos services.msc y presionamos Enter.**

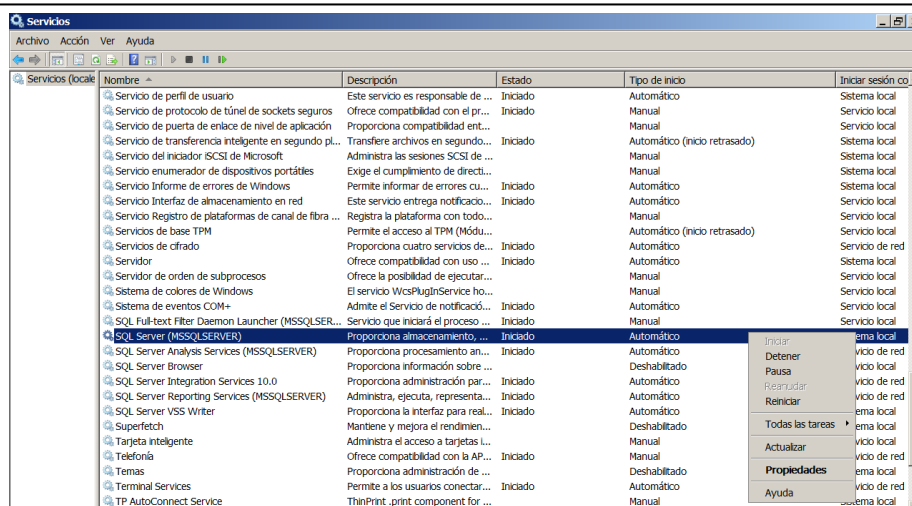


En el cuadro de diálogo Servicios se observan todos los servicios instalados en el equipo junto con su Estado, Descripción Tipo de inicio y otras características, entre los cuales aparece MSSQLSERVER (motor de la base de datos de la instancia predeterminada).

El servicio SQL Server se conoce como MSSQLSERVER si es la instancia predeterminada, o MSSQL\$nombreinstancia en el caso de haber hecho la instalación en una instancia con nombre.

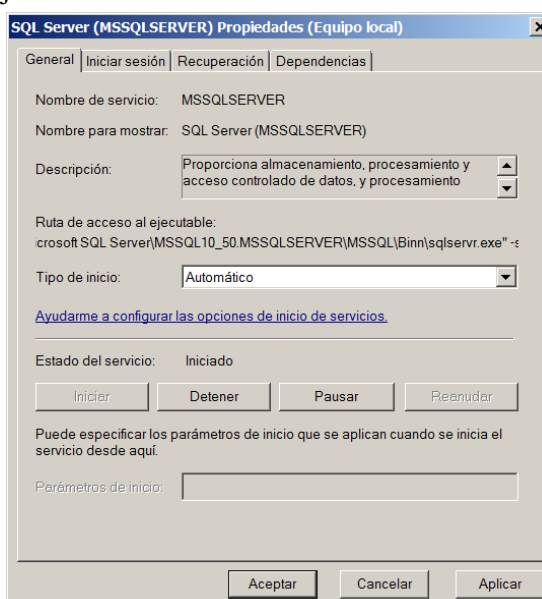
Para configurar la forma en que se inicia un servicio mediante la interfaz de Windows:

- En el panel de detalles, se hace clic con el **botón secundario en el servicio que se desee configurar**.



Tenemos **tres diferentes estados para cada uno de los servicios**:

- **Iniciado:** El servicio está en funcionamiento, los clientes podrán acceder a la información.
 - **Pausado:** El servicio no responde a las peticiones, se encuentra congelado y no parado, ya que, al reanudarlo, continuará en el mismo estado que se encontraba en el momento de pausarlo.
 - **Detenido:** El servicio se para, y deja de ocupar espacio de memoria. Cuando se reinicie, tomará los valores que se indiquen por defecto.
- Para **cambiar el tipo de inicio**, se hace clic en **Propiedades**.
- En la ficha General, en Tipo de inicio, se hace clic en Automático, Manual, Deshabilitado o Automático (inicio retrasado) según el modo de inicio que se desee.
 - **Automático:** Se inician junto con el sistema operativo.
 - **Manual:** Podemos iniciarlo y detenerlo manualmente cuando queramos u otro servicio puede hacerlo automáticamente.
 - **Deshabilitado:** No se puede iniciar manualmente ni otro servicio puede hacerlo.
 - **Automático (Inicio retrasado):** se inicia con el sistema, pero espera a que Windows esté cargado para ejecutarse.



- Para especificar la cuenta de usuario que el servicio puede utilizar para iniciar sesión se hace clic en la ficha **Iniciar sesión** y, a continuación, se realiza una de las siguientes acciones:

- Para que el servicio utilice la cuenta del sistema local (NT AUTHORITY\LocalSystem), se selecciona la opción *Cuenta del sistema local*.

En la mayoría de los casos, se recomienda no modificar la configuración Permitir que el servicio interactúe con el escritorio. Si se permite que el servicio interactúe con el escritorio, cualquier información que el servicio muestre en el escritorio también se mostrará también en el escritorio de un usuario interactivo. Un usuario malintencionado podría tomar el control del servicio o atacarlo desde el escritorio interactivo.

- Para el servicio utilice la cuenta del servicio local, se hace clic en *Esta cuenta* y, a continuación, se escribe NT AUTHORITY\LocalService.
- Para especificar que el servicio utilice la cuenta del servicio de red, se hace clic en *Esta cuenta* y, a continuación, se escribe NT AUTHORITY\NetworkService.

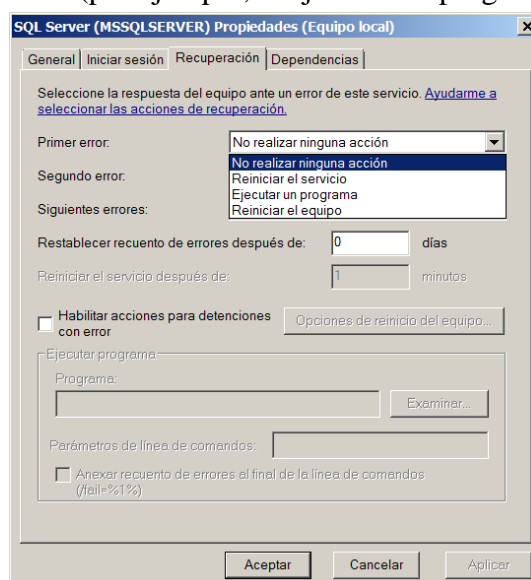
Si selecciona la cuenta del servicio local o la cuenta del servicio de red, la contraseña deberá estar en blanco. Estas cuentas están configuradas con una contraseña nula. Se ignoraré la información de contraseña que se proporcione.

- Para especificar otra cuenta, haga clic en *Esta cuenta*, se hace clic en *Examinar* y, a continuación, se especifica una cuenta de usuario en el cuadro de diálogo *Seleccione usuario*. Cuando se termine, se hace clic en *Aceptar*.

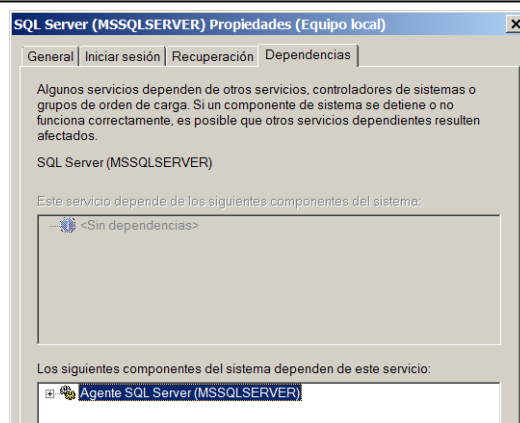
Se escribe la contraseña de la cuenta de usuario en Contraseña y en Confirmar la contraseña y, a continuación, se hace clic en Aceptar.

Es recomendable que las cuentas de usuario que se utilizan para iniciar sesión como un servicio tengan activada la casilla **La contraseña nunca expira** en sus cuadros de diálogo Propiedades y que tengan contraseñas seguras.

- Se puede utilizar la pestaña Recuperación para configurar acciones de recuperación para llevar a cabo si se produce un error en un servicio; por ejemplo, reiniciar el servicio automáticamente o reiniciar el equipo, o en el segundo error del servicio, o cuando se produzcan errores posteriores (por ejemplo, al ejecutar un programa o un script)



- En la pestaña Dependencia pueden verse los servicios que dependen del servicio seleccionado. Si este falla pueden verse afectados estos servicios dependientes.

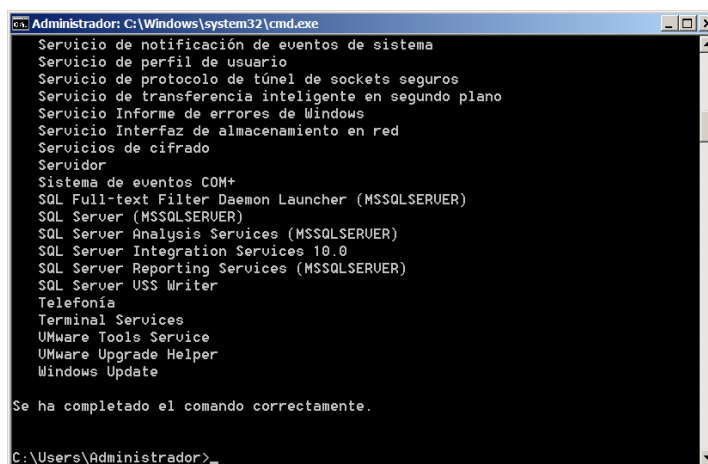


8.2 Desde la línea de comandos:

Se puede iniciar, detener y poner cualquier servicio de SQL Server con el **comando NET** que proporciona el servicio de Windows.

- **net stop** servicio : detener un servicio.
- **net start** servicio: iniciar servicio
- **net start**: lista los servicios que están activos.
- **net pause** servicio: pausar la instancia predeterminada de SQL Server
- **net continue** servicio: reanudar la instancia predeterminada de SQL Server desde el símbolo del sistema

Ej: de ejecución de net start:



Por ejemplo, para iniciar, parar, detener o reanudar la instancia predeterminada de SQL Server:

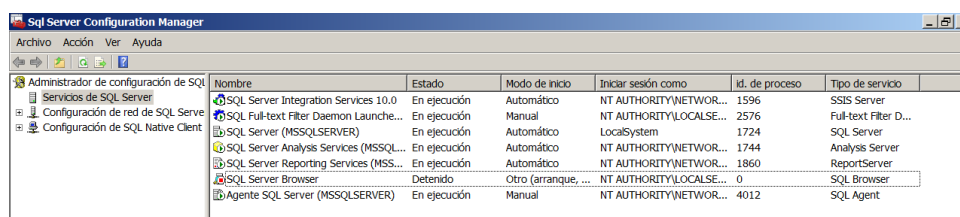
- **net stop mssqlserver**: detener la instancia predeterminada de SQL Server.
- **net start mssqlserver**: iniciar la instancia predeterminada de SQL Server
- **net start**: Para averiguar si el servicio SQL Server está activo.
- **net pause mssqlserver**: pausar la instancia predeterminada de SQL Server
- **net continue mssqlserver**: reanudar la instancia predeterminada de SQL Server desde el símbolo del sistema

8.3 Utilizar el Administrador de configuración de SQL Server.

El Administrador de configuración de SQL Server administra los servicios relacionados con SQL Server.

Se puede utilizar el Administrador de configuración de SQL Server para las siguientes tareas relacionadas con los servicios:

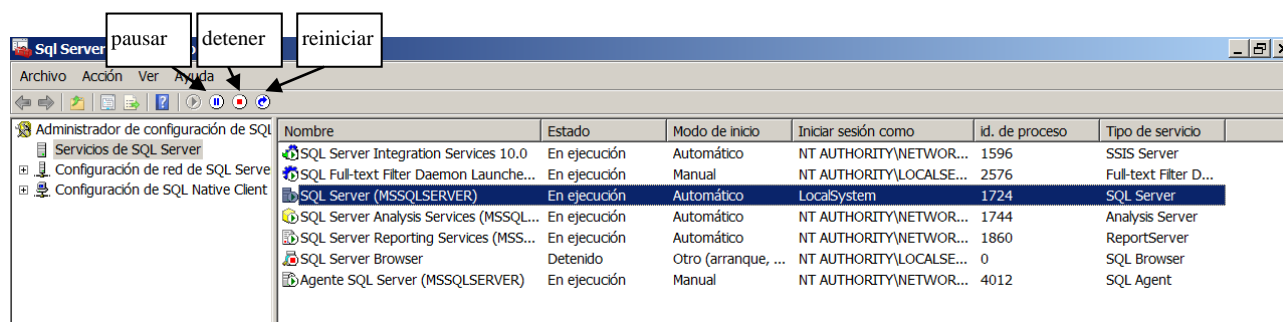
- Iniciar, detener y pausar los servicios
- Configurar los servicios para que se inicien de forma automática o manual, para deshabilitarlos o para cambiar otras opciones de los servicios
- Cambiar las contraseñas de las cuentas utilizadas por los servicios de SQL Server
- Ver las propiedades de los servicios



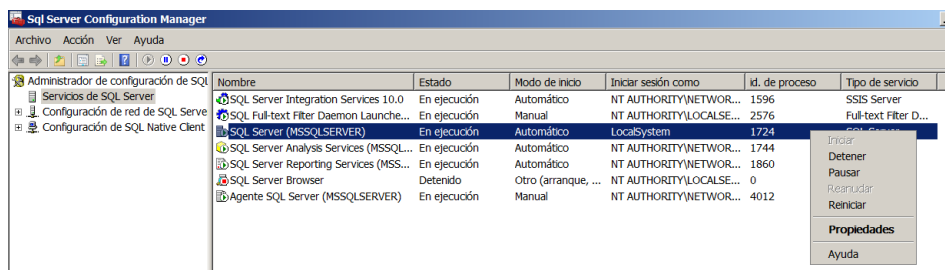
El *estado de los servicios* podemos comprobarlo también gracias al icono que aparece junto al nombre del servicio, un **cuadradillo rojo** indica que el **servicio se encuentra detenido**, mientras que un **triangulillo verde** indica **que el servicio está activo**.

Modificar el estado de los servicios.

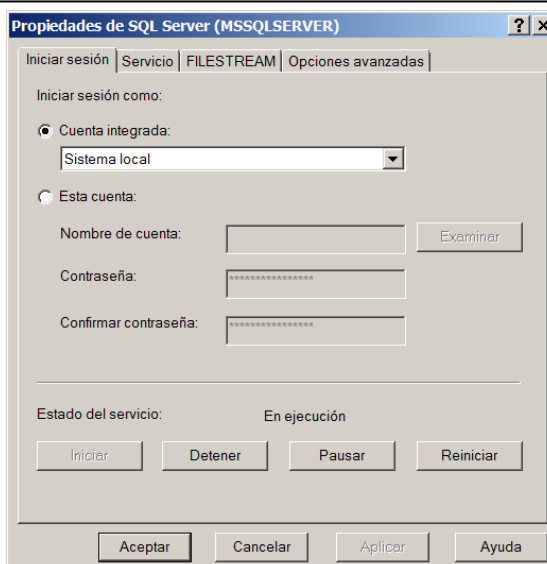
Para variar de un estado en otro, podemos realizarlo con la **barra de herramientas** que encontramos en la parte superior.



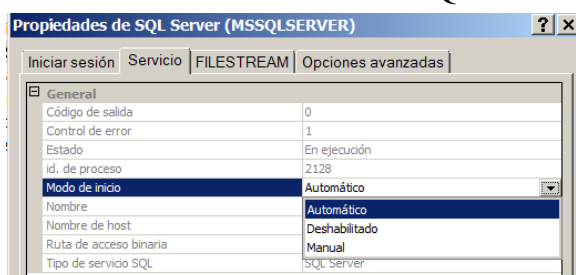
O bien, mediante el menú emergente que se muestra al pulsar con el botón derecho sobre el servicio del cual queremos variar su estado. En este menú tendremos unas opciones u otras en función del estado en el que se encuentre actualmente.



La opción **propiedades** no permite configurar el servicio: Configurar los servicios para que se inicien de forma automática o manual y cambiar las cuentas de inicio del servicio y sus contraseñas para los servicios de SQL Server



Pestaña **Servicio** permite indicar el inicio de los servicios de SQL Server



En el servicio del motor de base de datos, MSSQLSERVER también se puede habilitar la característica de FILESTREAM (si no se hizo durante la instalación) para que pueda trabajar SQL SERVER con archivos binarios de gran tamaño (**BLOB**) almacenados fuera de la base de datos o modificar esta opciones de FILESTREAM.

