



## Índice:

MÓDULO 1	9
INSTALACIÓN Y CONFIGURACIÓN DE SQL SERVER 2008	9
1- Preparando la instalación	10
Ediciones de SQLServer 2008	- 10
Requisitos de Hardware y Software para SQL Server 2008	- 10
Requisitos de procesador, sistema operativo y memoria	- 11
Instancias de SQL Server 2008	- 14
Instancia por defecto	- 14
Instancias con nombre	- 14
Opciones de Licenciamiento para SQL Server 2008	- 14
Consideraciones sobre seguridad para Servicios SQL Server	- 15
Usar una cuenta de usuario de dominio	- 15
Usar la cuenta de servicio local	- 15
Usar la cuenta de servicio de red	- 15
Intercalación (Collation)	- 15
Actualizar a SQL Server 2008	- 16
Utilizando el Asesor de actualizaciones (Upgrade Adviser) para preparar las actu 16 -	alizaciones
Análisis del Asesor de actualizaciones	- 16
Compatibilidad de la base de Datos	- 17
2- Instalación	
Instalación	- 18
Comprobador de Configuración del Sistema (System configuration checker)	- 18
Cómo instalar SQL Server 2008 desde el símbolo del sistema	- 18
Cómo instalar SQL Server 2008 usando el programa de instalación	- 18
Componentes de Instalación	- 21
Actualización de SQL Server	- 23
Instalación Desatendida	- 23
Creación del archivo .ini	- 23
Comenzando una instalación desatendida	- 24
3- Configuración	25
Administrador de Configuración (Configuration Manager)	- 25
Servicios SQL Server	- 25
Administrar Protocolos de Red	- 25
Configuración del Cliente de Red	- 26
SQL Server Management Studio	- 26
Nuevas funciones de las Secuencias de Comandos	- 27
Características del Explorador de objetos	- 27
SQLCMD	- 27
MÓDULO 2	29
RECUPERACIÓN DE DESASTRE	29
1- Planeamiento	30
Planear la estrategia de copias de seguridad y restauración	- 30





	Elegir el tipo de medio para la copia de seguridad Administrar medios de copia de seguridad	- 30 - 30
	Garantizar la disposición para afrontar desastres	- 31
	Revisar y reducir los posibles errores del usuario	- 31
	Modelo de Recuperación	- 32
	Recuperación simple	- 33
	Recuperación completa	- 33
2-	Copias de Seguridad	
	Tipos de Copias de Seguridad	- 34
	Copias de seguridad completas de bases de datos Copia de seguridad en el modelo de recuperación completa	- 34 - 34
	Minimizar el riesgo de pérdida de trabajo	- 34 - 35
	Copias de Seguridad del Registro de Transacciones	- 36
	La cadena de registros	- 36
	Truncamiento del registro con el modelo de recuperación completa y por medio de re	
	de operaciones masivas	- 37
	Cómo funciona el truncamiento del registro	- 37
	Cómo funcionan las copias de seguridad diferenciales	- 38
	Copias de seguridad diferenciales de archivos	- 39
	Copia de seguridad diferencial de grupos de archivos (filegroups)	- 40
	Copias de seguridad diferenciales parciales Crear una Copia de Seguridad	- 40 - 41
	Consideraciones para el operador de las Copias de Seguridad	- 42
	Compresión de Copias de Seguridad	- 42
3-	Restauración de Bases de Datos de Usuario	43
_	Cómo funcionan la restauración de copias de seguridad en SQL Server	- 43
	Conjunto de puestas al día	- 43
	Secuencias de restauración	- 43
	Fases de la restauración	- 43
	Fase de copia de datos	- 44
	Fase de rehacer (puesta al día)	- 44
	Fase de deshacer (revertir) y recuperación Opciones RECOVERY y NORECOVERY	- 45 - 45
	Escenarios de restauración	- 45 - 45
	Modelos de recuperación y operaciones de restauración admitidas	- 46
	Restaurar una copia de seguridad completa de la base de datos	- 47
	Restauración de base de datos completa (modelo de recuperación simple)	- 47
	Restauración por etapas de la base de datos (modelo de recuperación completa)	- 48
	Restauración con conexión de un archivo de sólo lectura (modelo de recuperación sim 49 -	ple)
	Aplicar copias de seguridad del registro de transacciones	- 50
	Registros de transacciones y recuperación	- 50
	Restaurar una Base de Datos usando SQL Server Management Studio	- 50
	Restaurar a un momento dado	- 51
	Restauración online	- 51
	Desconectar una base de datos o un archivo	- 52
M	ÓDULO 3	53
۸,	ANE IO DE SECUDIDAD	- 52



1- Seguridad de Base	- 54 -
Proteger SQL Server	- 54 -
Restringir el Acceso a la Red	- 54 -
Seguridad del Sistema Operativo	- 54 -
ooganaaa aar ciotoma operaare	0.
2- Introducción a la Seguridad de SQL Server	55 -
Entidades de Seguridad y Seguridad de objetos de base de dato	
Entidades de Seguridad (Principals)	- 55 -
Asegurables	- 55 -
Inicio de Sesión sa de SQL Server	- 56 -
La Función public	- 56 -
Funciones (Roles) de Nivel de Servidor (Server Fixed Roles)	- 56 -
Funciones (Roles) en el Nivel de Base de Datos	- 56 -
Jerarquía de Permisos	- 57 -
3- Implementación de Seguridad en SQL Server 2008	
Modo de Autenticación	- 59 -
Configurar el Modo de Autenticación	- 59 -
Conectar con Autenticación Windows	- 60 -
Ventajas de la Autenticación Windows	- 60 -
Conectar con Autenticación de SQL Server	- 60 -
Ventajas de la Autenticación de SQL Server	- 60 -
Desventajas de la Autenticación de SQL Server	- 60 -
Cuándo utilizar el Modo Mixto	- 61 -
Contraseñas	- 61 -
Aplicación de las Directivas de Contraseñas	- 62 -
Contraseñas Seguras	- 62 -
Inicios de Sesión en SQL Server (Login) Modificación de Inicios de Sesión	- 63 - - 65 -
Eliminación de Inicios de Sesión	- 65 - - 65 -
Asignación de una cuenta de inicio de sesión a una función fija o Usuarios de SQL Server	- 65 -
	- 66 -
Usuarios especiales de SQL Server Delegación	- 67 -
Configurando SQL Server para Delegación	- 67 - - 67 -
Permisos	- 67 -
Cadena de Propiedad	- 68 -
Credenciales	- 69 -
Oreachidates	00
4- Cifrado	70 -
Cifrado y Certificados	- 70 -
Jerarquía de Cifrado	- 70 -
Llave Maestra de Servicio (Service Master Key)	- 70 -
Llave Maestra de Base de Datos (Database Master Key)	- 70 -
Certificados	- 71 -
Claves Asimétricas	- 72 -
Claves Simétricas	- 73 -
MÓDULO 4	74 -
MONITOREO DE SQL SERVER	74 -
1- Monitorear la Actividad en Curso	75 -



Determinar la actividad de los usuarios	- 75 -
El Monitor de actividad (SQL Server Management Studio)	- 75 -
Monitor de Actividad de Trabajo	- 76 -
Para Ver la Actividad de Trabajo (SQL Server Management Studio)	- 76 -
Funciones y Vistas de Administración Dinámica	- 78 -
Vistas de Catálogo (Transact-SQL)	- 78 -
2- Monitor de Sistemas de Windows	- 79 -
Objetos de SQL Server	- 79 -
Consideraciones de Monitoreo de SQL Server	- 80 -
3- Analizador de SQL Server (SQL Server Profiler)	_ 21 _
Introducción	- 81 -
Terminología del Analizador de SQL Server	- 81 -
Usar el Analizador de SQL Server	- 82 -
Plantillas del Analizador de SQL Server	- 83 -
Guardar Trazas	- 83 -
Analizar el Rendimiento con Plantillas	- 84 -
4- Desencadenadores DDL (DDL Triggers)	85 -
Definición	- 85 -
Eventos DLL	- 85 -
Función EVENTDATA	- 86 -
5- Notificaciones de Eventos	88 -
Introducción	- 88 -
Conceptos Básicos de las Notificaciones de Eventos	- 88 -
Descripción de Notificaciones de Eventos frente a Desencadenadores	- 89 -
Notificaciones de Eventos frente a Traza de SQL	- 89 -
6- Auditoría SQL Server	91 -
SQL Server Audit	- 91 -
Crear un objeto de Auditoría	- 91 -
Crear una especificación de Auditoría	- 92 -
Habilitación de Auditorías	- 93 -
Leer los eventos	- 93 -
MÓDULO 5	- 95 -
AUTOMATIZANDO TAREAS ADMINISTRATIVAS	95 -
1- Automatizar las tareas administrativas (Agente SQL Server)	- 96 -
Herramientas para Automatizar la Administración	- 96 -
Planes de Mantenimiento	- 96 -
Iniciar el Asistente para Planes de Mantenimiento (SQL Server Management Stud	
Editar y crear Planes de Mantenimiento Manualmente	- 98 -
Agente SQL Server (SQL Server Agent)	- 99 -
Componentes	- 100 -
2- Configurar el Agente SQL Server	102 -
Establecer los Permisos Necesarios	- 102 -
Correo del Agente SQL Server	- 102 -
Correo Electrónico de Base de Datos (Database Mail)	- 103 -



	SQL Mail	- 104 -
3-	Trabajos y Operadores	
	Trabajos	- 105 -
	Organizar trabajos	- 105 -
	Propiedad de trabajos	- 105 -
	Crear un trabajo	- 106 -
	Crear pasos de trabajo	- 106 -
	Programas Ejecutables y Comandos del Sistema Operativo como Pasos de Trabajo	- 107 -
	Controlar varios Pasos del Trabajo	- 108 -
	Programar la Ejecución de un Trabajo	- 109 -
	Operadores Información de Contacto	- 110 - - 110 -
		- 110 -
	Requisitos para Notificar a un Operador Crear un Operador	- 111 -
	Supervisar la Actividad de Trabajo	- 111 -
	Sesiones del Agente SQL Server	- 112 -
	Monitor de Actividad de Trabajo	- 112 -
	Worldon de Addividad de Trabajo	112
4-	Alertas	
	Alertas	- 113 -
	Eventos  Calabaira de Randinia	- 113 -
	Seleccionar una Condición de Rendimiento	- 114 - - 115 -
	Seleccionar un Evento de WMI	- 116 -
	Alertar a Operadores	- 110 -
5-	Manejo de Servidores Múltiples	
	Administración multiservidor	- 117 -
	Consideraciones para Entornos Multiservidor	- 117 -
	Procesar Trabajos Multiservidor	- 118 -
6-	Seguridad del Agente SQL Server	119 -
	Conceder Acceso al Agente SQL Server	- 119 -
	Permisos de las Funciones Fijas de Base de Datos del Agente SQL Server	- 119 -
	Permisos de SQLAgentUserRole	- 119 -
	Permisos de SQLAgentReaderRole	- 120 -
	Permisos de SQLAgentOperatorRole	- 120 -
	Asignar a los Usuarios varias Funciones	- 120 -
	Cuentas de proxy del Agente SQL Server	- 120 -
	Creación de Proxies	- 120 -
M	ÓDULO 6	122 -
R	EPLICACIÓN	122 -
1.	Introducción a Replicación	<b>.</b> 123 -
'	Definición	- 123 - - 123 -
	Tipos de Replicación	- 123 -
	Replicación Transaccional	- 123 -
	Replicación de Mezcla	- 124 -
	Replicación de Instantáneas	- 125 -
	Modelo de Publicación de Replicación	- 126 -
	Publicador	- 127 -



Distribuidor	- 127 -
Suscriptores	- 127 -
Artículo	- 127 -
Publicación	- 128 -
Suscripción	- 128 -
2- Implementar Replicación	129 -
Implementación	- 129 -
Configurar la Replicación y Publicar Datos	- 129 -
Filtrar Datos Publicados	- 129 -
Procedimientos Almacenados de Réplica	- 131 -
Crear e Inicializar Suscripciones	- 132 -
Suscriptores que no son de SQL Server	- 133 -
Sincronizar Datos	- 133 -
Asistentes para Réplica	- 133 -
Asistente para Nueva Publicación	- 134 -
Asistente para Nuevas Suscripciones	- 135 -
Asistente para Configurar la Distribución	- 137 -
Asistente para la sincronización Web	- 138 -
3- Monitor de Réplica	- 139 -
Supervisión de la Réplica con el Monitor de Réplica	- 139 -
La interfaz del Monitor de réplica	- 139 -
Ver información de toda la topología	- 139 -
Ver información y realizar tareas relacionadas con publicadores	- 140 -
Ver información y realizar tareas relacionadas con suscripciones	- 140 -
4- Configurar Replicación en algunos Escenarios Comunes	
Replicar datos en un entorno de servidor a servidor	- 142 -
Replicar datos entre un servidor y los clientes	- 142 -
Replicación Transaccional Punto a Punto (Peer to Peer)	- 143 -
Implementar la Replicación por Internet	- 145 -
MÓDULO 7	- 146 -
MANTENIMIENTO DE ALTA DISPONIBILIDAD	146 -
1- Introducción a Alta Disponibilidad	
Factores que afectan la Disponibilidad:	- 147 -
Configurar una Alta Disponibilidad	- 147 -
Seleccionar una Solución de Alta Disponibilidad	- 147 -
2- Implementación de Server Clustering	149 -
Que es un Clúster?	- 149 -
Consideraciones antes de Instalar un Clúster de Conmutación por Error	- 149 -
Lista de comprobación previa a la instalación	- 150 -
Comprobar la Configuración del Sistema Operativo	- 150 -
Configurar el Servicio de Cluster Server de Microsoft	- 150 -
Instalar el Coordinador de Transacciones Distribuidas de Microsoft	- 151 -
Otras Consideraciones de Software	- 151 -
Consideraciones Relativas a la Red	- 152 -
Instalar un Clúster de Conmutación por Error	- 153 -
Elementos de una instancia de clúster de conmutación por error	- 153 -



	١	ı	

Asignar nombre a una Instancia de Clúster de Conmutación por Error	- 153 -
3- Implementando Reflejo de Base de Datos	154 -
Reflejo de la Base de Datos	- 154 -
Ventajas de la creación de reflejo de la base de datos	- 154 -
Funcionamiento de la creación de reflejo de la base de datos	- 154 -
Modos de funcionamiento	- 155 -
Seguridad de las Transacciones y Modos de Funcionamiento	- 156 -
Extremo de Creación de Reflejo de Base de Datos	- 157 -
Cómo Crear un Extremo de Reflejo para la Autenticación de Windows	- 157 -
Roles de servidor	- 157 -
Para Crear un Extremo de Reflejo utilizando la Autenticación de Windows	- 157 -
Sesiones de Creación de Reflejo de la Base de Datos	- 159 -
Sesiones Simultáneas	- 160 -
Requisitos Previos para una Sesión de Creación de Reflejo de la Base de Datos	- 161 -
Conmutación de Funciones durante una Sesión de Creación de Reflejo de la Base 161 -	de Datos-
Conmutación por Error Manual	- 162 -
Conmutación por Error Automática	- 163 -
Forzar Servicio (con posible pérdida de datos)	- 163 -
4- Trasvase de Registros (Log Shipping)	164 -
Definición	- 164 -
Operaciones del Trasvase de Registros	- 164 -
Configurar el Trasvase de Registros	- 164 -
Cambiar las funciones entre el servidor primario y secundario	- 168 -

# Módulo 1

# Instalación y Configuración de SQL Server 2008



#### 1- Preparando la instalación

#### Ediciones de SQLServer 2008

- Enterprise (x86, x64 e IA64): SQL Server Enterprise es una completa plataforma de datos que proporciona escalabilidad empresarial, rendimiento, alta disponibilidad y capacidades avanzadas de inteligencia empresarial para ejecutar aplicaciones seguras y esenciales para la empresa.
- Standard (x86 y x64): SQL Server Standard es una completa plataforma de datos que proporciona facilidad de uso y capacidad de administración. Esto incluye capacidades integradas de inteligencia empresarial para ejecutar aplicaciones de departamentos. SQL Server Standard for Small Business contiene todos los componentes técnicos y capacidades técnicas de SQL Server Standard y su uso está autorizado en un entorno de pequeña empresa con 75 equipos o menos.
- SQL Server 2008 Developer (x86, x64 e IA64): SQL Server 2008 Developer permite a los desarrolladores crear cualquier tipo de aplicación basada en SQL Server. Incluye toda la funcionalidad de SQL Server 2008 Enterprise, pero su uso está autorizado como sistema de desarrollo y pruebas, no como servidor de producción. SQL Server 2008 Developer es una opción ideal para las personas que crean y prueban aplicaciones. Puede actualizar SQL Server 2008 Developer para utilizarlo en producción.
- Workgroup (x86 y x64): SQL Server Workgroup es ideal para ejecutar bases de datos ubicadas en sucursales y proporciona una administración de datos confiable y una plataforma de informes que incluye capacidades de sincronización y de administración segura y remota.
- Web (x86, x64): SQL Server 2008 Web es una opción con un costo total de propiedad bajo para los hosts de web y los sitios web que proporciona capacidades de administración y escalabilidad para propiedades web, tanto de pequeña como de gran escala.
- SQL Server Express (x86 y x64), SQL Server Express with Tools (x86 y x64), SQL Server Express con Advanced Services (x86 y x64): La plataforma de bases de datos de SQL Server Express se basa en SQL Server 2008. Es también la sustitución de Microsoft Desktop Engine (MSDE). Gracias a su integración con Visual Studio, SQL Server Express facilita el desarrollo de aplicaciones controladas por datos que tienen una gran capacidad, ofrecen un almacenamiento seguro y se implementan con rapidez. SQL Server Express es gratuito y los ISV pueden redistribuirlo (según su contrato). SQL Server Express es ideal para conocer y crear pequeñas aplicaciones de servidor y de escritorio. Esta edición es la mejor opción para los fabricantes de software independientes, los desarrolladores no profesionales y los aficionados que crean aplicaciones cliente. Si necesita características de base de datos más avanzadas, SQL Server Express se puede actualizar sin problemas a versiones más sofisticadas de SQL Server.
- Compact 3.5 SP1 (x86), Compact 3.1 (x86): SQL Server Compact 3.5 es una base de datos gratuita e incrustada, ideal para crear aplicaciones independientes que se conectan ocasionalmente para dispositivos móviles, escritorios y clientes web en todas las plataformas de Windows.

#### Requisitos de Hardware y Software para SQL Server 2008

Cuando se planifica la instalación de SQL Server 2008 usted debe asegurarse de que la computadora en la que realizará la instalación cumple con los requerimientos mínimos y es adecuada para sus necesidades actuales y futuras. El hecho de no alcanzar los requerimientos mínimos puede ocasionar fallas en la instalación de algunos o de todos los componentes.





Componente	Degwieite
Componente  Marco de Trabajo	Requisito  El programa de instalación de SQL Server instala los siguientes componentes de software
	requeridos por el producto:
	.NET Framework 3.5 SP1
	SQL Server Native Client
	Archivos auxiliares para la instalación de SQL Server
Software	El programa de instalación de SQL Server requiere Microsoft Windows Installer 4.5 o una versión posterior
	Una vez instalados los componentes requeridos, el programa de instalación de SQL Server
	comprobará que el equipo en el que se ha instalado SQL Server 2008 también cumple los
	demás requisitos para su correcta instalación.
Software de red	Los requisitos de software de red para las versiones de 64 bits de SQL Server 2008 son
Contware de red	los mismos que para las versiones de 32 bits.
	Los sistemas operativos compatibles tienen el software de red integrado. Las instancias
	predeterminadas y con nombre independientes admiten los siguientes protocolos de red:
	Memoria compartida
	Canalizaciones con nombre
	TCP/IP
	• VIA
	Nota: La memoria compartida y VIA no se admiten en clústeres de conmutación por error.
Virtualización	SQL Server 2008 es compatible con entornos de máquina virtual que se ejecutan en la
	función Hyper-V de las ediciones Standard, Enterprise y Datacenter de Windows Server
	2008 R2 y Windows Server 2008. La máquina virtual debe ejecutarse en un sistema operativo compatible con la edición de SQL Server 2008 concreta que se cita más
	adelante en este tema.
	Además de los recursos requeridos por la partición primaria, a cada máquina virtual
	(partición secundaria) se debe proporcionar suficientes recursos de procesador, memoria y
	recursos de disco para su instancia de SQL Server 2008. Los requisitos se enumeran más
	adelante en este tema.3
	Dentro de la función Hyper-V de Windows Server 2008, se puede asignar un máximo de
	cuatro procesadores virtuales a máquinas virtuales que ejecuten las ediciones de 32 o 64
	bits de Windows Server 2008. Se pueden asignar como máximo 2 procesadores virtuales a
	equipos virtuales que ejecuten ediciones de 32 bits de Windows Server 2003. Para
	equipos virtuales que alojan otros sistemas operativos, se puede asignar como máximo un
0.6	procesador virtual a equipos virtuales.
Software de Internet	Para todas las instalaciones de SQL Server 2008 se requiere Microsoft Internet Explorer 6
	SP 1 o una versión posterior. Se requiere Internet Explorer 6 Service Pack 1 o una versión posterior para Microsoft Management Console (MMC), SQL Server Management Studio,
	Business Intelligence Development Studio, el componente Diseñador de informes de
	Reporting Services y la Ayuda HTML.
Disco Duro	Las necesidades de espacio en disco variarán con los componentes de SQL Server 2008
2.555 5410	que instale.
Pantalla	Las herramientas gráficas de SQL Server 2008 requieren VGA o una resolución mayor:
12	resolución mínima de 1.024 x 768 píxeles.
	the state of the s

#### Requisitos de procesador, sistema operativo y memoria

Estos requerimientos dependen de la versión que se desea instalar. A continuación se detallan los requerimientos de algunas versiones:

#### SQL Server 2008 Enterprise (64 bits) x64

Componente	Requisito		
Procesador	Tipo de procesador:  • Mínimo: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T  Velocidad de procesador:  • Mínimo: 1,4 GHz  • Recomendado: 2,0 GHz o más		
Sistema Operativo	Windows Server 2003 SP2 de 64 bits x64 Standard Windows Server 2003 SP2 de 64 bits x64 Datacenter Windows Server 2003 SP2 de 64 bits x64 Enterprise		



	Windows Server 2008 de 64 bits x64 Standard Windows Server 2008 de 64 bits x64 Standard sin Hyper-V Windows Server 2008 de 64 bits x64 Datacenter Windows Server 2008 de 64 bits x64 Datacenter sin Hyper-V Windows Server 2008 de 64 bits x64 Enterprise Windows Server 2008 de 64 bits x64 Enterprise sin Hyper-V Windows Server 2008 R2 de 64 bits x64 Web Windows Server 2008 R2 de 64 bits x64 Standard Windows Server 2008 R2 de 64 bits x64 Enterprise Windows Server 2008 R2 de 64 bits x64 Enterprise Windows Server 2008 R2 de 64 bits x64 Datacenter
Memoria	RAM:
	Mínimo: 512 MB
	Recomendado: 2,048 GB o más
	Máximo: máximo del sistema operativo

#### SQL Server 2008 Standard (64 bits) x64

Componente	Requisito
Procesador	Tipo de procesador:  • Mínimo: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T  Velocidad de procesador:  • Mínimo: 1,4 GHz  • Recomendado: 2,0 GHz o más
Sistema Operativo	Windows XP Professional x64 Windows Server 2003 SP2 de 64 bits x64 Standard Windows Server 2003 SP2 de 64 bits x64 Datacenter Windows Server 2003 SP2 de 64 bits x64 Enterprise Windows Vista Ultimate x64 Windows Vista Business x64 Windows Vista Business x64 Windows Server 2008 x64 Web Windows Server 2008 x64 Standard, Windows Server 2008 x64 Standard sin Hyper-V Windows Server 2008 x64 Datacenter, Windows Server 2008 x64 Datacenter sin Hyper-V Windows Server 2008 x64 Enterprise, Windows Server 2008 x64 Enterprise sin Hyper-V Windows Small Business Server 2008 Windows Server 2008 para Windows Essential Server Solutions Windows Server 2008 sin Hyper-V para Windows Essential Server Solutions Windows 7 de 64 bits x64 Ultimate Windows 7 de 64 bits x64 Professional Windows 2008 R2 de 64 bits x64 Web Windows 2008 R2 foundation Windows 2008 R2 de 64 bits x64 Standard Windows 2008 R2 de 64 bits x64 Enterprise Windows 2008 R2 de 64 bits x64 Datacenter
Memoria	RAM:
	Mínimo: 512 MB     Recommodado: 2.048 CB a más
	Recomendado: 2,048 GB o más     Máximo: máximo del sistema operativo
	iviaximo. maximo dei sistema operativo

#### SQL Server 2008 Enterprise (32 bits)

Componente	Requisito				
Procesador	Tipo de procesador:				
	Procesador compatible con Pentium III o superior				
	Velocidad de procesador:				
	Mínimo: 1,0 GHz				
	Recomendado: 2,0 GHz o más				
Sistema Operativo	Windows Server 2003 SP2 Small Business Server R2 Standard				
	Windows Server 2003 SP2 Small Business Server R2 Premium				



	Windows Server 2003 SP2 Standard			
	Windows Server 2003 SP2 Enterprise			
	Windows Server 2003 SP2 Datacenter			
	Windows Server 2003 Small Business Server SP2 Standard			
	Windows Server 2003 Small Business Server SP2 Premium			
	Windows Server 2003 SP2 de 64 bits x64 Standard			
	Windows Server 2003 SP2 de 64 bits x64 Datacenter			
	Windows Server 2003 SP2 de 64 bits x64 Enterprise			
	Windows Server 2008 Standard (con y sin Hyper-V)			
	Windows Server 2008 Web			
	Windows Server 2008 Datacenter			
	Windows Server 2008 Datacenter sin Hyper-V			
	Windows Server 2008 Enterprise			
	Windows Server 2008 Enterprise sin Hyper-V			
	Windows Server 2008 x64 Standard			
	Windows Server 2008 x64 Standard sin Hyper-V			
	Windows Server 2008 x64 Datacenter			
	Windows Server 2008 x64 Datacenter sin Hyper-V			
	Windows Server 2008 x64 Enterprise			
	Windows Server 2008 x64 Enterprise sin Hyper-V			
	Windows Server 2008 R2 de 64 bits x64 Web			
	Windows Server 2008 R2 de 64 bits x64 Standard			
	Windows Server 2008 R2 de 64 bits x64 Enterprise			
	Windows Server 2008 R2 de 64 bits x64 Datacenter			
Memoria	RAM:			
	Mínimo: 512 MB			
	Recomendado: 2,048 GB o más			
	Máximo: máximo del sistema operativo			

#### SQL Server 2008 Standard (32 bits)

Componente	Requisito
Procesador	Tipo de procesador:
	Procesador compatible con Pentium III o superior
	Velocidad de procesador:
	Mínimo: 1,0 GHz
	Recomendado: 2,0 GHz o más
Sistema Operativo	Windows XP Professional SP2
	Windows XP SP2 Tablet
	Windows XP x64 Professional
	Windows XP SP2 Media Center 2002
	Windows XP SP2 Media Center 2004
	Windows XP Media Center 2008
	Windows XP Professional Reduced Media
	Windows Server 2003 SP2 Small Business Server R2 Standard
	Windows Server 2003 SP2 Small Business Server R2 Premium
	Windows Server 2003 SP2 Standard
	Windows Server 2003 SP2 Enterprise
	Windows Server 2003 SP2 Datacenter
	Windows Server 2003 SP2 Small Business Server Standard
	Windows Server 2003 SP2 Small Business Server Premium
	Windows Server 2003 SP2 de 64 bits x64 Standard
	Windows Server 2003 SP2 de 64 bits x64 Datacenter
	Windows Server 2003 SP2 de 64 bits x64 Enterprise
	Windows Vista Ultimate
	Windows Vista Enterprise
	Windows Vista Business
	Windows Vista Ultimate x64
	Windows Vista Enterprise x64
	Windows Vista Business x64
	Windows Server 2008 Web
	Windows Server 2008 Standard Server
	Windows Server 2008 Standard Server sin Hyper-V
	Windows Server 2008 Datacenter
	Windows Server 2008 Datacenter sin Hyper-V
	Windows Server 2008 Enterprise
	Windows Server 2008 Enterprise sin Hyper-V



	Windows Server 2008 x64 Standard					
	Windows Server 2008 x64 Standard sin Hyper-V					
	Windows Server 2008 x64 Datacenter					
	Windows Server 2008 x64 Datacenter sin Hyper-V					
	Windows Server 2008 x64 Enterprise					
	Windows Server 2008 x64 Enterprise sin Hyper-V					
	Windows Small Business Server 2008					
	Windows Server 2008 para Windows Essential Server Solutions					
	Windows Server 2008 sin Hyper-V para Windows Essential Server Solutions					
	Windows 7 Ultimate					
	Windows 7 Enterprise					
	Windows 7 Professional					
	Windows 7 de 64 bits x64 Ultimate					
	Windows 7 de 64 bits x64 Enterprise					
	Windows 7 de 64 bits x64 Professional					
	Windows Server 2008 R2 de 64 bits x64 Web					
	Windows Server 2008 R2 Foundation					
	Windows Server 2008 R2 de 64 bits x64 Standard					
	Windows Server 2008 R2 de 64 bits x64 Enterprise					
	Windows Server 2008 R2 de 64 bits x64 Datacenter					
Memoria	RAM:					
	Mínimo: 512 MB					
	Recomendado: 2,048 GB o más					
	Máximo: máximo del sistema operativo					

#### Instancias de SQL Server 2008

Una instalación de SQL Server 2008 consiste de una o más instancias. Una instancia del motor de base de datos (database engine) de SQL Server, sea por defecto o con nombre, tiene sus propio conjunto de programas de instancia específicos y archivos de datos, así como un conjunto de archivos comunes compartidos con todas las instancias de la computadora.

Las instancias de otros componentes como Analysis Services o Reporting Service también tienen su propio conjunto de programas y archivos de datos.

Cada instancia opera independientemente de cualquier otra instancia de la misma computadora, y las aplicaciones pueden conectarse a cualquiera de las instancias.

#### Instancia por defecto

Esta instancia es identificada por el nombre de red de la computadora en la cual se está ejecutando. La instancia por defecto del servicio de SQL Server es MSSQLSERVER.

#### Instancias con nombre

Estas instancias son identificadas por el nombre de red de la computadora más un nombre de instancia con el formato: **nombrecomputadora/nombreinstancia**, como por ejemplo

MiPC\SQLINSTANCE2 para una instancia llamada SQLINSTANCE2 en una computadora llamada MiPC.

Una nueva instancia debe comenzar con una letra o guión bajo (\_), y puede contener números, letras u otros caracteres.

Cada instancia con nombre está formada por un conjunto diferente de servicios y puede tener diferentes asignaciones para intercalado, seguridad y otras opciones.

#### Opciones de Licenciamiento para SQL Server 2008

Microsoft SQL Server 2008 está disponible en base a tres modelos de licencias:

Licencia de servidor más una licencia de acceso de cliente (CAL) por dispositivo.
 Requiere una licencia para el equipo que ejecuta el producto servidor de Microsoft, y una CAL para cada dispositivo cliente.





- Licencia de servidor más una licencia de acceso de cliente (CAL) por usuario.
   Requiere una licencia para el equipo que ejecuta el producto servidor de Microsoft, y una CAL para cada usuario.
- Licencia por procesador. Requiere una única licencia por cada CPU en el entorno de sistema operativo que ejecuta SQL Server. Esta licencia incluye un acceso ilimitado de dispositivos cliente.

#### Consideraciones sobre seguridad para Servicios SQL Server

Para que SQL Server y el Agente SQL Server se ejecuten como servicios de Windows, SQL Server y el Agente SQL Server deben estar asignados a una cuenta de usuario de Windows. Las cuentas pueden ser: de usuario, de dominio o de sistema. Se puede asignar la misma cuenta de Windows a todos los servicios de SQL Server o bien configurar una cuenta para cada servicio.

A continuación se describen recomendaciones en la utilización de los diferentes tipos de cuenta:

#### Usar una cuenta de usuario de dominio

Es posible que sea preferible una cuenta de usuario de dominio cuando el servicio debe interactuar con los servicios de red. Muchas actividades de servidor a servidor sólo se pueden realizar con una cuenta de usuario de dominio, por ejemplo:

- Llamadas a procedimiento remoto.
- Réplica.
- Copias de seguridad en unidades de red.
- Combinaciones heterogéneas en las que intervienen orígenes de datos remotos.
- Características de correo del Agente SQL Server y SQL Mail. Esta restricción se aplica si utiliza Microsoft Exchange. La mayoría de los otros sistemas de correo también requieren que los clientes (como el servicio SQL Server y el servicio del Agente SQL Server) se ejecuten con cuentas con acceso a la red.

#### Usar la cuenta de servicio local

La cuenta de servicio local es una cuenta integrada especial parecida a una cuenta de usuario autenticado. La cuenta de servicio local tiene el mismo nivel de acceso a los recursos y objetos que los miembros del grupo Usuarios. Este acceso limitado ayuda a proteger el sistema si procesos o servicios individuales se ven afectados. Los servicios que se ejecutan como cuenta de servicio local tienen acceso a los recursos de red como una sesión nula sin credenciales.

#### Usar la cuenta de servicio de red

La cuenta de servicio de red es una cuenta integrada especial parecida a una cuenta de usuario autenticado. La cuenta de servicio de red tiene el mismo nivel de acceso a los recursos y objetos que los miembros del grupo Usuarios. Los servicios que se ejecutan como la cuenta de servicio de red tienen acceso a los recursos de red a través de las credenciales de la cuenta de equipo.

#### Intercalación (Collation)

La configuración de intercalación de SQL Server 2008 depende del tipo de instalación. Normalmente, se debe elegir una intercalación de SQL Server que admita la configuración regional del sistema de Windows que más se suela usar en la organización. En la mayoría de los casos, un equipo ejecuta la configuración regional del sistema de Windows que coincide con los requisitos de idioma del usuario. Por consiguiente, el programa de instalación de SQL





Server detecta automáticamente la configuración regional del sistema de Windows y selecciona la intercalación de SQL Server adecuada. Para mantener la compatibilidad con las versiones anteriores, la intercalación del idioma inglés (EE.UU.) predeterminada es SQL Latin1 General\*.

Las intercalaciones de SQL Server 2008 controlan lo siguiente:

- La página de códigos que se utiliza para almacenar datos no Unicode en SQL Server.
- Las reglas que rigen la forma en que SQL Server ordena y compara los caracteres que se almacenan en tipos de datos no Unicode. El programa de instalación de SQL Server detectará la intercalación de Windows en el equipo donde se instala SQL Server. A continuación, seleccionará automáticamente la intercalación para la instancia de SQL Server. Los órdenes y las comparaciones pueden ser diferentes para caracteres Unicode y no Unicode.

Muchas de las actividades de servidor a servidor pueden causar un error o dar resultados incoherentes si la configuración de intercalación no es coherente en los distintos servidores. Para evitarlo, seleccione una configuración regional de Windows o una intercalación de SQL Server (SQL\_\*) que coincida con la configuración de intercalación de las otras instancias de SQL Server.

SQL Server 2008 permite configurar las intercalaciones en los niveles siguientes:

- Servidor
- Base de datos
- Columna
- Expresión

#### Actualizar a SQL Server 2008

Puede actualizar directamente las instancias de SQL Server anteriores. Puede realizar la mayoría de las operaciones de actualización mediante el programa de instalación; no obstante, algunos componentes admiten o necesitan la migración de aplicaciones o soluciones una vez ejecutado el programa de instalación.

## Utilizando el Asesor de actualizaciones (Upgrade Adviser) para preparar las actualizaciones

El Asesor de actualizaciones de SQL Server ayuda a preparar las actualizaciones de SQL Server 2008. Esta herramienta analiza los componentes instalados de las versiones anteriores de SQL Server y genera un informe que identifica los problemas que han de solucionarse antes o después de la actualización.

Al ejecutar el Asesor de actualizaciones, se mostrará la página de inicio del Asesor de actualizaciones. Desde la página Inicio, podrá ejecutar las siguientes herramientas:

- Asistente para análisis del Asesor de actualizaciones
- Visor de informes del Asesor de actualizaciones
- Avuda del Asesor de actualizaciones

La primera vez que utilice el Asesor de actualizaciones, ejecute el Asistente para análisis del Asesor de actualizaciones para analizar los componentes de SQL Server. Una vez que el asistente haya finalizado el análisis, podrá ver los informes resultantes en el Visor de informes del Asesor de actualizaciones. Los informes incluyen vínculos a información de la Ayuda del Asesor de actualizaciones que le ayudará a solucionar los problemas conocidos o a paliar su efecto.

#### Análisis del Asesor de actualizaciones

El Asesor de actualizaciones analiza los siguientes componentes de SQL Server:

- Database Engine (Motor de base de datos)
- Analysis Services





- Notification Services
- Reporting Services
- Integration Services, anteriormente conocido como Servicios de transformación de datos (DTS)

Un analizador dedicado se ejecuta en el contexto del Asesor de actualizaciones para cada componente de SQL Server. El análisis examina los objetos accesibles para los analizadores individuales, como secuencias de comandos, procedimientos almacenados, desencadenadores y archivos de traza. El Asesor de actualizaciones no puede analizar aplicaciones de escritorio ni procedimientos almacenados cifrados.

La salida de cada analizador es un informe XML sobre dicho componente. Podrá ver el informe XML mediante el visor de informes del Asesor de actualizaciones.

#### Compatibilidad de la base de Datos

Algunos comportamientos en la base de datos SQL Server 2008 son diferentes a los utilizados en versiones anteriores de SQL Server.

Si existen aplicaciones en uso que utilicen una versión anterior de SQL Server es necesario que usted asigne el nivel de compatibilidad de SQL Server 2008 para su correcto funcionamiento.

Para signar un nivel de compatibilidad de una base de datos SQL Server debe hacerlo utilizando el comando ALTER DATABASE con la cláusula **SET COMPATIBILITY\_LEVEL** .

#### d

#### 2- Instalación

#### Instalación

El Asistente para la instalación de SQL Server se basa en Windows Installer. Proporciona un árbol de características único para instalar todos los componentes de SQL Server:

- SQL Server Database Engine (Motor de base de datos de SQL Server)
- Analysis Services
- Reporting Services
- Integration Services
- Replicación
- Herramientas de administración
- Documentación

**Nota**: SQL Server está disponible en ediciones de 32 bits y de 64 bits. Las ediciones de 64 bits y de 32 bits de SQL Server se instalan a través del Asistente para la instalación o en un símbolo del sistema.

De forma predeterminada, las bases de datos y el código de ejemplo no se instalan como parte del programa de instalación de SQL Server. Para instalar las bases de datos y el código de ejemplo para las ediciones de SQL Server 2008 que no son Express, vea el sitio web de CodePlex.

Independientemente de si utiliza el Asistente para la instalación de SQL Server o el símbolo del sistema para instalar SQL Server, el proceso del programa de instalación incluirá los siguientes pasos:

- Planear una instalación de SQL Server: Revisar los requisitos de hardware y software, Revisar los requisitos del Comprobador de configuración del sistema y los problemas de bloqueo. Revisar las consideraciones de seguridad para una instalación de SQL Server.
- Configurar SQL Server 2008: Después de que el programa de instalación finalice, puede configurar SQL Server mediante utilidades gráficas o del símbolo del sistema.

#### Comprobador de Configuración del Sistema (System configuration checker)

Como parte de la instalación de SQL Server 2008, el Comprobador de configuración del sistema (SCC) examina el equipo en el que se haya instalado Microsoft SQL Server 2008. El SCC comprueba las condiciones que impiden una instalación correcta de SQL Server. Antes de que el programa de instalación inicie el Asistente para la instalación de SQL Server 2008, el SCC recupera el estado de cada elemento de comprobación, compara el resultado con las condiciones necesarias y proporciona instrucciones para eliminar los problemas de bloqueo. Todos los elementos de comprobación del SCC están habilitados por la red; las comprobaciones se pueden ejecutar en un equipo local, así como en situaciones remotas o de clúster.

#### Cómo instalar SQL Server 2008 desde el símbolo del sistema

Instalar una instancia nueva de SQL Server en el símbolo del sistema le permite especificar las características que se instalarán y cómo se deberían configurar. También puede especificar la interacción silenciosa, básica o completa con la interfaz de usuario del programa de instalación. Ejemplo de sintaxis:

setup.exe /q /ACTION=RemoveNode /INSTANCENAME="<Insert Instance Name>"

#### Cómo instalar SQL Server 2008 usando el programa de instalación

 Inserte el disco de instalación de SQL Server. Desde la carpeta raíz, haga doble clic en setup.exe. Para realizar la instalación desde un recurso compartido de red, localice la carpeta raíz de dicho recurso y, a continuación, haga doble clic en setup.exe. Si



- aparece el cuadro de diálogo Programa de instalación de Microsoft SQL Server 2008, haga clic en Aceptar para instalar los requisitos previos y, a continuación, haga clic en Cancelar para salir de la instalación de SQL Server 2008.
- 2. Si aparece el cuadro de diálogo de instalación de .NET Framework 3.5 SP1, active la casilla para aceptar el contrato de licencia de .NET Framework 3.5 SP1. Haga clic en Siguiente. Para salir de la instalación de SQL Server 2008, haga clic en Cancelar. Cuando se complete la instalación de .NET Framework 3.5 SP1, haga clic en Finalizar.
- 3. También se requiere **Windows Installer 4.5**, que se puede instalar con el Asistente para la instalación. Si se le solicita que reinicie el equipo, hágalo y, a continuación, reinicie el archivo setup.exe de SQL Server 2008.
- 4. Una vez instalados los requisitos previos, el Asistente para la instalación ejecutará el Centro de instalación de SQL Server. Para crear una nueva instalación de SQL Server 2008, haga clic en Nueva instalación independiente de SQL Server o agregar características a una instalación existente.
- 5. El Comprobador de configuración del sistema ejecutará una operación de detección en su equipo. Para continuar, haga clic en Aceptar. Se crean los archivos de registro de la instalación.
- 6. En la página **Clave del producto**, seleccione un botón de opción para indicar si está instalando una edición gratuita de SQL Server o una versión de producción del producto que tiene una clave de PID.
- 7. En la página **Términos de licencia**, lea el contrato de licencia y active la casilla para aceptar los términos y condiciones de la licencia.
- 8. El Asistente para la instalación instalará los requisitos previos de SQL Server si aún no están en el equipo. Entre ellos, figuran: .Net Framework 3.5 SP1, SQL Server Native Client, Archivos auxiliares del programa de instalación de SQL Server. Para instalar los requisitos previos, haga clic en Instalar.
- 9. El **Comprobador de configuración** del sistema comprobará el estado del sistema de su equipo antes de seguir con la instalación.
- 10. En la página Selección de características, seleccione los componentes de la instalación. Después de seleccionar el nombre de la característica se muestra una descripción de cada grupo de componentes en el panel derecho. Puede activar una combinación de casillas.
  - Si desea especificar un directorio personalizado para los componentes compartidos, use el campo situado en la parte inferior de la página Selección de características. Para cambiar la ruta de instalación de los componentes compartidos, actualice el nombre de ruta en el campo situado en la parte inferior del cuadro de diálogo o haga clic en Examinar para moverse a un directorio de instalación. La ruta de instalación predeterminada es C:\Archivos de programa\Microsoft SQL Server\100\.
- 11. En la página Configuración de instancia, especifique si desea instalar una instancia predeterminada o una instancia con nombre. Id. de instancia: de forma predeterminada, el nombre de instancia se utiliza como identificador de la instancia. Se usa para identificar los directorios de instalación y las claves del Registro para la instancia de SQL Server. Es así en las instancias predeterminadas y en las instancias con nombre. En el caso de una instancia predeterminada, el nombre y el identificador de la citada instancia serían MSSQLSERVER. Para utilizar un identificador de instancia no predeterminado, active la casilla Id. de instancia y proporcione un valor.
  - Características e instancias detectadas: la cuadrícula muestra las instancias de SQL Server que están en el equipo en el que se ejecuta el programa de instalación. Si ya hay una instancia predeterminada instalada en el equipo, debe instalar una instancia con nombre de SQL Server 2008.
- 12. La página **Requisitos de espacio en disco** calcula el espacio en disco necesario para las características que ha especificado. A continuación, compara el espacio necesario con el espacio en disco disponible.



- 13. El flujo de trabajo en el resto del tema depende de las características que haya especificado en la instalación. Dependiendo de las selecciones, es posible que no vea todas las páginas.
- 14. En la página Configuración del servidor **Cuentas de servicio**, especifique las cuentas de inicio de sesión para los servicios de SQL Server. Los servicios reales que se configuran en esta página dependen de las características que se van a instalar.

Puede asignar la misma cuenta de inicio de sesión a todos los servicios de SQL Server, o configurar cada cuenta de servicio individualmente. También puede especificar si los servicios se inician automática o manualmente, o si están deshabilitados. Microsoft recomienda que configure de forma individual las cuentas de servicio para proporcionar los privilegios mínimos para cada servicio, donde a los servicios de SQL Server se les conceden los permisos mínimos que necesitan para completar sus tareas. Para obtener más información, vea Configuración SQL Server – Cuentas de servicio y Configurar cuentas de servicio de Windows.

Para especificar la misma cuenta de inicio de sesión para todas las cuentas de servicio en esta instancia de SQL Server, las credenciales se proporcionan en los campos de la parte inferior de la página.

Nota de seguridad: No utilice una contraseña en blanco. Utilice una contraseña segura.

Cuando termine de especificar la información de inicio de sesión para los servicios de SQL Server, haga clic en Siguiente.

- 15. Utilice la ficha Configuración del servidor **Intercalación** para especificar intercalaciones no predeterminadas para el Motor de base de datos y Analysis Services.
- 16. Use la página Configuración del Motor de base de datos Aprovisionamiento de cuentas para especificar lo siguiente:
  - Modo de Seguridad: seleccione la autenticación de Windows o la autenticación de modo mixto para su instancia de SQL Server. Si selecciona la autenticación de modo mixto, debe proporcionar una contraseña segura para la cuenta de administrador del sistema de SQL Server integrada. Una vez que un dispositivo establezca una conexión correcta con SQL Server, el mecanismo de seguridad será el mismo para la autenticación de Windows y para el modo mixto.
  - Administradores de SQL Server: debe especificar al menos un administrador del sistema para la instancia de SQL Server. Para agregar la cuenta en la que se ejecuta el programa de instalación de SQL Server, haga clic en Agregar usuario actual. Para agregar o quitar cuentas de la lista de administradores del sistema, haga clic en Agregar o en Quitar y, a continuación, modifique la lista de usuarios, grupos o equipos que tendrán privilegios de administrador para la instancia de SQL Server.

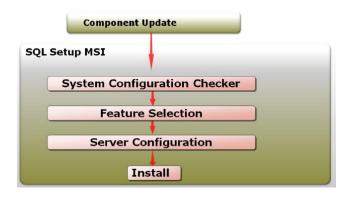
Cuando haya terminado de modificar la lista, haga clic en Aceptar. Compruebe la lista de administradores en el cuadro de diálogo de configuración. Cuando la lista esté completa, haga clic en Siguiente.

- 17. Use la página Configuración del Motor de base de datos **Directorios de datos** para especificar los directorios de instalación no predeterminados. Para instalar en los directorios predeterminados, haga clic en Siguiente.
  - **Importante**: Si especifica los directorios de instalación no predeterminados, asegúrese de que las carpetas de instalación sean únicas para esta instancia de SQL Server. Ninguno de los directorios de este cuadro de diálogo se debe compartir con los de otras instancias de SQL Server.
- 18. Use la página Configuración del Motor de base de datos **FILESTREAM** para habilitar FILESTREAM para la instancia de SQL Server.
- 19. Use la página Configuración de **Analysis Services** Aprovisionamiento de cuentas para especificar los usuarios o las cuentas que tendrán permisos de administrador para

Analysis Services. Debe especificar al menos un administrador del sistema para Analysis Services. Para agregar la cuenta en la que se ejecuta el programa de instalación de SQL Server, haga clic en Agregar usuario actual. Para agregar o quitar cuentas de la lista de administradores del sistema, haga clic en Agregar o Quitar y, a continuación, modifique la lista de usuarios, grupos o equipos que tendrán privilegios de administrador para Analysis Services.

Cuando haya terminado de modificar la lista, haga clic en Aceptar. Compruebe la lista de administradores en el cuadro de diálogo de configuración. Cuando la lista esté completa, haga clic en Siguiente.

- 20. Use la página Configuración de Analysis Services Directorios de datos para especificar los directorios de instalación no predeterminados.
- 21. Use la página Configuración de **Reporting Services** para especificar el tipo de instalación de Reporting Services que se creará. Entre las opciones posibles se encuentran las siguientes:
  - Configuración predeterminada del modo nativo
  - Configuración predeterminada del modo de SharePoint
  - Instalación de Reporting Services sin configurar
- 22. En la página Informes de errores y de uso, especifique la información que desee enviar a Microsoft y que ayudará a mejorar SQL Server. De forma predeterminada, las opciones para los informes de errores y el uso de características están habilitadas.
- 23. El Comprobador de configuración del sistema ejecutará uno o varios conjuntos de reglas para validar la configuración del equipo con las características de SQL Server que ha especificado.
- 24. La página **Listo** para instalar muestra una vista de árbol de las opciones de instalación que se especificaron durante la instalación. Para continuar, haga clic en Instalar.
- 25. La página **Progreso** de la instalación muestra el estado para que pueda supervisar el progreso de la instalación durante la ejecución del programa de instalación.
- 26. Después de la instalación, la página Importación completada proporciona un vínculo al archivo de registro de resumen para la instalación y otras notas importantes. Para completar el proceso de instalación de SQL Server, haga clic en Cerrar.
- 27. Si el programa indica que se reinicie el equipo, hágalo ahora. Es importante leer el mensaje del Asistente para la instalación tras finalizar el programa de instalación.



#### Componentes de Instalación

En el panel **Componentes para instalar**, se muestra una descripción de cada grupo de componentes al seleccionarlo. Puede activar una combinación de casillas de verificación.



Grupo de componentes	Para instalar estos componentes y características				
Servicios de bases de datos de SQL Server	SQL Server Database Engine (Motor de base de datos de SQL Server) incluye las siguientes tecnologías:				
	<ul> <li>Database Engine (Motor de base de datos) es el servicio principal para almacenar, procesar y proteger los datos.</li> </ul>				
	<ul> <li>La réplica es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de base de datos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia.</li> </ul>				
	<ul> <li>La búsqueda de texto proporciona funcionalidad para realizar consultas de texto en datos simples basados en caracteres contenidos en tablas de SQL Server.</li> </ul>				
	<ul> <li>Herramientas para administrar datos relacionales y XML.</li> </ul>				
Analysis Services	Analysis Services incluye las herramientas para crear y administrar aplicaciones de procesamiento analítico en línea (OLAP) y de minería de datos.				
Reporting Services	Reporting Services incluye componentes de servidor y de cliente para crear, administrar e implementar informes tabulares, matriciales, gráficos y de forma libre. Reporting Services también es una plataforma extensible que puede utilizarse para desarrollar aplicaciones de informes.				
Notification Services	Notification Services es una plataforma para desarrollar e implementar aplicaciones que envíen información personalizada puntualmente a los suscriptores de una gran variedad de dispositivos.				
Integration Services	Integration Services es un conjunto de herramientas gráficas y objetos programables para mover, copiar y transformar datos.				
Componentes de la estación de trabajo, Libros en pantalla y herramientas de desarrollo	Instala componentes para la comunicación entre clientes y servidores, incluidas bibliotecas de red para DB-Library, OLEDB para OLAP, ODBC, ADODB y ADOMD+.  Herramientas de administración				
	<ul> <li>SQL Server Management Studio (SSMS), nuevo en Microsoft SQL Server 2008, es un entorno integrado para obtener acceso, configurar, administrar y desarrollar todos los componentes de SQL Server. SSMS reúne las características del Administrador corporativo, el Analizador de consultas y Analysis Manager, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno que proporciona acceso para SQL Server a los desarrolladores y administradores de todos los niveles de conocimiento.</li> <li>El Administrador de configuración de SQL Server proporciona administración de configuración básica para los servicios, protocolos de servidor, protocolos de cliente y alias de cliente de SQL Server.</li> <li>Analizador de SQL Server proporciona una interfaz gráfica de usuario para supervisar una instancia de Database Engine (Motor de base de datos) o una instancia de Analysis Services.</li> <li>El Asistente para la optimización de Database Engine (Motor de base de datos) crea conjuntos óptimos de índices, vistas indizadas y particiones.</li> <li>El Monitor de réplica permite realizar un seguimiento del estado y del rendimiento de las publicaciones y las suscripciones en una topología de réplica.</li> <li>Características de cliente SQLXML</li> </ul>				
	DOGUTTOTRAGIUT				



- Los Libros en pantalla de SQL Server son la documentación principal de SQL Server 2008.
- Kits de desarrollo de software

Herramientas de desarrollo

Business Intelligence Development Studio es un entorno de desarrollo integrado para las soluciones de Analysis Services, Reporting Services y Integration Services.

#### Actualización de SQL Server

Existen dos formas de actualizar versiones de SQL Server. Cada una tiene sus beneficios y sus limitaciones y es apropiada para cada circunstancia:

- In-place Upgrades: Ocurre cuando una versión ya instalada es reemplazada por una nueva versión. Es una forma de actualización sencilla y automática pero riesgosa. Si la actualización falla es difícil volver al estado anterior de instalación. Se debe tener en cuenta también que no solo la actualización puede fallar, sino que podría ocurrir que la aplicación que utiliza las bases de datos se comporten diferente con esta nueva versión y se requiera volver a la instalación anterior hasta que puedan hacerse los ajustes necesarios en la nueva versión.
  - La ventaja de este tipo de instalación es que no requiere hardware adicional, ni tener que redireccionar las aplicaciones clientes a un nuevo servidor.
- Side by Side Upgrades: Es una alternativa más segura, ya que la instalación original se mantiene sin modificaciones. Sin embargo esta opción necesita más hardware y más trabajo de instalación. Una vez instalado el SQL Server es necesario copiar las bases de datos a la nueva ubicación lo cual requiere espacio en disco para esta operación. Normalmente este tipo de actualización se efectúa en servidores diferentes aunque es posible hacerla en el mismo servidor.
- **Híbrida:** Es una opción que utiliza algunos elementos de cada una de las opciones anteriores. Por ejemplo en vez de copiar todas las bases de datos a la nueva instalación, estas pueden ser adjuntadas directamente a esta nueva versión

#### Instalación Desatendida

Usted puede realizar una instalación desatendida creando un archivo .ini que contenga la información de setup y ejecutando el **setup.exe** desde la línea de comandos. Entender cómo realizar una instalación desatendida puede ayudarlo a distribuir múltiples instalaciones de SQL Server idénticas.

#### Creación del archivo .ini

- 1. Inserte el disco de instalación de SQL Server. Desde la carpeta raíz, haga doble clic en **Setup.exe**. Para realizar la instalación desde un recurso compartido de red, localice la carpeta raíz de dicho recurso y, a continuación, haga doble clic en Setup.exe.
- 2. Siga los pasos del asistente hasta llegar a la página Listo para instalar. La ruta de acceso al archivo de configuración se especifica en la sección que así lo indica en la página Listo para instalar.
- 3. Cancele la instalación sin completarla realmente, para generar el archivo INI.

También puede utilizar cualquier editor de texto, por ejemplo Notepad.







#### Comenzando una instalación desatendida

Para comenzar una instalación desatendida utilice una sentencia similar a la siguiente en la línea de comando:

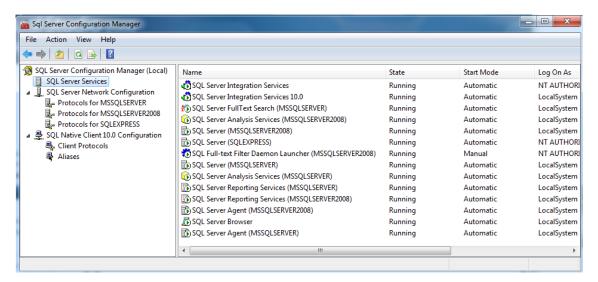
Setup.exe /ConfigurationFile=MyConfigurationFile.INI



#### 3- Configuración

#### Administrador de Configuración (Configuration Manager)

El Administrador de configuración de SQL Server es una herramienta para administrar los servicios asociados a SQL Server, para configurar los protocolos de red utilizados por SQL Server y para administrar la configuración de conectividad de red de los equipos cliente de SQL Server.



#### **Servicios SQL Server**

Usted puede utilizar SQL Server Configuration Manager para iniciar, pausar, detener o restaurar servicios Windows asociados a SQL Server. También puede configurar los servicios para que controlen sus modos de inicio y sus cuentas de servicio, así como propiedades avanzadas como por ejemplo parámetros de inicio.

**Importante:** para cambiar las cuentas de servicio utilice SQL Server Configuration Manager en lugar de la consola de Windows Services porque SQL Server Configuration Manager aplica automáticamente los permisos de registro requeridos para la cuenta que usted especifique.

#### Administrar Protocolos de Red

El Administrador de configuración de SQL Server permite configurar protocolos de red de servidor y cliente, así como opciones de conectividad. Una vez habilitados los protocolos correctos, no suele ser necesario cambiar las conexiones de red del servidor. Sin embargo, puede utilizar el Administrador de configuración de SQL Server si necesita volver a configurar las conexiones del servidor de modo que SQL Server escuche en un protocolo de red, puerto o canalización concretos.

El Administrador de configuración de SQL Server permite administrar protocolos de red de cliente y servidor, lo que incluye la posibilidad de exigir el cifrado del protocolo, ver las propiedades del alias o habilitar o deshabilitar un protocolo.

El Administrador de configuración de SQL Server también permite ver información sobre las instancias del clúster de conmutación por error, aunque se debe utilizar el Administrador de clústeres para algunas acciones como el inicio y la detención de los servicios.





#### Configuración del Cliente de Red

Cuando se instala SQL Server Configuration Manager en una computadora cliente, usted puede utilizarlo para manejar la librería **Cliente Nativo SQL** (SQL Native Client) estableciendo el orden de prioridad de protocolos de red y alianzas de servidor.

#### **SQL Server Management Studio**

SQL Server Management Studio es un entorno integrado para obtener acceso a todos los componentes de SQL Server, configurarlos, administrarlos y desarrollarlos. Combina un amplio grupo de herramientas gráficas con una serie de editores de script para ofrecer acceso a SQL Server a programadores y administradores de todos los niveles de especialización.

SQL Server Management Studio combina las características del Administrador corporativo, el Analizador de consultas y Analysis Manager, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno. Además, SQL Server Management Studio funciona con todos los componentes de SQL Server, como Reporting Services, Integration Services y SQL Server Compact 3.5. Los programadores obtienen una experiencia familiar y los administradores de bases de datos una única herramienta completa que combina herramientas gráficas fáciles de usar con funciones de script enriquecidos.

SQL Server Management Studio incluye las siguientes características generales:

- Compatibilidad con la mayoría de las tareas administrativas de SQL Server.
- Un entorno único integrado para administración y edición de SQL Server Database Engine (Motor de base de datos de SQL Server).
- Nuevos cuadros de diálogo para la administración de objetos de SQL Server Database Engine (Motor de base de datos de SQL Server), Analysis Services, Reporting Services, Notification Services y SQL Server Compact 3.5 SP1, lo que permite ejecutar las acciones inmediatamente, enviarlas a un editor de código o escribirlas en scripts para ejecutarlas posteriormente.
- Cuadros de diálogo no modales y de tamaño variable que permiten obtener acceso a varias herramientas mientras un cuadro de diálogo está abierto.
- Un cuadro de diálogo común de programación que permite realizar acciones de los cuadros de diálogo de administración en otro momento.
- Exportación e importación del registro de servidor de SQL Server Management Studio desde un entorno de Management Studio a otro.
- Guardado o impresión de archivos de plan de presentación XML o de interbloqueo generados por el Analizador de SQL Server, revisión posterior o envío a los administradores para su análisis.
- Un nuevo cuadro de mensaje de error e informativo que presenta mucha más información, permite enviar a Microsoft un comentario sobre los mensajes, copiar mensajes en el Portapapeles y enviar fácilmente los mensajes por correo electrónico al equipo de soporte.
- Un explorador Web integrado para una rápida exploración de MSDN o la Ayuda en pantalla.
- Integración de la Ayuda de comunidades en línea.
- Un tutorial sobre SQL Server Management Studio para ayudarle a aprovechar las ventajas de las numerosas características nuevas y a que sea más productivo de forma inmediata.
- Un nuevo monitor de actividad con filtro y actualización automática.
- Interfaces de Correo electrónico de base de datos integradas





#### Nuevas funciones de las Secuencias de Comandos

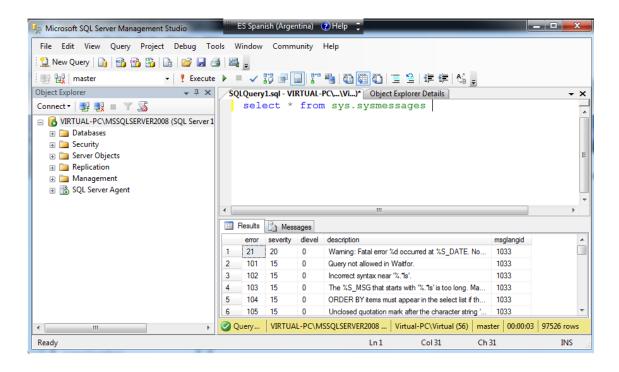
El Editor de código de SQL Server Management Studio contiene editores de scripts integrados para crear scripts **Transact-SQL**, **MDX**, **DMX**, **XML/A** y **XML**. Ofrece las características siguientes:

- Ayuda dinámica para el acceso inmediato a la información relevante mientras se trabaja.
- Un amplio conjunto de plantillas y la posibilidad de crear plantillas personalizadas.
- Compatibilidad con la escritura y modificación de consultas o scripts sin necesidad de conexión a un servidor.
- Compatibilidad con scripts para consultas y scripts SQLCMD.
- Una nueva interfaz para ver resultados XML.
- Control de código fuente integrado para proyectos de scripts y soluciones compatible con el almacenamiento y la conservación de copias de scripts a medida que evolucionan.
- Compatibilidad de Microsoft IntelliSense con instrucciones MDX

#### Características del Explorador de objetos

El Explorador de objetos de SQL Server Management Studio es una herramienta integrada para ver y administrar objetos en todo tipo de servidores. Ofrece las características siguientes:

- Filtrado por todo o parte de un nombre, esquema o fecha.
- Llenado asincrónico de objetos, con la posibilidad de filtrar objetos según sus metadatos.
- Acceso al Agente SQL Server en los servidores de réplica para administración.



#### **SQLCMD**

Puede utilizar la utilidad **sqlcmd**, una utilidad de símbolo del sistema de Microsoft, para ejecutar scripts e instrucciones Transact-SQL ad hoc. Para utilizar sqlcmd de manera interactiva o crear archivos de scripts que pueda ejecutar con sqlcmd, es necesario que esté familiarizado con Transact-SQL. La utilidad sqlcmd tiene normalmente los usos siguientes:



- En el entorno sqlcmd, puede especificar instrucciones Transact-SQL de forma interactiva del mismo modo que se utilizan en el símbolo del sistema. Los resultados aparecen en la ventana del símbolo del sistema, a menos que especifique otra cosa.
- Puede enviar un trabajo sqlcmd especificando una sola instrucción Transact-SQL para ejecutarla o bien indicando a la utilidad un archivo de script que contenga instrucciones Transact-SQL para ejecutar. sqlcmd puede conectarse a versiones anteriores de SQL Server.

#### Ejemplo:

En la línea de comandos ejecutar:

 $sqlcmd - E - S < nombre Servidor > - d < Nombre Base Datos > - q "CREATE PROCEDURE Person. SP\_Sqlcmd AS SELECT FirstName, LastName FROM Person. Person WHERE LastName LIKE 'Whi%';"$ 

# Módulo 2

# Recuperación de desastre



#### .

#### 1- Planeamiento

#### Planear la estrategia de copias de seguridad y restauración

Al administrar una base de datos de SQL Server, es importante estar preparado para la recuperación de desastres potenciales. Es necesario un plan de restauración y de copia de seguridad correctamente diseñado y probado para poder recuperar las copias de seguridad de SQL Server de las bases de datos después de un desastre.

Además, para garantizar que todos los sistemas y datos puedan recuperar rápidamente su funcionamiento normal en caso de un desastre natural, es necesario crear un plan de recuperación de desastres. Durante la elaboración de este plan es preciso tener en cuenta los escenarios de distintos tipos de desastres que pueden afectar a su negocio, incluidos los desastres naturales, como un incendio, y los desastres técnicos, como los errores en dos discos de una matriz RAID-5. Cuando cree un plan de recuperación de desastres, identifique y prepare todos los pasos necesarios para hacer frente a cada tipo de desastre. Debe realizar la comprobación práctica de los pasos de recuperación de cada escenario. Se recomienda que compruebe el plan de recuperación de desastres mediante la simulación de un desastre natural.

Durante el diseño del plan de copia de seguridad y restauración, es necesario realizar el diseño del plan de recuperación de desastres según el entorno y las necesidades del negocio.

Lo ideal es que el plan de recuperación de desastres indique el tiempo que durará la recuperación y el estado final de las bases de datos que los usuarios pueden esperar. Por ejemplo, puede determinar que, tras la adquisición del hardware especificado, la recuperación debe completarse en 48 horas y sólo se garantizarán los datos hasta finales de la semana previa al incidente.

Un plan de recuperación de desastres se puede estructurar de diferentes maneras y puede contener muchos tipos de información. Entre los tipos de planes de recuperación de desastres se incluyen los siguientes:

- Un plan para adquirir el hardware.
- Un plan de comunicación.
- Una lista de las personas con las que ponerse en contacto si se produce un desastre.
- Instrucciones para ponerse en contacto con las personas implicadas en la respuesta al desastre.
- Información acerca del propietario de la administración del plan.
- Una lista de comprobación de las tareas necesarias para cada escenario de recuperación. Para facilitar la revisión de la evolución de la recuperación de desastres, ponga a cada tarea una inicial a medida que se vayan completando y anote la hora de finalización en la lista de comprobación.

#### Elegir el tipo de medio para la copia de seguridad

SQL Server puede generar copias en discos rígidos o cintas. Los discos locales o a través de la red son el medio más común para guardar copias de seguridad. Cuando la copia se genera en cinta el dispositivo debe estar instalado localmente al servidor de SQL Server. Las copias en cinta van a discontinuarse en versiones futuras del SQL Server.

Una sola copia puede tener hasta 64 destinos diferentes.

#### Administrar medios de copia de seguridad

Se recomienda que el plan de copias de seguridad estipule cómo se han de administrar los medios de copia de seguridad, por ejemplo:

 Un plan de seguimiento y administración para almacenar y reciclar conjuntos de copias de seguridad.



- Una programación para sobrescribir el medio de copia de seguridad.
- En un entorno multiservidor, la decisión de utilizar copias de seguridad centralizadas o distribuidas.
- Un modo de realizar un seguimiento de la vida útil del medio.
- Un procedimiento para minimizar los efectos de la pérdida de un conjunto o medio de copia de seguridad, por ejemplo, la pérdida de una cinta.
- La decisión de guardar los conjuntos de copia de seguridad dentro o fuera del sitio, y un análisis de cómo afectaría esta decisión al tiempo de recuperación.

#### Garantizar la disposición para afrontar desastres

Para asegurarse de que está preparado para hacer frente a desastres, se recomienda que realice las siguientes tareas de forma periódica:

- Compruebe los procedimientos de copia de seguridad y recuperación antes de que se produzca un error real. Las comprobaciones le ayudan a asegurarse de que cuenta con las copias de seguridad necesarias para recuperarse de diversos errores, que sus procedimientos están perfectamente definidos y documentados y que cualquier operario cualificado puede ejecutarlos rápidamente y sin problemas.
- Para que la cantidad de datos perdidos sea mínima, realice periódicamente copias de seguridad de las bases de datos y los registros de transacciones. Se recomienda realizar copias de seguridad del sistema y de las bases de datos de los usuarios.
- Mantenga los registros del sistema de manera segura. Conserve registros de todos los Service Pack instalados en Microsoft Windows y SQL Server. Conserve registros de las bibliotecas de red usadas y del modo de seguridad. Asimismo, si SQL Server se ejecuta en autenticación de modo mixto (modo de autenticación de SQL Server y de Windows), registre la contraseña de sa.
- En otro servidor, evalúe por anticipado los pasos que debe seguir para la recuperación de un desastre, modifíquelos según sea necesario para ajustarlos a su entorno de servidor local y compruebe los pasos modificados.
- Conserve una secuencia de comandos con la funcionalidad básica a fin de evaluar rápidamente la capacidad mínima.

#### Revisar y reducir los posibles errores del usuario

Uno de los escenarios de recuperación más difíciles es recuperarse de un error de usuario importante, como quitar objetos de la base de datos de forma accidental. En esta sección se enumeran herramientas que le pueden ayudar a revisar y en algunos casos a regular los cambios efectuados a las bases de datos.

- Desencadenadores del lenguaje de definición de datos (DDL): Estos desencadenadores se pueden crear para revisar y regular algunos cambios del esquema de base de datos. Los desencadenadores DDL activan procedimientos almacenados en respuesta a una variedad de instrucciones DDL. Estas instrucciones son básicamente las que empiezan por CREATE, ALTER y DROP. El ámbito de un desencadenador DDL puede ser una base de datos determinada o una instancia de servidor completa.
- Notificaciones de eventos: Las notificaciones de eventos se ejecutan en respuesta a una variedad de instrucciones DDL de Transact-SQL y eventos de la traza de SQL, y envían información acerca de esos eventos a un servicio de Service Broker. Se pueden programar notificaciones de eventos para muchos de los eventos capturados por Traza de SQL, pero, en lugar de usarlas para crear trazas, puede usar dichas notificaciones para realizar una acción en una instancia de SQL Server 2008 como respuesta a eventos. Dado que las notificaciones de eventos se ejecutan de forma asincrónica, estas acciones no consumen recursos definidos por la transacción inmediata.



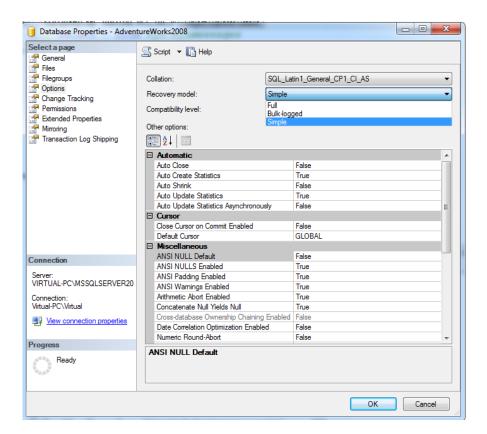
- Agente SQL Server: Se trata de un servicio de Windows que ejecuta tareas administrativas programadas, denominadas trabajos. El Agente SQL Server utiliza SQL Server para almacenar información de los trabajos. Entre otras cosas, el Agente SQL Server puede ejecutar un trabajo en respuesta a un evento concreto; por ejemplo, errores que tienen un nivel de gravedad o un número de mensaje específicos.
- Traza de SQL (Trace): La traza de SQL proporciona procedimientos almacenados del sistema Transact-SQL para crear trazas sobre clases de eventos seleccionadas por el usuario en una instancia del Motor de base de datos de SQL Server. Puede utilizar estos procedimientos almacenados del sistema desde sus propias aplicaciones para crear trazas manualmente.

#### Modelo de Recuperación

Los modelos de recuperación se han diseñado para controlar el mantenimiento del registro de transacciones. Existen tres modelos de recuperación: simple (Simple), completa (Full) y por medio de registros de operaciones masivas (BULK\_LOGGED). Normalmente, en las bases de datos se usa el modelo de recuperación completa o el modelo de recuperación simple.

En la tabla siguiente se resumen estos modelos de recuperación.

Modelo de recuperación	Descripción	Riesgo de pérdida de trabajo	¿Recuperación hasta un momento dado?
Simple	Sin copias de seguridad de registros.  Recupera automáticamente el espacio de registro para mantener al mínimo los requisitos de espacio, eliminando, en esencia, la necesidad de administrar el espacio del registro de transacciones.	Los cambios realizados después de la copia de seguridad más reciente no están protegidos. En caso de desastre, es necesario volver a realizar dichos cambios.	Sólo se puede recuperar hasta el final de una copia de seguridad.
Completa	Requiere copias de seguridad de registros.  No se pierde trabajo si un archivo de datos se pierde o resulta dañado.  Se puede recuperar hasta cualquier momento, por ejemplo, antes del error de aplicación o usuario.	Normalmente ninguno.  Si el final del registro resulta dañado, se deben repetir los cambios realizados desde la última copia de seguridad de registros.	Se puede recuperar hasta determinado momento, siempre que las copias de seguridad se hayan completado hasta ese momento.
Por medio de registros de operaciones masivas	Requiere copias de seguridad de registros.  Complemento del modelo de recuperación completa que permite operaciones de copia masiva de alto rendimiento.  Reduce el uso del espacio de registro mediante el registro masivo de la mayoría de las operaciones masivas.	Si el registro resulta dañado o se han realizado operaciones masivas desde la última copia de seguridad de registros, se pueden repetir los cambios desde esa última copia de seguridad.  En caso contrario, no se pierde el trabajo.	Se puede recuperar hasta el final de cualquier copia de seguridad. No admite recuperaciones a un momento dado.



#### Recuperación simple

Utilícelo si se dan todas las condiciones siguientes:

- La recuperación al momento del error no es necesaria. Si se pierde o se daña la base de datos, no le importa perder todas las actualizaciones realizadas entre el error y la copia de seguridad anterior.
- No le importa perder algunos datos del registro.
- No desea realizar copias de seguridad del registro de transacciones ni restaurarlo, y prefiere confiar exclusivamente en las copias de seguridad completas y diferenciales.

#### Recuperación completa

Utilice este modelo y, opcionalmente, también el **modelo de recuperación por medio de registros de operaciones masivas**, si se da cualquiera de las condiciones siguientes:

- Desea poder recuperar todos los datos.
- Si la base de datos incluye varios grupos de archivos y desea realizar una restauración por etapas de los grupos de archivos secundarios de lectura y escritura, y opcionalmente, de los de sólo lectura.
- Debe poder realizar una recuperación hasta el momento del error.
- Desea poder restaurar páginas individuales.
- Le resulta aceptable incurrir en los costes administrativos de las copias de seguridad del registro de transacciones



#### d

#### 2- Copias de Seguridad

#### Tipos de Copias de Seguridad

- Copias de seguridad completas de bases de datos
- Copias de Seguridad del Registro de Transacciones
- Copias de Seguridad Diferenciales
- Copias de Seguridad de Archivos o Grupos de Archivos

#### Copias de seguridad completas de bases de datos

Una copia de seguridad completa de la base de datos crea una copia de seguridad de toda la base de datos, que incluye parte del registro de transacciones para que se pueda recuperar la copia de seguridad completa de la base de datos. Las copias de seguridad completas representan la base de datos en el momento en que finalizó la copia de seguridad.

Las copias de seguridad de bases de datos son fáciles de utilizar. Una copia de seguridad completa de una base de datos contiene todos los datos de la base de datos. Para las bases de datos pequeñas, de las que se puede hacer una copia de seguridad con rapidez, la práctica recomendada es utilizar copias de seguridad completas de la base de datos. Sin embargo, a medida que la base de datos aumenta de tamaño, las copias de seguridad completas requieren una mayor cantidad de tiempo y espacio de almacenamiento. Por ello, para una base de datos grande, puede que desee complementar las copias de seguridad completas con copias de seguridad diferenciales.

#### Copia de seguridad en el modelo de recuperación completa

En el modelo de recuperación completa se usan copias de seguridad de registros para evitar la pérdida de datos en la mayor parte de los casos de error y es necesario realizar copias de seguridad y restaurar el registro de transacciones. La ventaja de usar las copias de seguridad de registros reside en que permite restaurar una base de datos a cualquier momento de una copia de seguridad de registros (**recuperación a un momento dado**). Si consideramos que se puede realizar una copia de seguridad del registro activo después de que ocurra un desastre, se podrá restaurar la base de datos al momento del error sin perder datos. Las desventajas de usar las copias de seguridad de registros son que requieren espacio de almacenamiento y aumentan la duración y la complejidad de las restauraciones.

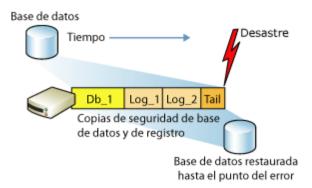
En las bases de datos en que se usa con frecuencia el modelo de recuperación completa, se pueden optimizar algunas operaciones masivas utilizando temporalmente el modelo de recuperación por medio de registros de operaciones masivas. El modelo de recuperación por medio de registros de operaciones masivas impone varias restricciones que hacen que no sea adecuado para su uso diario.

#### Ejemplo:

En la siguiente ilustración se muestra la estrategia de copia de seguridad más fácil con el modelo de recuperación completa. En la ilustración se han realizado una copia de seguridad de base de datos, Db\_1, y dos copias de seguridad de registros rutinarias, Log\_1 y Log\_2. Algún tiempo después de la copia de seguridad de registros Log\_2, se pierden datos de la base de datos. Antes de restaurar estas tres copias de seguridad, el administrador de la base de datos debe realizar una copia de seguridad del registro activo (el final del registro). Entonces, el administrador de la base de datos restaura Db\_1, Log\_1 y Log\_2 sin recuperar la base de datos. A continuación, el administrador de la base de datos restaura y recupera la copia de seguridad de registros después del error (Tail). Así se recupera la base de datos al momento del error, incluidos todos los datos.





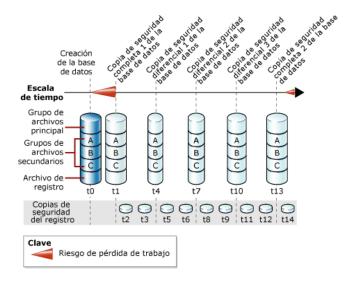


#### Minimizar el riesgo de pérdida de trabajo

Una vez que finaliza la primera copia de seguridad completa de la base de datos y se inician las copias de seguridad periódicas de registros, el riesgo potencial de pérdida de trabajo se limita al tiempo transcurrido entre el momento en que se daña la base de datos y la copia de seguridad periódica de registros más reciente. Por lo tanto, recomendamos que realice copias de seguridad de registros con suficiente frecuencia para mantener el riesgo de pérdida de trabajo dentro de los límites establecidos por sus requisitos empresariales.

Cuando se produce un error, puede intentar realizar una **copia de seguridad del registro después del error** (el registro del que aún no se ha realizado una copia de seguridad). Si la copia de seguridad del registro después del error se realiza sin problemas, puede evitar cualquier pérdida de trabajo restaurando la base de datos hasta el momento del error.

Puede utilizar una serie de copias de seguridad de registros para poner al día una base de datos hasta cualquier momento que se encuentre en una de las copias de seguridad de registros. Para minimizar el riesgo, recomendamos programar copias de seguridad de registros rutinarias. Tenga en cuenta que para minimizar el tiempo de restauración, puede complementar cada copia de seguridad completa con una serie de copias de seguridad diferenciales de los mismos datos.



En esta ilustración, antes de la primera copia de seguridad de la base de datos, existe un riesgo potencial de pérdida de trabajo en la base de datos (de la hora t0 a la hora t1). Por tanto, las copias de seguridad de registros rutinarias reducen el riesgo de pérdida de trabajo a



...

la posibilidad de perder los cambios realizados después de la última copia de seguridad de registros (realizada a la hora t14 en esta ilustración). Si se produce un error, el administrador de la base de datos debe intentar realizar inmediatamente una copia de seguridad del registro activo (el final del registro). Si esta copia de seguridad de registros después del error se realiza correctamente, la base de datos se puede restaurar hasta el momento del error.

#### Copias de Seguridad del Registro de Transacciones

En los modelos de recuperación completa y por medio de registros de operaciones masivas, es necesario realizar copias de seguridad periódicas de los registros de transacciones (**copias de seguridad de registros**) para recuperar datos. Gracias a las copias de seguridad de registros es posible recuperar la base de datos en el punto en que se haya producido el error o en un momento dado. Es aconsejable realizar copias de seguridad de registros suficientemente regulares para ajustarse a los requisitos de su empresa, específicamente a la tolerancia a la pérdida de trabajo que una unidad de registro dañada podría provocar. La frecuencia adecuada para realizar copias de seguridad de registros varía en función de la tolerancia al riesgo de pérdida de trabajo y, por otra parte, de la cantidad de copias de seguridad de registros que puede almacenar, administrar y, potencialmente, restaurar. Una copia de seguridad de registros cada 15 ó 30 minutos puede ser suficiente. Si su empresa necesita minimizar el riesgo de pérdida de trabajo, piense en la posibilidad de realizar copias de seguridad de registros más frecuentemente. Al realizar copias de seguridad de registros con más frecuencia tendrá la ventaja añadida de que la frecuencia del truncamiento del registro será mayor, por lo que los archivos o archivos de registro serán más pequeños.

Antes de crear la primera copia de seguridad de registros, debe crear una copia de seguridad completa, como una copia de seguridad de la base de datos o la primera de un conjunto completo de copias de seguridad de archivos. La restauración de una base de datos utilizando únicamente copias de seguridad de archivos puede llegar a ser un proceso complejo. Por lo tanto, es recomendable que comience con una copia de seguridad de la base de datos completa si es posible. Posteriormente, será necesario realizar copias de seguridad del registro de transacciones con regularidad. De esta forma, no sólo se minimiza el riesgo de pérdida de trabajo, sino que también se permite el **truncamiento del registro de transacciones**. Normalmente, el registro de transacciones se trunca tras cada copia de seguridad de registros convencional.

#### La cadena de registros

Una secuencia continua de copias de seguridad de registros se denomina cadena de registros. Una cadena de registros empieza con una copia de seguridad completa de la base de datos. Por lo general, una nueva cadena de registros sólo se inicia cuando se realiza una copia de seguridad de la base de datos por primera vez o después de cambiar del modelo de recuperación simple al modelo de recuperación completa o por medio de registros de operaciones masivas.

Para restaurar una base de datos al momento del error, es preciso que la cadena de registros esté intacta. De esta forma, es necesario que una secuencia ininterrumpida de las copias de seguridad del registro de transacciones se extienda hasta el momento del error. El lugar en el que esta secuencia de registros debe comenzar depende del tipo de copias de seguridad de datos que esté restaurando: de base de datos, parcial o de archivos. En las copias de seguridad de base de datos o parciales, la secuencia de copias de seguridad de registros debe extenderse desde el final de la copia de seguridad de base de datos o parcial. En un conjunto de copia de seguridad de archivos, la secuencia de copias de seguridad de registros debe comenzar desde el principio del conjunto completo de copias de seguridad de archivos.

Si sólo utiliza copias de seguridad de archivos, es necesario realizar una copia de seguridad del registro desde el principio de la primera copia de seguridad de archivos completa. Es posible comenzar a realizar copias de seguridad de registros inmediatamente después de la primera copia de seguridad copia de seguridad de archivos completa de archivos. Es



recomendable comenzar en ese momento, dado que la primera copia de seguridad de registros puede tardar mucho tiempo. Mientras se realiza la copia de seguridad del registro, puede realizar copias de seguridad de otros archivos. Para restaurar la base de datos sólo con copias de seguridad de archivos, el conjunto de copias de seguridad completas de archivos debe ampliarse con una o más copias de seguridad de registros que cubran el intervalo entre la primera copia de seguridad de archivos y la última.

## Truncamiento del registro con el modelo de recuperación completa y por medio de registros de operaciones masivas

En el modelo de recuperación completa o en el modelo de recuperación por medio de registros de operaciones masivas, no se puede truncar la parte inactiva del registro hasta que se hayan capturado todas sus entradas de registro en una copia de seguridad de registros. Esto es necesario para mantener la **cadena de registros**, una serie de entradas de registro que tienen una secuencia continúa de números de secuencia de registro (**LSN**). El registro se trunca cuando se realiza la copia de seguridad del registro de transacciones, siempre que se cumplan las siguientes condiciones:

- Se ha producido un punto de comprobación (Checkpoint) desde la última copia de seguridad del registro. Un punto de comprobación es esencial pero no suficiente para truncar el registro en el modelo de recuperación completa o el modelo de recuperación por medio de registros de operaciones masivas. Después de un punto de comprobación, el registro se mantiene intacto, al menos hasta la siguiente copia de seguridad del registro de transacciones.
- Ningún otro factor impide el truncamiento del registro.
   Por lo general, si se hacen copias de seguridad con regularidad, el espacio de registro se libera con la misma frecuencia para su uso futuro. Sin embargo, otros factores, como una transacción de ejecución prolongada, pueden impedir temporalmente el truncamiento del registro.
- La instrucción BACKUP LOG no específica WITH NO\_TRUNCATE, WITH NO\_LOG o WITH COPY ONLY.

#### Cómo funciona el truncamiento del registro

El truncamiento no reduce el tamaño del archivo de registro físico. Reducir el tamaño físico de un archivo de registro requiere la reducción del archivo.

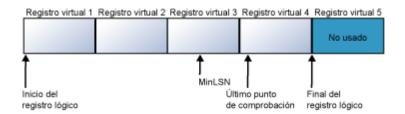
El registro de transacciones es un archivo de registro circular. Cuando se crea la base de datos, el archivo de registro lógico empieza por el principio del archivo de registro físico. Las nuevas entradas del registro se agregan al final del registro lógico y se expanden hacia el final del archivo físico. El registro de transacciones de una base de datos está asignado a uno o varios archivos físicos. El Motor de base de datos de SQL Server segmenta cada archivo de registro físico internamente en una serie de archivos de registro virtuales. El truncamiento del registro libera el espacio en el registro lógico a base de eliminar archivos de registro virtuales inactivos desde el principio del registro lógico.

Los archivos de registro virtuales son las unidades de espacio que se puede reutilizar. Sólo se pueden truncar los archivos de registro que contienen únicamente entradas de registro inactivas. La parte activa del registro de transacciones, el registro activo, no puede truncarse ya que se necesita para recuperar la base de datos. El punto de comprobación más reciente define el registro activo. El registro se puede truncar hasta ese punto de comprobación.

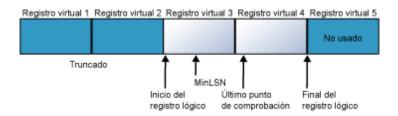
Cuando se lleva a cabo el punto de comprobación, la parte inactiva del registro de transacciones se marca como reutilizable. A partir de ese momento, se puede liberar la parte inactiva mediante el truncamiento del registro. El truncamiento libera los archivos de registro virtuales para su reutilización. Finalmente, cuando se escribe una nueva entrada en un registro virtual libre, ese archivo de registro virtual pasa de nuevo a estar activo.



En la siguiente ilustración se muestra un registro de transacciones antes y después del truncamiento. En la primera ilustración se muestra un registro de transacciones que no se ha truncado nunca. El registro lógico tiene actualmente cuatro archivos de registro virtuales en uso. El registro lógico empieza por delante del primer archivo de registro virtual y termina en el registro virtual 4. La entrada MinLSN se encuentra en el registro virtual 3. Los registros virtuales 1 y 2 sólo contienen entradas de registro inactivas. Estas entradas pueden truncarse. El registro virtual 5 no se utiliza aún y no forma parte del registro lógico actual.



En la segunda ilustración se muestra el registro después del truncamiento. Se han liberado los registros virtuales 1 y 2 para su reutilización. El registro lógico empieza ahora en el inicio del registro virtual 3. El registro virtual 5 no se utiliza aún y no forma parte del registro lógico actual.



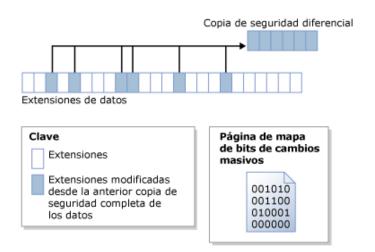
#### Cómo funcionan las copias de seguridad diferenciales

Una copia de seguridad diferencial se basa en la copia de seguridad completa más reciente existente. Esto se denomina **base** del diferencial. Una copia de seguridad diferencial incluye sólo los datos que han cambiado desde la última base diferencial.

El tamaño de una copia de seguridad diferencial depende de la cantidad de datos que han cambiado desde la base. Como regla general, cuanto más antigua sea una base, más grande será una nueva copia de seguridad diferencial. Una copia de seguridad diferencial captura el estado de las extensiones modificadas en el momento en que se crea la copia de seguridad. Si crea una serie de copias de seguridad diferenciales, es probable que una extensión actualizada con frecuencia contenga datos diferentes en cada una de las copias diferenciales. A medida que se incrementa el tamaño de las copias de seguridad diferenciales, la restauración de una copia de seguridad diferencial puede incrementar sensiblemente el tiempo necesario para restaurar una base de datos. Por ello, recomendamos que realice una copia de seguridad completa a intervalos definidos para establecer una nueva base diferencial para los datos. Por ejemplo, cada semana podría realizar una copia de seguridad completa de toda la base de datos (es decir, una copia de seguridad completa de la base de datos) seguida de una serie de copias de seguridad diferenciales de la base de datos realizadas periódicamente durante la semana.

En la siguiente ilustración se muestra cómo funciona una copia de seguridad diferencial. La ilustración muestra 24 extensiones de datos, 6 de las cuales han cambiado. La copia de seguridad diferencial sólo contiene estas 6 extensiones de datos. La operación de copia de

seguridad diferencial se basa en una página de mapa de bits que contiene un bit por cada extensión. Por cada extensión actualizada desde que se creó la base, el bit se establece en 1 en el mapa de bits.



Una copia de seguridad diferencial que se realiza poco después de su base tiende a ser sustancialmente más pequeña que la base diferencial. Así, se gana espacio de copia de seguridad y tiempo.

Sin embargo, a medida que la base de datos cambia, la diferencia entre la base de datos y una base diferencial concreta aumenta. Cuanto más tiempo pasa entre una copia de seguridad diferencial y su base, más probabilidades hay de que la copia de seguridad diferencial sea más grande. Esto significa que las copias de seguridad diferenciales pueden, a la larga, asemejarse en tamaño a la base diferencial. La copia de seguridad diferencial grande pierde las ventajas de una copia de seguridad más rápida y de menor tamaño.

En el momento de la restauración, antes de restaurar una copia de seguridad diferencial, debe restaurar su base. A continuación, sólo necesita restaurar la copia diferencial más reciente para poner al día la base de datos hasta el momento en que se creó la copia de seguridad diferencial. Por lo general, restaurará la copia de seguridad completa más reciente seguida de la copia de seguridad diferencial más reciente que está basada en la copia de seguridad completa.

Al crear y restaurar copias de seguridad diferenciales, SQL Server 2008 trata la base de datos como un conjunto de archivos. Esto afecta al contenido de las copias de seguridad diferenciales y al modo en que se utilizan en combinación con las copias de seguridad de bases de datos y de archivos. SQL Server Database Engine (Motor de base de datos de SQL Server) está diseñado para tratar los escenarios habituales fácilmente y sin ningún comportamiento inesperado.

#### Copias de seguridad diferenciales de archivos

Una copia de seguridad diferencial de archivos necesita una copia de seguridad completa de archivos como base.

La copia de seguridad diferencial de archivos supone una forma rápida de crear copias de seguridad de archivos actuales y, además, ocupa poco espacio. En el modelo de recuperación simple, las copias de seguridad diferenciales de archivos se habilitan únicamente para los grupos de archivos de sólo lectura. En el modelo de recuperación completa, las copias de seguridad diferenciales de archivos están permitidas en todos los grupos de archivos que tengan una base diferencial. Las copias de seguridad diferenciales de archivos pueden



disminuir significativamente el tiempo de recuperación reduciendo la porción del registro de transacciones que debe restaurarse.

Se recomienda utilizar las copias de seguridad diferenciales de archivos en las siguientes situaciones:

- A algunos archivos se les hacen copias de seguridad con menor frecuencia que a otros.
- Los archivos son grandes y los datos se actualizan con poca frecuencia, o bien se actualizan los mismos datos de forma repetida.

**Importante:** Evite el uso de copias de seguridad de bases de datos diferenciales y de archivos diferenciales en la misma base de datos.

#### Copia de seguridad diferencial de grupos de archivos (filegroups)

Las copias de seguridad diferenciales de una base de datos de lectura y escritura funcionan correctamente aunque uno de los grupos de archivos fuera de sólo lectura cuando se creó la base diferencial. Como con cualquier copia de seguridad diferencial de una base de datos de lectura y escritura, el Motor de base de datos de SQL Server registra la base diferencial en el archivo principal.

**Nota:** Para una base de datos de sólo lectura, el mapa de bits diferencial del archivo principal no se puede actualizar durante una copia de seguridad.

En un grupo de archivos principalmente de sólo lectura, la mayor parte del tiempo el grupo de archivos es de sólo lectura pero, ocasionalmente, pasa a ser de lectura y escritura durante breves períodos de mantenimiento. Por ejemplo, un archivo de grupos que normalmente es de sólo lectura se puede establecer temporalmente como de lectura y escritura para permitir que los archivos se importen de forma masiva y, posteriormente, se restablece a sólo lectura.

Una vez se hayan completado las actualizaciones, puede proteger los datos nuevos haciendo una copia de seguridad del grupo de archivos. La recomendación para esta copia de seguridad varía en función de cuánto ha cambiado el archivo:

- Si el grupo de archivos ha cambiado significativamente y sigue siendo de lectura y escritura, realice una copia de seguridad completa de archivos. Dado que el grupo de archivos sigue siendo de lectura y escritura, la operación de copia de seguridad puede restablecer el mapa de bits diferencial para preparar una serie de copias de seguridad diferenciales nuevas. A continuación, cambie el grupo de archivos a sólo lectura nuevamente y cree inmediatamente una copia de seguridad diferencial de archivos para poder restaurar el grupo de archivos, que ahora es de sólo lectura.
- Si el grupo de archivos no ha cambiado mucho desde la última copia de seguridad completa de archivos, asumiendo que existe una copia de seguridad completa de archivos como base diferencial, establezca el grupo de archivos como de sólo lectura de nuevo inmediatamente y, a continuación, cree una copia de seguridad diferencial de archivos.

**Nota:** La propiedad **IsReadOnly** se establece en un grupo de archivos, no en archivos individuales. Si un grupo de archivos es de sólo lectura (es decir, si la propiedad **IsReadOnly** es **TRUE** para el grupo de archivos), todos los archivos del grupo son de sólo lectura.

#### Copias de seguridad diferenciales parciales

Las copias de seguridad diferenciales parciales se utilizan sólo con las copias de seguridad parciales. Una copia de seguridad diferencial parcial únicamente registra las extensiones de datos que han cambiado en grupos de archivos desde la copia de seguridad parcial anterior,



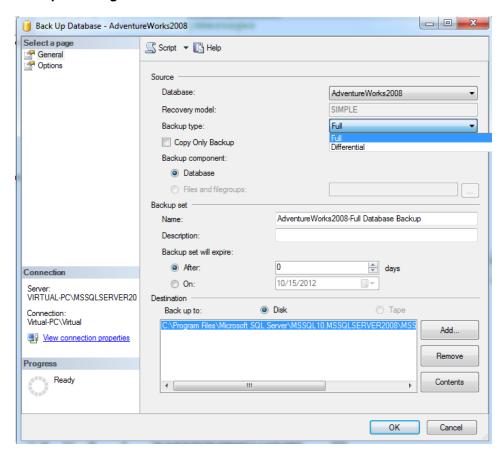
...

que se conoce como la base para la diferencial. Si sólo han cambiado algunos de los datos capturados en la copia de seguridad parcial, la copia de seguridad diferencial parcial será menor que la base y más rápida de crear. En una base de datos grande, una copia de seguridad diferencial facilita la realización de copias de seguridad frecuentes de los datos, lo que disminuye el riesgo de pérdida de datos.

Sin embargo, la restauración a partir de copias de seguridad diferenciales parciales implicará necesariamente más pasos y más tiempo que la restauración a partir de una copia de seguridad parcial. Además, el proceso de restauración es más complejo dado que intervienen dos archivos de copia de seguridad.

Las copias de seguridad diferenciales parciales se utilizan con una base diferencial única.

#### Crear una Copia de Seguridad



La sintaxis de **BACKUP** necesaria para crear una copia de seguridad:

```
BACKUP DATABASE { database_name | @database_name_var } TO <backup_device> [ ,...n ] [ <MIRROR TO clause> ] [ next-mirror-to ] [ WITH { DIFFERENTIAL | <general_WITH_options> [ ,...n ] } ] [;]
```

#### Ejemplo backup Completo:

BACKUP DATABASE [AdventureWorks2008]





TO DISK = N'L:\SQLBackup\AdventureWorks2008.bak' WITH INIT

#### Ejemplo de Copia del Registro de Transacciones

```
BACKUP LOG AdventureWorks
TO AdvWorksLog;
```

#### Ejemplo de Backup Diferencial

```
BACKUP DATABASE [AdventureWorks2008]

TO DISK = N'L:\SQLBackup\AdventureWorks2008.bak'

WITH DIFFERENTIAL, INIT
```

#### Consideraciones para el operador de las Copias de Seguridad

El operador de las copias de seguridad deberá ser miembro de alguna de las funciones fijas de seguridad mencionadas a continuación:

- sysadmin (función fija de servidor)
- db\_owner (función fija de base de datos)
- db backupoperator (función fija de base de datos)

#### Compresión de Copias de Seguridad

La creación de copias de seguridad comprimidas sólo se admite en SQL Server 2008 Enterprise y versiones posteriores, pero a partir de SQL Server 2008, todas las ediciones pueden restaurar una copia de seguridad comprimida.

La compresión de las copias de seguridad está sujeta a las siguientes restricciones:

- Las copias de seguridad comprimida y sin comprimir no pueden coexistir al mismo tiempo en un mismo conjunto de medios.
- Las versiones anteriores de SQL Server no pueden leer copias de seguridad comprimidas.
- NTbackups no puede compartir una cinta con copias de seguridad comprimidas de SQL Server.

Dado que el tamaño de una copia de seguridad comprimida es menor que el de una sin comprimir de los mismos datos, normalmente la compresión de una copia de seguridad requiere menos operaciones de E/S en los dispositivos y, por consiguiente, suele aumentar significativamente la velocidad de creación de la copia. También aumenta la velocidad de la restauración.

En la instalación, la compresión de la copia de seguridad está deshabilitada de forma predeterminada. Puede invalidar el valor predeterminado de nivel servidor al crear una copia de seguridad única o programar una serie de copias de seguridad rutinarias.

Para cambiar el valor predeterminado en el nivel de servidor

- Transact-SQL: Utilice el procedimiento almacenado sp\_configure para establecer el valor de valor redeterminado de compresión de copia de seguridad y, a continuación, ejecutar la instrucción RECONFIGURE
- SQL Server Management Studio: Utilice la página Configuración de base de datos del cuadro de diálogo Propiedades del Servidor.



#### ...

#### 3- Restauración de Bases de Datos de Usuario

#### Cómo funcionan la restauración de copias de seguridad en SQL Server

La restauración es el proceso de copiar datos desde una copia de seguridad y aplicar transacciones registradas a los datos para ponerlos al día hasta el punto de recuperación de destino. Una copia de seguridad de datos o diferencial contiene suficientes registros de transacciones para permitir poner al día las transacciones activas como parte de la restauración de cada copia de seguridad. Cada copia de seguridad contiene suficientes registros para revertir las transacciones no confirmadas y llevar la base de datos a un estado coherente con la transacción y utilizable. El proceso de poner al día las transacciones no confirmadas, si las hay, y poner la base de datos en conexión se conoce como recuperación.

#### Conjunto de puestas al día

El proceso de aplicar cambios registrados a los datos de una base de datos para poner los datos al día se conoce como **poner al día**. El conjunto de todos los datos restaurados se conoce como el **conjunto de puestas al día**. Un conjunto de puestas al día se define con la restauración de una o más copias de seguridad completa, como una copia de seguridad de base de datos o parcial, o un conjunto de copias de seguridad de archivos. Si una instrucción **RESTORE** especifica grupos de archivos, archivos o páginas, sólo se incluyen estos elementos en el conjunto de puestas al día. De lo contrario, se incluirán en el conjunto de puestas al día todos los archivos de la copia de seguridad que se está restaurando. Si la copia de seguridad completa contiene entradas de registro, los datos restaurados se pondrán al día mediante este registro.

**Nota:** Si especifica un grupo de archivos durante la restauración, ésta engloba todo el grupo de archivos tal como existe actualmente. Esto incluye los archivos agregados al grupo de archivos desde que se realizó la copia de seguridad.

Para copias de seguridad diferenciales, si se agregaron archivos a la base de datos desde la base diferencial, la restauración de una copia de seguridad diferencial podría sobrescribir páginas del conjunto de puestas al día con datos de la copia de seguridad diferencial.

La restauración de una copia de seguridad diferencial sólo actualiza una página si ésta está en el conjunto de puestas al día; la página se incluye en la copia de seguridad y la instrucción **RESTORE** muestra la página o su archivo o no muestra ningún archivo ni página.

Con los modelos de recuperación completa y de recuperación por medio de registros de operaciones masivas, se debe realizar una copia de seguridad independiente del registro. Después de restaurar copias de seguridad de datos y (opcionalmente) diferenciales, en general debería restaurar las copias de seguridad de registros subsiguientes para llevar la base de datos al punto de error. Restaurar una copia de seguridad de registros pone al día todas las páginas del conjunto de puestas al día.

#### Secuencias de restauración

Cada escenario de restauración se implementa mediante uno o varios pasos de restauración (operaciones), lo que se denomina **secuencia de restauración**. Cada operación corresponde a una instrucción **RESTORE** de **Transact-SQL** independiente. Una secuencia de restauración mueve los datos afectados a través de una o varias fases de la restauración.

#### Fases de la restauración

Una restauración es un proceso de varias fases. Las fases posibles de una restauración incluyen las fases de copia de datos, rehacer (puesta al día) y deshacer (revertir):



- La fase de copia de datos implica copiar todos los datos, el registro y las páginas de índice desde el medio de copia de seguridad de una base de datos a los archivos de la base de datos.
- La fase de rehacer aplica las transacciones registradas a los datos copiados desde la copia de seguridad para poner al día esos datos hasta el punto de recuperación. Normalmente, en este punto una base de datos tiene transacciones no confirmadas y se encuentra en un estado inutilizable. En ese caso, se requiere una fase de deshacer como parte de la recuperación de la base de datos.
- La fase de deshacer, que es la primera parte de la recuperación, revierte cualquier transacción no confirmada y hace que la base de datos esté disponible para los usuarios. Después de la fase de reversión, no se pueden restaurar las copias de seguridad subsiguientes.

#### Fase de copia de datos

La primera fase de todo proceso de restauración es la fase de copia de datos. La fase de copia de datos inicializa el contenido de la base de datos, los archivos o las páginas que se restauran. Esta fase se realiza mediante las operaciones de restaurar la base de datos, restaurar archivos y restaurar páginas utilizando copias de seguridad completas o diferenciales. La fase de copia de datos implica copiar datos de una o más copias de seguridad completas y, de forma opcional, copias de seguridad diferenciales y, a continuación, restablecer el contenido de la base de datos, los archivos o las páginas afectados en el momento en que fueron capturados por esas copias de seguridad.

El archivo o la página más antigua del conjunto de puestas al día determina el punto de inicio de la siguiente fase: rehacer (puesta al día).

#### Fase de rehacer (puesta al día)

Rehacer (o puesta al día) es el proceso de rehacer los cambios registrados hasta los datos del conjunto de puestas al día para avanzar los datos en el tiempo. Para llevar a cabo la puesta al día, el Motor de base de datos de SQL Server procesa las copias de seguridad de registros conforme se restauran, empezando por el registro contenido en las copias de seguridad completas.

La restauración evita las puestas al día innecesarias. Generalmente, si los datos eran de sólo lectura cuando se realizó la copia de seguridad y han permanecido como de sólo lectura, la puesta al día es innecesaria y se omite.

El objetivo de la puesta al día es devolver los datos a su estado original en el **punto de recuperación**. El punto de recuperación es el punto hasta el que el usuario especifica que debe recuperarse el conjunto de datos. Con el modelo de recuperación completa, puede especificar el punto de recuperación como un momento determinado, una transacción marcada o un número de secuencia de registro. Con el modelo de recuperación por medio de registros de operaciones masivas, sólo puede realizar la restauración a un momento dado si no se ha realizado ninguna operación masiva desde la copia de seguridad de registros anterior.

En la fase de rehacer, los datos siempre se ponen al día hasta un punto que es **coherente para rehacer** con el estado de la base de datos en el punto de recuperación. Todos los datos se han puesto al día hasta un punto en el que se puede realizar la operación de deshacer.

El archivo principal define el estado de la base de datos, como se indica a continuación:

• Si el archivo principal se está restaurando, el punto de recuperación determina el estado de toda la base de datos. Por ejemplo, si una base de datos se está recuperando a un momento dato justo antes de que la tabla se quitara accidentalmente, toda la base de datos debe restaurarse al mismo momento dado.



• Si el archivo principal no se está restaurando, el estado de la base de datos es conocido y los datos restaurados se ponen al día hasta un punto de recuperación transaccionalmente coherente con la base de datos. SQL Server lo exige.

Sin embargo, la base de datos puede contener cambios realizados por transacciones que no están confirmadas en el punto de recuperación. Para la restauración con conexión, los datos se recuperan a un momento dado coherente con el estado actual de la parte conectada de la base de datos.

Una copia de seguridad diferencial avanza hasta el punto en que se realizó la copia de seguridad diferencial. Las páginas del conjunto de puestas al día se sobrescriben con páginas más recientes de la copia de seguridad diferencial.

#### Fase de deshacer (revertir) y recuperación

Después de que la fase de rehacer haya puesto al día todas las transacciones del registro, una base de datos suele contener los cambios realizados por las transacciones no confirmadas en el punto de recuperación. Esto convierte los datos puestos al día en transaccionalmente incoherentes. El proceso de recuperación abre el registro de transacciones para identificar las transacciones no confirmadas. Las transacciones no confirmadas se deshacen mediante la reversión, a menos que mantengan bloqueos que eviten que otras transacciones vean datos transaccionalmente incoherentes. Este paso se denomina fase de deshacer (o revertir). Si los datos ya son transaccionalmente coherentes al inicio del proceso de recuperación, la fase de deshacer se omitirá. Después de que la base de datos sea transaccionalmente coherente, la recuperación conecta la base de datos.

#### Opciones RECOVERY y NORECOVERY

Una instrucción **RESTORE** específica termina después de la fase de rehacer o continúa con la fase de deshacer, según si la instrucción especificó **WITH NORECOVERY**, de la manera siguiente:

- WITH RECOVERY: Recupera la base de datos e incluye las fases de rehacer y deshacer; no se pueden restaurar otras copias de seguridad. Es el valor predeterminado.
  - Si el conjunto de puestas al día no se ha puesto al día lo suficiente para ser coherente con la base de datos, la fase de deshacer no se podrá producir. El Motor de base de datos emitirá un error y la recuperación se detendrá. Si todo el conjunto de puestas al día es coherente con la base de datos, se realizará la recuperación y se podrá conectar la base de datos.
- WITH NORECOVERY: Omite la fase de deshacer para preservar las transacciones no
  confirmadas. Al omitir la fase de deshacer, se pueden restaurar otras copias de
  seguridad para poner al día la base de datos a un momento posterior. A veces,
  RESTORE WITH NORECOVERY pone al día los datos hasta donde son coherentes
  con la base de datos. En estos casos, el Motor de base de datos emite un mensaje
  informativo indicando que el conjunto de puestas al día se puede recuperar mediante la
  opción RECOVERY.

#### Escenarios de restauración

Un escenario de restauración en SQL Server es el proceso de restaurar datos de una o más copias de seguridad y, a continuación, recuperar la base de datos. Los escenarios de restauración compatibles dependen del modelo de recuperación de la base de datos y de la edición de SQL Server 2008.

La siguiente tabla presenta los posibles escenarios de restauración compatibles para diferentes modelos de recuperación.





Escenario de restauración	Modelo de recuperación simple	Modelos de recuperación completa o por medio de registros de operaciones masivas
Restauración completa de la base de datos	Es la estrategia de restauración básica. Una restauración de base de datos completa puede implicar simplemente la restauración y recuperación de una copia de seguridad completa de base de datos. Por otra parte, una restauración de base de datos completa puede consistir en restaurar una copia de seguridad completa de base de datos y, luego, restaurar y recuperar una copia de seguridad diferencial.	Es la estrategia de restauración básica. Una restauración completa de una base de datos supone restaurar una copia de seguridad completa de base de datos y, opcionalmente, una copia de seguridad diferencial (si existe), además de restaurar todas las copias de seguridad de registros posteriores (en orden secuencial). La restauración completa de base de datos finaliza al recuperar la última copia de seguridad de registros y restaurarla (RESTORE WITH RECOVERY).
Restauración de archivos*	Restauración de uno o más archivos de sólo lectura dañados, sin restaurar la base de datos completa. La restauración de archivos está disponible sólo si la base de datos tiene como mínimo un grupo de archivos de sólo lectura.	Restaura uno o más archivos, sin restaurar la base de datos completa. La restauración de archivos puede realizarse mientras la base de datos está desconectada o, en algunas versiones de SQL Server 2008, cuando está conectada. Durante la restauración de archivos, los grupos de archivos en los que se incluyen los archivos en cuestión permanecen siempre desconectados.
Restauración de páginas	No aplicable	Restaura una o más páginas dañadas. La restauración de páginas puede realizarse mientras la base de datos está desconectada o, en algunas versiones de SQL Server 2008, cuando está conectada. Durante la restauración de páginas, las páginas que se están restaurando permanecen siempre desconectadas.  Es preciso que haya disponible una cadena intacta de copias de seguridad de registros, hasta el archivo de registro actual, y deben
Restauración por	Restauración y recuperación de la base de	aplicarse todas a fin de actualizar la página según el archivo de registro actual.  Restauración y recuperación de la base de
etapas (de grupos de archivos)*	datos por etapas a nivel de grupo de archivos, empezando por el grupo de archivos principal y todos los grupos de archivos secundarios de lectura y escritura.	datos por etapas a nivel del grupo de archivos, empezando por el grupo de archivos principal.

#### Modelos de recuperación y operaciones de restauración admitidas

Las operaciones de restauración disponibles para una base de datos dependen de su modelo de recuperación. En la tabla siguiente se resumen todos los modelos de recuperación y las diferentes situaciones de restauración en las que funcionarían.

Operación de restauración	Modelo de recuperación completa	Modelo de recuperación por medio de registros de operaciones	Modelo de recuperación simple



		masivas	
Recuperación de datos	Recuperación completa (si el registro está disponible).	Existe el riesgo de perder algunos datos.	Se perderán los datos desde la última copia de seguridad completa o diferencial.
Restauración a un momento dado	Cualquier momento cubierto por las copias de seguridad de registros.	No está permitida si la copia de seguridad de registros contiene algún cambio registrado de forma masiva.	No compatible.
Restauración de archivos*	Totalmente compatible.	A veces*	Sólo está disponible para archivos secundarios de sólo lectura.
Restauración de páginas*	Totalmente compatible.	A veces	Ninguna.
Restauración por etapas (de grupos de archivos)*	Totalmente compatible.	A veces	Sólo está disponible para archivos secundarios de sólo lectura.

<sup>\*</sup> Sólo disponible en SQL Server 2008 Enterprise Edition.

#### Restaurar una copia de seguridad completa de la base de datos

La sintaxis **RESTORE** básica para restaurar una copia de seguridad de la base de datos es:

RESTORE DATABASE nombre\_basedatos FROM dispositivo [ WITH NORECOVERY ]

**Nota:** Use **WITH NORECOVERY** si también desea restaurar una copia de seguridad diferencial de la base de datos.

#### Restauración de base de datos completa (modelo de recuperación simple)

El objetivo de una restauración completa de la base de datos es restaurar la base de datos completa. Durante el proceso de restauración, la base de datos completa se encuentra sin conexión. Antes de que ninguna parte de la base de datos tenga conexión, se recuperan todos los datos a un punto coherente en el que todas las partes de la base de datos se encuentran en el mismo momento y en el que no existe ninguna transacción sin confirmar.

En el modelo de recuperación simple, no se puede restaurar la base de datos a un momento concreto de una copia de seguridad específica.

Una restauración completa de base de datos con el modelo de recuperación simple implica una o dos instrucciones **RESTORE**, en función de si se debe restaurar o no una copia de seguridad diferencial de la base de datos.

Si sólo usa copias de seguridad completas de la base de datos, restaure sólo la copia de seguridad más reciente. Si también usa una copia de seguridad diferencial de la base de datos, restaure la copia de seguridad completa más reciente de la base de datos sin recuperar la base de datos y, a continuación, restaure la copia de seguridad diferencial más reciente de la base de datos y recupere la base de datos.

La base de datos se restaura a su estado de copia de seguridad completa de base de datos. Al recuperar una base de datos, se recomienda especificar explícitamente la opción **RECOVERY** por motivos de claridad, aunque es la opción predeterminada.





#### Ejemplo:

En el siguiente ejemplo se muestra primero cómo usar la instrucción **BACKUP** para crear una copia de seguridad completa y diferencial de la base de datos AdventureWorks2008. A continuación, se restauran estas copias de seguridad una después de la otra.

**Nota:** En el ejemplo se comienza con una instrucción **ALTER DATABASE** que establece el modelo de recuperación como **SIMPLE**.

```
USE master;
-- Establece el modelo Simple
ALTER DATABASE AdventureWorks2008 SET RECOVERY SIMPLE;
-- Back up Completo de la base de datos AdventureWorks.
BACKUP DATABASE AdventureWorks 2008
TO DISK = 'Z:\SQLServerBackups\AdventureWorks.bak'
WITH FORMAT;
GO
--Crea Backuo diferencial.
BACKUP DATABASE AdventureWorks 2008
TO DISK = 'Z:\SQLServerBackups\AdventureWorks.bak'
WITH DIFFERENTIAL;
GO
-- Restauración completa del backup completo.
RESTORE DATABASE AdventureWorks2008
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks.bak'
WITH FILE=1, NORECOVERY;
-- Restauración diferencial.
RESTORE DATABASE AdventureWorks 2008
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks.bak'
WITH FILE=2, RECOVERY;
```

#### Restauración por etapas de la base de datos (modelo de recuperación completa)

En una secuencia de restauración por etapas se restaura y recupera una base de datos en fases en el nivel del grupo de archivos, empezando con los grupos de archivos principales, los de lectura y escritura, y los secundarios.

En este ejemplo, la base de datos adb se restaura en un nuevo equipo después de un desastre. La base de datos está utilizando el modelo de recuperación completa, por lo que, antes de iniciar la restauración, debe crearse una copia de seguridad de registros después del error de la base de datos. Antes del desastre, todos los grupos de archivos están conectados. El grupo de archivos B es de sólo lectura. Se deben restaurar todos los grupos de archivos secundarios, pero por orden de importancia: A (el de mayor importancia), C y, por último, B. En este ejemplo, hay cuatro copias de seguridad de registros, incluida la copia de seguridad de registros después del error.

Antes de restaurar la base de datos, el administrador debe realizar una copia de seguridad de registros después del error. Dado que la base de datos está dañada, esta copia de seguridad debe crearse con la opción **NO\_TRUNCATE**:

```
BACKUP LOG adb TO tailLogBackup WITH NORECOVERY, NO TRUNCATE
```

La copia de seguridad de registros después del error es la última copia de seguridad que se aplica en las secuencias de restauración siguientes.



#### a. Restauración parcial del grupo de archivos principal y secundario A:

```
RESTORE DATABASE adb FILEGROUP='Primary' FROM backup1
WITH PARTIAL, NORECOVERY
RESTORE DATABASE adb FILEGROUP='A' FROM backup2
WITH NORECOVERY
RESTORE LOG adb FROM backup3 WITH NORECOVERY
RESTORE LOG adb FROM backup4 WITH NORECOVERY
RESTORE LOG adb FROM backup5 WITH NORECOVERY
RESTORE LOG adb FROM backup5 WITH NORECOVERY
RESTORE LOG adb FROM tailLogBackup WITH RECOVERY
```

#### b. Restauración con conexión del grupo de archivos C.

En este momento, el grupo de archivos principal y el secundario A están conectados. Todos los archivos de los grupos de archivos B y C están pendientes de recuperación y sin conexión. Los mensajes de la última instrucción **RESTORE LOG** del paso 1 indican que la reversión de las transacciones en las que interviene el grupo de archivos C se ha diferido porque este grupo de archivos no está disponible. Las operaciones periódicas pueden continuar, pero estas transacciones mantienen los bloqueos y no se truncará el registro hasta que se pueda completar la reversión. En la segunda secuencia de restauración, el administrador de la base de datos restaura el grupo de archivos C:

```
RESTORE DATABASE adb FILEGROUP='C'
FROM backup2a WITH NORECOVERY
RESTORE LOG adb FROM backup3 WITH NORECOVERY
RESTORE LOG adb FROM backup4 WITH NORECOVERY
RESTORE LOG adb FROM backup5 WITH NORECOVERY
RESTORE LOG adb FROM tailLogBackup WITH RECOVERY
```

En este momento, los grupos de archivos principal, A y C están conectados. Los archivos del grupo B permanecen pendientes de recuperación, con el grupo de archivos sin conexión. Las transacciones diferidas se han resuelto y se trunca el registro.

#### c. Restauración con conexión del grupo de archivos B.

En la tercera secuencia de restauración, el administrador de la base de datos restaura el grupo de archivos B. La copia de seguridad del grupo de archivos B se realizó después de cambiar el grupo a sólo lectura, por lo que no es necesario ponerlo al día durante la recuperación.

```
RESTORE DATABASE adb FILEGROUP='B'
FROM backup2b WITH RECOVERY
```

## Restauración con conexión de un archivo de sólo lectura (modelo de recuperación simple)

El modelo de recuperación simple permite restaurar un archivo de sólo lectura con conexión si existe una copia de seguridad realizada después de que el archivo pasara a ser de sólo lectura por última vez.

En este ejemplo, la base de datos adb contiene tres grupos de archivos. El grupo de archivos A es de lectura y escritura, mientras que los grupos de archivos B y C son de sólo lectura. Inicialmente, los tres están conectados.

Se debe restaurar el archivo de sólo lectura b1 del grupo de archivos B. El administrador de la base de datos puede restaurarlo mediante una copia de seguridad realizada después de que el archivo pasara a ser de sólo lectura. Durante la restauración, el grupo de archivos B permanecerá sin conexión, pero el resto de la base de datos estará conectada.







Para restaurar el archivo, el administrador de la base de datos utiliza la siguiente secuencia de restauración:

RESTORE DATABASE adb FILE='b1' FROM filegroup\_B\_backup
WITH RECOVERY

Ahora el archivo está conectado.

#### Aplicar copias de seguridad del registro de transacciones

Este tema sólo es relevante para el modelo de recuperación completa o para el modelo de recuperación optimizado para cargas masivas de registros.

En este tema se describe la aplicación de copias de seguridad del registro de transacciones como parte de la restauración de una base de datos de SQL Server. Para aplicar una copia de seguridad del registro de transacciones, deben cumplirse los requisitos siguientes:

- Primero debe restaurarse la copia de seguridad diferencial de la base de datos o la copia de seguridad completa inmediatamente anterior de la base de datos.
- Todos los registros de transacciones creados después de esa copia de seguridad completa o diferencial de la base de datos deben restaurarse en orden cronológico. Si se pierde o se daña una copia de seguridad del registro de transacciones en esta cadena de registros, sólo puede restaurar los registros de transacciones anteriores al registro de transacciones que falta.
- Todavía no se ha recuperado la base de datos. No se puede recuperar la base de datos hasta que se haya aplicado el registro de transacciones final. Si recupera la base de datos después de restaurar una de las copias de seguridad intermedias del registro de transacciones, anterior al final de la cadena de registros, no podrá restaurar la base de datos más allá de ese punto sin reiniciar toda la secuencia de restauración, empezando por la copia de seguridad completa de la base de datos.

#### Registros de transacciones y recuperación

Cuando termina la operación de restauración y recupera la base de datos, la recuperación revierte todas las transacciones incompletas. Este paso se conoce como la **fase de deshacer**. Revertir es necesario para restaurar la integridad de la base de datos. Después de la reversión, la base de datos pasa a estar en línea y no se pueden aplicar más copias de seguridad del registro de transacciones a la base de datos.

Es aconsejable restaurar todas las copias de seguridad de registros mediante WITH NORECOVERY:

RESTORE LOG nombre\_basedatos FROM <dispositivo> WITH NORECOVERY

A continuación, después de restaurar la última copia de seguridad de registros, recupere la base de datos en una operación aparte:

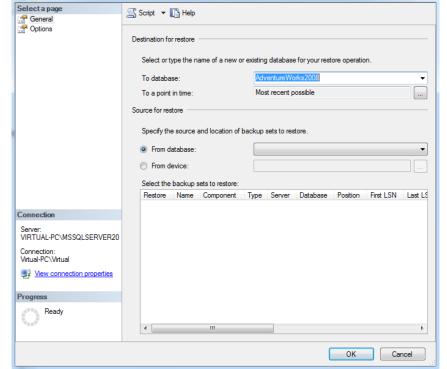
RESTORE DATABASE nombre\_basedatos WITH RECOVERY

Restaurar una Base de Datos usando SQL Server Management Studio





- - X



#### Restaurar a un momento dado

Restore Database - AdventureWorks2008

```
RESTORE DATABASE AdventureWorks
FROM AdventureWorksBackups
WITH FILE=3, NORECOVERY;

RESTORE LOG AdventureWorks
FROM AdventureWorksBackups
WITH FILE=4, NORECOVERY, STOPAT = 'Apr 15, 2020 12:00 AM';

RESTORE LOG AdventureWorks
FROM AdventureWorks
FROM AdventureWorksBackups
WITH FILE=5, NORECOVERY, STOPAT = 'Apr 15, 2020 12:00 AM';
RESTORE DATABASE AdventureWorks WITH RECOVERY;
```

#### Restauración online

Sólo se admite la restauración con conexión en SQL Server 2008 Enterprise Edition. En esta versión, la restauración de un archivo, una página o por etapas es con conexión de manera predeterminada.

La restauración de datos mientras la base de datos está conectada se denomina **restauración con conexión**. Se considera que una base de datos está conectada siempre que el grupo de archivos principal esté conectado, aunque alguno de los grupos de archivos secundarios esté sin conexión. En todos los modelos de recuperación se puede restaurar un archivo sin conexión mientras la base de datos está conectada. En el modelo de recuperación completa, también se pueden restaurar páginas mientras la base de datos está conectada.

Durante una operación de restauración de archivos con conexión, los archivos que se estén restaurando y su grupo de archivos están sin conexión. Si algunos de dichos archivos están





conectados cuando se inicia una restauración con conexión, la primera instrucción de la restauración desconecta el grupo de archivos al que pertenece el archivo. Por el contrario, durante una restauración con conexión de una página, sólo esa página está desconectada.

El escenario de restauración con conexión implica los siguientes pasos básicos:

- o Restaure los datos.
- Restaure el registro utilizando WITH RECOVERY para la última restauración del registro. Así, se conectan los datos restaurados.

A veces, una transacción sin confirmar no se puede revertir porque los datos necesarios para la operación de reversión están sin conexión durante el inicio. En ese caso, la transacción se difiere.

#### Desconectar una base de datos o un archivo

Si no desea utilizar la restauración con conexión, puede desconectar la base de datos antes de iniciar la secuencia de restauración; para ello, puede usar uno de los métodos siguientes:

 En todos los modelos de recuperación puede desconectar la base de datos utilizando la siguiente instrucción ALTER DATABASE:

ALTER DATABASE database\_name SET OFFLINE

 Si lo desea, en el modelo de recuperación completa, puede forzar que la restauración de un archivo o una página sea sin conexión; para ello, utilizando la siguiente instrucción BACKUP LOG la base de datos se pone en el estado de restauración:

BACKUP LOG database name WITH NORECOVERY.

Siempre que la base de datos permanezca sin conexión, todas las restauraciones serán sin conexión.

## Módulo 3

# Manejo de Seguridad



#### ...

#### 1- Seguridad de Base

#### **Proteger SQL Server**

La protección de SQL Server implica:

- Seguridad de la plataforma y de la red: La plataforma de SQL Server incluye el hardware físico y los sistemas de redes que conectan los clientes con los servidores de base de datos, así como los archivos binarios que se utilizan para procesar solicitudes de base de datos.
- Seguridad Física: Las recomendaciones de seguridad física limitan de forma estricta el acceso al servidor físico y a los componentes de hardware. Por ejemplo, use salas cerradas de acceso restringido para el hardware de servidor de base de datos y los dispositivos de red. Además, limite el acceso a los medios de copia de seguridad almacenándolos en una ubicación segura fuera de las instalaciones. La implementación de la seguridad de la red física comienza por mantener a los usuarios no autorizados fuera de la red.

#### Restringir el Acceso a la Red

Los usuarios y administradores deben ser autenticados por el sistema operativo Microsoft Windows para poder obtener acceso a los datos de Microsoft SQL Server 2008. El sistema operativo Windows permite a todos los usuarios autenticados obtener acceso a un servidor miembro desde la red. Sin embargo, de forma predeterminada, los miembros del grupo **Everyone** también tienen permiso para obtener acceso a los servidores miembro. El grupo **Everyone** incluye los usuarios anónimos. Utilice la herramienta Directiva de seguridad local para quitar al grupo **Everyone** el derecho de obtener acceso al equipo desde la red. Esta herramienta reside en el grupo Herramienta administrativas del equipo.

#### Seguridad del Sistema Operativo

Los Service Packs y las actualizaciones del sistema operativo incluyen mejoras de seguridad importantes. Aplique todas las revisiones y actualizaciones al sistema operativo después de probarlas con las aplicaciones de base de datos.

Los **firewalls** también proporcionan formas eficaces de implementar la seguridad. Lógicamente, un firewall es un separador o limitador del tráfico de red, que puede configurarse para aplicar la directiva de seguridad de datos de su organización. El uso de un firewall aumenta la seguridad del sistema operativo ya que proporciona un punto de arranque en el que pueden centrarse las medidas de seguridad.

La reducción de la superficie es una medida de seguridad que implica detener o deshabilitar componentes no utilizados. La reducción de la superficie ayuda a mejorar la seguridad al proporcionar menos accesos para ataques potenciales al sistema. La clave para limitar la superficie de SQL Server consiste en ejecutar los servicios requeridos con "privilegios mínimos" mediante la concesión de los derechos necesarios únicamente a los servicios y usuarios.

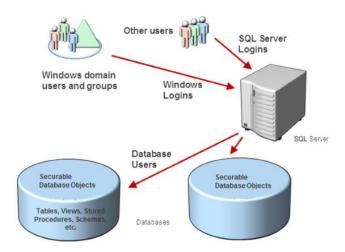
Si el sistema SQL Server usa los Servicios de **Internet Information Server** (IIS), es necesario realizar pasos adicionales para proteger la superficie de la plataforma.



### 2- Introducción a la Seguridad de SQL Server

#### Entidades de Seguridad y Seguridad de objetos de base de datos

Las **entidades de seguridad** son los individuos, grupos y procesos que tienen acceso a SQL Server. Los **asegurables** son el servidor, la base de datos y los objetos incluidos en la base de datos. Cada uno de estos elementos dispone de un conjunto de permisos que pueden configurarse para minimizar aún más la superficie de SQL Server.



#### **Entidades de Seguridad (Principals)**

Las entidades de seguridad son entidades que pueden solicitar recursos de SQL Server. Igual que otros componentes del modelo de autorización de SQL Server, las entidades de seguridad se pueden organizar en jerarquías. El ámbito de influencia de una entidad de seguridad depende del ámbito de su definición: Windows, servidor o base de datos; y de si la entidad de seguridad es indivisible o es una colección. Un Inicio de sesión de Windows es un ejemplo de entidad de seguridad indivisible y un Grupo de Windows es un ejemplo de una del tipo colección. Toda entidad de seguridad tiene un identificador de seguridad (SID).

Entidades de seguridad a nivel de Windows:

- Inicio de sesión del dominio de Windows
- Inicio de sesión local de Windows

Entidades de seguridad a nivel de SQL Server

• Inicio de sesión de SQL Server

Entidades de seguridad a nivel de bases de datos

- Usuario de base de datos
- Función de base de datos
- Función de aplicación

#### Asegurables

Los asegurables son los recursos cuyo acceso es regulado por el sistema de autorización del Motor de base de datos de SQL Server. Algunos asegurables pueden estar incluidos en otros, con lo que se crean jerarquías anidadas denominadas "ámbitos" que a su vez se pueden asegurar. Los ámbitos asegurables son **servidor**, **base de datos** y **esquema**.





#### Inicio de Sesión sa de SQL Server

El inicio de sesión **sa** de SQL Server es una entidad de seguridad de nivel de servidor. Se crea de forma predeterminada cuando se instala una instancia.

#### La Función public

Además de las funciones fijas de servidor mencionadas en la tabla anterior, cada instancia de SQL contiene una función fija de servidor especial denominada **public** de la cual todos los inicios de sesión son miembros.

A la función public se le concede el permiso VIEW ANY DATABASE.

#### Funciones (Roles) de Nivel de Servidor (Server Fixed Roles)

Las funciones fijas de servidor abarcan todo el ámbito del servidor. Cada miembro de una función fija de servidor puede agregar otros inicios de sesión a esa misma función.

Las funciones fijas de servidor se pueden asignar a los permisos más específicos que se incluyen en SQL Server 2008. La siguiente tabla describe la asignación de funciones fijas de servidor a permisos.

Función fija de servidor	Permiso en el servidor
bulkadmin	Se le concede: ADMINISTER BULK OPERATIONS
dbcreator	Se le concede: CREATE DATABASE
diskadmin	Se le concede: ALTER RESOURCES
processadmin	Se le concede: ALTER ANY CONNECTION, ALTER SERVER STATE
securityadmin	Se le concede: ALTER ANY LOGIN
serveradmin	Se le concede: ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
setupadmin	Se le concede: ALTER ANY LINKED SERVER
sysadmin	Se le concede con la opción GRANT: CONTROL SERVER

#### Funciones (Roles) en el Nivel de Base de Datos

Las funciones fijas de base de datos se definen en el nivel de base de datos y existen en cada una de ellas. Las funciones fijas de base de datos se pueden asignar a permisos más detallados que se incluyen en SQL Server 2008. La tabla siguiente describe la asignación de funciones fijas de base de datos a permisos.

Función fija de base de datos	Permiso en la base de datos	Permiso en el servidor
db_accessadmin	Concedido: ALTER ANY USER, CREATE SCHEMA	Concedido: VIEW ANY DATABASE
db_accessadmin	Concedido con la opción GRANT: CONNECT	
db_backupoperator	Concedido: BACKUP DATABASE, BACKUP LOG, CHECKPOINT	Concedido: VIEW ANY DATABASE
db_datareader	Concedido: SELECT	Concedido:



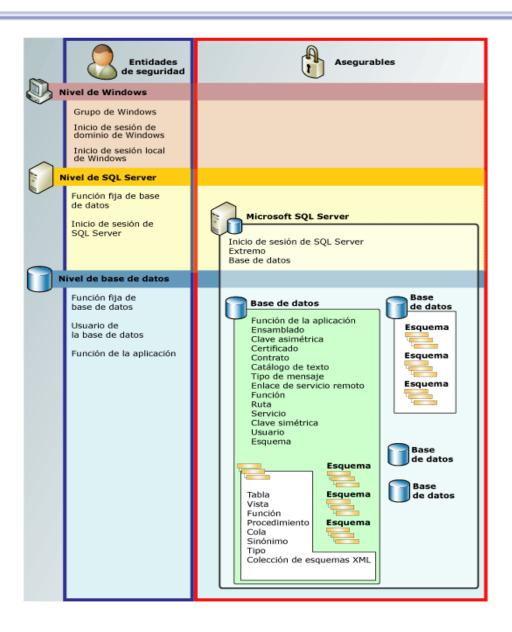
		VIEW ANY DATABASE
db_datawriter	Concedido: DELETE, INSERT, UPDATE	Concedido: VIEW ANY DATABASE
db_ddladmin	Concedido: ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT, NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULLTEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES	Concedido: VIEW ANY DATABASE
db_denydatareader	Denegado: SELECT	Concedido: VIEW ANY DATABASE
db_denydatawriter	Denegado: DELETE, INSERT, UPDATE	
db_owner	Concedido con la opción GRANT: CONTROL	Concedido: VIEW ANY DATABASE
db_securityadmin	Concedido: ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION	Concedido: VIEW ANY DATABASE

#### Jerarquía de Permisos

El Motor de base de datos de SQL Server 2008 administra un conjunto jerárquico de entidades que se pueden proteger mediante permisos. Estas entidades se conocen como **asegurables**. Los asegurables más importantes son los servidores y las bases de datos, aunque se pueden establecer permisos discretos en niveles menores. SQL Server regula las acciones de las entidades de seguridad en los asegurables comprobando que se les han concedido los permisos adecuados.

En la ilustración siguiente se muestran las relaciones entre las jerarquías de permisos del Motor de base de datos.







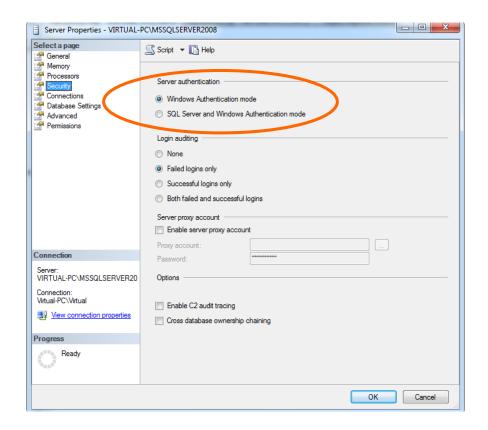
#### 3- Implementación de Seguridad en SQL Server 2008

#### Modo de Autenticación

Durante la instalación, debe seleccionar un modo de autenticación para el Motor de base de datos. Hay dos modos posibles

- Modo de autenticación de Windows
- Modo mixto.

El modo de autenticación de Windows habilita la autenticación de Windows y deshabilita la autenticación de SQL Server. El modo mixto habilita tanto la autenticación de Windows como la de SQL Server. La autenticación de Windows está disponible siempre y no se puede deshabilitar.



#### Configurar el Modo de Autenticación

Si selecciona la autenticación de modo mixto durante la instalación, debe proporcionar una contraseña segura, y confirmarla después, para la cuenta de administrador del sistema de SQL Server integrada denominada  $\mathbf{sa}$ . La cuenta  $\mathbf{sa}$  se conecta con la autenticación de SQL Server. Si selecciona la autenticación de Windows durante la instalación, el programa de instalación crea la cuenta  $\mathbf{sa}$  para la autenticación de SQL Server pero se deshabilita. Si después cambia a la autenticación de modo mixto y desea utilizar la cuenta  $\mathbf{sa}$ , debe habilitar la cuenta. Cualquier cuenta de SQL Server o de Windows se puede configurar como del administrador del sistema. Dado que la cuenta  $\mathbf{sa}$  es muy conocida y a menudo es el objetivo de usuarios malintencionados, no la habilite a menos que la aplicación lo requiera. Nunca establezca una contraseña en blanco o con poca seguridad para la cuenta  $\mathbf{sa}$ .





#### ...

#### Conectar con Autenticación Windows

Cuando un usuario se conecta a través de una cuenta de usuario de Microsoft Windows, SQL Server valida el nombre de cuenta y la contraseña con el token de la entidad de seguridad de Windows del sistema operativo. Esto significa que Windows confirma la identidad del usuario. SQL Server no pide la contraseña y no realiza la validación de identidad. La autenticación de Windows es el modo de autenticación predeterminado y es mucho más seguro que la autenticación de SQL Server. 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 y admite la expiración de las contraseñas. Una conexión realizada utilizando la autenticación de Windows se denomina a veces conexión de confianza, porque SQL Server confía en las credenciales proporcionadas por Windows.

Utilice este modo en entornos de red en los cuales los usuarios son autenticados a través de cuentas de usuario de Windows.

#### Ventajas de la Autenticación Windows

- Permite agregar grupos de usuarios a SQL Server agregando una única cuenta de usuario
- Permite a los usuarios acceder rápidamente a SQL Server sin tener que recordar cuenta y contraseña.

#### Conectar con Autenticación de SQL Server

Al utilizar la autenticación de SQL Server, los inicios de sesión se crean en SQL Server, que no se basa en las cuentas de usuario de Windows. El nombre de usuario y la contraseña se crean utilizando SQL Server y se almacenan en SQL Server. Los usuarios que se conectan utilizando la autenticación de SQL Server deben proporcionar sus credenciales (inicio de sesión y contraseña) cada vez que se conectan. Al utilizar la autenticación de SQL Server, debe establecer contraseñas seguras para todas las cuentas de SQL Server.

#### Ventajas de la Autenticación de SQL Server

- Permite a SQL Server admitir las aplicaciones anteriores y las que proporcionan terceros y requieren la autenticación de SQL Server.
- Permite que SQL Server admita entornos con sistemas operativos mixtos, en los que un dominio de Windows no autentica a todos los usuarios.
- Permite a los usuarios conectarse desde dominios desconocidos o que no son de confianza. Por ejemplo, una aplicación en la que los clientes establecidos se conectan con los inicios de sesión de SQL Server asignados para recibir el estado de sus pedidos.
- Permite que SQL Server admita aplicaciones basadas en WEB en las que los usuarios crean sus propias identidades.
- Permite a los desarrolladores de software distribuir sus aplicaciones utilizando una jerarquía de permisos compleja basada en los inicios de sesión conocidos y preestablecidos de SQL Server.

#### Desventajas de la Autenticación de SQL Server

Si un usuario del dominio de Windows tiene un inicio de sesión y una contraseña para Windows, aún debe proporcionar otro inicio de sesión y contraseña (SQL Server) para conectarse. Hacer el seguimiento de varios nombres y contraseñas es difícil para muchos usuarios. Tener que proporcionar las credenciales de SQL Server cada vez que se conectan a la base de datos puede resultar molesto.





- La autenticación de SQL Server no puede utilizar el protocolo de seguridad de Kerberos.
- Windows proporciona directivas de contraseñas adicionales que no están disponibles para los inicios de sesión de SQL Server.

#### Cuándo utilizar el Modo Mixto

Utilice este modo cuando tenga que permitir conectarse a SQL Server a usuarios o aplicaciones que no tengan credenciales Windows.

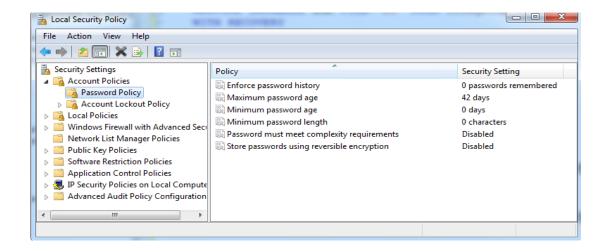
#### Contraseñas

Hay tres directivas de contraseñas opcionales para los inicios de sesión de SQL Server.

- El usuario debe cambiar la contraseña en el siguiente inicio de sesión
   Exige que el usuario cambie la contraseña la próxima vez que se conecte. SQL Server
   Management Studio proporciona la capacidad de cambiar la contraseña. Otros
   programadores de software deberían proporcionar esta característica si se utiliza esta
   opción.
- Exigir expiración de contraseña. La directiva de vigencia máxima de la contraseña del equipo se exige para los inicios de sesión de SQL Server.
- Exigir directivas de contraseñas. Las directivas de contraseñas de Windows del equipo se exigen para los inicios de sesión de SQL Server. Esto incluye la longitud y complejidad de las contraseñas. Esta funcionalidad depende de la API NetValidatePasswordPolicy, que sólo está disponible en Windows Server 2003 y versiones posteriores.

Para determinar las directivas de las contraseñas del equipo local

- En el menú Inicio, haga clic en Ejecutar.
- En el cuadro de diálogo Ejecutar, escriba secpol.msc y, a continuación, haga clic en Aceptar.
- En la aplicación Configuración de seguridad local, expanda Configuración de seguridad, expanda Directivas de cuenta y, a continuación, haga clic en Directiva de contraseñas.







#### Aplicación de las Directivas de Contraseñas

La aplicación de la directiva de contraseñas se puede configurar independientemente para cada inicio de sesión de SQL Server. Utilice ALTER LOGIN para configurar las opciones de la directiva de contraseñas de un inicio de sesión de SQL Server.

Se aplican las siguientes reglas a la configuración de la aplicación de directivas de contraseñas:

- Cuando el valor de **CHECK POLICY** se cambia a ON, ocurre lo siguiente:
  - **CHECK\_EXPIRATION** también se establece en ON, salvo que se haya establecido de forma explícita en OFF.
  - o El historial de contraseñas se inicializa con el valor hash de la contraseña actual.
- Cuando el valor de **CHECK POLICY** se cambia a OFF, ocurre lo siguiente:
  - CHECK EXPIRATION también se establece en OFF.
  - Se borra el historial de contraseñas.
  - Se restablece el valor de lockout time.

Algunas combinaciones de opciones de directiva no se admiten.

- Si se especifica MUST\_CHANGE, CHECK\_EXPIRATION y CHECK\_POLICY deben establecerse en ON. De lo contrario, se producirá un error en la instrucción.
- Si CHECK\_POLICY se establece en OFF, CHECK\_EXPIRATION no puede establecerse en ON. Se producirá un error en una instrucción ALTER LOGIN que combine estas dos opciones.

**Importante:** CHECK\_EXPIRATION y CHECK\_POLICY sólo se aplican en Windows Server 2003 y versiones posteriores.

Cuando se ejecuta SQL Server en Windows 2000, puede establecer **CHECK\_POLICY** = ON para evitar la creación de contraseñas que sean:

- Nulas o vacías
- Iguales al nombre del equipo o el inicio de sesión
- Una de las siguientes: "password", "admin", "administrator", "sa", "sysadmin"

#### Contraseñas Seguras

Las contraseñas pueden constituir el vínculo más débil de una implementación de seguridad de servidor. Debe tener siempre mucho cuidado a la hora de elegir una contraseña. Una contraseña segura presenta las siguientes características:

- Tiene una longitud de 8 caracteres como mínimo.
- Contiene una combinación de letras, números y símbolos.
- No es una palabra que pueda encontrarse en el diccionario.
- No es el nombre de un comando.
- No es el nombre de una persona.
- No es el nombre de un usuario.
- No es el nombre de un equipo.
- Se cambia con frecuencia.
- Presenta diferencias notables con respecto a contraseñas anteriores.

Las contraseñas de SQL Server pueden contener hasta 128 caracteres, entre los que se pueden incluir letras, símbolos y dígitos. Dado que los inicios de sesión, nombres de usuario, funciones y contraseñas se utilizan con frecuencia en instrucciones Transact-SQL, determinados símbolos deberán estar incluidos entre comillas dobles (") o corchetes ([]). Utilice estos delimitadores en las instrucciones Transact-SQL cuando el inicio de sesión, usuario, función o contraseña de SQL Server presente las siguientes características:

- Contiene o comienza por un carácter de espacio.
- Comienza por el carácter \$ o @.





Si se utiliza en una cadena de conexión OLE DB u ODBC, el inicio de sesión o la contraseña no deben contener ninguno de los siguientes caracteres: [] {}() , ; ? \*! @. Estos caracteres se utilizan para inicializar una conexión o para separar valores de conexión.

#### Inicios de Sesión en SQL Server (Login)

La mayoría de usuarios de Windows necesitan un inicio de sesión de SQL Server para conectarse a SQL Server. A continuación se describen distintas opciones para crear un inicio de sesión en SQL Server:

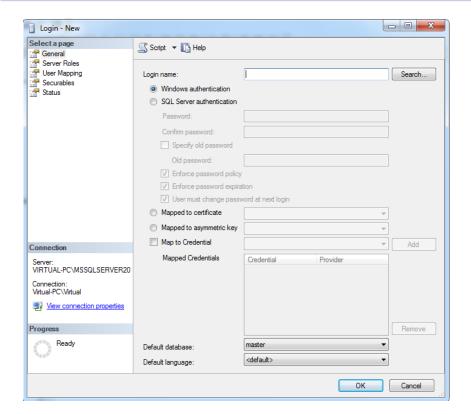
- En SQL Server Management Studio, abra el Explorador de objetos y expanda la carpeta de la instancia de servidor en la que se creará el nuevo inicio de sesión.
- Haga clic con el botón secundario en la carpeta Seguridad, seleccione Nuevo y, a continuación, Inicio de sesión.
- En la página **General**, escriba el nombre de un usuario de Windows en el cuadro **Nombre de inicio de sesión**.

Para crear un inicio de sesión de SQL Server que use la autenticación de Windows (SQL Server Management Studio)

- Seleccione Autenticación de Windows.
- Haga clic en Aceptar.

Para crear un inicio de sesión de SQL Server que use la autenticación de SQL Server (SQL Server Management Studio)

- Seleccione Autenticación de SQL Server. La autenticación de Windows es la opción más segura.
- Escriba una contraseña de inicio de sesión.
- Seleccione las opciones de la directiva de contraseñas que deben aplicarse al nuevo inicio de sesión. En general, exigir la directiva de contraseñas es la opción más segura.
- Haga clic en Aceptar.



Para crear un inicio de sesión de SQL Server que use la autenticación de Windows con Transact-SQL

CREATE LOGIN <nombre de usuario Windows> FROM WINDOWS GO

Para crear un inicio de sesión de SQL Server que use la autenticación de SQL Server con Transact-SQL

El ejemplo siguiente crea un inicio de sesión para un usuario determinado y asigna una contraseña. La opción MUST\_CHANGE requiere que los usuarios cambien la contraseña la primera vez que se conectan al servidor.

CREATE LOGIN <nombre login> WITH PASSWORD = '<password>' MUST\_CHANGE; GO

#### Ejemplos:

```
USE master
GO
CREATE LOGIN [Adventure-Works\Mary] FROM WINDOWS;
GO

CREATE LOGIN <login_name> WITH PASSWORD = '<enterStrongPasswordHere>'
MUST_CHANGE;
GO
```





#### Modificación de Inicios de Sesión

Se pueden modificar inicios de sesión desde sus propiedades en el Explorador de Objetos de SQL Management Studio o bien utilizando **ALTER LOGIN**.

Una utilización común de ALTER LOGIN es por ejemplo para desbloquear un inicio de sesión que fue bloqueado porque expiró su contraseña como se muestra a continuación:

ALTER LOGIN Alice WITH PASSWORD = 'NewPa\$\$w0rd' UNLOCK

#### Eliminación de Inicios de Sesión

Se pueden eliminar un inicio de sesión con clic derecho sobre él en el Explorador de Objetos y luego elegir **Delete**, o bien utilizando **DROP LOGIN**, como se muestra en el siguiente ejemplo:

DROP LOGIN Alice

#### Asignación de una cuenta de inicio de sesión a una función fija de servidor

Usted puede utilizar las propiedades de inicio de sesión en el SQL Management Studio o bien el procedimiento almacenado de sistema **sp\_addsrvrolemember** para agregar un inicio de sesión como miembro de una función fija de servidor. Solo los miembros de funciones fijas de servidor pueden ejecutar este procedimiento almacenado.

Cuando usted agrega una cuenta de inicio de sesión a una función fija de servidor, la cuenta adquiere los permisos asociados a esa función.

Al realizarlo tenga en cuenta lo siguiente:

- Usted no puede agregar, modificar o eliminar las funciones fijas de servidor.
- Cualquier miembro de una función fija de servidor puede agregar otra cuenta de inicio de sesión a esa función.
- El procedimiento almacenado de sistema sp\_addsrvrolemember no puede ser ejecutado dentro de una transacción definida por el usuario
- Usted puede utilizar el procedimiento almacenado de sistema sp\_dropsrvrolemember para eliminar un miembro de una función fija de servidor.

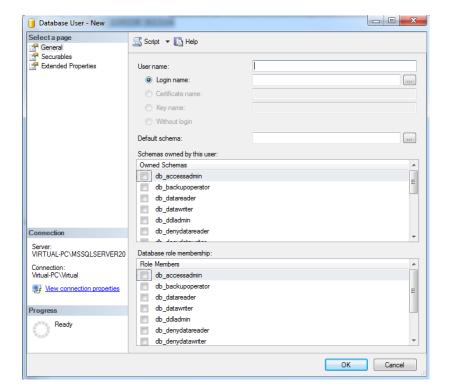
#### Usuarios de SQL Server

Para crear un usuario de base de datos mediante SQL Server Management Studio (se asume que existe el correspondiente inicio de sesión de SQL Server)

- En SQL Server Management Studio, abra el Explorador de objetos y expanda la carpeta **Bases de datos**.
- Expanda la base de datos en la que se va a crear el usuario de la misma.
- Haga clic con el botón secundario en la carpeta Seguridad, seleccione Nuevo y, a continuación, haga clic en Usuario.
- En la página General, escriba un nombre para el usuario en el cuadro Nombre de usuario.
- En el cuadro Nombre de inicio de sesión, escriba el nombre de un inicio de sesión de SQL Server para asignarlo al usuario de la base de datos.
- Haga clic en Aceptar.







Para crear un usuarios de bases de datos mediante Transact-SQL

```
USE <nombre_base_datos>
GO
CREATE USER <nombre usuario> FOR LOGIN <nombre login>;
GO
```

#### Ejemplos:

```
CREATE LOGIN John
WITH PASSWORD = '340$Uuxwp7Mcxo7Khy';
USE AdventureWorks2008;
CREATE USER John FOR LOGIN John

CREATE LOGIN Mary
WITH PASSWORD = '8fdKJl3$nlNv3049jsKK';
USE AdventureWorks2008;
CREATE USER Mary FOR LOGIN Mary
WITH DEFAULT_SCHEMA = Marketing;
```

#### Usuarios especiales de SQL Server

Cuenta de usuario dbo: dbo o propietario de base de datos, es una cuenta de usuario con permisos implícitos para realizar todas las actividades en la base de datos. Los miembros de la función fija del servidor sysadmin se asignan automáticamente a dbo. La cuenta de usuario dbo se confunde a menudo con la función fija de base de datos db\_owner. El ámbito de db\_owner es una base de datos y el ámbito de sysadmin es el servidor completo. La pertenencia a la función db\_owner no proporciona privilegios de usuario dbo.



• Cuenta de usuario guest: Después de que un usuario se haya autenticado y se le haya permitido iniciar sesión en una instancia de SQL Server, debe existir una cuenta de usuario independiente en cada base de datos a la que tenga acceso el usuario. Si se exige una cuenta de usuario en cada base de datos, se impide que los usuarios se conecten a una instancia de SQL Server y puedan tener acceso a todas las bases de datos de un servidor. La existencia de una cuenta de usuario guest en la base de datos evita este requisito, ya que permite que un inicio de sesión sin cuenta de usuario de base de datos tenga acceso a una base de datos. La cuenta guest es una cuenta integrada en todas las versiones de SQL Server. De forma predeterminada, está deshabilitada en las bases de datos nuevas. Si está habilitada, se puede deshabilitar mediante la revocación de su permiso CONNECT, que se lleva a cabo con la ejecución de la instrucción REVOKE CONNECT FROM GUEST de Transact-SQL.

**Nota:** Se debe evitar el uso de la cuenta **guest**, ya que todos los inicios de sesión que no dispongan de permisos de base de datos propios obtendrán los permisos de base de datos concedidos a esta cuenta. Si debe usar la cuenta **guest**, concédale los permisos mínimos.

#### Delegación

SQL Server y Windows pueden estar configurados para permitir que una instancia de SQL Server se conecte a otra instancia de SQL Server en el mismo contexto de u usuario autenticado por Windows. Esta técnica se denomina **delegación**.

Por ejemplo, un procedimiento almacenado en una base de datos en el Servidor1 podría consultar una tabla en una base de datos en el Servidor2. Cuando se utiliza delegación, el procedimiento almacenado utiliza la identidad del usuario que lo llamó cuando requiera datos del Servidor2.

#### Configurando SQL Server para Delegación

Para utilizar seguridad de delegación de cuenta, SQL Server debe tener un Nombre Principal de Servidor (Service Principal Name - SPN) asignado por el administrador de cuenta de dominio de Windows Server 2003/2008 a la cuenta de servicio de SQL Server y estar utilizando TCP/IP. Si el servicio de SQL Server se está ejecutando bajo una cuenta de Sistema Local, automáticamente el servicio SQL Server registrará un SPN al iniciar el servicio y lo desregistrará al cerrar SQL Server.

#### **Permisos**

**GRANT** concede permisos en un elemento que puede protegerse a una entidad de seguridad. Se puede utilizar la instrucción **REVOKE** para retirar permisos concedidos y la instrucción **DENY** para evitar que una entidad de seguridad obtenga un permiso específico mediante una instrucción GRANT.

La concesión de un permiso quita DENY o REVOKE de ese permiso en el elemento que puede protegerse especificado. Si se deniega el mismo permiso en un ámbito superior que contiene el elemento que puede protegerse, DENY tiene prioridad. No obstante, revocar el permiso concedido en un ámbito superior no tiene prioridad.

Los permisos en nivel de base de datos se conceden en el ámbito de la base de datos especificada. Si un usuario necesita permisos para objetos de otra base de datos, cree la cuenta del usuario en la otra base de datos o conceda a la cuenta del usuario acceso a la otra base de datos y a la base de datos actual.

**WITH GRANT OPTION**: Indica que la entidad de seguridad también podrá conceder el permiso especificado a otras entidades de seguridad.







#### Ejemplo:

```
GRANT ALTER ANY DATABASE
TO [AdventuWorks2008\Mary]

GRANT SELECT ON Person.Address
TO [AdventureWorks2008\Rosa];
GO

GRANT SELECT ON Person.Address
TO Usuario1 WITH GRANT OPTION;

DENY SELECT ON OBJECT::Person.Address
TO Rosa;
GO

REVOKE SELECT ON OBJECT::Person.Address
FROM Rosa;
GO
```

#### Cadena de Propiedad

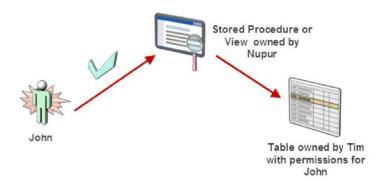
Cuando varios objetos de base de datos obtienen acceso unos a otros de forma secuencial, la secuencia se denomina cadena. Aunque estas cadenas no existen de manera independiente, cuando SQL Server recorre los eslabones de una cadena, SQL Server evalúa los permisos de los objetos que la componen de manera distinta a si se estuviese obteniendo acceso a los objetos por separado. Estas diferencias tienen implicaciones importantes en lo que se refiere a administrar la seguridad.

Cuando se obtiene acceso a un objeto mediante una cadena, SQL Server primero compara al propietario del objeto con el propietario del objeto de llamada. Este es el eslabón anterior de la cadena. Si ambos objetos tienen el mismo propietario, los permisos del objeto de referencia no se evalúan.

El uso de las cadenas de propietarios es una herramienta muy poderosa, debido a que un DBA podría en principio solo otorgarle permisos sobre un SP, sin embargo los usuarios nunca tiene permisos sobre las tablas u objetos relacionados, por consiguiente las únicas acciones que podrían hacer son las que se establecieron dentro del SP.







#### Credenciales

Una credencial es un registro que contiene la información de autenticación (credenciales) necesaria para conectarse a un recurso situado fuera de SQL Server. Esta información la utiliza SQL Server internamente. La mayoría de las credenciales incluyen un nombre de usuario y una contraseña de Windows.

La información almacenada en una credencial permite al usuario que se ha conectado a SQL Server mediante la autenticación de SQL Server obtener acceso a recursos situados fuera de la instancia del servidor. Cuando el recurso externo es Windows, el usuario se autentica como el usuario de Windows especificado en la credencial. Se puede asignar una única credencial a varios inicios de sesión de SQL Server. Sin embargo, un inicio de sesión de SQL Server sólo se puede asignar a una credencial.

Las credenciales del sistema se crean de forma automática y se asocian a extremos específicos. Los nombres de las credenciales del sistema comienzan por dos signos de número (##).

Usted puede crear una credencial desde SQL Management Studio (Solapa Security) o bien utilizando CREATE CREDENTIAL.

```
CREATE CREDENTIAL nombre credencial WITH IDENTITY = 'nombre_identity' [, SECRET = 'secreto']
```

#### Argumentos:

- **Nombre credencial**: especifica el nombre de la credencial a ser creada. No puede comenzar con el símbolo #.
- **Nombre\_identity**: Especifica el nombre de la cuenta que se utilizará para la conexión fuera del servidor.
- secreto: Especifica una palabra secreta para la autenticación. Esta cláusula es
  opcional.

#### Ejemplo

```
CREATE CREDENTIAL AlterEgo WITH IDENTITY = 'Mary5',
SECRET = '<EnterStrongPasswordHere>';
ALTER LOGIN User1
WITH CREDENTIAL = AlterEgo;
```



#### 4- Cifrado

#### Cifrado y Certificados

El cifrado no resuelve los problemas de control de acceso. Sin embargo, mejora la seguridad debido a que limita la pérdida de datos, incluso en el caso poco probable de que se superen los controles de acceso. Por ejemplo, si el equipo host de base de datos no está configurado correctamente y un pirata informático obtiene datos confidenciales, como números de tarjetas de crédito, esa información robada resulta inservible si está cifrada.

#### Jerarquía de Cifrado

SQL Server 2008 cifra los datos con una infraestructura de cifrado jerárquico y administración de claves. Cada capa cifra la capa inferior utilizando una combinación de certificados, claves asimétricas y claves simétricas.

El cifrado jerárquico está basado en una clave maestra de servicio (**Service Master Key**), esta clave es generada automáticamente cuando se instala SQL Server 2008.

El motor de base de datos utiliza la clave maestra de servicio para cifrar los siguientes objetos:

- Passwords de Servidores Vinculados (Linked Server Passwords)
- Cadenas de Conexión (Connection Strings)
- Credenciales de Cuentas (Account Credentials)
- Todas las Claves maestras de la base de datos

A la llave maestra de servicio se le debe hacer backup y almacenarla en un sitio seguro y fuera de línea. Esto para poder administrar más fácilmente ya sea hacer copias de seguridad o restaurar la clave maestra de servicio en caso de que sea necesario.

#### Llave Maestra de Servicio (Service Master Key)

Es una llave **Triple DES** en la raíz de SQL Server 2008 y es generada automáticamente la primera vez que es necesaria y está asegurada por la API de protección de datos de Windows (DPAPI).

#### Llave Maestra de Base de Datos (Database Master Key)

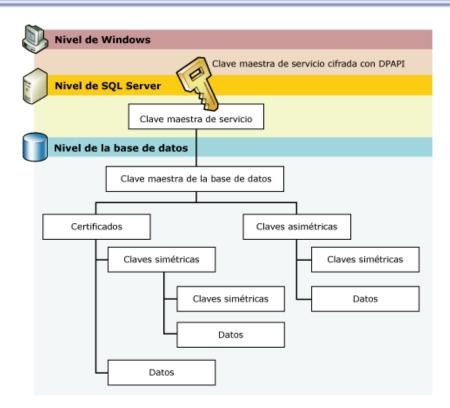
Es una llave simétrica **Triple DES** que puede utilizarse para proteger las llaves privadas de certificados y llaves asimétricas en una base de datos.

Cuando se crea una llave Maestra de Base de Datos ésta es cifrada utilizando el algoritmo Triple DES y una contraseña provista por el usuario.

SQL Server almacena una copia de la llave maestra de base de datos en la base de datos master y a su vez es cifrada con la llave maestra de servicio. Otra copia es almacenada en la base de datos cifrada con una contraseña.

La siguiente ilustración muestra que cada capa de la jerarquía de cifrado cifra la capa que tiene debajo. La capa superior, la llave maestra de servicio, se cifra con la API de protección de datos (DPAPI) de Windows.





SQL Server 2008 le ofrece los mecanismos siguientes para el cifrado:

- Certificados
- Claves asimétricas
- Claves simétricas

#### Certificados

Un certificado de clave pública, normalmente denominado sólo certificado, es una instrucción firmada digitalmente que enlaza el valor de una clave pública con la identidad de la persona, dispositivo o servicio que tiene la clave privada correspondiente. Las entidades emisoras de certificados (CA) son las encargadas de emitir y firmar los certificados. La entidad que recibe un certificado de una CA es el sujeto de ese certificado. Por lo general, los certificados contienen la siguiente información.

- La clave pública del sujeto.
- La información que identifica al sujeto, como el nombre y la dirección de correo electrónico.
- El periodo de validez. Es decir, el periodo de tiempo durante el que el certificado se considera válido.
  - Un certificado sólo es válido durante el periodo de tiempo que se especifica en el mismo; todos los certificados contienen una fecha **Válido desde** y otra **Válido hasta**. Estas fechas establecen los límites del período de validez. Cuando el período de validez de un certificado ha transcurrido, es necesario que el sujeto del certificado caducado solicite uno nuevo.
- Información de identificador del emisor.
- La firma digital del emisor. Esta firma da fe de la validez de las obligaciones entre la clave pública y la información de identificador del sujeto. (El proceso de firmar



digitalmente la información conlleva a transformar la información, así como cierta información privada que conserva el remitente, en una etiqueta denominada firma.)

Una de las principales ventajas de los certificados es que liberan a los hosts de la necesidad de establecer contraseñas para sujetos individuales. En su lugar, el host simplemente establece la confianza en un emisor de certificados, que a continuación puede firmar un número ilimitado de certificados.

Cuando un host, por ejemplo, un servidor Web seguro, designa a un emisor como entidad emisora raíz de confianza, el host implícitamente confía en las directivas que el emisor ha utilizado para establecer las obligaciones de los certificados que emite. En efecto, el host confía en que el emisor ha comprobado la identidad del sujeto del certificado. Un host designa a un emisor como entidad emisora raíz de confianza presentando el certificado autofirmado del emisor, que contiene la clave pública de éste, en el almacén de certificados de la entidad emisora de certificados raíz de confianza del equipo host. Las entidades emisoras de certificados intermedias o subordinadas sólo son de confianza si tienen una ruta válida de certificación procedente de una entidad emisora de certificados raíz.

El emisor puede revocar un certificado antes de que caduque. La revocación cancela las obligaciones que una clave pública tiene con una identidad que se exprese en el certificado. Cada emisor mantiene una lista de revocación de certificados que los programas pueden utilizar cuando estén comprobando la validez de un certificado determinado.

Dentro de las posibilidades de los certificados, se puede cargar desde un ensamblado o bien desde un archivo que contenga el certificado, como así también crear uno desde SQL 2008. Para la creación de estos certificados SQL dispone de la sentencia **CREATE CERTIFICATE**.

Ejemplo de creación de un Certificado con contraseña y fecha de expiración:

```
CREATE CERTIFICATE Shipping04
ENCRYPTION BY PASSWORD = 'pGFD4bb925DGvbd2439587y'
WITH SUBJECT = 'Sammamish Shipping Records',
EXPIRY DATE = '20121231';
```

#### Claves Asimétricas

Una clave asimétrica se compone de una clave privada y su correspondiente clave pública. Cada clave puede descifrar los datos que cifra la otra. Una clave asimétrica se puede utilizar para cifrar una clave simétrica para almacenar en una base de datos.

Las claves asimétricas son entidades que pueden protegerse a nivel de la base de datos. Por defecto, la clave privada es protegida por la **Master Key**; de no existir esta última, es necesaria una contraseña. La clave privada puede ser de 512, 1024 ó 2048 bits. El cifrado y descifrado asimétricos consumen una cantidad grande de recursos pero proporcionan un nivel de seguridad superior al del cifrado simétrico.

Para crear claves asimétricas se utiliza CREATE ASYMMETRIC KEY.

Ejemplo: En el siguiente ejemplo se crea una tabla para grabar datos cifrados con una clave asimétrica



GO

```
Claves Simétricas
```

DECLARE @KeyId int

) ON [PRIMARY]

-- Crea clave asimétrica

WITH ALGORITHM = RSA 2048

CREATE ASYMMETRIC KEY ClaveAsym

ENCRYPTION BY PASSWORD = '\$\$PEPE\$\$'

SET @KeyId = AsymKey\_Id('ClaveAsym')
INSERT INTO Tabla (Codigo, Descripcion)

-- Agrega un registro con la descripcion cifrada

Una clave simétrica es una clave que se utiliza para el cifrado y el descifrado. El cifrado y el descifrado con una clave simétrica son más rápidos y adecuados para usarlos de forma rutinaria con datos confidenciales de una base de datos.

VALUES (1, EncryptByAsymKey(@KeyId, 'mi descripción cifrada'))

Al crear una clave simétrica, ésta debe ser cifrada utilizando una de 4 opciones: certificado, contraseña, clave simétrica o clave asimétrica. Esta clave puede tener más de un cifrado de cada tipo.

Si se utiliza contraseña para cifrar la clave simétrica, entonces en lugar de utilizar la clave pública de la base de datos (Master Key) se debe utilizar el algoritmo Triple DES. Para la creación de las claves simétricas se utiliza CREATE SIMMETRIC KEY.

### Ejemplo:

```
USE AdventureWorks2008

GO

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'p@ssword'

CREATE CERTIFICATE MyCertificate
WITH SUBJECT = 'My certificate',
START_DATE = '20120131'

GO

CREATE SYMMETRIC KEY my_symmetric_key
WITH ALGORITHM = TRIPLE_DES
ENCRYPTION BY CERTIFICATE MyCertificate
GO

OPEN SYMMETRIC KEY my_symmetric_key
DECRYPTION BY CERTIFICATE MyCertificate

DECLARE @KeyId uniqueidentifier
SET @KeyId = Key_GUID('my_symmetric_key')
INSERT INTO Tabla (Codigo, Descripcion)
VALUES (2, EncryptByKey(@KeyId, 'mi descripción cifrada'))
```

**Nota:** Es muy importante realizar un backup de los certificados, como así también de las **Master Key**; para ello, SQL 2008 dispone de las sentencias **BACKUP CERTIFICATE**, **BACKUP MASTER KEY**.



# Módulo 4

# Monitoreo de SQL Server





# 1- Monitorear la Actividad en Curso

### Determinar la actividad de los usuarios

Se puede monitorear la actividad de cada usuario para identificar el tipo y la ubicación de las transacciones de bloqueo o el rendimiento lento de Microsoft SQL Server.

La supervisión de la actividad de los usuarios ayuda a identificar tendencias como los tipos de transacciones ejecutadas por los usuarios, el número de consultas ineficaces y ad hoc, y los tipos de transacciones que requieren la mayor parte de los recursos. Para recopilar información estadística sobre los usuarios, utilice el **Analizador de SQL Server**, el **Monitor de actividades del SQL Server** o el **Monitor de sistema**.

# El Monitor de actividad (SQL Server Management Studio)

Utilice el Monitor de actividad para obtener información sobre los procesos de SQL Server y el modo en que éstos afectan a la instancia actual de SQL Server.

La página Monitor de actividad tiene las secciones siguientes:

- Información general: muestra pantalla gráficas del porcentaje de tiempo de procesador, el número de tareas en espera, las operaciones de E/S de base de datos (megabytes por segundo) y las solicitudes por lote (número por segundo).
- Tareas de usuario activas: muestra información de las conexiones de usuario activas del Motor de base de datos de SQL Server.
- Esperas de recursos: muestra información sobre el estado de espera.
- E/S de archivo de datos: muestra información de E/S de los archivos de registro y de datos de la base de datos.
- Consultas costosas recientes: muestra información sobre las consultas más costosas.

Para ver el Monitor de actividad en SQL Server 2008, el usuario debe tener el permiso **VIEW SERVER STATE**.

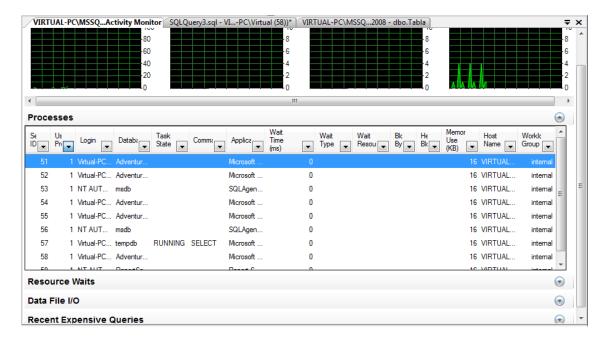
Para detener (KILL) un proceso, el usuario debe ser miembro de las funciones de servidor fijas sysadmin o processadmin.

Para abrir el Monitor de actividad en SQL Server Management Studio, en la barra de herramientas estándar de SQL Server Management Studio, haga clic en el icono del Monitor de actividad. O en el Explorador de objetos, haga clic con el botón secundario en el nombre de instancia y, a continuación, seleccione Monitor de actividad.









# Monitor de Actividad de Trabajo

El Monitor de Actividad de Trabajo permite ver la tabla **sysjobactivity** mediante SQL Server Management Studio. Puede ver todos los trabajos del servidor, o bien puede definir filtros para limitar el número de trabajos mostrados. También puede ordenar la información sobre los trabajos haciendo clic en un encabezado de columna de la cuadrícula **Actividad de Trabajo del Agente**. Por ejemplo, al seleccionar el encabezado de columna **Última ejecución**, puede ver los trabajos en el orden en que se ejecutaron por última vez. Al volver a hacer clic en el encabezado de columna, el orden de los trabajos cambia entre ascendente y descendente basándose en la fecha en que se ejecutaron por última vez.

El Monitor de actividad de trabajo le permite realizar las siguientes tareas:

- Iniciar y detener trabajos.
- Ver las propiedades del trabajo.
- Ver el historial de un determinado trabajo.
- Actualizar la información de la cuadrícula Actividad de trabajo del agente manualmente o establecer un intervalo de actualización automático haciendo clic en Ver configuración de actualización.

Utilice el Monitor de actividad de trabajo cuando desee localizar los trabajos que están programados para su ejecución, el último resultado de los trabajos que se han ejecutado durante la sesión actual y localizar los trabajos que se están ejecutando actualmente o que están inactivos. Si el servicio del Agente SQL Server tiene un error inesperado, puede determinar los trabajos que se estaban ejecutando buscando en la sesión anterior del Monitor de Actividad de Trabajo.

# Para Ver la Actividad de Trabajo (SQL Server Management Studio)

Cuando se inicia el servicio del Agente Microsoft SQL Server, se crea una nueva sesión y la tabla **sysjobactivity** de la base de datos **msdb** se llena con todos los trabajos definidos que existen. En esta tabla se registra la actividad y el estado actuales de los trabajos. Puede utilizar el Monitor de Actividad de trabajo del Agente SQL Server para ver el estado actual de los





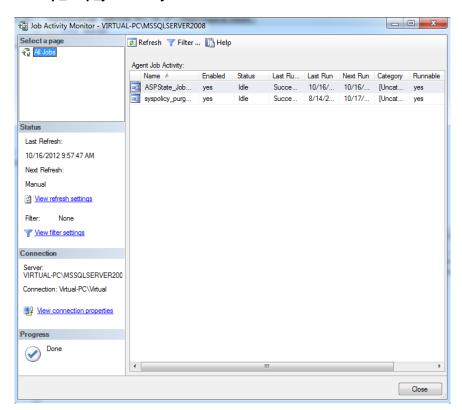
trabajos. Si el servicio del Agente SQL Server finaliza inesperadamente, puede consultar la tabla **sysjobactivity** para ver qué trabajos se estaban ejecutando cuando finalizó el servicio.

```
use msdb
select * from sysjobactivity
```

Para ver la actividad de los trabajos

- En el Explorador de objetos, conéctese a una instancia del Motor de base de datos de SQL Server y expándala.
- Expanda Agente SQL Server.
- Haga clic con el botón secundario en Monitor de Actividad de Trabajo y haga clic en Ver actividad de trabajo.
- En el Monitor de actividad de trabajo, puede ver los detalles de cada trabajo definido para este servidor.
- Haga clic con el botón secundario en un trabajo para iniciarlo, detenerlo, habilitarlo o
  deshabilitarlo, actualizar el estado que se muestra en el Monitor de Actividad de
  Trabajo, eliminarlo o ver su historial o sus propiedades. Para iniciar, detener, habilitar o
  deshabilitar, o actualizar varios trabajos, seleccione varias filas y haga clic con el botón
  secundario en la selección.
- Para actualizar el Monitor de actividad de trabajo, haga clic en Actualizar. Para ver menos filas, haga clic en Filtro y escriba los parámetros del filtro.

También puede ver la actividad de trabajo de la sesión actual mediante el procedimiento almacenado **sp\_help\_jobactivity**.





# Funciones y Vistas de Administración Dinámica

Las funciones y vistas de administración dinámica devuelven información sobre el estado del servidor que se puede utilizar para controlar el estado de una instancia del servidor, para diagnosticar problemas y para optimizar el rendimiento.

Hay dos tipos de funciones y vistas de administración dinámica:

- Funciones y vistas de administración dinámica con ámbito en el servidor. Se requiere el permiso VIEW SERVER STATE en el servidor.
- Funciones y vistas de administración dinámica con ámbito en la base de datos. Se requiere el permiso VIEW DATABASE STATE en la base de datos.

Todas las funciones y vistas de administración dinámica existen en el esquema **sys** y siguen la convención de nomenclatura siguiente: **dm\_\***. Cuando utilice una función o vista de administración dinámica, debe agregar un prefijo al nombre de la función o vista mediante el esquema **sys**.

Se muestran a continuación algunas de las vistas de administración dinámica más comunes:

Vista de Administración Dinámica	Descripción
sys.dm_db_partition_stats	Devuelve información de página y recuento de filas de cada partición en la base de datos actual.
sys.dm_exec_sessions	Devuelve una fila por cada sesión autenticada en SQL Server. <b>sys.dm_exec_sessions</b> es una vista de ámbito de servidor que muestra información acerca de todas las conexiones de usuario activas y las tareas internas.
sys.dm_io_pending_io_requests	Devuelve una fila para cada petición de E/S pendiente de SQL Server.
sys.dm_os_memory_pools	Devuelve una fila para cada almacén de objetos en la instancia de SQL Server.
sys.dm_os_threads	Devuelve una lista de todos los subprocesos del sistema operativo SQL Server que se están ejecutando en el proceso de SQL Server.
sys.dm_broker_queue_monitors	Devuelve una fila por cada monitor de cola en la instancia. Un monitor de cola administra la activación de una cola.
sys.dm_tran_locks	Devuelve información acerca de los recursos del administrador de bloqueos activos actualmente.

#### Eiemplo

```
select * from sys.dm tran locks
```

# Vistas de Catálogo (Transact-SQL)

Las vistas de catálogo devuelven información utilizada por el Motor de base de datos de SQL Server 2008 de Microsoft. Se recomienda utilizar las vistas de catálogo porque son la interfaz más general para los metadatos del catálogo y proporcionan el método más eficaz para obtener, transformar y presentar formas personalizadas de esta información. Todos los metadatos del catálogo disponibles para el usuario se exponen mediante las vistas de catálogo. Las vistas de catálogo no contienen información sobre los datos de catálogo de réplica, copia de seguridad, plan de mantenimiento de bases de datos o Agente SQL Server.

# Ejemplo:

```
select * from sys.sysobjects
select * from sys.tables
```

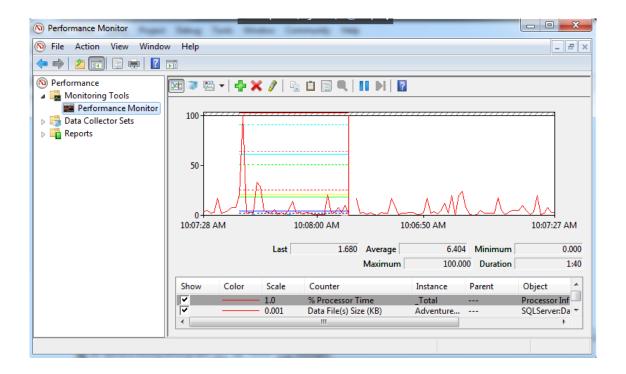


# ...

# 2- Monitor de Sistemas de Windows

Utilice el Monitor de sistema para supervisar el uso de los recursos del sistema. Recopile y vea datos de rendimiento en tiempo real en forma de contadores para recursos de servidor, como el uso del procesador y la memoria, y para muchos recursos de Microsoft SQL Server, como los bloqueos y las transacciones.

Para iniciar el Monitor de sistema en Windows en el menú **Inicio**, haga clic en **Ejecutar**, escriba **perfmon** en el cuadro de texto y haga clic en **Aceptar**. O seleccione la opción **Rendimiento** de las Herramientas Administrativas del Panel de Control.



# Objetos de SQL Server

SQL Server incluye objetos y contadores que puede utilizar el **Monitor de Sistema** para monitorear la actividad de los equipos en los que se ejecute una instancia de SQL Server. Un objeto es cualquier recurso de SQL Server como, por ejemplo, un bloqueo de SQL Server o un proceso de Windows XP. Cada objeto contiene uno o más contadores que determinan diversos aspectos de los objetos que se van a Monitorear. Por ejemplo, el objeto **Bloqueos de SQL Server** contiene los contadores **Número de interbloqueos/seg.** y **Tiempos de espera de bloqueos/seg.** 

Algunos objetos tienen varias instancias si existen varios recursos de un determinado tipo en el equipo. Por ejemplo, el tipo de objeto **Procesador** tendrá varias instancias si un sistema contiene varios procesadores. El tipo de objeto **Bases de datos** tiene una instancia para cada base de datos de SQL Server. Algunos tipos de objetos (por ejemplo, el objeto **Administrador de memoria**) tienen sólo una instancia. Si un tipo de objeto tiene varias instancias, puede agregar contadores para realizar un seguimiento de las estadísticas relativas a cada instancia o, en muchos casos, de todas las instancias a la vez. Los contadores de la instancia predeterminada aparecen con el formato **SQLServer:<object name>**. Los contadores de las instancias con nombre aparecen con el formato **MSSQL\$<instance name>:<counter name>** o **SQLAgent\$<instance name>:<counter name>**.







Al agregar o quitar contadores en el gráfico y guardar la configuración del gráfico, puede especificar los objetos y contadores de SQL Server que se supervisan al iniciar el Monitor de sistema.

Puede configurar el Monitor de sistema para que muestre las estadísticas de cualquier contador de SQL Server. Además, puede establecer un valor de umbral para cualquier contador de SQL Server y generar posteriormente una alerta cuando un contador supere dicho umbral.

### Consideraciones de Monitoreo de SQL Server

Al monitorear SQL Server y el sistema operativo Microsoft Windows para investigar problemas relacionados con el rendimiento, hay tres áreas principales en las que debe concentrarse inicialmente:

- Actividad del disco
- Uso del procesador
- Uso de la memoria

Puede resultar útil monitorear el sistema operativo Windows y los contadores de SQL Server al mismo tiempo para determinar las posibles correlaciones entre el rendimiento de SQL Server y el de Windows. Por ejemplo, la supervisión simultánea de los contadores de E/S de disco de Windows y los contadores del Administrador de búfer de SQL Server puede mostrar el comportamiento del sistema en su totalidad.

La supervisión de un equipo en el que se ejecuta el Monitor de sistema puede afectar un poco al rendimiento del equipo. Por tanto, registre los datos del Monitor de Sistema en otro disco o en otro equipo para reducir así el efecto en el equipo que está supervisando, o bien ejecute el Monitor de Sistema desde un equipo remoto. Supervise sólo los contadores en los que esté interesado. Si supervisa demasiados contadores, la sobrecarga de uso de los recursos se agrega al proceso de supervisión y afecta al rendimiento del equipo que se está supervisando.

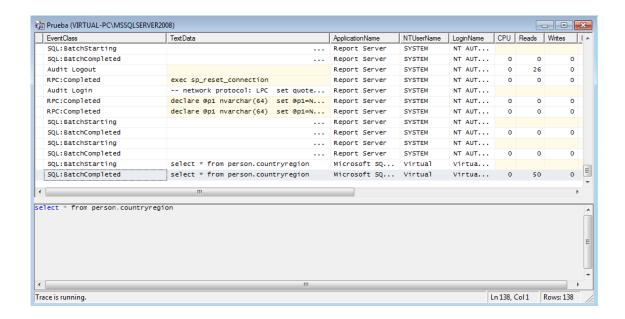
# d

# 3- Analizador de SQL Server (SQL Server Profiler)

#### Introducción

El Analizador de SQL Server de Microsoft es una interfaz gráfica de usuario de la Traza de SQL que se utiliza para supervisar una instancia del Motor de base de datos de SQL Server o de Analysis Services. Puede capturar y guardar datos acerca de cada evento en un archivo o en una tabla para analizarlos posteriormente. Por ejemplo, puede supervisar un entorno de producción para ver qué procedimientos almacenados afectan negativamente al rendimiento al ejecutarse demasiado lentamente.

Para ejecutar Analizador de SQL Server, en el menú Inicio, elija Todos los programas, Microsoft SQL Server 2008, Herramientas de rendimiento y, a continuación, haga clic en Analizador de SQL Server. También, dentro del Management Studio, menú Herramientas.



### Terminología del Analizador de SQL Server

Para utilizar el Analizador de SQL Server, debe comprender la terminología que describe cómo funciona la herramienta.

- Evento: Un evento es una acción generada dentro de una instancia del Motor de base de datos de SQL Server. Por ejemplo:
  - o Conexiones, errores y desconexiones de inicio de sesión.
  - Instrucciones SELECT, INSERT, UPDATE y DELETE de Transact-SQL.
  - o Estado de lotes de RPC (llamada a procedimiento remoto).
  - Inicio o finalización de procedimientos almacenados.
  - o Inicio o finalización de instrucciones incluidas en procedimientos almacenados.
  - Inicio o finalización de lotes SQL.
  - o Errores escritos en el registro de errores de SQL Server.
  - Bloqueos adquiridos o liberados en objetos de base de datos.
  - Cursores abiertos.
  - Comprobaciones de permisos de seguridad.

Todos los datos generados por un evento se muestran en la traza en una sola fila. Esta fila está interceptada por columnas de datos que describen el evento de forma detallada.



- Clase de evento: Una clase de evento es un tipo de evento del cual se puede realizar un seguimiento. La clase de evento contiene todos los datos que puede comunicar un evento. Por ejemplo:
  - SQL:BatchCompleted
  - Audit Login
  - Audit Logout
  - Lock:Acquired
  - Lock:Released
- Categoría de eventos: Una categoría de eventos define cómo se agrupan los eventos en el Analizador de SQL Server. Por ejemplo, todas las clases de eventos de bloqueo se agrupan dentro de la categoría de eventos Bloqueos. Sin embargo, las categorías de eventos sólo existen en el Analizador de SQL Server. Este término no refleja cómo se agrupan los eventos del motor.
- Columna de datos: Una columna de datos es un atributo de una clase de evento capturada en la traza. Como la clase de evento determina el tipo de datos que se pueden recopilar, no se aplicarán todas las columnas de datos a todas las clases de evento. Por ejemplo, en una traza que capture la clase de evento Lock:Acquired, la columna de datos BinaryData contiene el valor del ld. o la fila de la página bloqueada, pero la columna de datos Integer Data no contiene ningún valor porque no es aplicable a la clase de evento que se captura.
- Plantilla: Una plantilla define la configuración predeterminada de una traza. En concreto, incluye las clases de evento que desea supervisar con el Analizador de SQL Server. Por ejemplo, puede crear una plantilla que especifique los eventos, las columnas de datos y los filtros que desea utilizar. Las plantillas no se ejecutan, sino que se guardan como archivos con la extensión .tdf. Una vez guardada, una plantilla controla los datos de la traza que se capturan cuando se inicia una traza basada en la plantilla en cuestión.
- Traza: Una traza captura datos basándose en clases de evento, columnas de datos y filtros seleccionados. SQL Server ofrece dos formas de incluir en una traza una instancia de SQL Server: puede hacerlo con el Analizador de SQL Server o con procedimientos almacenados del sistema.
- Filtro: Al crear una traza o una plantilla, puede definir criterios para filtrar los datos recopilados por el evento. Para que las trazas no sean demasiado grandes, puede filtrarlas de forma que sólo se recopile un subconjunto de los datos del evento. Por ejemplo, puede limitar los nombres de usuario de Microsoft Windows de la traza a usuarios específicos, con lo que reducirá los datos de salida.
  - Si no se establece un filtro, se devolverán todos los eventos de las clases de evento seleccionadas en el resultado de la traza.

### Usar el Analizador de SQL Server

El Analizador de SQL Server muestra el modo en que SQL Server resuelve las consultas internamente. Esto permite a los administradores ver exactamente las instrucciones Transact-SQL o las Expresiones multidimensionales que se envían al servidor y cómo el servidor tiene acceso a la base de datos o al cubo para devolver los conjuntos de resultados.

Mediante el Analizador de SQL Server puede hacer lo siguiente:

- Crear una traza que se base en una plantilla que se puede reutilizar.
- Observar el resultado de la traza a medida que se ejecuta la traza
- Almacenar el resultado de una traza en una tabla
- Iniciar, detener, pausar y modificar el resultado de la traza según sea necesario
- Reproducir el resultado de la traza





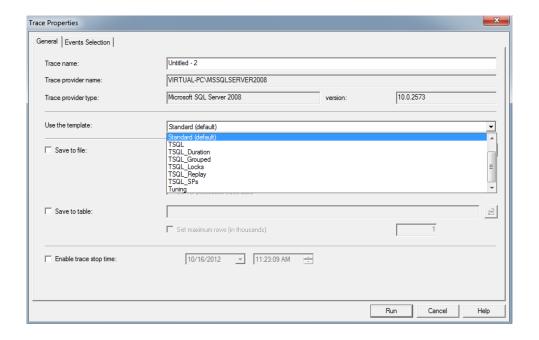
Utilice el Analizador de SQL Server para supervisar únicamente los eventos en los que está interesado. Si las trazas son demasiado grandes, puede filtrarlas a partir de la información que desea, de forma que sólo se recopile un subconjunto de los datos del evento. Si se supervisan demasiados eventos, aumentará la sobrecarga del servidor y el proceso de supervisión, y podría hacer que el archivo o la tabla de traza crezcan demasiado, especialmente cuando el proceso de supervisión se realiza durante un período prolongado de tiempo.

### Plantillas del Analizador de SQL Server

Puede utilizar el Analizador de SQL Server para crear plantillas que definan las clases de evento y columnas de datos que desea incluir en las trazas. Después de definir y guardar la plantilla, ejecute una traza que registre los datos de cada clase de evento que ha seleccionado. Una sola plantilla puede utilizarse en varias trazas puesto que la plantilla no se ejecuta como tal

El Analizador de SQL Server incluye plantillas de traza predefinidas para que pueda configurar fácilmente las clases de evento que seguramente necesitará para trazas concretas. La plantilla **Standard**, por ejemplo, le ayuda a crear una traza genérica para registrar inicios y cierres de sesión, lotes finalizados e información de conexión. Esta plantilla permite ejecutar trazas sin modificarlas o como punto de inicio para plantillas adicionales con configuraciones de evento distintas.

Además de las trazas de las plantillas predefinidas, el Analizador de SQL Server también permite crearlas a partir de una plantilla en blanco que no contenga ninguna clase de evento de manera predeterminada. Puede resultar útil utilizar la plantilla de traza en blanco cuando una traza planeada no se parece a la configuración de ninguna de las plantillas predefinidas.



# **Guardar Trazas**

Guarde los datos de los eventos capturados en un **archivo** o una **tabla** de SQL Server cuando necesite analizar o reproducir los datos capturados más adelante. Utilice un archivo de traza para lo siguiente:





- Utilice un archivo de traza o una tabla de traza para crear una carga de trabajo a fin de utilizarla como entrada para el Asistente para la Optimización del Motor de Base de Datos.
- Utilice un archivo de traza para capturar eventos y enviar el archivo de traza a un proveedor de asistencia técnica para su análisis.
- Utilice las herramientas de procesamiento de consultas de SQL Server para tener acceso a los datos o verlos en el Analizador de SQL Server. Sólo pueden tener acceso directo a la tabla de traza los miembros de la función fija de servidor sysadmin o el creador de la tabla.

**Nota:** La captura de datos de traza en una tabla resulta una operación más lenta que la captura de datos de traza en un archivo. Una alternativa es capturar los datos de traza en un archivo, abrir el archivo de traza y, después, guardar la traza como una tabla de traza. Cuando utilice un archivo de traza, el Analizador de SQL Server guardará los datos de eventos capturados (no las definiciones de traza) en un archivo de traza (\*.trc) del Analizador de SQL Server. La extensión se agrega automáticamente al final del archivo al guardarlo, independientemente de si se ha especificado otra extensión.

#### Analizar el Rendimiento con Plantillas

Si supervisa SQL Server con frecuencia, utilice plantillas para analizar el rendimiento. Las plantillas capturan los mismos datos de eventos cada vez y utilizan la misma definición de traza para supervisar los mismos eventos. No tendrá que definir las clases de eventos y las columnas de datos cada vez que cree una traza. Además, se puede proporcionar una plantilla a otro usuario para supervisar eventos específicos de SQL Server. Por ejemplo, un proveedor de asistencia técnica puede proporcionar una plantilla a un cliente. El cliente puede utilizar la plantilla para capturar los datos de eventos necesarios, que posteriormente enviará al proveedor de asistencia técnica para que los analice.



# d

# 4- Desencadenadores DDL (DDL Triggers)

#### Definición

Los desencadenadores DDL, al igual que los desencadenadores habituales, activan procedimientos almacenados como respuesta a un evento. Sin embargo, a diferencia de los desencadenadores DML, no se activan como respuesta a las instrucciones UPDATE, INSERT o DELETE de una tabla o vista. En cambio, sí se activan en respuesta a diversos eventos del lenguaje de definición de datos (DDL). Estos eventos corresponden principalmente a instrucciones Transact-SQL que comienzan por las palabras clave CREATE, ALTER y DROP. Determinados procedimientos almacenados del sistema que realizan operaciones de estilo DDL también pueden activar desencadenadores DDL.

Los desencadenadores DDL pueden utilizarse para tareas administrativas como auditar y regular las operaciones de base de datos.

Utilice los desencadenadores DDL cuando:

- Desee evitar determinados cambios en el esquema de base de datos.
- Desee que ocurra algún evento en la base de datos como respuesta a un cambio realizado en el esquema de base de datos.
- Desee registrar cambios o eventos del esquema de base de datos.

Los desencadenadores DDL sólo se activan cuando se ejecutan las instrucciones DDL que los desencadenan. Los desencadenadores DDL no se pueden utilizar como desencadenadores INSTEAD OF.

En el siguiente ejemplo se muestra el uso de un desencadenador DDL para evitar que se modifique o quite una tabla de una base de datos:

```
CREATE TRIGGER safety
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
PRINT 'Debe desactivar los desencadenadores "de seguridad" antes de modificar o eliminar bases de datos'
ROLLBACK;
```

Los desencadenadores DDL pueden activarse en respuesta a un evento de Transact-SQL procesado en la base de datos actual o en el servidor actual. El ámbito del desencadenador depende del evento.

Para ver los desencadenadores DDL que está disponible en la base de datos de ejemplo AdventureWorks2008, en el Explorador de objetos de SQL Server Management Studio, abra la carpeta **Database Triggers**, que se encuentra en la carpeta **Programmability** de la base de datos.

### **Eventos DLL**

Los eventos DDL se pueden utilizar para activar un desencadenador DDL o una notificación de eventos. Tenga en cuenta que cada evento corresponde a una instrucción o un procedimiento almacenado de Transact-SQL, con la sintaxis modificada para que incluya un carácter de subrayado ( ) entre las palabras clave.

Algunos ejemplos: CREATE\_DATABASE, CREATE\_TABLE, CREATE\_PROCEDURE, ALTER TABLE, DROP TABLE, etc.





#### Función EVENTDATA

La información acerca de un evento que activa un desencadenador DDL se captura mediante la función EVENTDATA. Esta función devuelve un valor **xml**. El esquema XML incluye información acerca de lo siguiente:

- La hora del evento.
- El ld. de proceso del sistema (SPID) de la conexión en la cual se ha ejecutado el desencadenador.
- El tipo de evento que ha activado el desencadenador.

En función del tipo de evento, el esquema incluirá información adicional, como la base de datos en la que se ha producido el evento, el objeto en el que se ha producido el evento y la instrucción Transact-SQL del evento.

Por ejemplo, el siguiente desencadenador DDL se crea en la base de datos de ejemplo AdventureWorks2008 :

```
CREATE TRIGGER CrearTabla
ON DATABASE
FOR CREATE_TABLE
AS
PRINT ' Temas de CREATE TABLE'
SELECT
EVENTDATA().value('(/EVENT_INSTANCE/TSQLCommand/CommandText)[1]','nvar char(max)')
RAISERROR ('No pueden crearse nuevas tablas en esta base de datos', 16, 1)
ROLLBACK;
```

Se ejecutará la siguiente instrucción CREATE TABLE:

```
CREATE TABLE NewTable (Column1 int);
```

La instrucción EVENTDATA() del desencadenador DDL captura el texto de la instrucción CREATE TABLE que no se admite. Esto se realiza utilizando una instrucción **XQuery** en los datos **xml** generados por EVENTDATA y recuperando el elemento <CommandText>.

Puede utilizar la función EVENTDATA para crear un registro de eventos. En el siguiente ejemplo, una tabla se crea para almacenar la información del evento. A continuación, se crea un desencadenador DDL en la base de datos actual que llena la tabla con la siguiente información siempre que tiene lugar un evento DDL en la base de datos:

- La hora del evento (mediante la función GETDATE).
- El usuario de la base de datos contra cuya sesión se ha producido el evento (mediante la función **CURRENT\_USER**).
- El tipo de evento.
- La instrucción Transact-SQL que contenía el evento.

Una vez más, los dos últimos elementos se capturan mediante **XQuery** con los datos **xml** generados por **EVENTDATA**.

```
CREATE TABLE ddl_log (PostTime datetime, DB_User nvarchar(100), Event nvarchar(100), TSQL nvarchar(2000));
GO

CREATE TRIGGER log
ON DATABASE
```



```
d
```

```
FOR DDL_DATABASE_LEVEL_EVENTS

AS

DECLARE @data XML

SET @data = EVENTDATA()

INSERT ddl_log (PostTime, DB_User, Event, TSQL)

VALUES (GETDATE(), CONVERT(nvarchar(100), CURRENT_USER),

@data.value('(/EVENT_INSTANCE/EventType)[1]', 'nvarchar(100)'),

@data.value('(/EVENT_INSTANCE/TSQLCommand)[1]', 'nvarchar(2000)'));

GO

-- Probar el desencadenador

CREATE TABLE TestTable (a int)

DROP TABLE TestTable;

GO

SELECT * FROM ddl_log;

GO
```

# d

# 5- Notificaciones de Eventos

### Introducción

Las notificaciones de eventos envían información acerca de los eventos al servicio **Service Broker**. Se pueden programar notificaciones de eventos para muchos de los eventos capturados por Traza de SQL, pero dichas notificaciones pueden utilizarse para realizar una acción en una instancia de SQL Server 2008 como respuesta a eventos, en lugar de utilizarse para crear trazas. Como las notificaciones de eventos se ejecutan asincrónicamente, no consumen los recursos definidos por la transacción inmediata.

# Conceptos Básicos de las Notificaciones de Eventos

Las notificaciones de eventos se ejecutan como respuesta a una variedad de instrucciones del lenguaje de definición de datos (DDL) Transact-SQL y eventos de Traza de SQL enviando información acerca de esos eventos a un servicio de **Service Broker**.

Las notificaciones de eventos se pueden usar para realizar lo siguiente:

- Registrar y revisar cambios o actividades que se producen en la base de datos.
- Realizar una acción en respuesta a un evento de una forma asincrónica en lugar de sincrónica.

Las notificaciones de eventos pueden ofrecer una alternativa de programación a los desencadenadores DDL y a la Traza SQL.

Las notificaciones de eventos se ejecutan asincrónicamente, fuera del alcance de una transacción. Por consiguiente, a diferencia de los desencadenadores DDL, las notificaciones de eventos se pueden usar dentro de una aplicación de bases de datos para responder a eventos sin utilizar los recursos definidos por la transacción inmediata.

A diferencia de la Traza de SQL, las notificaciones de eventos se pueden utilizar para realizar una acción en una instancia de SQL Server como respuesta a un evento de Traza de SQL.

Cuando se crea una notificación de eventos, se abren una o más conversaciones de **Service Broker** entre una instancia de SQL Server y el servicio de destino que se especifica. Normalmente, las conversaciones permanecen abiertas mientras existe la notificación de eventos como objeto de la instancia de servidores. En algunos casos de error, las conversaciones se pueden cerrar antes de que se quite la notificación de eventos. Esas conversaciones nunca se comparten entre notificaciones de eventos. Cada notificación de eventos tiene sus propias conversaciones exclusivas. Al finalizar una conversación explícitamente se impide que el servicio de destino reciba más mensajes y la conversación no se vuelve a abrir la próxima vez que se activa la notificación de eventos.

La información de eventos se proporciona a **Service Broker** como una variable de tipo **xml** que proporciona información acerca de cuándo se produce un evento, el objeto de la base de datos afectado, la instrucción de lote Transact-SQL implicada y otra información.

Los datos de eventos pueden ser utilizados por aplicaciones que se ejecutan junto con SQL Server para realizar un seguimiento del progreso y tomar decisiones.

Ejemplo: la siguiente notificación de eventos envía un aviso a un servicio determinado cada vez que se emite una instrucción ALTER TABLE en la base de datos de ejemplo AdventureWorks2008 :

```
USE AdventureWorks2008

GO

CREATE EVENT NOTIFICATION NotifyALTER_T1

ON DATABASE

FOR ALTER_TABLE

TO SERVICE '//Adventure-Works.com/ArchiveService',

'8140a771-3c4b-4479-8ac0-81008ab17984';
```





# Descripción de Notificaciones de Eventos frente a Desencadenadores

En la siguiente tabla, se comparan y se establecen diferencias entre los desencadenadores y las notificaciones de eventos.

DESENCADENADORES	NOTIFICACIONES DE EVENTOS
Los desencadenadores DML responden a eventos DML (de lenguaje de manipulación de datos). Los desencadenadores DDL responden a eventos DDL (de lenguaje de definición de datos).	Las notificaciones de eventos responden a eventos DDL y a un subconjunto de eventos de traza de SQL.
Los desencadenadores pueden ejecutar un código administrado de Transact-SQL o de Common Language Runtime (CLR).	Las notificaciones de eventos no ejecutan códigos. En cambio, envían mensajes <b>xml</b> a un servicio de Service Broker.
Los desencadenadores se procesan de manera sincrónica, dentro del ámbito de las transacciones que los accionan.	Las notificaciones de eventos se pueden procesar de manera asincrónica y no se ejecutan en el ámbito de las transacciones que las accionan.
El consumidor de un desencadenador está estrechamente unido al evento que acciona el desencadenador.	El consumidor de una notificación de eventos está desvinculado del evento que acciona la notificación.
Los desencadenadores se deben procesar en el servidor local.	Las notificaciones de eventos se pueden procesar en un servidor remoto.
Los desencadenadores se pueden revertir.	Las notificaciones de eventos no se pueden revertir.
Los nombres de los desencadenadores DML pertenecen al ámbito del esquema. Los nombres de los desencadenadores DDL pertenecen al ámbito de la base de datos o del servidor.	Los nombres de las notificaciones de eventos pertenecen al ámbito del servidor o de la base de datos. Las notificaciones de eventos de un evento QUEUE_ACTIVATION pertenecen al ámbito de una cola específica.
Los desencadenadores DML pertenecen al mismo propietario que el de las tablas a las que fueron aplicados.	El propietario de una notificación de eventos de una cola puede ser diferente del propietario del objeto al que fue aplicada.
Los desencadenadores admiten la cláusula EXECUTE AS.	Las notificaciones de eventos no admiten la cláusula EXECUTE AS.
La información del evento del desencadenador DDL se puede capturar con la función EVENTDATA, que devuelve un tipo de dato <b>xml</b> .	Las notificaciones de eventos envían información de evento <b>xml</b> a un servicio de Service Broker. La información utiliza el formato del mismo esquema que el de la función EVENTDATA.
Los metadatos acerca de los desencadenadores se encuentran en las vistas de catálogo sys.triggers y sys.server_triggers.	Los metadatos sobre las notificaciones de eventos se encuentran en las vistas de catálogo sys.event_notifications y sys.server_event_notifications.

# Notificaciones de Eventos frente a Traza de SQL

En la siguiente tabla se compara y contrasta el uso de notificaciones de eventos y de la Traza de SQL para supervisar eventos de servidor.

Traza de SQL	Notificaciones de eventos
Traza SQL Trace no genera carga de rendimiento asociada con transacciones. El empaquetado de los datos es eficaz.	Existe una carga de rendimiento asociada con la creación de datos de eventos con formato XML y con el envío de notificaciones de eventos.
Traza SQL puede supervisar y realizar un seguimiento de	Los notificaciones de eventos pueden supervisar un



cualquier clase de evento.	subconjunto de clases de eventos de seguimiento y también todos los eventos del lenguaje de definición de datos (DDL).
Puede personalizar qué columnas de datos se crean en un evento de seguimiento.	El esquema de datos de eventos con formato XML devuelto por las notificaciones de eventos es fijo.
Los eventos de traza generador por DDL siempre se generan, independientemente de si la instrucción DDL se revierte.	Las notificaciones de eventos no se activan si el evento de la instrucción DDL correspondiente se revierte.
La administración del flujo intermedio de los datos de eventos de traza implica llenar y administrar archivos de traza o tablas de traza.	La administración intermedia de los datos de notificación de eventos se consigue automáticamente mediante las colas de Service Broker.
Las trazas deben reiniciarse cada vez que se reinicia el servidor.	Después de registrarse, las notificaciones de eventos persisten en ciclos de servidor y participan en transacciones.
Tras reiniciarse, la activación de las trazas no se puede controlar. Las horas de detención y filtrado se pueden usar para especificar cuándo se inician. Se obtiene acceso a las trazas sondeando el archivo de trazas correspondiente.	Las notificaciones de eventos se pueden controlar utilizando la instrucción WAITFOR sobre la cola que recibe el mensaje generado por la notificación de eventos. Se puede obtener acceso a ellas sondeando la cola.
ALTER TRACE es el permiso mínimo necesario para crear una traza. También se requiere el permiso para crear un archivo de traza en el equipo correspondiente.	El permiso mínimo depende del tipo de notificación de eventos que se está creando. El permiso RECEIVE también es necesario en la cola correspondiente.
Las trazas se pueden recibir remotamente.	Las notificaciones de eventos se pueden recibir remotamente.
Los eventos de traza se implementan utilizando procedimientos almacenados del sistema.	Las notificaciones de eventos se implementan utilizando una combinación de Motor de base de datos de SQL Server y Service Broker, y de instrucciones Transact-SQL.
Se puede obtener acceso a los datos de eventos de traza mediante programación consultando la tabla de traza correspondiente, analizando el archivo de traza o utilizando la clase TraceReader de los objetos de administración de SQL Server (SMO).	Se obtiene acceso a los datos de eventos mediante programación emitiendo XQuery sobre los datos de eventos con formato XML, o utilizando las clases SMO Event.

# d

# 6- Auditoría SQL Server

#### **SQL Server Audit**

Una auditoría es la combinación de varios elementos en un único paquete para un grupo específico de acciones de servidor o de base de datos. Los componentes de **SQL Server Audit** se unen para producir una salida denominada auditoría, de la misma manera que una definición de informe combinada con gráficos y elementos de datos da como resultado un informe.

El objeto SQL Server Audit recopila una única instancia de acciones y grupos de acciones en el nivel de servidor o de base de datos para su supervisión. La auditoría se realiza en el nivel de instancia de SQL Server. Es posible tener varias auditorías por cada instancia de SQL Server. Cuando se define una auditoría, se especifica la ubicación para los resultados generados. Éste es el destino de la auditoría. La auditoría se crea en un estado deshabilitado y no audita automáticamente ninguna acción. Una vez habilitada la auditoría, el destino de la auditoría recibe los datos de la misma.

Los resultados de una auditoría se envían a un destino, que puede ser un archivo, el registro de eventos de seguridad de Windows o el registro de eventos de aplicación de Windows. (La operación de escribir en el registro de seguridad no está disponible en Windows XP). Los registros se deben revisar y archivar periódicamente para asegurar que el destino tiene espacio suficiente para escribir registros adicionales.

Puede utilizar SQL Server Management Studio o Transact-SQL para definir una auditoría. Una vez creada y habilitada la auditoría, el destino comenzará a recibir entradas.

Puede leer los registros de eventos de Windows mediante la utilidad **Visor de eventos en Windows**. Para los destinos de archivo, puede utilizar tanto el **Visor del archivo de registros** en SQL Server Management Studio como la función **fn\_get\_audit\_file** para leer el archivo de destino.

El proceso general de creación y uso de una auditoría es el siguiente:

- Cree una auditoría y defina el destino.
- Cree una especificación de auditoría de servidor o una especificación de auditoría de base de datos que se asigne a la auditoría. Habilite la especificación de auditoría.
- Habilite la auditoría.
- Lea los eventos de auditoría mediante el Visor de eventos de Windows, el Visor del archivo de registros o la función fn\_get\_audit\_file.

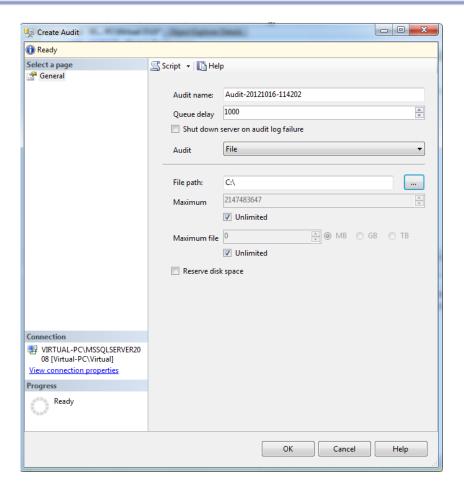
# Crear un objeto de Auditoría

Dentro del Management Studio, desplegar la solapa **Seguridad** y presionar botón derecho sobre la solapa **Audits**. Puede usar también la sentencia Transact-SQL **CREATE SERVER AUDIT** 

# Ejemplo:

```
CREATE SERVER AUDIT Prueba_Audit
TO APPLICATION_LOG
WITH ( QUEUE_DELAY = 1000, ON_FAILURE = SHUTDOWN);
```





### Crear una especificación de Auditoría

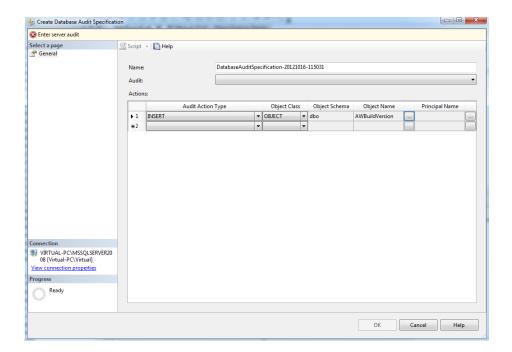
Una vez creado es necesario crear una especificación de Servidor o de Base de Datos (Server or Database Audit Specification). Dentro del Management Studio, seleccionar la solapa Seguridad de alguna base de datos, desplegarla y presionar botón derecho en New DataBase Audit Specification. Para crear una especificación de servidor, seleccionar la opción New Server Audit Especification en la misma solapa de la opción Audits.

También puede usar las sentencias CREATE SERVER AUDIT SPECIFICATION o CREATE DATABASE AUDIT

```
CREATE SERVER AUDIT SPECIFICATION HIPPA_Audit_Specification
FOR SERVER AUDIT HIPPA_Audit
ADD (FAILED_LOGIN_GROUP);

CREATE DATABASE AUDIT SPECIFICATION AWSensitiveDMLAudit
FOR SERVER AUDIT AWDMLAudit
ADD (SELECT , INSERT , UPDATE, DELETE
ON HumanResources.EmployeePayHistory
BY PUBLIC),
ADD (SELECT , INSERT , UPDATE, DELETE
ON Sales.CreditCard
BY PUBLIC)
WITH (STATE = ON)
GO
```





# Habilitación de Auditorías

Una vez creados la auditoría y la especificación hay que habilitarlas. Para ellos presionar botón derecho sobre el nombre del objeto de auditoría y seleccionar **Enable Audit**. Luego, de la misma forma, habilitar la especificación.

# Leer los eventos

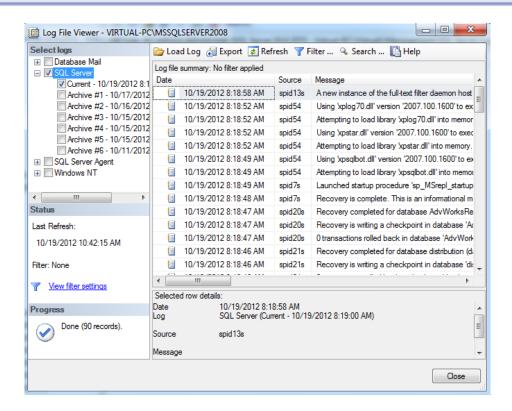
Para abrir el Visor del archivo de registros, expandir la solapa **Management** (Administración) dentro del Management Studio, presionar botón derecho sobre **SQL Server Logs**, seleccionar **View** (Ver)

También puede usar la función fn\_get\_audit\_file.

SQL también provee varias vistas dinámicas para trabajar con auditoria: sys.server\_audits, sys.server\_audit\_specifications, Sys.database\_audit\_specifications, etc.

```
SELECT * FROM sys.fn_get_audit_file
('c:\Prueba Audit',default,default);
```





# Módulo 5

# Automatizando Tareas Administrativas



# 1- Automatizar las tareas administrativas (Agente SQL Server)

Microsoft SQL Server le permite automatizar las tareas administrativas. Para automatizar la administración, se definen las tareas administrativas previsibles y, después, se especifican las condiciones en las que se produce cada tarea. El uso de la administración automatizada para controlar las tareas y eventos habituales le permite disponer de tiempo para realizar otras funciones administrativas.

# Herramientas para Automatizar la Administración

SQL Server incluye las siguientes herramientas para ayudarle a automatizar la administración:

- SQL Server Management Studio: Puede utilizar SQL Server Management Studio para automatizar la administración mediante la creación de trabajos, alertas, operadores y servidores proxy en el nodo Agente SQL Server del Explorador de objetos.
- Asistente para planes de mantenimiento: El Asistente para planes de mantenimiento es una utilidad que puede ayudarle a crear trabajos, alertas y operadores para automatizar una instancia de SQL Server. Le ayuda a configurar las tareas de mantenimiento principales para asegurarse de que la base de datos funciona bien, se realiza una copia de seguridad regular de la misma y no tiene incoherencias. El Asistente para planes de mantenimiento crea uno o varios trabajos del Agente SQL Server que realizan estas tareas en servidores locales o en servidores de destino en un entorno multiservidor. La ejecución puede tener lugar a intervalos programados o a petición.

#### Planes de Mantenimiento

Para crear o administrar planes de mantenimiento, debe ser miembro de la función fija de servidor **sysadmin**. Tenga en cuenta que el Explorador de objetos sólo muestra planes de mantenimiento si el usuario es miembro de dicha función fija.

Si desea crear o administrar planes de mantenimiento en un entorno de varios servidores, necesitará una configuración adicional.

Los planes de mantenimiento se pueden crear para realizar las tareas siguientes:

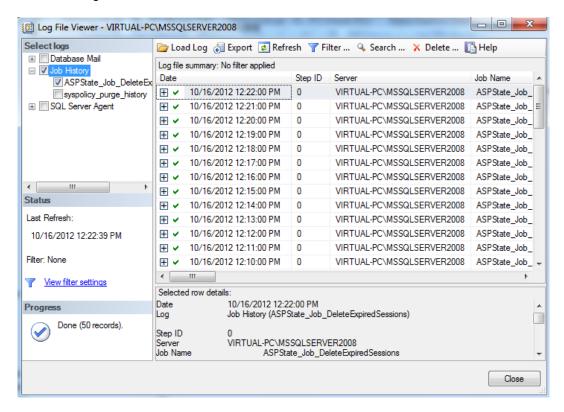
- Reorganizar los datos de las páginas de datos y de índices mediante una nueva generación de los índices con un nuevo factor de relleno. Al volver a crear índices con un nuevo factor de relleno se asegura que las páginas de la base de datos contienen una cantidad de datos y espacio libre distribuidos por igual. También permite un crecimiento más rápido en el futuro.
- Comprimir archivos de datos mediante la eliminación de las páginas de base de datos que estén vacías.
- Actualizar las estadísticas de los índices para asegurarse de que el optimizador de consultas dispone de información actualizada acerca de la distribución de los valores de los datos en las tablas. Esto permite al optimizador de consultas elegir el método más adecuado para obtener acceso a los datos, ya que dispone de más información acerca de los datos almacenados en la base de datos. Aunque SQL Server actualiza periódicamente las estadísticas de los índices de forma automática, esta opción puede obligar a que se actualicen inmediatamente.
- Realizar comprobaciones de coherencia interna de los datos y de las páginas de datos de la base de datos para asegurarse de que no se han dañado debido a un problema de software o del sistema.
- Realizar copias de seguridad de la base de datos y de los archivos de registro de transacciones. Las copias de seguridad de la base de datos y del registro pueden mantenerse durante un período especificado. Esto le permite crear un historial de copias de seguridad para utilizarlo si tiene que restaurar la base de datos a una fecha





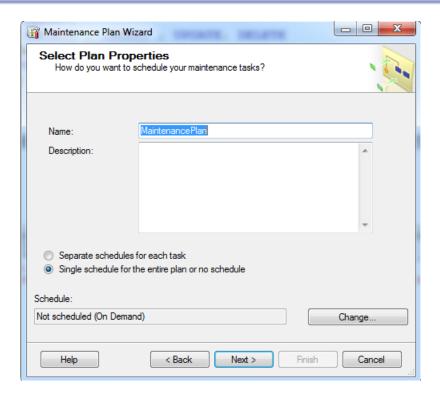
- anterior a la de la última copia de seguridad de la base de datos. También puede realizar copias de seguridad diferenciales.
- Ejecutar trabajos del Agente SQL Server. Esta tarea se puede utilizar para crear trabajos que realicen una serie de acciones y, también, para crear los planes de mantenimiento para ejecutar los trabajos.

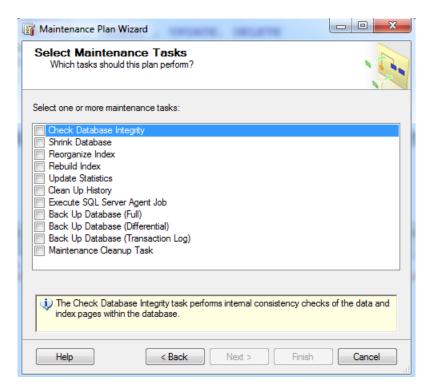
Los resultados generados por las tareas de mantenimiento pueden escribirse en forma de informe en un archivo de texto, o bien escribirse en las tablas del plan de mantenimiento, sysmaintplan\_log y sysmaintplan\_logdetail, en msdb. Para ver los resultados en el visor del archivo de registros, haga clic con el botón secundario en Planes de mantenimiento y, a continuación, haga clic en Ver historial.



# Iniciar el Asistente para Planes de Mantenimiento (SQL Server Management Studio) Para iniciar el Asistente para planes de mantenimiento:

- Expanda el servidor.
- Expanda la carpeta Administración.
- Haga clic con el botón secundario en Planes de mantenimiento y seleccione
   Asistente para planes de mantenimiento. Esto inicia el asistente y podrá seguir los pasos y crear un plan personalizado para cumplir sus requisitos de mantenimiento.





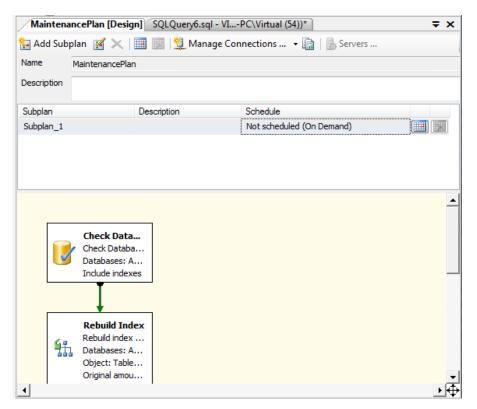
# Editar y crear Planes de Mantenimiento Manualmente

Usted puede editar un plan existente con la herramienta **Diseñador de Plan de Mantenimiento** disponible dentro de SQL Server Management Studio.





Esta herramienta provee una interfaz gráfica permitiéndole reordenar tareas, agregar nuevas tareas y definir un flujo de trabajo, indicando cómo manejar el éxito o la falla de tareas.



# Agente SQL Server (SQL Server Agent)

El Agente SQL Server es un servicio de Microsoft Windows que ejecuta tareas administrativas programadas, denominadas **trabajos (Jobs)**. El Agente SQL Server utiliza SQL Server para almacenar información de los trabajos. Los trabajos contienen uno o más pasos. Cada paso contiene su propia tarea; por ejemplo, realizar una copia de seguridad de una base de datos. El Agente SQL Server puede ejecutar un trabajo según una programación, como respuesta a un evento específico o a petición. Por ejemplo, si desea realizar una copia de seguridad de todos los servidores de la organización todos los días entre semana después del horario de trabajo, puede automatizar esta tarea. Programe la copia de seguridad para que se ejecute después de las 22:00 h de lunes a viernes; si la copia de seguridad encuentra un problema, el Agente SQL Server puede registrar el evento y notificárselo.

**Nota:** De forma predeterminada, el servicio Agente SQL Server está deshabilitado al instalar SQL Server 2008, a menos que el usuario elija de forma explícita iniciarlo automáticamente.

Para automatizar la administración, siga estos pasos:

- Establezca las tareas administrativas o eventos del servidor que se realizan con regularidad y si estas tareas o eventos se pueden administrar mediante programación. Una tarea es una buena candidata a la automatización si consta de una secuencia de pasos predecible y se produce en un momento específico o en respuesta a un evento concreto.
- Defina un conjunto de trabajos, programaciones, alertas y operadores usando SQL Server Management Studio, secuencias de comandos Transact-SQL u objetos de administración de SQL Server (SMO).



Ejecute los trabajos del Agente SQL Server que haya definido.

Si ejecuta varias instancias de SQL Server, utilice la administración multiservidor para automatizar las tareas comunes a todas las instancias.

# Componentes

El Agente SQL Server utiliza los siguientes componentes para definir las tareas que se realizarán, cuándo se llevarán a cabo y cómo se informará de si se han realizado correctamente o no. El Agente SQL Server también proporciona seguridad para la administración automática.

- Trabajos: Un trabajo es una serie especificada de acciones que realiza el Agente SQL Server. Utilice los trabajos para definir tareas administrativas de manera que se ejecuten una o más veces, y se pueda supervisar si se realizan o no correctamente. Un trabajo se puede ejecutar en un servidor local o en varios servidores remotos. Existen varias maneras de ejecutar trabajos:
  - Conforme a una o más programaciones.
  - Como respuesta a una o varias alertas.
  - Ejecutando el procedimiento almacenado sp start job.

Cada acción de un trabajo es un **paso de trabajo**. Por ejemplo, un paso de trabajo puede consistir en la ejecución de una instrucción Transact-SQL, la ejecución de un paquete SSIS o la emisión de un comando en un servidor de Analysis Services. Los pasos de trabajo se administran como parte de un trabajo.

Cada paso se ejecuta en un contexto de seguridad específico. En el caso de los pasos de trabajo que utilizan Transact-SQL, use la instrucción **EXECUTE AS** para establecer el contexto de seguridad para éstos. Para los demás tipos de pasos de trabajo, utilice una cuenta de proxy para establecer el contexto de seguridad.

- **Programaciones**: Una programación específica cuándo se ejecuta un trabajo. Se puede ejecutar más de un trabajo en la misma programación y se puede aplicar más de una programación al mismo trabajo. Una programación puede definir las condiciones siguientes del momento en el que se ejecuta un trabajo:
  - Cuando se inicia el Agente SQL Server.
  - Cuando el uso de la CPU del equipo se encuentre en un nivel que se haya definido como inactivo.
  - Una vez, a una hora y una fecha específicas.
  - Periódicamente.
- Alertas: Una alerta es una respuesta automática a un evento específico. Por ejemplo, un evento puede ser el inicio de un trabajo o que los recursos del sistema alcancen un umbral específico. Debe definir las condiciones en las que se genera una alerta. Una alerta puede responder a una de las condiciones siguientes:
  - Eventos de SQL Server
  - Condiciones de rendimiento de SQL Server
  - Eventos de Instrumental de administración de Microsoft Windows (WMI) en el equipo en el que se ejecuta el Agente SQL Server

Una alerta puede realizar las acciones siguientes:

- Notificar a uno o varios operadores
- Ejecutar un trabajo
- Operadores: Los operadores definen información de contacto para las personas responsables del mantenimiento de una o varias instancias de SQL Server. En algunas compañías, las responsabilidades de operador están asignadas a una sola persona. En compañías con varios servidores, muchas personas comparten las responsabilidades de operador. Un operador no contiene información de seguridad y no define una entidad de seguridad.







SQL Server puede notificar a los operadores de alertas mediante una o varias de las opciones siguientes:

- Correo electrónico
- Localizador (por correo electrónico)
- NET SEND

**Nota:** Para enviar notificaciones mediante **NET SEND**, se debe iniciar el servicio Windows Messenger en el equipo en el que reside el Agente SQL Server.

Para enviar a los operadores notificaciones por correo electrónico o localizador, deberá configurar el Agente SQL Server para utilizar Correo electrónico de base de datos o SQL Mail. Puede definir un operador como alias de un grupo de personas. De esta manera, todos los miembros de este alias pueden recibir notificaciones al mismo tiempo.

# 2- Configurar el Agente SQL Server

Puede especificar algunas opciones de configuración para el Agente SQL Server durante la instalación de SQL Server. El conjunto completo de opciones de configuración del Agente SQL Server sólo está disponible en SQL Server Management Studio, SMO (objetos de administración de SQL Server) o los procedimientos almacenados del Agente SQL Server.

**Nota:** Haga clic en el **Agente SQL Server** en el Explorador de objetos de SQL Server Management Studio para administrar trabajos, operadores, alertas y el servicio del Agente SQL Server. No obstante, el Explorador de objetos sólo muestra el nodo del Agente SQL Server si tiene permiso para utilizarlo.

El Agente SQL Server almacena la mayor parte de la información de configuración en las tablas que residen en la base de datos **msdb**. El Agente SQL Server utiliza los objetos de credenciales de SQL Server para almacenar la información de autenticación para los servidores proxy.

### **Establecer los Permisos Necesarios**

Para realizar sus funciones, el Agente SQL Server debe configurarse de modo que utilice las credenciales de una cuenta que sea miembro de la función fija de servidor **sysadmin** en SQL Server.

# Correo del Agente SQL Server

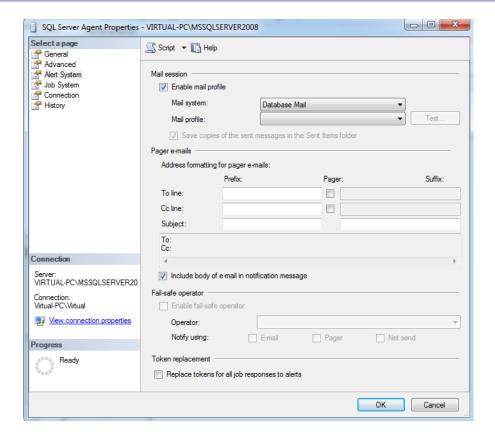
El Agente SQL Server incluye la posibilidad de enviar correo electrónico. Puede configurar el correo del Agente SQL Server para enviar mensajes de correo electrónico a operadores predefinidos cuando:

- Se desencadene una alerta. Las alertas se pueden configurar para enviar una notificación por correo electrónico acerca de eventos específicos que se produzcan. Por ejemplo, es posible configurar alertas para que avisen a un operador acerca de un determinado evento de la base de datos o una situación del sistema operativo que precise una acción inmediata.
- Se lleve a cabo o no se complete una tarea programada, como una copia de seguridad de la base de datos o un evento de réplica. Por ejemplo, puede usar el correo del Agente SQL Server para notificar a los operadores si se produce un error durante el procesamiento a fin de mes.

Es posible enviar mensajes de correo electrónico a una lista de destinatarios para informarles del estado de los trabajos programados, lo que les permitiría emprender una acción. Por ejemplo, puede configurar el correo del Agente SQL Server para enviar mensajes de correo electrónico cuando termine de hacerse una copia de seguridad.

De forma predeterminada, el correo del Agente SQL Server está desactivado. Para configurarlo, utilice el panel **Sistema de alerta** del cuadro de diálogo **Propiedades** de **Agente SQL Server**. Tenga en cuenta que el correo del Agente SQL Server sólo es necesario para la notificación de alertas y la notificación automática cuando se completa un trabajo. Cada paso de un trabajo puede enviar un mensaje de correo electrónico, independientemente de si el correo del Agente SQL Server está activado. Por ejemplo, un paso de trabajo de Transact-SQL puede usar el Correo electrónico de base de datos para enviar el resultado de una consulta a la lista de destinatarios.





#### Correo Electrónico de Base de Datos (Database Mail)

El Correo electrónico de base de datos permite a las aplicaciones de base de datos enviar mensajes de correo electrónico a los usuarios. Los mensajes enviados pueden incluir resultados de consultas y archivos de cualquier recurso de la red. El Correo electrónico de base de datos está diseñado para proporcionar confiabilidad, escalabilidad, seguridad y compatibilidad.

### Características:

- No se requiere Microsoft Outlook ni MAPI (Interfaz de programación de aplicaciones de mensajería) extendida. El Correo electrónico de base de datos utiliza el protocolo estándar SMTP (Protocolo simple de transferencia de correo) para enviar correo electrónico.
- Aislamiento de procesos. Para minimizar el impacto en SQL Server, el componente que entrega el correo electrónico se ejecuta fuera de SQL Server, en un proceso independiente. SQL Server continuará almacenando en cola los mensajes de correo electrónico incluso si el proceso externo se detiene o genera un error. Los mensajes en cola se enviarán cuando el proceso externo o el servidor SMTP se encuentren en línea.
- Cuentas de conmutación por error. Los perfiles del Correo electrónico de base de datos permiten especificar más de un servidor SMTP. Si un servidor SMTP no está disponible, se puede enviar el correo mediante otro.
- Compatibilidad con clústeres. El Correo electrónico de base de datos es una aplicación para clústeres y es totalmente compatible con éstos
- Varios perfiles. Permite crear varios perfiles en una instancia de SQL Server. También se puede seleccionar el perfil del Correo electrónico de base de datos para enviar el mensaje.



- Varias cuentas. Cada perfil puede incluir varias cuentas de conmutación por error. Se pueden configurar varios perfiles con distintas cuentas para distribuir el correo electrónico entre varios servidores de correo.
- Compatibilidad con 64 bits. Es totalmente compatible con las versiones de 64 bits de SQL Server.
- Para enviar correo electrónico de base de datos, debe pertenecer a la función de la base de datos DatabaseMailUserRole en la base de datos msdb.
- Registro. Registra la actividad de correo electrónico en SQL Server, en el registro de eventos de aplicación de Microsoft Windows y en la base de datos msdb.
- Auditoría. Conserva copias de los mensajes y datos adjuntos enviados en la base de datos msdb. Puede auditar fácilmente la utilización del Correo electrónico de base de datos y revisar los mensajes conservados.

#### SQL Mail

SQL Mail utiliza componentes de cliente de **MAPI** extendida de una aplicación de correo electrónico externa (por ejemplo, **Microsoft Outlook**) para enviar y recibir correo electrónico. Por lo tanto, para utilizar SQL Mail, debe instalar una aplicación de correo electrónico compatible con MAPI extendida en el equipo que ejecute SQL Server. SQL Server utiliza los componentes de MAPI extendida proporcionados por la aplicación de correo electrónico para comunicarse con el servidor de correo electrónico.

Esta característica se quitará en una versión futura de Microsoft SQL Server.



# 3- Trabajos y Operadores

# **Trabajos**

Un trabajo es una serie específica de operaciones que el Agente SQL Server realiza secuencialmente. Un trabajo puede realizar una amplia variedad de actividades, incluidas secuencias de comandos Transact-SQL, aplicaciones de símbolo del sistema, secuencias de comandos de Microsoft ActiveX, paquetes de Integration Services, comandos y consultas de Analysis Services o tareas de réplica. Los trabajos pueden ejecutar tareas repetitivas o que se pueden programar, y pueden notificar automáticamente a los usuarios el estado del trabajo mediante alertas, lo cual simplifica en gran medida la administración de SQL Server.

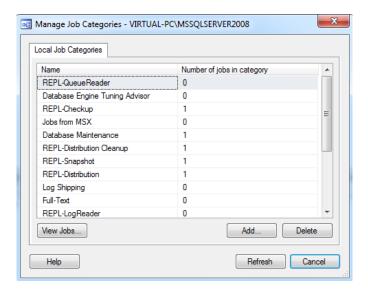
Para crear un trabajo, el usuario debe ser miembro de una de las funciones fijas de base de datos del Agente SQL Server o de la función fija de servidor **sysadmin**. Sólo pueden editar el trabajo el propietario de éste o los miembros de la función **sysadmin**.

Se puede escribir un trabajo para que se ejecute en la instancia local de SQL Server o en varias instancias de una empresa. Para ejecutar trabajos en varios servidores, debe configurar al menos un servidor principal, y uno o más servidores de destino.

# Organizar trabajos

Las categorías de trabajo le ayudan a organizar los trabajos para poder filtrarlos y agruparlos fácilmente. Por ejemplo, puede organizar todos los trabajos de copia de seguridad de las bases de datos en la categoría Mantenimiento de bases de datos. También puede crear sus propias categorías.

Las categorías multiservidor existen sólo en los servidores principales. Sólo hay una categoría de trabajo predeterminada disponible en un servidor principal: [Sin categoría (Multiservidor)]. Cuando se descarga un trabajo multiservidor, su categoría se cambia a Trabajos del servidor principal en el servidor de destino.



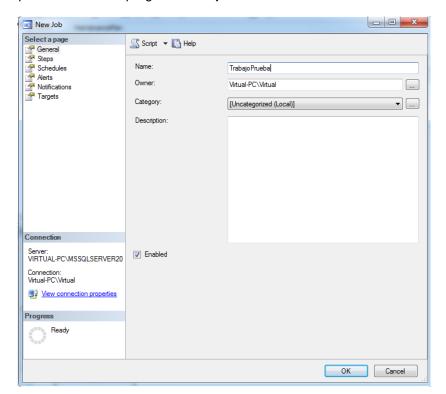
# Propiedad de trabajos

Por razones de seguridad, sólo puede cambiar la definición del trabajo el propietario de éste o un miembro de la función **sysadmin**. Los miembros de la función **sysadmin** pueden asignar el valor de propiedad de trabajo a otros usuarios, y pueden ejecutar cualquier trabajo, independientemente del propietario del trabajo.



# Crear un trabajo

- En el Explorador de objetos, conéctese a una sesión del Motor de base de datos de SQL Server y expándala.
- Expanda Agente SQL Server.
- Haga clic con el botón secundario en Trabajos (job) y, a continuación, haga clic en Nuevo trabajo.
- En la página General, en el cuadro Nombre, escriba un nombre para el trabajo.
- Desactive la casilla de verificación Habilitado si no desea que el trabajo se ejecute inmediatamente después de su creación. Por ejemplo, deshabilite un trabajo si desea probarlo antes de programar su ejecución.



Para agregar pasos de trabajo, programas, alertas y notificaciones que puedan enviarse a los operadores, seleccione las diferentes solapas de esta opción.

### Crear pasos de trabajo

Los pasos de trabajo son acciones que el trabajo realiza en una base de datos o en un servidor. Cada trabajo debe estar formado por un paso, como mínimo. Los pasos de trabajo pueden ser:

- Programas ejecutables y comandos del sistema operativo.
- Instrucciones Transact-SQL, incluidos los procedimientos almacenados y los procedimientos almacenados extendidos.
- Secuencias de comandos Microsoft ActiveX.
- Tareas de réplica.
- Tareas de Analysis Services.
- Paquetes de Integration Services.

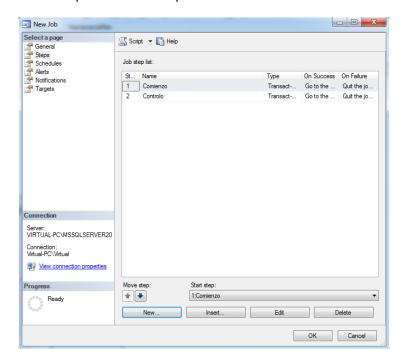
Todos los pasos de trabajo se ejecutan en un contexto de seguridad determinado. Si en el paso de trabajo se especifica un **proxy**, se ejecuta en el contexto de seguridad de la credencial del



æ

proxy. Si en el paso de trabajo no se especifica un proxy, se ejecuta en el contexto de la cuenta de servicio del Agente SQL Server. Sólo los miembros de la función de servidor fija **sysadmin** pueden crear trabajos en los que no se especifique un proxy de forma explícita.

Puesto que los pasos de trabajo se ejecutan en el contexto de un usuario específico de Microsoft Windows, dicho usuario debe disponer de los permisos y la configuración necesarios para que se ejecute el paso de trabajo. Por ejemplo, si crea un trabajo que requiere una letra de unidad o una ruta de acceso UNC (Convención de nomenclatura universal), los pasos de trabajo se pueden ejecutar con la cuenta de usuario de Microsoft Windows durante la comprobación de las tareas. Sin embargo, el usuario de Windows para el paso de trabajo debe tener también los permisos y configuraciones de letra de unidad necesaria, o acceso a la unidad requerida. De lo contrario, se producirá un error en el paso de trabajo. Para evitar este problema, asegúrese de que el proxy para cada paso de trabajo dispone de los permisos necesarios para la tarea que realiza dicho paso.



# Programas Ejecutables y Comandos del Sistema Operativo como Pasos de Trabajo

Los programas ejecutables y comandos del sistema operativo se pueden utilizar como pasos de trabajo. Los archivos pueden tener las extensiones .bat, .cmd, .com o .exe.

Si utiliza un programa ejecutable o un comando del sistema operativo como paso de trabajo, debe especificar:

- El código de salida del proceso que se devuelve si el comando se ha ejecutado correctamente.
- El comando que se debe ejecutar. Para ejecutar un comando del sistema operativo, se trata simplemente del propio comando.
- En un programa externo, es el nombre del programa y los argumentos para el programa, por ejemplo:
  - C:\Archivos de programa\Microsoft SQL Server\90\Tools\Binn\sqlcmd.exe -e -q "sp\_who"

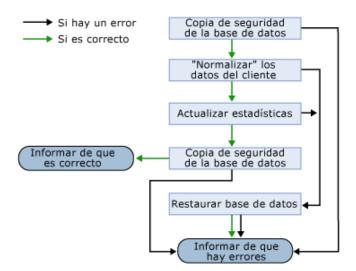
ai.

**Nota:** Debe proporcionar la ruta de acceso completa del archivo ejecutable si éste no se encuentra en un directorio especificado en la ruta de acceso del sistema o la ruta de acceso del usuario con el que se ejecuta el paso de trabajo.

# Controlar varios Pasos del Trabajo

Si el trabajo está formado por varios pasos, debe especificar el orden de ejecución de los pasos del trabajo. Esto se denomina control de flujo. En cualquier momento puede agregar nuevos pasos del trabajo y reorganizar el flujo de los pasos; los cambios se aplicarán la próxima vez que se ejecute el trabajo.

Esta ilustración muestra el control de flujo de un trabajo de copia de seguridad de una base de datos.



El primer paso es Copia de seguridad de la base de datos. Si este paso genera un error, el Agente SQL Server informa del error al operador que se ha definido que reciba la notificación. Si el paso Copia de seguridad de la base de datos es correcto, el trabajo pasa al siguiente paso: "Normalizar" los datos del cliente. Si este paso genera un error, el Agente SQL Server avanza a Restaurar base de datos. Si "Normalizar" los datos del cliente es correcto, el trabajo avanza al siguiente paso: Actualizar estadísticas y, así sucesivamente, hasta que el paso final da como resultado un informe correcto o con errores.

Se define una acción de control de flujo para la ejecución satisfactoria o con errores de cada paso del trabajo. Debe especificar una acción que se deberá realizar cuando un paso del trabajo se ejecute correctamente y la acción que se llevará a cabo cuando se ejecute con errores. También puede definir el número de reintentos y el intervalo entre ellos para los pasos del trabajo que no se han ejecutado correctamente.

Los pasos del trabajo deben ser independientes. Es decir, un trabajo no puede pasar datos, valores booleanos o numéricos entre pasos del trabajo. Sin embargo, puede pasar valores de un paso del trabajo de Transact-SQL a otro si utiliza tablas permanentes o tablas temporales globales. También puede pasar valores de pasos del trabajo que ejecuten programas ejecutables de un paso del trabajo a otro mediante archivos. Por ejemplo, el programa ejecutado mediante un paso del trabajo escribe un archivo y el programa ejecutado por un paso del trabajo posterior lee el archivo.

# Programar la Ejecución de un Trabajo

La programación de trabajos del Agente SQL Server consiste en definir las condiciones que provocan el inicio de la ejecución de los trabajos sin intervención del usuario. Puede programar que un trabajo se ejecute automáticamente creando una nueva programación para el trabajo, o adjuntando una programación existente al trabajo.

Hay dos maneras de crear una programación:

- Crear la programación mientras se está creando un trabajo.
- Crear la programación en el Explorador de objetos.

Una vez creada una programación, puede adjuntarla a varios trabajos, aun cuando la programación se haya creado para un trabajo concreto. También puede separar las programaciones de los trabajos.

Una programación puede basarse en tiempo o en un evento. Por ejemplo, puede programar un trabajo para que se ejecute en los momentos siguientes:

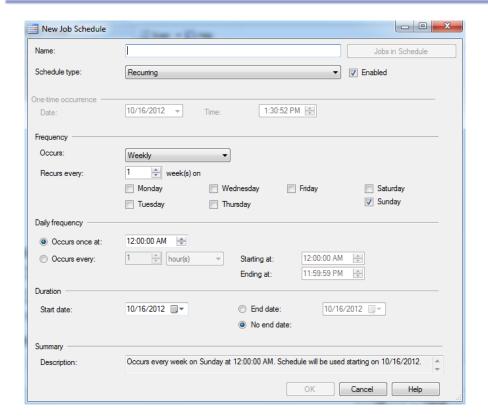
- Cuando se inicia el Agente SQL Server.
- Cuando el uso de la CPU del equipo se encuentre en un nivel que se haya definido como inactivo.
- Una vez, a una hora y una fecha específicas.
- Periódicamente.

Como alternativa a las programaciones de trabajo, también puede crear una alerta que responda a un evento ejecutando un trabajo.

**Nota**: Sólo se puede ejecutar una instancia del trabajo cada vez. Si intenta ejecutar un trabajo manualmente mientras se está ejecutando en el momento programado, el Agente SQL Server rechazará la petición.

Para impedir que un trabajo programado se ejecute, debe realizar una de las siguientes acciones:

- Deshabilitar la programación.
- Deshabilitar el trabajo.
- Separar la programación del trabajo.
- Detener el servicio del Agente SQL Server.
- Eliminar la programación.



#### **Operadores**

El servicio Agente SQL Server admite la notificación de administradores a través de operadores. Los operadores son alias para personas o grupos que pueden recibir una notificación electrónica cuando los trabajos finalizan o se activa una alerta. Los atributos principales de un operador son:

- Nombre del operador
- Información de contacto

Se recomienda definir operadores antes de definir alertas.

Cada operador debe tener asignado un nombre. Los nombres de los operadores deben ser únicos en la instancia de SQL Server y no pueden tener más de 128 caracteres.

#### Información de Contacto

La información de contacto de un operador define cómo se va a notificar a dicho operador. Se puede notificar a los operadores mediante correo electrónico, localizador o el comando **NET SEND**:

- Notificación mediante correo electrónico: La notificación por correo electrónico envía un mensaje de correo electrónico al operador. Para la notificación por correo electrónico debe proporcionar una dirección de correo electrónico al operador.
- Notificación mediante localizador: Este tipo de notificación se implementa mediante el correo electrónico. Para la notificación por localizador debe proporcionar una dirección de correo electrónico en la que el operador recibirá los mensajes del localizador. Para establecer la notificación mediante localizador, debe instalar en el servidor de correo un software que procese el correo entrante y lo convierta en mensajes de localizador. Este software es propiedad de los proveedores de servicios de localización.

 Notificación mediante NET SEND: Envía un mensaje al operador mediante el comando NET SEND. Si utiliza este comando, especifique el destinatario (el equipo o el usuario) de un mensaje de red.

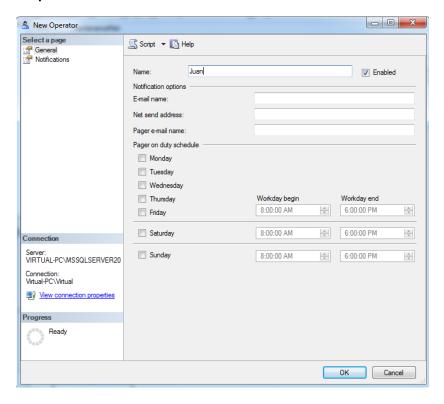
**Nota:** El comando net send utiliza **Microsoft Windows Messenger**. Para enviar alertas correctamente, este servicio debe ejecutarse tanto en el equipo en el que SQL Server se está ejecutando como en el equipo que utiliza el operador.

# Requisitos para Notificar a un Operador

Debe configurar al menos uno de los elementos siguientes para poder notificar a un operador:

- Para enviar un mensaje de correo electrónico mediante la funcionalidad Correo electrónico de base de datos, debe tener acceso a un servidor de correo electrónico que admita SMTP.
- Para enviar un mensaje de correo electrónico mediante la funcionalidad SQL Mail (MAPI extendido), debe tener acceso a un servidor Microsoft Exchange e instalar el cliente Microsoft Outlook y Microsoft Exchange en el equipo en el que se ejecuta SQL Server.
- Para notificar mediante un localizador, debe disponer de hardware o software de otros fabricantes para enviar mensajes de localizador a correo electrónico.
- Para utilizar NET SEND, el operador debe haber iniciado sesión en el equipo especificado y el equipo especificado debe permitir la recepción de mensajes desde Windows Messenger.

#### Crear un Operador



#### Supervisar la Actividad de Trabajo

Para supervisar la actividad de los trabajos puede usar las siguientes herramientas



- Sesiones del Agente SQL Server
- Monitor de Actividades de Trabajo

# Sesiones del Agente SQL Server

El Agente SQL Server crea una sesión cada vez que se inicia el servicio. Al crear una sesión, la tabla **sysjobactivity** de la base de datos **msdb** se rellena con todos los trabajos definidos existentes. Esta tabla mantiene la última actividad para los trabajos cuando se reinicia el Agente SQL Server. Cada sesión registra la actividad de trabajo normal del Agente SQL Server desde el inicio del trabajo hasta que termina. La información acerca de estas sesiones se almacena en la tabla **syssessions** de la base de datos **msdb**.

### Monitor de Actividad de Trabajo

Ver información en el capítulo anterior

# 4- Alertas

#### Alertas

Microsoft SQL Server genera eventos que se incluyen en el registro de aplicación de Microsoft Windows. El Agente SQL Server lee el registro de aplicación y compara los eventos con las alertas definidas. Cuando el Agente SQL Server encuentra una coincidencia, activa una alerta, que es una respuesta automatizada a un evento. Además de supervisar los eventos de SQL Server, el Agente SQL Server también puede supervisar las condiciones de rendimiento y los eventos de Instrumental de Administración de Windows (WMI).

Para definir una alerta, debe especificar:

- El nombre de la alerta.
- El evento o condición de rendimiento que desencadena la alerta.
- La acción que el Agente SQL Server realiza como respuesta al evento o condición de rendimiento.

Cada alerta debe tener un nombre. Los nombres de las alertas deben ser exclusivos en la instancia de SQL Server y no pueden tener más de 128 caracteres.

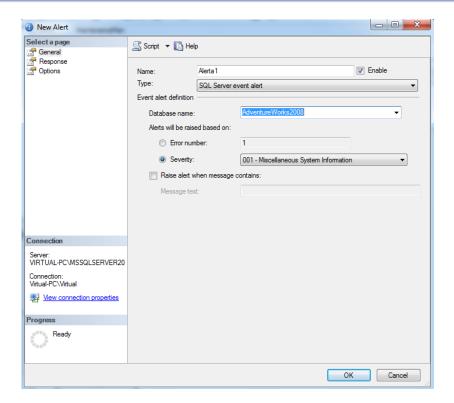
#### **Eventos**

Una alerta responde a un tipo de evento específico. Las alertas responden a los siguientes tipos de evento:

- Eventos de SQL Server
- Condiciones de rendimiento de SQL Server
- Eventos de WMI

El tipo de evento determina los parámetros que se utilizan para especificar el evento preciso. Puede especificar una alerta para que se produzca en respuesta a uno o más eventos. Utilice los siguientes parámetros para especificar los eventos que desencadenan una alerta:

- Número de error: El Agente SQL Server activa una alerta cuando se produce un error específico. Por ejemplo, puede especificar el número de error 2571 para responder a los intentos no autorizados de invocar comandos de consola de base de datos (DBCC).
- **Nivel de gravedad**: El Agente SQL Server activa una alerta cuando se produce un error de la gravedad específica. Por ejemplo, puede específicar el nivel de gravedad 15 para responder a errores de sintaxis en instrucciones Transact-SQL.
- Base de datos: El Agente SQL Server sólo activa una alerta cuando el evento tiene lugar en una base de datos determinada. Esta opción se aplica además del número de error o el nivel de gravedad.
- Texto del evento: El Agente SQL Server activa una alerta cuando el evento especificado contiene una cadena de texto determinada en el mensaje de evento. Por ejemplo, puede definir una alerta que responda a los mensajes que contienen el nombre de una tabla o restricción determinada.

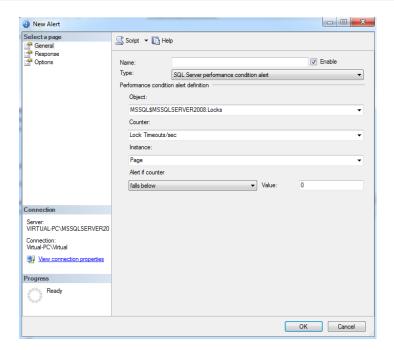


# Seleccionar una Condición de Rendimiento

Puede especificar una alerta para que se active en respuesta a una condición de rendimiento determinada. En este caso, debe especificar el contador de rendimiento que se supervisa, un umbral para la alerta y el comportamiento que el contador debe mostrar si la alerta tiene lugar. Para establecer una condición de rendimiento, debe definir los siguientes elementos en la página **General** del cuadro de diálogo **Nueva alerta** o Propiedades de alerta del Agente SQL Server:

- Objeto: El objeto es el área de rendimiento que se supervisa.
- Contador: Un contador es un atributo del área que se supervisa.
- Instancia: La instancia de SQL Server define la instancia específica (si la hay) del atributo que se va a supervisar.
- Alertar si el contador y Valor: El umbral de la alerta y el comportamiento que genera la alerta. El umbral es un número. El comportamiento puede ser: está por debajo de, es igual a o está por encima de un número especificado en Valor. El Valor es un número que describe el contador de condición de rendimiento. Por ejemplo, para establecer una alerta que tendrá lugar para el objeto de rendimiento SQLServer:Locks cuando pasen 30 minutos del Tiempo de espera de bloqueos, deberá elegir está por encima de y especificar 30 para el valor.

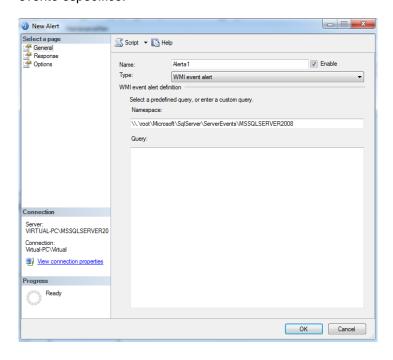
**Nota:** Se muestrean periódicamente los datos de rendimiento, lo que puede causar una pequeña demora (unos segundos) entre el momento en que se alcanza el umbral y la activación de la alerta relativa al rendimiento.



### Seleccionar un Evento de WMI

Puede especificar que una alerta tenga lugar como respuesta a un determinado evento de WMI. Para seleccionar un evento de WMI, debe definir lo siguiente en la página General del cuadro de diálogo Nueva alerta o Propiedades de alerta del Agente SQL Server:

- **Espacio de nombres**: El Agente SQL Server se registra como un cliente de WMI en el espacio de nombres de WMI que se proporciona para consultar los eventos.
- Consulta: El Agente SQL Server utiliza la instrucción de Lenguaje de consulta de Instrumental de administración de Windows (WQL) proporcionada para identificar el evento específico.



# **Alertar a Operadores**

Puede elegir a qué operadores notificará en respuesta a una alerta. Por ejemplo, puede asignar responsabilidades rotativas para notificar a los operadores mediante la programación de alertas. Por ejemplo, se notifica a A de las alertas que se producen los lunes, miércoles o viernes, y a B de las que se producen los martes, jueves y sábados. Si no se puede notificar a ninguno de los dos operadores, o si la alerta se produce en domingo, se notificará al operador a prueba de errores.

Los operadores pueden recibir notificaciones mediante uno o varios de los siguientes métodos:

- Notificación mediante correo electrónico
- Notificación mediante localizador
- Notificación mediante NET SEND

# 5- Manejo de Servidores Múltiples

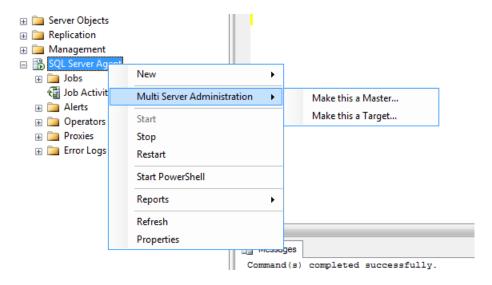
#### Administración multiservidor

La administración **multiservidor** requiere que se configure un servidor principal y uno o más servidores de destino. Los trabajos que se van a procesar en todos los servidores de destino se definen primero en el servidor principal y luego se descargan en los servidores de destino. Para crear un entorno multiservidor, utilice el Asistente para servidor principal.

El cifrado **SSL** (Capa de sockets seguros) y la validación de certificados completos se habilitan para las conexiones entre los servidores principales y los servidores de destino de forma predeterminada.

El asistente le guía por los siguientes pasos:

- Comprobar la configuración de seguridad del servicio del Agente SQL Server y del servicio SQL Server en todos los servidores que van a ser servidores de destino.
   Se recomienda ejecutar ambos servicios en cuentas de dominio de Microsoft Windows.
- Crear un operador de servidor principal (MSXOperator) en el servidor principal.
   MSXOperator es el único operador que puede recibir notificaciones de trabajos multiservidor.
- Iniciar el servicio del Agente SQL Server en el servidor principal.
- Dar de alta uno o más servidores como servidores de destino.



Si tiene un gran número de servidores de destino, evite definir el servidor principal en un servidor de producción. De lo contrario, el tráfico del servidor de destino puede ralentizar el rendimiento en el servidor de producción. Si también reenvía eventos a este servidor principal dedicado, puede centralizar la administración en un servidor.

**Nota:** Para utilizar el procesamiento de trabajos multiservidor, la cuenta de servicio del Agente SQL Server debe ser miembro de la función **TargetServersRole** de la base de datos **msdb** del servidor principal. El Asistente para servidor principal agrega automáticamente la cuenta de servicio a esta función como parte del proceso de alta.

#### Consideraciones para Entornos Multiservidor

Considere lo siguiente cuando cree un entorno multiservidor:





- Cada servidor de destino notifica únicamente a un servidor principal. Para dar de alta un servidor de destino en otro servidor principal, primero debe darlo de baja en el servidor principal actual.
- Si desea cambiar el nombre de un servidor de destino, debe darlo de baja antes de cambiar el nombre y volver a darlo de alta después del cambio.
- Si desea anular una configuración multiservidor, debe dar de baja todos los servidores de destino del servidor principal.
- Si desea distribuir planes de mantenimiento, necesitará una configuración adicional.

### **Procesar Trabajos Multiservidor**

Un trabajo multiservidor es un trabajo que ejecuta un servidor principal en uno o más servidores de destino. Cada servidor de destino sondea periódicamente al servidor principal, descarga una copia de cualquier nuevo trabajo asignado al servidor de destino y, a continuación, se desconecta. El servidor de destino ejecuta el trabajo de manera local y, a continuación, se vuelve a conectar al servidor principal para cargar el estado del resultado del trabajo una vez finalizado.



# ...

# 6- Seguridad del Agente SQL Server

El Agente SQL Server permite al administrador de la base de datos ejecutar cada paso de trabajo en un contexto seguro que sólo tiene los permisos necesarios para realizar ese paso de trabajo, que está determinado por un **servidor proxy** del Agente SQL Server. Para establecer los permisos para un paso de trabajo concreto, cree un proxy que disponga de los permisos necesarios y, a continuación, asigne ese proxy al paso de trabajo. Se puede especificar un servidor proxy en más de un paso de trabajo. Para los pasos de trabajo que necesitan los mismos permisos se utiliza el mismo proxy.

### Conceder Acceso al Agente SQL Server

Para utilizar el Agente SQL Server, los usuarios deben ser miembros de una o más de las siguientes funciones fijas de base de datos:

- SQLAgentUserRole
- SQLAgentReaderRole
- SQLAgentOperatorRole

Estas funciones se almacenan en la base de datos **msdb**. De manera predeterminada, ningún usuario es miembro de estas funciones de base de datos. La pertenencia a estas funciones se debe conceder explícitamente. Los usuarios que sean miembros de la función fija de servidor **sysadmin** tienen acceso total al Agente SQL Server, y no necesitan ser miembros de estas funciones fijas de base de datos para utilizar el Agente SQL Server. Si un usuario no es miembro de una de estas funciones de base de datos ni de la función **sysadmin**, el nodo del Agente SQL Server no estará disponible para ellos cuando se conecten con SQL Server mediante SQL Server Management Studio.

Los miembros de estas funciones de base de datos pueden ver y ejecutar trabajos que les pertenecen, así como crear pasos de trabajos que se ejecuten como una cuenta de proxy existente.

#### Permisos de las Funciones Fijas de Base de Datos del Agente SQL Server

Los permisos de las funciones de base de datos del Agente SQL Server son concéntricos: las funciones con más privilegios heredan los permisos de las funciones con menos privilegios en los objetos del Agente SQL Server (incluidos alertas, operadores, trabajos, programaciones y servidores proxy). Por ejemplo, si a los miembros de la función **SQLAgentUserRole** con menos privilegios se les ha concedido el acceso al proxy\_A, los miembros de las funciones **SQLAgentReaderRole** y **SQLAgentOperatorRole** tendrán automáticamente acceso a este proxy incluso si no se les ha concedido explícitamente el acceso al proxy\_A. Esto puede tener implicaciones de seguridad, que se describen en las siguientes secciones sobre cada función.

# Permisos de SQLAgentUserRole

Es la función con menos privilegios de todas las funciones fijas de base de datos del Agente SQL Server. Sólo dispone de permisos para operadores, trabajos locales y programaciones de trabajos. Los miembros de **SQLAgentUserRole** sólo tienen permisos en los trabajos locales y en las programaciones de trabajos que les pertenecen. No pueden utilizar trabajos multiservidor, ni pueden cambiar la propiedad de un trabajo para obtener acceso a trabajos que todavía no les pertenecen. Los miembros de **SQLAgentUserRole** pueden ver una lista de los servidores proxy disponibles únicamente en el cuadro de diálogo Propiedades de paso de trabajo de SQL Server Management Studio. Para los miembros de **SQLAgentUserRole** sólo está visible el nodo Trabajos del Explorador de objetos de SQL Server Management Studio.



# Permisos de SQLAgentReaderRole

La función **SQLAgentReaderRole** incluye todos los permisos de **SQLAgentUserRole** así como permisos para ver la lista de trabajos multiservidor disponibles, sus propiedades y su historial. Los miembros de esta función también pueden ver la lista de trabajos y programaciones de trabajos disponibles y sus propiedades, y no sólo los trabajos y programaciones de trabajos que les pertenecen. Los miembros de **SQLAgentReaderRole** no pueden cambiar la propiedad de un trabajo para obtener acceso a trabajos que no les pertenezcan ya. Para los miembros de **SQLAgentReaderRole** sólo está visible el nodo Trabajos del Explorador de objetos de SQL Server Management Studio.

# Permisos de SQLAgentOperatorRole

Es la función con más privilegios de todas las funciones fijas de base de datos del Agente SQL Server. Incluye todos los permisos de **SQLAgentUserRole** y **SQLAgentReaderRole**. Los miembros de esta función también pueden ver las propiedades de operadores y servidores proxy, así como enumerar los servidores proxy y alertas disponibles en el servidor.

Los miembros de **SQLAgentOperatorRole** tienen permisos adicionales en los trabajos locales y en las programaciones. Pueden ejecutar, detener o iniciar todos los trabajos locales, y pueden eliminar el historial de trabajos de cualquier trabajo local del servidor. También pueden habilitar o deshabilitar todos los trabajos locales y programaciones del servidor. Para los miembros de **SQLAgentOperatorRole** están visibles los nodos Trabajos, Alertas, Operadores y Servidores proxy del Explorador de objetos de SQL Server Management Studio. El único nodo que no está visible para los miembros de esta función es el nodo Registros de errores.

#### Asignar a los Usuarios varias Funciones

Los miembros de la función fija de seguridad **sysadmin** tienen acceso a toda la funcionalidad del Agente SQL Server. Si un usuario no es miembro de la función **sysadmin**, pero es miembro de más de una función fija de base de datos del Agente SQL Server, es importante recordar el modelo de permisos concéntricos de estas funciones. Debido a que las funciones con más privilegios siempre contienen todos los permisos de las funciones con menos privilegios, un usuario que sea miembro de más de una función automáticamente tendrá los permisos asociados con la función con más privilegios de la que sea miembro.

#### Cuentas de proxy del Agente SQL Server

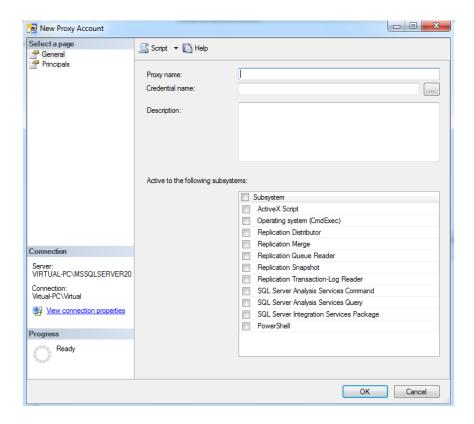
Un proxy del Agente SQL Server define el contexto de seguridad de un paso de trabajo. Proporciona al Agente SQL Server acceso a las credenciales de seguridad de un usuario de Microsoft Windows. Cada proxy se puede asociar a uno o más subsistemas. Un paso de trabajo que utilice el proxy puede obtener acceso a los subsistemas especificados usando el contexto de seguridad del usuario de Windows. Antes de que el Agente SQL Server ejecute un paso de trabajo que utilice un proxy, suplanta las credenciales definidas en el proxy y, a continuación, ejecuta el paso de trabajo usando este contexto de seguridad.

#### Creación de Proxies

- En el **Explorador de objetos**, expanda un servidor.
- Expanda Agente SQL Server.
- Haga clic con el botón secundario en Servidores proxy o simplemente Proxies y seleccione Nuevo proxy.
- En la página General del cuadro de diálogo Nueva cuenta de proxy, especifique el nombre del proxy, el nombre de la credencial y la descripción del nuevo proxy.
   Recuerde que debe crear una credencial antes de crear un proxy si todavía no hay uno disponible.



- Seleccione el **subsistema** correspondiente para este proxy.
- En la página **Entidades de seguridad**, agregue o quite inicios de sesión o funciones para conceder o quitar el acceso a la cuenta de proxy



# Módulo 6

# Replicación



# .

# 1- Introducción a Replicación

#### Definición

La réplica es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de base de datos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia. La réplica permite distribuir datos entre diferentes ubicaciones y entre usuarios remotos o móviles mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet.

Es útil dividir la réplica en dos amplias categorías:

- Replicar datos en un servidor para un entorno de servidor
- Replicar datos entre un servidor y los clientes.

# Tipos de Replicación

Microsoft SQL Server 2008 proporciona los siguientes tipos de réplica para utilizar en las aplicaciones distribuidas:

- Réplica transaccional (Transactinal replication).
- Réplica de mezcla (Merge replication).
- Réplica de instantáneas (Snapshot replication).

El tipo de réplica que se elige para una aplicación depende de muchos factores, como el entorno físico de la réplica, el tipo y la cantidad de datos que se desean replicar y si los datos se actualizan en el suscriptor. El entorno físico incluye el número y la ubicación de los equipos que participan en la réplica, y si estos equipos son clientes (estaciones de trabajo, equipos portátiles o dispositivos de mano) o servidores.

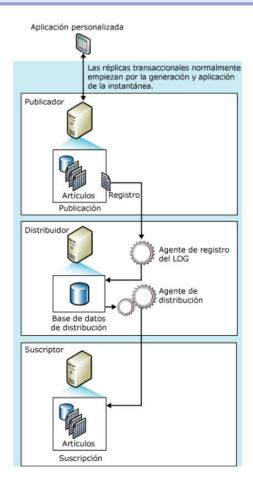
#### Replicación Transaccional

La replicación transaccional se implementa con el Agente de instantáneas, el Agente de registro del LOG y el Agente de distribución de SQL Server. El Agente de instantáneas prepara archivos de instantáneas que contienen esquemas y datos de las tablas y objetos de base de datos publicados, almacena los archivos en la carpeta de instantáneas y registra los trabajos de sincronización en la base de datos de distribución del distribuidor.

El Agente de registro del LOG supervisa el registro de transacciones de cada base de datos configurada para la replicación transaccional y copia las transacciones marcadas para ser replicadas desde el registro de transacciones a la base de datos de distribución, que actúa como una cola de almacenamiento y reenvío confiable. El Agente de distribución copia los archivos de instantáneas iniciales de la carpeta de instantáneas y las transacciones almacenadas en las tablas de la base de datos de distribución a los suscriptores.

Los cambios incrementales realizados en el publicador se transfieren a los suscriptores de acuerdo con la programación del Agente de distribución, que se puede ejecutar continuamente para que la latencia sea mínima o a intervalos programados. Puesto que los datos deben cambiarse en el publicador (cuando se utiliza la replicación transaccional sin las opciones de actualización inmediata ni de actualización en cola), se evita que se produzcan conflictos de actualización. Al final, todos los suscriptores disponen de los mismos valores que el publicador. Si se utilizan las opciones de actualización inmediata o de actualización en cola con la replicación transaccional, las actualizaciones pueden realizarse en el suscriptor y, con la actualización en cola, pueden producirse conflictos.

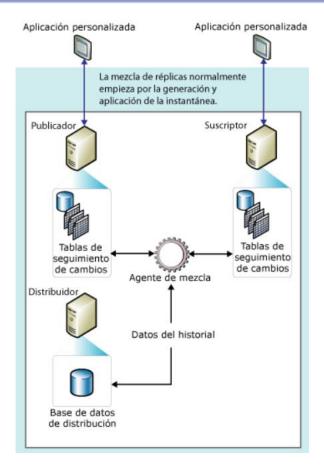




# Replicación de Mezcla

El Agente de instantáneas y el Agente de mezcla de SQL Server implementan la replicación de mezcla. Si la publicación no se filtra o utiliza filtros estáticos, el Agente de instantáneas crea una sola instantánea. Si la publicación utiliza filtros con parámetros, el Agente de instantáneas crea una instantánea para cada partición de datos. El Agente de mezcla aplica las instantáneas iniciales a los suscriptores. También mezcla los cambios incrementales de los datos que tienen lugar en el publicador o en los suscriptores después de la creación de la instantánea inicial y detecta y resuelve los conflictos según las reglas que configure.

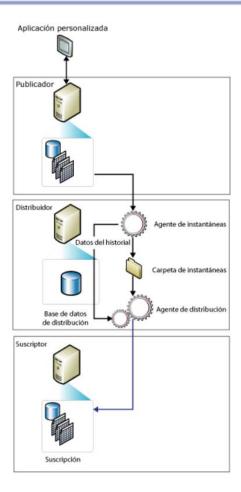
La réplica de mezcla permite que diferentes sitios funcionen de forma autónoma y, después, mezclen las actualizaciones en un solo resultado uniforme. Puesto que las actualizaciones tienen lugar en más de un nodo, los mismos datos pueden haber sido actualizados por el publicador y por más de un suscriptor. Por tanto, se pueden producir conflictos cuando las actualizaciones se mezclan y la réplica de mezcla proporciona varias formas de controlar los conflictos.



#### Replicación de Instantáneas

De forma predeterminada, los tres tipos de replicación utilizan una instantánea para inicializar suscriptores. El Agente de instantáneas de SQL Server siempre genera los archivos de instantáneas, pero el agente que entrega los archivos varía según el tipo de replicación que se utilice. La replicación de instantáneas y la replicación transaccional utilizan el Agente de distribución para entregar los archivos, mientras que la replicación de mezcla utiliza el Agente de mezcla de SQL Server. El Agente de instantáneas se ejecuta en el distribuidor. El Agente de distribución y el Agente de mezcla se ejecutan en el distribuidor para las suscripciones de inserción o en los suscriptores para las suscripciones de extracción.

Las instantáneas se pueden generar y aplicar inmediatamente tras la creación de la suscripción o de acuerdo con una programación establecida en el momento de crear la publicación. El Agente de instantáneas prepara archivos de instantáneas que contienen el esquema y los datos de las tablas y objetos de base de datos publicados, almacena los archivos en la carpeta de instantáneas del publicador y registra la información de seguimiento en la base de datos de distribución del distribuidor. Al configurar un distribuidor se especifica la carpeta de instantáneas predeterminada, pero puede especificar una ubicación alternativa para una publicación en lugar de, o además de, la predeterminada.



# Modelo de Publicación de Replicación

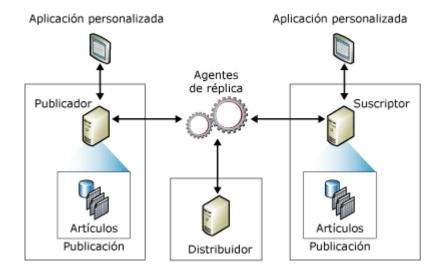
La réplica utiliza una metáfora del sector editorial para representar los componentes de una topología de réplica, que incluyen el publicador, el distribuidor, los suscriptores, las publicaciones, los artículos y las suscripciones. Resulta útil pensar en la réplica de Microsoft SQL Server como si fuera una revista:

- El **publicador** (editor) de una revista produce una o más publicaciones.
- Una publicación contiene artículos.
- El publicador distribuye la revista directamente o a través de un **distribuidor**.
- Los **suscriptores** reciben las publicaciones a las que se han suscrito.

Aunque la metáfora de la revista es útil para comprender la réplica, es importante señalar que la réplica de SQL Server incluye funciones que no están representadas en esta metáfora, en particular, la posibilidad de que un suscriptor realice actualizaciones y de que un publicador envíe cambios incrementales a los artículos de una publicación.

Una topología de réplica define la relación entre los servidores y las copias de los datos, y aclara la lógica que determina cómo fluyen los datos entre los servidores. Hay varios procesos de réplica (denominados agentes) que son responsables de copiar y mover los datos entre el publicador y los suscriptores. En la siguiente ilustración se muestra información general acerca de los componentes y procesos que participan en la réplica.





#### **Publicador**

El publicador es una instancia de base de datos que permite que los datos estén disponibles para otras ubicaciones a través de la réplica. El publicador puede tener una o más publicaciones, cada una de las cuales representa un conjunto de objetos y datos relacionados lógicamente para replicar.

#### Distribuidor

El distribuidor es una instancia de base de datos que funciona como almacén para datos específicos de réplica asociados con uno o más publicadores. Cada publicador está asociado con una sola base de datos (conocida como la base de datos de distribución) en el distribuidor. La base de datos de distribución almacena los datos de estado de la réplica, metadatos acerca de la publicación y, en algunos casos, funciona como cola para los datos que se transfieren del publicador a los suscriptores. En muchos casos, una sola instancia de servidor de bases de datos funciona como publicador y como distribuidor. Esto se conoce como un distribuidor local. Cuando el publicador y el distribuidor se configuran en instancias distintas del servidor de bases de datos, el distribuidor se denomina un distribuidor remoto.

#### Suscriptores

Un suscriptor es una instancia de base de datos que recibe datos replicados. Un suscriptor puede recibir datos de varios publicadores y publicaciones. En función del tipo de réplica elegida, el suscriptor también puede devolver los datos modificados al publicador o volver a publicar los datos en otros suscriptores.

#### **Artículo**

Un artículo identifica un objeto de base de datos incluido en una publicación. Una publicación puede contener diferentes tipos de artículos, como tablas, vistas, procedimientos almacenados y otros objetos. Cuando las tablas se publican como artículos, se pueden usar filtros para restringir las columnas y filas de datos que se envían a los suscriptores.





# á

# **Publicación**

Una publicación es un conjunto de uno o más artículos de una base de datos. La agrupación de varios artículos en una publicación permite especificar más fácilmente un conjunto de objetos y datos de bases de datos relacionados lógicamente, que se replican como una unidad.

### Suscripción

Una suscripción es una solicitud de una copia de una publicación que se entrega a un suscriptor. La suscripción define qué publicación se recibirá, dónde y cuándo. Hay dos tipos de suscripciones: de inserción y de extracción.

# d

# 2- Implementar Replicación

#### Implementación

El proceso de implementación de la réplica varía según el tipo de réplica y las opciones elegidas pero, en general, la réplica se compone de las fases siguientes:

- Configurar la réplica y publicar datos
- Crear e inicializar suscripciones
- Sincronizar datos

En este tema se proporciona información de cada fase, con vínculos a descripciones más detalladas. Además de conocer los pasos necesarios para configurar la réplica, es importante conocer las consideraciones de:

- Implementación.
- Seguridad.
- Rendimiento.
- Copia de seguridad y restauración.

# Configurar la Replicación y Publicar Datos

La implementación de la réplica comienza cuando se configura el publicador y el distribuidor. El distribuidor desempeña una función principal en la réplica transaccional; en cambio, tiene un papel más limitado en la réplica de instantáneas y de mezcla, donde se utiliza sólo para tareas del historial del agente y supervisión e informes de errores. Normalmente, la réplica de mezcla y la réplica de instantáneas utilizan un distribuidor que se ejecuta en el mismo equipo que el publicador, mientras que la réplica transaccional puede utilizar un distribuidor remoto, especialmente si el publicador es un sistema OLTP de alto rendimiento.

Después de configurar el publicador y el distribuidor, puede crear publicaciones basadas en los datos, subconjuntos de datos y objetos de base de datos. Cuando se crea una publicación, debe especificar:

- Los datos y los objetos de base de datos que desea replicar.
- Qué tipo y opciones de réplica utilizará, incluido el filtro.
- Dónde se almacenarán los archivos de instantáneas y cuándo se producirá la sincronización inicial, independientemente de que entregue manualmente el conjunto de datos inicial.
- Otras propiedades que se deben establecer para la publicación.

#### **Filtrar Datos Publicados**

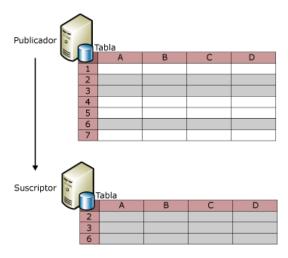
Al filtrar artículos de tabla podrá crear particiones de los datos que se publicarán. Si filtra los datos publicados, podrá:

- Minimizar la cantidad de datos que se envían a través de la red.
- Reducir la cantidad de espacio de almacenamiento necesario en el suscriptor.
- Personalizar las publicaciones y las aplicaciones en función de los requisitos de cada suscriptor.
- Evitar o reducir los conflictos si los suscriptores actualizan datos, ya que pueden enviarse particiones de datos diferentes a varios suscriptores (dos suscriptores distintos no actualizarán los mismos valores de datos).
- Evitar la transmisión de datos reservados. Se pueden utilizar filtros de fila y filtros de columna para restringir el acceso de un suscriptor a los datos.

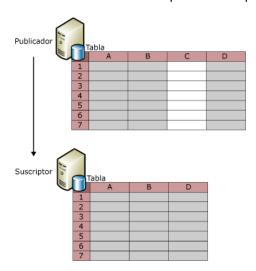
La Réplica ofrece cuatro Tipos de Filtros:



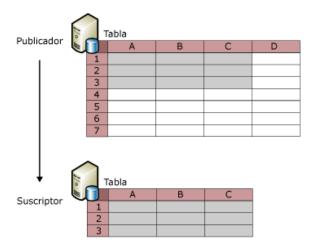
- Filtros de fila estáticos: que están disponibles con todos los tipos de réplica. Mediante los filtros de fila estáticos, puede especificar un subconjunto de filas para publicarlo. Todos los suscriptores de una publicación filtrada reciben el mismo subconjunto de filas para la tabla filtrada.
  - En la siguiente ilustración se muestra una tabla publicada que está filtrada para que solamente se incluyan las filas 2, 3 y 6 en la publicación.



• **Filtros de columna:** que están disponibles con todos los tipos de réplica. Al utilizar los filtros de columna, puede seleccionar un subconjunto de columnas para publicarlo. En la siguiente ilustración se muestra una publicación que filtra la columna C.



También puede utilizar conjuntamente el filtrado de filas y columnas, como se ilustra a continuación.



- Filtros de fila con parámetros: que están disponibles solamente con la réplica de mezcla. Al utilizar los filtros de fila con parámetros, puede seleccionar un subconjunto de filas para publicarlo. A diferencia de los filtros estáticos que envían el mismo subconjunto de filas a cada suscriptor, los filtros de fila con parámetros utilizan un valor de datos suministrado por el suscriptor para enviar a los suscriptores diferentes subconjuntos de filas.
- Filtros de combinación: que están disponibles solamente con la réplica de mezcla. Al utilizar los filtros de combinación, puede ampliar un filtro de fila de una tabla publicada a otra.

# Procedimientos Almacenados de Réplica

Los procedimientos almacenados del sistema de réplica se documentan y están disponibles como un método para realizar tareas únicas como la implementación de la réplica y el uso de archivos de proceso por lotes y secuencias de comandos. Si desea agregar un control de programación de la réplica para una aplicación o realizar tareas de réplica repetidas como la sincronización de suscripciones, se recomienda utilizar las interfaces de programación suministradas con los objetos de administración de réplica (RMO).

**Importante:** Sólo se admiten los procedimientos almacenados de réplica documentados en los Libros en pantalla de SQL Server. Los procedimientos almacenados no documentados sólo se utilizan para los componentes de réplica internos y no se pueden utilizar para administrar la réplica.

En la tabla siguiente se muestran los principales procedimientos almacenados de réplica:

Procedimiento almacenado	Descripción
sp_replicationdboption	Establece una opción de base de datos de réplica para la base de datos especificada. Este procedimiento almacenado se ejecuta en el publicador o el suscriptor de cualquier base de datos.
sp_addlogreader_agent	Agrega un Agente de registro del LOG a una base de datos dada. Este procedimiento almacenado se ejecuta en el publicador de la base de datos de publicaciones.
sp_addqreader_agent	Agrega un Agente de lectura de cola a un distribuidor específico. Este procedimiento almacenado se ejecuta en el distribuidor de la base de datos de distribución o en el publicador de la base de datos de publicaciones.



sp_addpublication	Crea una publicación de instantáneas o transaccional. Este procedimiento almacenado se ejecuta en el publicador de la base de datos de publicaciones.
sp_addpublication_snapshot	Crea el Agente de instantáneas en la publicación especificada. Este procedimiento almacenado se ejecuta en el publicador de la base de datos de publicaciones.
sp_grant_publication_access	Agrega un inicio de sesión a la lista de acceso de la publicación. Este procedimiento almacenado se ejecuta en el publicador de la base de datos de publicaciones.
sp_addarticle	Crea un artículo y lo agrega a una publicación. Este procedimiento almacenado se ejecuta en el publicador de la base de datos de publicaciones.
sp_articlefilter	Filtra los datos que se publican en función de un artículo de tabla. Este procedimiento almacenado se ejecuta en el publicador de la base de datos de publicaciones.
sp_articleview	Crea la vista que define el artículo publicado cuando una tabla se filtra horizontal o verticalmente. Esta vista se utiliza como el origen filtrado del esquema y los datos de las tablas de destino. Con este procedimiento almacenado sólo pueden modificarse artículos sin suscripciones. Este procedimiento almacenado se ejecuta en el publicador de la base de datos de publicaciones.

#### Crear e Inicializar Suscripciones

Una suscripción es una petición de copia de datos y objetos de base de datos en una publicación. Una suscripción define qué publicación se recibirá, dónde y cuándo. Al planear suscripciones, tenga en cuenta dónde se realizará el proceso del agente. El tipo de suscripción que elige controla dónde se ejecuta el agente. Con una suscripción de inserción, el Agente de Mezcla o el Agente de Distribución se ejecutan en el distribuidor, mientras que en una suscripción de extracción los agentes se ejecutan en los suscriptores. Después de crear una suscripción, no se puede cambiar de un tipo a otro.

Después de crear una publicación, puede crear suscripciones y configurar opciones adicionales.

Tanto si utiliza la réplica de instantáneas, transaccional o de mezcla, la réplica crea de forma predeterminada una instantánea inicial del esquema y los datos de la publicación y, a continuación, la guarda en la ubicación de la carpeta de instantáneas que ha especificado. Después de crear la suscripción, la instantánea inicial se aplica en función de la programación indicada al crear la publicación. Puede pasar por alto uno o varios pasos de la instantánea si el suscriptor ya tiene un conjunto de datos inicial o si desea aplicarla manualmente.

Suscripción	Características	Se utiliza si
Suscripción de inserción	Con una suscripción de inserción, el publicador propaga los cambios al suscriptor sin que éste lo pida. Los cambios pueden insertarse en los suscriptores a petición, de manera continua o según una programación. De forma predeterminada, el Agente de distribución o el Agente de mezcla se ejecutan en el distribuidor.	<ul> <li>Normalmente, los datos se sincronizarán de forma continua o con una frecuencia determinada.</li> <li>Las publicaciones requieren que el movimiento de datos sea casi en tiempo real.</li> </ul>



		<ul> <li>La sobrecarga del procesador en el distribuidor no afecta al rendimiento.</li> <li>Se utiliza frecuentemente en la réplica de instantáneas y transaccional.</li> </ul>
Suscripción de extracción	En una suscripción de extracción, el suscriptor solicita los cambios efectuados en el publicador. Las suscripciones de extracción permiten al usuario del suscriptor determinar cuándo se sincronizan los cambios en los datos. El Agente de distribución o el Agente de mezcla se ejecutan en el suscriptor.	<ul> <li>Los datos se sincronizarán, generalmente, a petición o en función de una programación, en lugar de hacerlo de forma continuada.</li> <li>La publicación dispone de un gran número de suscriptores y/o la ejecución de todos los agentes en el distribuidor supone un uso demasiado intensivo de recursos.</li> <li>Los suscriptores son autónomos, están desconectados o se desplazan. Los suscriptores determinan cuándo se conectarán y sincronizarán los cambios.</li> <li>Se utiliza frecuentemente en la réplica de mezcla.</li> </ul>

# Suscriptores que no son de SQL Server

Oracle e IBM DB2 pueden suscribirse a publicaciones de instantáneas y transaccionales con suscripciones de inserción.

#### **Sincronizar Datos**

La sincronización es el proceso de propagación de los datos entre el publicador y los suscriptores después de haber aplicado el conjunto de datos inicial en el suscriptor. En la réplica de instantáneas, la sincronización significa volver a aplicar la instantánea en el suscriptor, de modo que los datos y esquemas de la base de datos de suscripciones sean coherentes con la base de datos de publicaciones. En la réplica transaccional, la sincronización de datos significa que las modificaciones de datos, como las inserciones, actualizaciones y eliminaciones, se distribuyen entre el publicador y los suscriptores (y desde los suscriptores de vuelta al publicador en caso de suscripciones de actualización). En la réplica de mezcla, la sincronización significa que las modificaciones de datos realizadas en varios sitios se mezclan, los conflictos se detectan y resuelven, y los datos finalmente adoptan los mismos valores en todos los sitios.

### Asistentes para Réplica

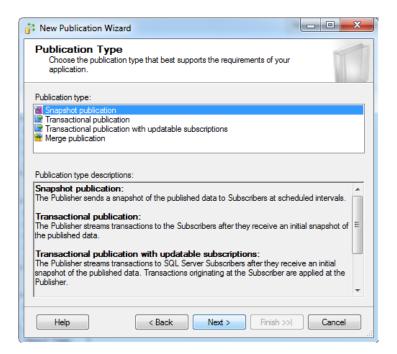
La réplica proporciona asistentes para simplificar la implementación de la réplica. A través de la carpeta Réplica en SQL Server Management Studio puede tener acceso a los asistentes para réplica.



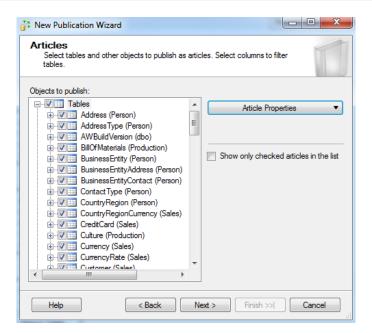
### Asistente para Nueva Publicación

Use el Asistente para nueva publicación para especificar lo siguiente:

- Que la instancia del publicador también actuará como el distribuidor (si va a utilizar un distribuidor remoto, éste ya debe estar configurado con el Asistente para configurar la distribución).
- La ubicación predeterminada de los archivos de instantáneas.
- La base de datos de publicaciones.
- El tipo de publicación que se va a crear (de instantánea, transaccional, transaccional con suscripciones actualizables o de mezcla).
- Los datos y los objetos de la base de datos (artículos) que se incluirán en la publicación.
- Los filtros de fila estáticos y los filtros de columna para todos los tipos de publicaciones, y los filtros de fila con parámetros y los filtros de combinación para publicaciones de mezcla
- La programación del Agente de instantáneas.
- Las cuentas con las que se ejecutarán los siguientes agentes: el Agente de instantáneas para todas las publicaciones, el Agente de registro del LOG para todas las publicaciones transaccionales, el Agente de lectura de cola para publicaciones transaccionales que permiten suscripciones de actualización.
- Un nombre y una descripción para la publicación.



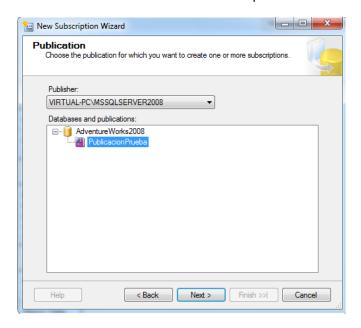




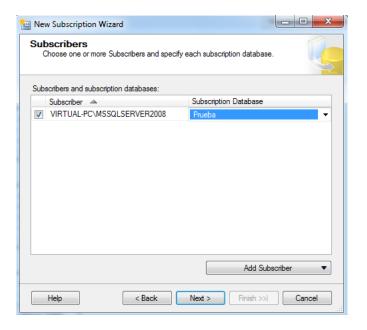
### **Asistente para Nuevas Suscripciones**

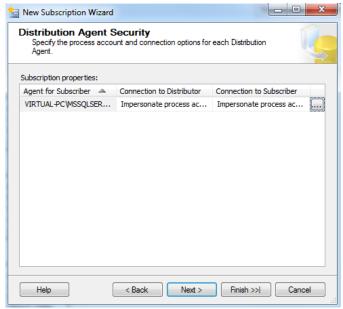
Use el Asistente para nuevas suscripciones para especificar lo siguiente:

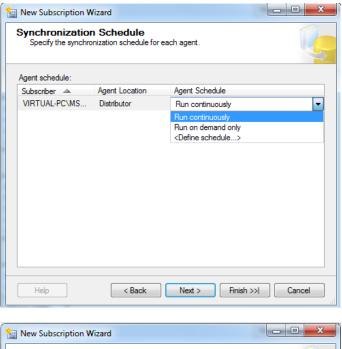
- La publicación a la que desea suscribirse.
- Dónde deben ejecutarse el Agente de distribución o el Agente de mezcla.
- Las cuentas con las que se ejecutarán y realizarán las conexiones el Agente de distribución o el Agente de mezcla.
- Uno o varios suscriptores que van a recibir datos publicados y la base de datos de suscripciones que recibirá los datos publicados en cada suscriptor.
- Si la suscripción debe inicializarse y, si es así, cuándo.
- Las programaciones de los agentes para definir la frecuencia con la que se propagarán las actualizaciones al suscriptor.
- Información adicional basada en los valores de publicación.

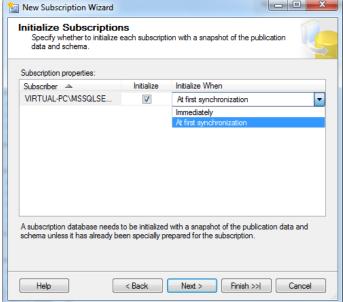












# Asistente para Configurar la Distribución

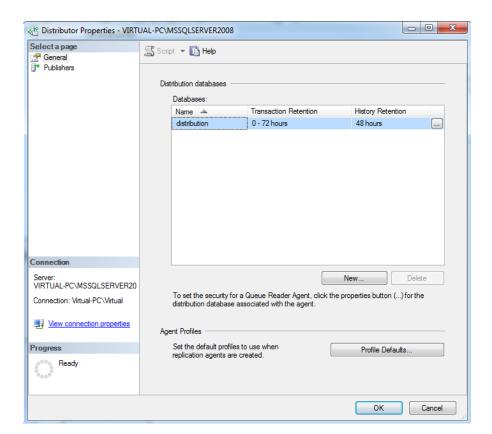
Use el Asistente para configurar la distribución para lo siguiente:

- Configurar un distribuidor remoto.
- Especificar la ubicación predeterminada de los archivos de instantáneas.

#### Nota:

También puede utilizar este asistente para configurar un distribuidor local, pero normalmente esto se realiza mediante el Asistente para nueva publicación la primera vez que se crea una publicación en un publicador.





# Asistente para la sincronización Web

Use el Asistente para la sincronización Web para configurar un servidor con Microsoft Internet Information Services (IIS) para la sincronización Web, que permite sincronizar suscripciones a publicaciones de mezcla a través de una conexión de Internet o intranet.

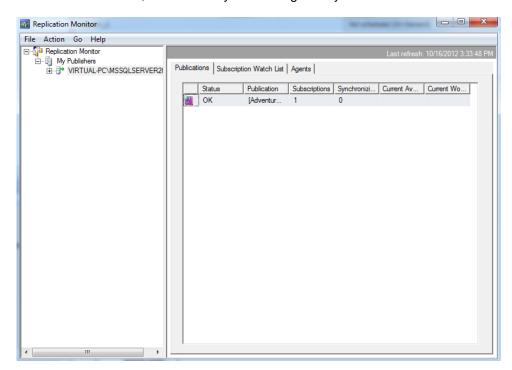
# 3- Monitor de Réplica

# Supervisión de la Réplica con el Monitor de Réplica

El Monitor de réplica de Microsoft SQL Server es una herramienta gráfica que le permite supervisar el estado general de una topología de réplica. El Monitor de réplica proporciona información detallada sobre el estado y rendimiento de las publicaciones y suscripciones Para supervisar la réplica, un usuario debe ser miembro de la función fija de servidor **sysadmin** en el distribuidor o miembro de la función fija de base de datos **replmonitor** en la base de datos de distribución. Un administrador del sistema puede agregar cualquier usuario a la función **replmonitor** que permite a un usuario ver la actividad de réplica en el Monitor de réplica; sin embargo, el usuario no puede administrar la réplica.

#### La interfaz del Monitor de réplica

El Monitor de réplica de Microsoft SQL Server presenta una vista centrada en el publicador de toda la actividad de réplica en un formato de dos paneles. Se agrega un publicador al monitor en el panel izquierdo y, en el panel derecho, el monitor muestra información sobre el publicador, sus publicaciones, las suscripciones a esas publicaciones y los diversos agentes de réplica. Además de presentar información de la topología de réplica, el Monitor de réplica permite realizar varias tareas, como iniciar y detener agentes y validar datos.



### Ver información de toda la topología

En el panel izquierdo del Monitor de réplica se muestran los grupos de publicadores, los publicadores y las publicaciones.

- El estado del sistema de réplica es relativamente correcto si no existen iconos de error en los nodos del panel izquierdo. Para obtener una vista más completa del estado del sistema, también debe comprobar la ficha Lista de supervisión de suscripciones, en la que se muestra información sobre las suscripciones que pueden requerir atención.
- Un agente no funciona en un momento determinado porque no está programado para ejecutarse o porque se ha producido un error. Si se ha producido un error, se muestra



un icono de error en los nodos correspondientes del panel izquierdo. Por ejemplo, si el Agente de instantáneas de una publicación se detiene debido a un error, se muestra un icono de error en el grupo de publicador, el publicador y los nodos de publicación. En la ficha Advertencias y agentes de la publicación se muestra información resumida del Agente de instantáneas; haga doble clic en el Agente de instantáneas de esta ficha para obtener información detallada del error.

# Ver información y realizar tareas relacionadas con publicadores

El Monitor de réplica muestra información sobre publicadores en tres fichas:

- **Ficha Publicaciones:** Esta ficha proporciona información resumida de todas las publicaciones en un publicador.
- Ficha Lista de Supervisión de Suscripciones: Esta ficha está diseñada para mostrar información sobre suscripciones de todas las publicaciones disponibles en el publicador seleccionado. Puede filtrar la lista de suscripciones para ver errores, advertencias y las suscripciones que tienen un rendimiento bajo. Esta ficha permite también: tener acceso a las propiedades de la suscripción, tener acceso a información detallada sobre el agente o los agentes asociados con una suscripción, y reinicializar y validar suscripciones.
- Ficha Trabajos Comunes: En esta ficha se muestra información sobre los trabajos utilizados por todos los tipos de réplica. Esta ficha permite también: tener acceso a información detallada sobre los trabajos e iniciar y detener cada trabajo.

El Monitor de réplica también proporciona un menú contextual para el nodo de **publicador**. En el panel izquierdo, haga clic con el botón secundario en un publicador para:

- Editar la configuración del Monitor de réplica para el publicador.
- Quitar el publicador del Monitor de réplica.
- · Ver y editar perfiles de agente.
- Conectarse o desconectarse del distribuidor que almacena información sobre el publicador.

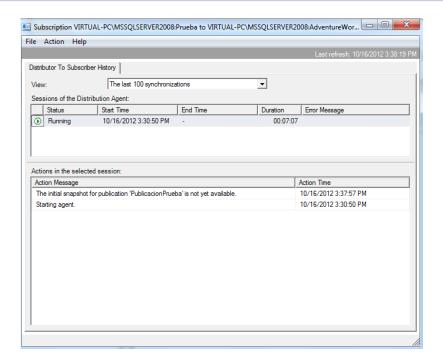
El Monitor de réplica también proporciona un menú contextual para el nodo de **publicaciones**. En el panel izquierdo, haga clic con el botón secundario en una publicación para:

- Reinicializar todas las suscripciones en una publicación.
- Validar todas las suscripciones en una publicación.
- Generar una instantánea para una publicación.
- Ver y editar las propiedades de una publicación.

# Ver información y realizar tareas relacionadas con suscripciones

El Monitor de réplica muestra información sobre suscripciones en varias fichas diferentes. Haga doble clic en una suscripción en el Monitor de Réplica para tener acceso a información en una ventana de detalles.



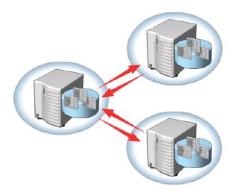


# 4- Configurar Replicación en algunos Escenarios Comunes

# Replicar datos en un entorno de servidor a servidor

Por lo general, los datos se replican entre servidores para proporcionar compatibilidad con las siguientes aplicaciones y requisitos:

- Mejorar la escalabilidad y la disponibilidad: Mantener copias de los datos que se actualicen de forma continua permite escalar la actividad de lectura entre varios servidores. La redundancia que produce el mantener varias copias de los mismos datos es crucial durante el mantenimiento del sistema (ya sea planeado o no planeado).
- Almacenamiento de datos e informes: Los servidores de almacenamiento de datos e
  informes utilizan con frecuencia datos de los servidores de procesamiento de
  transacciones en línea (OLTP). Utilice la replicación para mover datos entre los
  servidores OLTP y los sistemas de informes y ayuda para la toma de decisiones.
- Integrar datos de varios sitios: Con frecuencia, los datos se transfieren desde las
  oficinas remotas y se consolidan en una oficina central. De la misma forma, es posible
  replicar los datos en las oficinas remotas.
- Integrar datos heterogéneos: Algunas aplicaciones dependen de datos que se envían o se reciben de bases de datos distintas de SQL Server. Utilice la replicación para integrar datos de bases de datos que no sean de SQL Server.
- Descargar procesos por lotes: Las operaciones por lotes consumen con frecuencia demasiados recursos para ejecutarse en un servidor OLTP. Utilice la replicación para descargar el procesamiento a un servidor de proceso por lotes dedicado.

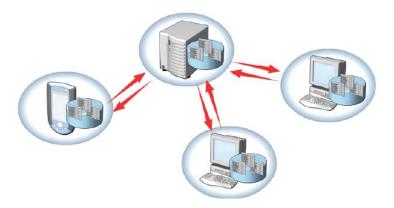


#### Replicar datos entre un servidor y los clientes

Replican datos entre servidores y clientes, incluidas estaciones de trabajo, equipos portátiles, Tablet PC y dispositivos. Los datos normalmente se replican entre servidores y clientes para admitir las siguientes aplicaciones:

- Intercambiar datos con usuarios móviles: Muchas aplicaciones requieren que los datos estén disponibles para usuarios remotos, como personal de ventas, repartidores, etc. Estas aplicaciones incluyen las de administración de relaciones con los clientes, automatización del personal de ventas y automatización del personal de campo.
- Aplicaciones de punto de venta (POS) para el consumidor: Las aplicaciones POS, como los terminales de caja de salida y los cajeros automáticos, requieren que los datos se repliquen desde sitios remotos a un sitio central.
- Integrar datos de varios sitios: Las aplicaciones a menudo integran datos de varios sitios. Por ejemplo, una aplicación de apoyo a oficinas regionales puede requerir que los datos fluyan en una o en ambas direcciones entre las oficinas regionales y la oficina central.





# Replicación Transaccional Punto a Punto (Peer to Peer)

La réplica transaccional de punto a punto está diseñada para aplicaciones que pueden leer o modificar los datos en cualquiera de las bases de datos que participan en la réplica. Por ejemplo, una aplicación para realizar compras por Internet es muy adecuada para la réplica de punto a punto: el rendimiento de la aplicación se puede mejorar distribuyendo entre varias bases de datos las consultas que leen los datos. Además, si alguno de los servidores que alojan las bases de datos no está disponible, se puede programar una aplicación para dirigir el tráfico a los otros servidores, que contienen copias idénticas de los datos. El rendimiento de lectura mejora porque la actividad se puede repartir entre todos los nodos. El rendimiento acumulado de actualización, inserción y eliminación para la topología es similar al de un solo nodo, porque al final todos los cambios se propagan a todos los nodos.

Todos los nodos de una topología de punto a punto son pares: cada nodo publica y suscribe al mismo esquema y los mismos datos. Los cambios (inserciones, actualizaciones y eliminaciones) pueden realizarse en todos los nodos. La réplica reconoce cuándo un cambio se ha aplicado a un nodo determinado, lo que evita cambios cíclicos en los nodos más de una vez.

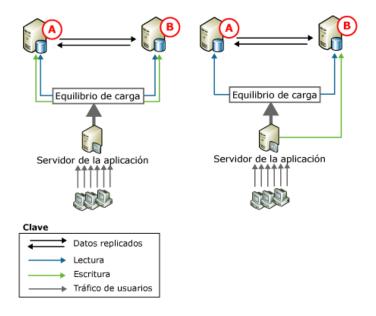
**Nota:** Las aplicaciones personalizadas que tienen acceso a los datos y los modifican deben asegurarse de que las inserciones, actualizaciones y eliminaciones tienen particiones, de forma que las modificaciones en una fila determinada con origen en un nodo se sincronicen con el resto de bases de datos de la topología antes de que la fila se modifique en otro nodo diferente. Si una aplicación ejecuta modificaciones simultáneas que producen conflictos en una fila determinada en varios nodos, utilice la réplica de mezcla, que se adapta perfectamente para controlar conflictos.

La réplica transaccional estándar supone suscriptores de sólo lectura y tiene una estructura jerárquica: normalmente un solo publicador publica datos para uno o varios suscriptores. La réplica transaccional estándar también admite una jerarquía de republicación: las actualizaciones se entregan desde un publicador a un conjunto de suscriptores de republicación, que a su vez entregan las actualizaciones a un conjunto final de suscriptores de nodo hoja. Las suscripciones de actualización ofrecen la posibilidad de que los suscriptores inserten cambios de nuevo en el publicador, pero la organización aún es jerárquica porque los cambios siguen la estructura jerárquica cuando se mueven entre los suscriptores y los publicadores. Al contrario que la réplica transaccional de sólo lectura y la réplica transaccional con suscripciones de actualización, las relaciones entre nodos en una topología de réplica de punto a punto son relaciones entre pares en vez de jerárquicas, donde cada nodo contiene un esquema y datos idénticos.

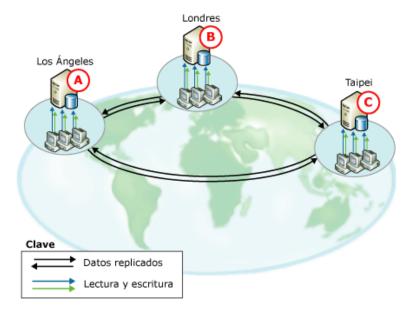


Aunque las actualizaciones pueden realizarse donde participan varias bases de datos, es importante comprender que las topologías de punto a punto no necesitan ni permiten las opciones de publicación de actualización en cola o inmediata.

# Topología en la que participan dos bases de datos:



# Topología en la que participan tres o más bases de datos





#### Implementar la Replicación por Internet

La réplica de datos por Internet permite a usuarios desconectados y remotos obtener acceso a los datos cuando los necesiten, por medio de una conexión a Internet. Puede llevar a cabo la réplica de datos por Internet mediante:

- Una red virtual privada (VPN): La tecnología de Red privada virtual (VPN) permite a los usuarios que trabajan en su casa, oficinas subsidiarias, clientes remotos y otras empresas conectarse a una red corporativa por Internet, al tiempo que se mantiene una comunicación segura. Los usuarios pueden utilizar la Autenticación de Windows como si estuvieran en una red de área local (LAN). Todos los tipos de réplica de Microsoft SQL Server pueden replicar datos a través de una red privada virtual (VPN), pero se puede usar la sincronización Web si se está usando la réplica de mezcla, porque con la sincronización Web no es necesario usar una VPN.
- Sincronización Web de la réplica de mezcla: La sincronización Web para la réplica de mezcla permite replicar datos utilizando el protocolo HTTPS y es útil en los siguientes escenarios:
  - o Sincronizar datos de usuarios móviles a través de Internet
  - Sincronizar datos entre bases de datos de Microsoft SQL Server a través de un firewall corporativo

Por ejemplo, un representante de ventas puede utilizar la sincronización Web durante sus viajes. El Agente de mezcla de cada equipo portátil tiene una dirección URL de Internet que apunta a los componentes de réplica instalados en un equipo en el que se ejecutan los Servicios de Microsoft Internet Information Server (IIS). Estos componentes sincronizan el suscriptor con el publicador. Ahora cada representante se puede conectar a través de cualquier conexión de Internet disponible sin utilizar una conexión remota de acceso telefónico y puede cargar y descargar los datos que desee. La conexión de Internet utiliza SSL (Capa de sockets seguros), por lo que no es necesaria una red privada virtual (VPN).

Todos los tipos de réplica de Microsoft SQL Server pueden replicar datos mediante una VPN, pero debe considerar la sincronización Web si utiliza la réplica de mezcla.



# Módulo 7

# Mantenimiento de Alta Disponibilidad



# 1- Introducción a Alta Disponibilidad

#### Factores que afectan la Disponibilidad:

La disponibilidad se ocupa de asegurar que las aplicaciones y servicios de una organización sigan funcionando ante una falla. Para implementar soluciones de alta disponibilidad usted debe analizar y comprender los diferentes tipos de fallas a los que está expuesto su sistema y planificar una estrategia de disponibilidad adecuada.

- Fallas de Software: Una aplicación consistirá de muchos componentes de software, la
  mayoría de los cuales serán requeridos para el funcionamiento de la aplicación.
  Además de las aplicaciones de cliente y servidor, puede haber aplicaciones
  intermedias y de servidor Web. Los sistemas operativos en todos los nodos son
  cruciales para que una aplicación funcione debidamente, inclusive servicios y
  aplicaciones no relacionadas pueden causar fallas si utilizan demasiados recursos.
- Fallas de Componentes de Hardware: Considere los efectos de la falla de cada componente de hardware en su sistema. Discos duros, procesadores, memoria y tarjetas de red pueden causar problemas particulares, pero otros componentes como ventiladores y fuentes de energía pueden ser igualmente esenciales para el correcto funcionamiento de su sistema.
- Falla de Red: Una falla de red puede deberse a un cable defectuoso. Una red basada en TCP/IP utiliza Servidores Domain Naming System (DNS) y Dynamic Host Configuration Protocol (DHCP), así como routers y otros elementos. Todos ellos pueden fallar o tener problemas de configuración.
- Falla de energía o desastre natural: Resulta relativamente sencillo proteger por períodos cortos contra estas fallas, pero hay eventos naturales que pueden afectar un sitio entero por largos períodos.

#### Configurar una Alta Disponibilidad

Una solución de alta disponibilidad enmascara los efectos de un error de hardware o software y mantiene la disponibilidad de las aplicaciones a fin de minimizar el tiempo de inactividad que perciben los usuarios.

SQL Server 2008 ofrece varias opciones para crear una alta disponibilidad para un servidor o una base de datos. Entre las opciones de alta disponibilidad figuran las siguientes:

- Clúster de conmutación por error
- Creación de reflejo de la base de datos
- Trasvase de registros
- Replicación

**Importante**: Es fundamental disponer de una estrategia de copia de seguridad y restauración bien diseñada e implementada para utilizar una solución de alta disponibilidad.

#### Seleccionar una Solución de Alta Disponibilidad

En la siguiente lista figuran algunas consideraciones que deben tenerse en cuenta para elegir una solución de alta disponibilidad.

El clúster de conmutación por error y la creación de reflejo de la base de datos ofrecen:

- Detección y conmutación por error automáticas
- Conmutación por error manual
- Redireccionamiento de clientes transparente

El **clúster de conmutación por error** presenta las siguientes restricciones:

Funciona en el ámbito de la instancia de servidor



- Exige hardware certificado
- No permite la generación de informes en espera
- Utiliza una sola copia de la base de datos
- No protege frente a errores de disco

#### La creación de **reflejo de la base de datos** ofrece las siguientes ventajas:

- Se aplica en la base de datos
- Utiliza una sola copia duplicada de la base de datos. En caso de necesitarse más copias, puede utilizar la solución de trasvase de registros en la base de datos además de la creación de reflejo de la base de datos.
- Utiliza servidores estándar
- Proporciona una capacidad de generación de informes limitada en el servidor reflejado mediante instantáneas de base de datos
- Cuando funciona de forma sincrónica, garantiza una pérdida de datos cero gracias a la confirmación retardada de la base de datos principal

La creación de reflejo de la base de datos supone un aumento considerable de la disponibilidad con relación al nivel que se conseguía anteriormente con SQL Server y constituye una alternativa fácil de administrar al clúster de conmutación por error.

El **trasvase de registros** puede ser un complemento o una alternativa a la creación de reflejo de la base de datos. Aunque son similares conceptualmente, la creación de reflejo de la base de datos asincrónica y el trasvase de registros tienen diferencias importantes. El trasvase de registros ofrece las siguientes capacidades distintivas:

- Admite varias bases de datos secundarias en varias instancias de servidor para una base de datos principal única.
- Permite un retraso especificado por el usuario entre el momento en que el servidor primario realiza una copia de seguridad del registro de la base de datos primaria y el momento en que los servidores secundarios deben restaurar la copia de seguridad del registro. Un retraso más largo puede ser útil, por ejemplo, si los datos se cambian en la base de datos principal de manera accidental. Si se detecta el cambio accidental rápidamente, un retraso puede permitirle recuperar datos aún sin modificar de una base de datos secundaria antes de que el cambio se refleje en ella.
  Si se compara con el tiempo más breve necesario para que el trasvase de registros refleje un cambio en una base de datos secundaria, la creación de reflejo de la base de
  - refleje un cambio en una base de datos secundaria, la creación de reflejo de la base de datos asincrónica tiene la ventaja potencial de un tiempo menor entre el momento en que se realiza un cambio determinado en la base de datos primaria y el momento en que dicho cambio surte efecto en la base de datos reflejada. Una ventaja de la creación de reflejo de la base de datos respecto al trasvase de registros es que el modo de alta seguridad es una configuración "sin pérdida de datos" compatible como estrategia de conmutación por error simple.

#### La **replicación** ofrece las siguientes ventajas:

- Permite el filtrado en la base de datos para obtener un subconjunto de datos en las bases de datos secundarias, dado que funciona en el ámbito de la base de datos.
- Permite más de una copia redundante de la base de datos.
- Permite la disponibilidad y escalabilidad en tiempo real entre varias bases de datos, además de admitir las actualizaciones con particiones.
- Asegura una disponibilidad completa de las bases de datos secundarias para las opciones de informes entre otras, sin recuperación de consultas.



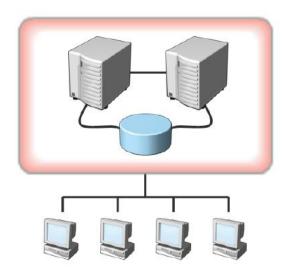
# 2- Implementación de Server Clustering

#### Que es un Clúster?

Un clúster de conmutación proporciona alta disponibilidad a una instancia completa de SQL Server. Un clúster de conmutación por error es una combinación de uno o varios nodos, o servidores, con dos o más discos compartidos. Aplicaciones como SQL Server y Notification Services se instalan cada una en un grupo de clústeres de Servicios de Cluster Server de Microsoft (MSCS), conocido como grupo de recursos. En todo momento, cada grupo de recursos es propiedad de un único nodo del clúster. El servicio de aplicación tiene un nombre virtual que es independiente de los nombres del nodo y al que nos referimos como nombre de instancia de clúster de conmutación por error. Una aplicación puede conectarse con la instancia de clúster de conmutación por error si hace referencia al nombre de la misma. La aplicación no necesita saber qué nodo aloja a la instancia de clúster de conmutación por error.

Una instancia de clúster de conmutación por error de SQL Server aparece en la red como un equipo individual, pero ofrece funciones para la conmutación por error entre nodos si el nodo actual deja de estar disponible. Por ejemplo, durante un error de hardware no relacionado con el disco, un error del sistema operativo o una actualización planeada del mismo, puede configurar una instancia de SQL Server en un nodo de un clúster de conmutación por error para que conmute a cualquier otro nodo del grupo de discos.

Un clúster de conmutación por error no implica protección ante errores de disco. Puede utilizar el clúster de conmutación por error para reducir el tiempo de inactividad del sistema y garantizar una mayor disponibilidad de la aplicación. El clúster de conmutación por error es compatible con SQL Server 2008 Enterprise Edition, Developer Edition y, con algunas restricciones, en la versión Standard Edition.



#### Consideraciones antes de Instalar un Clúster de Conmutación por Error

Antes de instalar un clúster de conmutación por error de SQL Server 2008, debe seleccionar el hardware y el sistema operativo en el que se ejecutará SQL Server 2008. También debe configurar el Servicio de Cluster Server de Microsoft (MSCS), así como revisar las consideraciones relativas a la red, la seguridad y el resto del software que se ejecutará en el clúster de conmutación por error.



#### Lista de comprobación previa a la instalación

Antes de comenzar el proceso de instalación del clúster de conmutación por error, revise los siguientes elementos:

#### Comprobar la Configuración del Sistema Operativo

- Asegúrese de que el sistema operativo esté correctamente instalado y diseñado para admitir los clústeres de conmutación por error.
- Habilite el Proveedor de servicios de cifrado de Windows (*CSP*) en Windows Server 2003. Si el servicio CSP se detiene o deshabilita en cualquier nodo del clúster, se produce un error en la instalación de SQL Server con un mensaje de error de requisito del logotipo de Windows.
- Habilite el servicio Programador de tareas en todos los sistemas operativos para la instalación remota y de clúster. Si el Programador de tareas está deshabilitado, la instalación de SQL Server producirá el error 1058.
- SQL Server 2008 admite puntos de montaje; las instalaciones en clúster de SQL Server están limitadas al número de letras de unidad disponibles. Si utiliza sólo una letra de unidad para el sistema operativo, se limita a un máximo de 25 instancias de SQL Server por clúster de conmutación por error.
- Un volumen montado, o punto de montaje, le permite utilizar una sola letra de unidad para hacer referencia a muchos discos o volúmenes. Si tiene una letra de unidad D: para un disco o volumen normal, puede conectar o "montar" discos o volúmenes adicionales como directorios en la letra de unidad D: sin que dichos discos o volúmenes adicionales exijan letras de unidad propias.

#### Configurar el Servicio de Cluster Server de Microsoft

- El Servicio de Cluster Server de Microsoft (MSCS) debe configurarse al menos en un nodo del clúster de servidores. MSCS sólo se admite si está instalado en una configuración de hardware cuya compatibilidad con el software MSCS haya sido comprobada. Además, debe ejecutar SQL Server 2008 Enterprise Edition o Standard Edition junto con MSCS. SQL Server 2008 Enterprise Edition admite clústeres de conmutación por error con un máximo de 8 nodos. SQL Server 2008 Standard Edition admite clústeres de conmutación por error con 2 nodos.
- La DLL de recursos para el servicio SQL Server exporta dos funciones utilizadas por administrador de clústeres de MSC para comprobar disponibilidad del recurso SQL Server. Una comprobación simple, LooksAlive, consulta el estado del Administrador de control de servicios de Windows NT. Una comprobación más rigurosa, IsAlive, conecta a SQL Server como un sondeo del usuario para realizar una consulta simple. De forma predeterminada, LooksAlive se activa cada 5 segundos e IsAlive se activa cada 60 segundos. Los intervalos de sondeo de IsAlive y LooksAlive se pueden cambiar en el Administrador de clústeres de MSC en la ficha Avanzadas para el recurso SQL Server o utilizando el comando del símbolo del sistema cluster.exe.
- MSCS debe poder comprobar que la instancia agrupada de conmutación por error está en ejecución mediante la comprobación IsAlive. Esto requiere conectar al servidor mediante una conexión de confianza. De forma predeterminada, la cuenta que ejecuta el servicio de clúster está configurada como administrador en todos los nodos del clúster y el grupo BUILTIN\Administradores tiene permiso para iniciar sesión en SQL Server. Esta configuración sólo cambia si se cambian los permisos en los nodos del clúster. Si quita la cuenta BUILTIN\Administradores, asegúrese de que la cuenta con la que se ejecutan los Servicios de Cluster Server puede iniciar una sesión en SQL Server para la comprobación de IsAlive. En caso contrario, se producirá un error en la comprobación de IsAlive. Como mínimo, la cuenta de los Servicios de Cluster Server





(MSCS) debe tener privilegios **public** en SQL Server para poder ejecutar **@@servername** de modo regular.

- Al instalar MSCS, es muy importante utilizar cuentas de servicio independientes para iniciar una sesión en MSCS y SQL Server. De lo contrario, la contraseña del servicio de clúster no se puede cambiar con el comando de clúster.
- Cuando se utiliza MSCS, un nodo debe tener el control del bus SCSI compartido antes de conectar el otro nodo. Si no es así, la conmutación por error de aplicaciones puede entrar en un estado pendiente de conexión e impedir la conmutación por error al otro nodo o producir un error total. Si el sistema de clúster tiene un proceso de instalación propio, debe utilizarse.

#### Instalar el Coordinador de Transacciones Distribuidas de Microsoft

Antes de instalar SQL Server 2008 en un clúster de conmutación por error, determine si debe crearse el recurso de clúster del Coordinador de Transacciones Distribuidas de Microsoft (MSDTC). Si sólo instala el Motor de base de datos, no será necesario el recurso de clúster de MSDTC. Si instala el Motor de base de datos y SSIS, Notification Services o componentes de la estación de trabajo, deberá instalar MSDTC. Este requisito se aplica a los sistemas operativos Windows 2000 y Windows Server 2003.

Las herramientas administrativas Servicios de componentes, proxy de MSDTC y administrador de transacciones de MSDTC se instalan en cada nodo del clúster de servidores basados en Windows. El clúster utiliza los Servicios de Cluster Server de Microsoft (MSCS) como parte de la instalación del clúster de servidores basados en Windows.

Tras instalar el sistema operativo y configurar el clúster, debe configurar MSDTC para que funcione en un clúster mediante el Administrador de clústeres. Si no logra crear el clúster de MSDTC, no se bloqueará el programa de instalación de SQL Server, pero la funcionalidad de la aplicación SQL Server puede verse afectada si MSDTC no se configura correctamente.

Todos los procesos que se ejecuten en cualquier nodo del clúster pueden utilizar MSDTC. Estos procesos simplemente llaman al proxy de MSDTC y éste reenvía automáticamente las llamadas de MSDTC al administrador de transacciones de MSDTC, que controla todo el clúster. Si se produce un error en el nodo que ejecuta el administrador de transacciones de MSDTC, éste se reinicia automáticamente en otro nodo del clúster. El administrador de transacciones recién reiniciado lee el archivo de registro de MSDTC en el disco del clúster compartido para determinar el resultado de las transacciones pendientes y recién completadas.

Los administradores de recursos se vuelven a conectar al administrador de transacciones y realizan la recuperación para determinar el resultado de las transacciones pendientes. Las aplicaciones se vuelven a conectar a MSDTC para poder iniciar las nuevas transacciones. Por ejemplo, suponga que el administrador de transacciones de MSDTC está activo en el sistema B. El programa de aplicación y el administrador de recursos del sistema A llaman al proxy de MSDTC. El proxy de MSDTC del sistema A reenvía todas las llamadas de MSDTC al administrador de transacciones de MSDTC del sistema B. Si se produce un error en el sistema B, el administrador de transacciones de MSDTC del sistema A tomará el control. Leerá todo el archivo de registro de MSDTC en el disco del clúster compartido, realizará la recuperación y, a continuación, actuará como administrador de transacciones de todo el clúster.

#### **Otras Consideraciones de Software**

- Asegúrese de que todos los nodos del clúster están configurados de forma idéntica, lo que incluye COM+, letras de unidad de disco y usuarios del grupo de administradores.
- Compruebe que la interconexión del clúster (latido) está configurada correctamente.



- Compruebe que ha borrado los registros del sistema en todos los nodos y ha consultado de nuevo los registros del sistema. Antes de continuar, asegúrese de que los registros no contienen mensajes de error.
- Para instalaciones de SQL Server 2008 en configuraciones simultáneas con versiones anteriores de SQL Server, los servicios de SQL Server 2008 deben usar cuentas que sólo se encuentran en el grupo de dominio global. Además, las cuentas utilizadas por los servicios de SQL Server 2008 no deben aparecer en el grupo local de administradores. Si no se sigue esta directriz, se producirán comportamientos inesperados con respecto a la seguridad.
- Si instala SQL Server 2008 en un grupo de clúster de Windows con varias unidades de disco y decide colocar los datos en una de las unidades, el recurso de SQL Server se configura para que sólo dependa de dicha unidad. Para colocar datos o registros en recursos de disco adicionales, primero debe agregar una dependencia al recurso de SQL Server para el disco adicional.
- Para utilizar el cifrado, instale el certificado del servidor con el nombre DNS completo del clúster MSCS en todos los nodos del clúster de conmutación por error de SQL Server. Por ejemplo, si tiene un clúster con dos nodos cuyos nombres son "Test1.DomainName.com" y "Test2.DomainName.com" y una instancia de clúster de conmutación por error de SQL Server denominada "Virtsql", debe obtener un certificado para "Virtsql.DomainName.com" e instalarlo en los nodos test1 y test2. A continuación, puede activar la casilla de verificación Forzar cifrado de protocolo en el Administrador de configuración de SQL Server para configurar el clúster de conmutación por error para el cifrado.

**Importante:** No active la casilla de verificación Forzar cifrado de protocolo hasta que haya instalado certificados en todos los nodos participantes de la instancia de clúster de conmutación por error.

- Compruebe que no tiene instalado software antivirus en el clúster MSCS.
- SQL Server 2008 no se admite en Terminal Server de Windows Server 2003.
- Compruebe que el disco en el que se instalará SQL Server no está comprimido. Si intenta instalar SQL Server en una unidad comprimida, se producirá un error en la instalación de SQL Server.
- Compruebe también que los nombres del grupo de clústeres existentes no contienen caracteres no compatibles. Cuando asigne nombre a un grupo de clústeres de la instalación de clúster de conmutación por error, no debe utilizar ninguno de los caracteres siguientes:
  - Operador menor que (<)</li>
  - Operador mayor que (>)
  - o Comillas dobles (")
  - o Comillas simples (')
  - Símbolo de "y" comercial (&)

#### Consideraciones Relativas a la Red

Antes de iniciar el programa de instalación de SQL Server, compruebe que ha desactivado **NetBIOS** para todas las tarjetas de red privada.

El nombre de red y la dirección IP del servidor SQL Server no deben utilizarse para ningún otro fin, por ejemplo el uso compartido de archivos. Si desea crear un recurso compartido de archivos, utilice un nombre de red y una dirección IP diferentes y únicos para el recurso.

**Importante:** Microsoft recomienda no utilizar recursos compartidos de archivos en unidades de datos, ya que pueden afectar al comportamiento y el rendimiento de SQL Server.

Aunque SQL Server 2008 admite canalizaciones con nombre y Sockets TCP/IP sobre TCP/IP en un clúster, Microsoft recomienda utilizar Sockets TCP/IP en una configuración de clúster.



#### Instalar un Clúster de Conmutación por Error

Para instalar un clúster de conmutación por error de Microsoft SQL Server 2008, debe crear y configurar una instancia de clúster de conmutación por error mediante la ejecución del programa de instalación de SQL Server.

#### Elementos de una instancia de clúster de conmutación por error

- Una instancia de clúster de conmutación por error puede ejecutarse en uno o varios equipos que sean nodos participantes de un clúster de conmutación por error. Sólo el sistema operativo puede limitar el número de nodos participantes.
- Una instancia de clúster de conmutación por error contiene:
  - Una combinación de uno o más discos en un grupo de clústeres de Servicios de Cluster Server de Microsoft (MSCS), conocido también como grupo de recursos. Cada grupo de recursos puede contener una instancia de SQL Server como máximo.
  - o Un nombre de red para la instancia de clúster de conmutación por error.
  - Una o más direcciones IP asignadas a la instancia de clúster de conmutación por error.
  - Una instancia de SQL Server 2008 que incluya SQL Server, el Agente SQL Server y el servicio de búsqueda de texto (FTS).

#### Asignar nombre a una Instancia de Clúster de Conmutación por Error

Una instancia de clúster de conmutación por error de SQL Server siempre aparece en la red como si se tratara de un solo equipo. Debe utilizar el nombre de la instancia de clúster de conmutación por error de SQL Server para conectarse al clúster de conmutación por error de SQL Server, no el nombre del equipo del nodo en el que se está ejecutando. De este modo se garantiza que siempre se pueda conectar a la instancia de clúster de conmutación por error mediante el mismo nombre, sea cual sea el nodo en el que se ejecute SQL Server.

El nombre de la instancia de clúster de conmutación por error debe ser único en el dominio. SQL Server no escucha en la dirección IP de los servidores locales. En realidad, SQL Server sólo escucha en la dirección IP virtual creada durante la instalación de la instancia de clúster de conmutación por error de SQL Server.

SQL Server depende de claves del **Registro** y nombres de servicio diferenciados dentro del clúster de conmutación por error para garantizar que SQL Server siga funcionando tras una conmutación por error. Por tanto, el nombre que proporcione a la instancia de SQL Server, incluida la instancia predeterminada, debe ser único en todos los nodos del clúster de conmutación por error. El uso de nombres de instancia únicos garantiza que las instancias de SQL Server configuradas para realizar la conmutación por error en un solo servidor tengan claves del Registro y nombres de servicio diferenciados.

Tenga en cuenta que la limitación de discos detectada en versiones anteriores de SQL Server no afecta a SQL Server 2008. Gracias a la compatibilidad con las unidades montadas en SQL Server 2008, cada instancia de SQL Server sólo necesita un disco de clúster para los archivos de datos.



#### .

# 3- Implementando Reflejo de Base de Datos

#### Reflejo de la Base de Datos

La creación de reflejo de la base de datos es una solución de software usada principalmente para aumentar la disponibilidad de una base de datos. La creación de reflejo se implementa en cada una de las bases de datos y sólo funciona con las que utilizan el **modelo de recuperación completa**. Los modelos de recuperación simple y por medio de registros de operaciones masivas no admiten la creación de reflejo de la base de datos. Por lo tanto, todas las operaciones masivas se registran siempre por completo. La creación de reflejo de una base de datos funciona con cualquier nivel de compatibilidad con bases de datos.

Nota: No es posible reflejar las bases de datos master, msdb, tempdb o model.

La creación de reflejo de la base de datos mantiene dos copias de una sola base de datos que debe residir en diferentes instancias del Motor de base de datos de SQL Server (instancias de servidor). Generalmente, estas instancias de servidor residen en equipos de diferentes ubicaciones. Una instancia de servidor da servicio a la base de datos para los clientes (el servidor principal), mientras la otra actúa como un servidor en espera semi activa o activa (el servidor reflejado), en función de la configuración y el estado de la sesión de creación de reflejo. Cuando una sesión de creación de reflejo de la base de datos está sincronizada, la creación de reflejo de la base de datos proporciona un servidor en espera activa que admite la conmutación por error rápida sin que se produzca ninguna pérdida de datos derivada de las transacciones confirmadas. Cuando la sesión no está sincronizada, el servidor reflejado suele estar disponible como servidor en espera semi activa (con posible pérdida de datos).

#### Ventajas de la creación de reflejo de la base de datos

La creación de reflejo de la base de datos es una estrategia sencilla que ofrece las siguientes ventajas:

- Aumenta la protección de los datos.
- La creación de reflejo de la base de datos proporciona una redundancia completa o
  casi completa de los datos, en función de si el modo de funcionamiento es el de alta
  seguridad o el de alto rendimiento.
- Incrementa la disponibilidad de una base de datos.
- Si se produce un desastre en el modo de alta seguridad con conmutación por error automática, la conmutación por error pone en conexión rápidamente la copia en espera de la base de datos, sin pérdida de datos. En los demás modos operativos, el administrador de bases de datos tiene la alternativa del servicio forzado (con una posible pérdida de datos) para la copia en espera de la base de datos.
- Mejora la disponibilidad de la base de datos de producción durante las actualizaciones.
- Para minimizar el tiempo de inactividad de una base de datos reflejada, puede actualizar secuencialmente las instancias de SQL Server que participan en una sesión de reflejo de la base de datos, incurriendo sólo en el tiempo de inactividad de una única conmutación por error. Este formulario de actualización se conoce como actualización sucesiva.

#### Funcionamiento de la creación de reflejo de la base de datos

Los dos servidores, **principal** y **reflejado**, se comunican y colaboran como asociados en una **sesión de creación de reflejo** de la base de datos. Los dos asociados realizan funciones complementarias en la sesión: la **función principal** y la **función de reflejo**. En cada momento, un asociado realiza la función principal y el otro realiza la función de reflejo. Cada asociado se describe como poseedor de su función actual. El asociado que posee la función principal se



denomina servidor principal y su copia de la base de datos es la base de datos principal actual. El asociado que posee la función de reflejo se denomina servidor reflejado y su copia de la base de datos es la base de datos reflejada actual. Cuando se implementa la creación de reflejo de la base de datos en un entorno de producción, la base de datos principal es la base de datos de producción.

La creación de reflejo de la base de datos implica rehacer cada operación de inserción, actualización y eliminación que ocurre desde la base de datos principal a la base de datos reflejada tan pronto como sea posible. Para rehacer estas operaciones se envía cada entrada del registro de transacciones activo al servidor reflejado que, en secuencia, aplica las entradas del registro a la base de datos reflejada lo más rápido posible. A diferencia de la réplica, que trabaja en el nivel lógico, la creación de reflejo de la base de datos trabaja en el registro físico.

#### Modos de funcionamiento

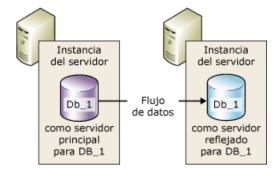
Una sesión de creación de reflejo de la base de datos se ejecuta en modo **sincrónico** o **asincrónico**. Con el funcionamiento asincrónico, las transacciones se confirman sin esperar a que el servidor reflejado escriba el registro en el disco, lo que maximiza el rendimiento. Con el funcionamiento sincrónico, una transacción confirmada se confirma en ambos asociados, pero a costa de aumentar la latencia de las transacciones.

Existen dos modos de funcionamiento de la creación de reflejo:

- Modo de Alta Seguridad: Admite el funcionamiento sincrónico. En el modo de alta seguridad, cuando se inicia una sesión, el servidor reflejado sincroniza la base de datos reflejada con la base de datos principal lo más rápido posible. Una vez sincronizadas las bases de datos, una transacción confirmada se confirma en ambos asociados, pero a costa de aumentar la latencia de las transacciones.
- Modo de Alto Rendimiento: Se ejecuta de forma asincrónica. El servidor reflejado intenta hacer frente a las entradas de registro enviadas por el servidor principal. La base de datos reflejada podría ir un poco retrasada con respecto a la base de datos principal; sin embargo, el espacio entre las bases de datos suele ser pequeño. No obstante, la diferencia puede ser considerable si el servidor principal soporta una gran carga de trabajo o el sistema del servidor reflejado se encuentra sobrecargado. En el modo de alto rendimiento, en cuanto el servidor principal envía una entrada de registro al servidor reflejado, el servidor principal envía una confirmación al cliente, sin esperar una confirmación del servidor reflejado. Esto significa que las transacciones se confirman sin esperar a que el servidor reflejado escriba el registro en el disco. Este funcionamiento asincrónico permite que el servidor principal se ejecute con la mínima latencia de transacciones, pero a riesgo de una pérdida potencial de datos.

Todas las sesiones de creación de reflejo de la base de datos sólo admiten un servidor principal y un servidor reflejado.

En la siguiente ilustración se muestra esta configuración.

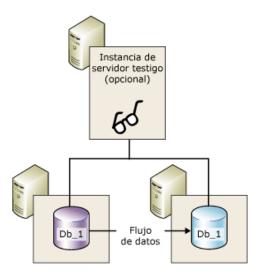




...

El modo de **alta seguridad con conmutación automática** por error requiere una tercera instancia de servidor denominada **testigo**. A diferencia de los dos asociados, el testigo no sirve a la base de datos. El testigo admite la conmutación automática por error al comprobar que el servidor principal se encuentre activo y en funcionamiento. El servidor reflejado inicia la conmutación automática por error sólo si éste y el testigo permanecen mutuamente conectados después de haberse desconectado del servidor principal.

En la siguiente ilustración se muestra una configuración que incluye un testigo.



Modo	Conmutación automática por error	Protección Completa de Pérdida de Datos
Alta Seguridad con Com. Errores Automáica	Si	SI
Alta Seguridad	No	Si
Alto Rendimiento	No	No

#### Seguridad de las Transacciones y Modos de Funcionamiento

Que el modo operativo sea asincrónico o sincrónico depende de la configuración de seguridad de las transacciones. Si utiliza exclusivamente SQL Server Management Studio para configurar la creación de reflejo de la base de datos, la configuración de seguridad de las transacciones se realiza automáticamente cuando selecciona el modo operativo.

Si utiliza Transact-SQL para configurar la creación de reflejo de la base de datos, deberá comprender cómo establecer la seguridad de las transacciones. El control de la seguridad de las transacciones está a cargo de la propiedad **SAFETY** de la instrucción ALTER DATABASE. En la base de datos que se va a reflejar, la opción **SAFETY** se establece en **FULL** o en **OFF**.

Si la opción **SAFETY** se establece en **FULL**, la operación de creación de reflejo de la base de datos es sincrónica, tras la fase inicial de sincronización. Si un testigo se establece en el modo de alta seguridad, la sesión admite la conmutación automática por error.

Si la opción **SAFETY** se establece en **OFF**, la operación de creación de reflejo de la base de datos es asincrónica. La sesión se ejecuta en modo de alto rendimiento y la opción **WITNESS** también debe establecerse en **OFF**.





#### Extremo de Creación de Reflejo de Base de Datos

La administración de conexiones en Microsoft SQL Server 2008 se basa en extremos. Un extremo es un objeto de SQL Server que permite a SQL Server comunicarse por la red. En la creación de un reflejo de una base de datos, una instancia de servidor requiere su propio extremo de creación de reflejo de la base de datos dedicado. Todas las conexiones de creación de reflejo en una instancia de servidor utilizan un único extremo de reflejo de la base de datos. Se trata de un extremo especial que se utiliza exclusivamente para recibir conexiones de creación de reflejo de la base de datos procedentes de otras instancias de servidor.

**Nota:** Las conexiones de cliente al servidor principal no usan el extremo de creación de reflejo de la base de datos.

Los extremos de creación de reflejo de la base de datos utilizan el Protocolo de control de transporte (**TCP**, Transmission Control Protocol) para enviar y recibir mensajes entre instancias de servidor en sesiones de creación de reflejo de la base de datos. Cada extremo de creación de reflejo de la base de datos se escucha en un número de puerto TCP diferente. El extremo de creación de reflejo de la base de datos de una instancia de servidor controla el puerto en el que dicha instancia escucha los mensajes de creación de reflejo de la base de datos de otras instancias de servidor.

#### Cómo Crear un Extremo de Reflejo para la Autenticación de Windows

Cada instancia de servidor de creación de reflejo de base de datos requiere un puerto de escucha único asignado al extremo de creación de reflejo de base de datos de la instancia. Una instancia de servidor sólo puede tener un extremo de creación de reflejo de base de datos, que tiene un puerto único. Un extremo de creación de reflejo de base de datos puede utilizar cualquier puerto disponible en el sistema local cuando se crea el extremo. Todas las sesiones de creación de reflejo de base de datos de una instancia de servidor escuchan en dicho puerto y todas las conexiones entrantes para la creación de reflejo de base de datos utilizan dicho puerto.

#### Roles de servidor

Los argumentos que aparecen a continuación son específicos de la opción **DATABASE\_MIRRORING**.

ROLE = { WITNESS | PARTNER | ALL }

Especifica la función o funciones de creación de reflejo de la base de datos que admite el extremo.

#### Argumentos:

- **WITNESS**: Permite al extremo realizar la función de un testigo en el proceso de creación del reflejo.
- PARTNER: Permite al extremo realizar la función de un asociado en el proceso de creación del reflejo.
- ALL: Permite al extremo realizar la función de un testigo y un asociado en el proceso de creación del reflejo.

#### Para Crear un Extremo de Reflejo utilizando la Autenticación de Windows

 Conecte con la instancia del servidor para la que desea crear un extremo de reflejo de la base de datos.





 Determine si ya existe un extremo de reflejo de la base de datos utilizando la siguiente instrucción:

```
SELECT name, role_desc, state_desc FROM
sys.database_mirroring_endpoints
```

**Importante:** Si ya existe un extremo de reflejo para la instancia del servidor, utilice ese extremo para todas las otras sesiones que establezca en la instancia del servidor.

 Para usar Transact-SQL para crear un extremo que se utilice con la autenticación de Windows, utilice la instrucción CREATE ENDPOINT. La instrucción toma la siguiente forma general:

```
CREATE ENDPOINT <Nombre_endpoint>
STATE=STARTED

AS TCP ( LISTENER_PORT = <Lista_Puertos_Escucha> )
FOR DATABASE_MIRRORING
(
[ AUTHENTICATION = WINDOWS [ <Metodo_Autorización> ]
]
[ [,] ENCRYPTION = REQUIRED
      [ ALGORITHM { <algoritmo> } ]
]
[,] ROLE = <rol> )
```

#### **Argumentos:**

- <Nombre\_endpoint>: Es un nombre exclusivo para el extremo de reflejo de la base de datos de la instancia del servidor.
- **STARTED:** Especifica que el extremo debe iniciarse y que debe empezar a escuchar las conexiones. Un extremo de reflejo de base de datos normalmente se crea en el estado STARTED. Opcionalmente, puede iniciar una sesión en el estado **STOPPED** (valor predeterminado) o **DISABLED**.
- **Lista\_Puertos\_Escucha>**: Es un número de puerto único en el que desea que el servidor escuche los mensajes de creación de reflejo de base de datos. Sólo se permite **TCP**; si se especifica cualquier otro protocolo se provoca un error. Un número de puerto sólo se puede usar una vez por sistema. Un extremo de creación de reflejo de base de datos puede utilizar cualquier puerto disponible en el sistema local cuando se crea el extremo. Para identificar los puertos que están usando los extremos TCP del sistema, utilice la siguiente instrucción Transact-SQL:

```
SELECT name, port FROM sys.tcp_endpoints
```

Importante: Cada instancia del servidor requiere sólo un puerto de escucha único.

- AUTHENTICATION: Para la autenticación de Windows es opcional, a menos que desee que el extremo utilice sólo NTLM o Kerberos para autenticar conexiones.
- <Metodo\_Autorización>: Especifica el método que se utiliza para autenticar conexiones como uno de los siguientes: NTLM, KERBEROS o NEGOTIATE. El valor predeterminado, NEGOTIATE, hace que el extremo utilice el protocolo de negociación de Windows para elegir NTLM o Kerberos. La negociación habilita conexiones con o sin autenticación, dependiendo del nivel de autenticación del extremo opuesto.
- ENCRYPTION se establece en REQUIRED de forma predeterminada. Esto significa
  que todas las conexiones con este punto final deben usar cifrado. No obstante, puede
  deshabilitar el cifrado o hacer que sea opcional en un extremo (DISABLED,
  SUPPORTED).



- <algoritmo>: Proporciona la opción de especificar estándares de cifrado para el extremo. El valor de <algoritmo> puede ser uno de los siguientes algoritmos o combinaciones de algoritmos: RC4, AES, AES RC4 o RC4 AES.
  - AES RC4 especifica que este extremo negociará el algoritmo de cifrado, dando preferencia al algoritmo AES. RC4 AES especifica que este extremo negociará el algoritmo de cifrado, dando preferencia al algoritmo RC4. Si ambos extremos especifican ambos algoritmos pero en órdenes diferentes, el extremo que acepta la conexión gana.
- <nol>
   <ln><nol>
   <ln><nol>

#### Ejemplo

```
CREATE ENDPOINT endpoint_mirroring
STATE = STARTED
AS TCP ( LISTENER_PORT = 7022 )
FOR DATABASE_MIRRORING (
    AUTHENTICATION = WINDOWS KERBEROS,
    ENCRYPTION = SUPPORTED,
    ROLE=ALL);
GO
```

#### Sesiones de Creación de Reflejo de la Base de Datos

La creación de reflejo de la base de datos se produce en el contexto de una sesión de creación de reflejo de la base de datos.

Cuando la base de datos reflejada esté preparada y las instancias del servidor estén configuradas, el propietario de la base de datos podrá iniciar la creación de reflejo de la base de datos. En cuanto se inicia la creación de reflejo, cada asociado comienza a mantener información de estado en su base de datos sobre esa base de datos y sobre el otro asociado y el testigo, si existen. Esta información de estado permite que las instancias de servidor mantengan una relación conocida como sesión de creación de reflejo de la base de datos. Durante una sesión de creación de reflejo de una base de datos, las instancias de servidor se supervisan entre sí. La información de estado se mantiene hasta que el propietario de la base de datos detiene la sesión.

Al inicio de una sesión de creación de reflejo de la base de datos, el servidor reflejado identifica el número de secuencia de registro (**LSN**) del último registro de transacciones aplicado en la base de datos reflejada y solicita al servidor principal el registro de transacciones de todas las transacciones posteriores, si las hay. En respuesta, el servidor principal envía al servidor reflejado las entradas del registro activo acumuladas desde último registro restaurado en la base de datos reflejada o enviado al servidor reflejado. El registro sin enviar que se ha acumulado en el disco lógico de la base de datos principal se conoce como cola de envío.

El servidor reflejado escribe inmediatamente el registro entrante en el disco, donde se mantiene hasta que se aplica a la base de datos reflejada. El registro que permanece en espera en el disco del reflejo se conoce como la cola rehecha. La parte del registro sin restaurar que espera en la cola rehecha es un indicador del tiempo necesario para conmutar por error a la base de datos reflejada.

El servidor principal continúa poniendo la base de datos principal a disposición de los clientes y de las conexiones de cliente. Tras el inicio de la creación de reflejo, cada vez que un cliente actualiza la base de datos principal, al escribir la transacción en el registro de la base de datos



principal, el servidor principal también envía esa entrada del registro al servidor reflejado. Ahí, el servidor reflejado escribe inmediatamente la entrada del registro en el disco como la última entrada de la cola rehecha.

En segundo plano, el servidor reflejado rehace el registro en la base de datos reflejada, entrada por entrada, empezando por la entrada del registro más antigua, tan pronto como sea posible. Al rehacer el registro, se aplican secuencialmente las entradas del registro en cola a la base de datos reflejada, comenzando por la entrada más antigua. Cada entrada del registro sólo se rehace una vez. A medida que el servidor reflejado rehace el registro, la base de datos reflejada se pone al día de forma continua. Si el servidor principal trunca o reduce el registro de la base de datos principal, el servidor reflejado también reduce el registro en el mismo punto de la secuencia.

Generalmente, al rehacer el registro, la base de datos reflejada se pone inmediatamente al nivel de la base de datos principal. El modo operativo de la sesión determina si la base de datos reflejada alcanza totalmente el nivel de la base de datos principal. En el modo de alta seguridad sincrónico, el servidor principal espera para confirmar las transacciones nuevas hasta que se graban en el disco del registro del servidor reflejado. Cuando las entradas del registro acumuladas se han enviado al servidor reflejado, la base de datos reflejada se sincroniza con la principal.

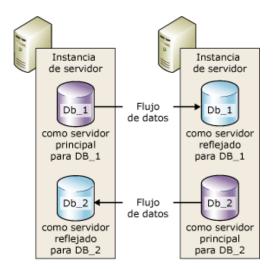
Durante una sesión, si el servidor principal no puede enviar todas las entradas del registro inmediatamente, las entradas del registro sin enviar se acumulan en la cola de envío. En el modo de alta seguridad sincrónico, después de la sincronización, las nuevas entradas del registro sin enviar se acumulan sólo cuando la creación del reflejo se pone en pausa o se suspende. Por el contrario, en el modo de alto rendimiento asincrónico, el registro sin enviar se acumula siempre que el servidor reflejado se retrasa durante la creación del reflejo, además de cuando el reflejo se pone en pausa o se suspende. La cantidad del registro sin enviar es un indicador de la posible pérdida de datos en el caso de que el servidor principal no funcione.

**Nota:** Si se producen errores al rehacer, el servidor reflejado pausa la sesión poniendo la base de datos en el estado **SUSPENDED**. Para poder reanudar la sesión, el propietario de la base de datos debe resolver la causa del error.

#### Sesiones Simultáneas

Una determinada instancia de servidor puede participar en varias sesiones simultáneas de creación de reflejo de base de datos (una por cada base de datos reflejada) con la misma instancia o con instancias de servidor distintas. Con frecuencia, una instancia de servidor sirve exclusivamente como asociado o como testigo en todas las sesiones de creación de reflejo de la base de datos. Sin embargo, puesto que cada sesión depende de las demás, una instancia de servidor puede actuar como asociado en algunas sesiones y como testigo en otras.

En la siguiente ilustración se muestran dos instancias de servidor que participan como asociados en dos sesiones de creación de reflejo. Una sesión es para una base de datos llamada Db\_1 y la otra sesión es para una base de datos llamada Db\_2.



Cada una de las bases de datos es independiente de las demás. Por ejemplo, una instancia de servidor puede ser inicialmente el servidor reflejado de dos bases de datos. Si en una de esas bases de datos se produce una conmutación por error, la instancia de servidor se convierte en el servidor principal de la base de datos en la que se ha realizado la conmutación por error y, a la vez, sigue siendo el servidor reflejado de la otra base de datos.

#### Requisitos Previos para una Sesión de Creación de Reflejo de la Base de Datos

Para que pueda comenzar una sesión de creación de reflejo, el propietario de la base de datos o el administrador del sistema deben crear la base de datos reflejada, configurar los extremos e inicios de sesión, y en algunos casos, crear y configurar certificados.

La creación de una base de datos reflejada requiere como mínimo que se realice una copia de seguridad completa de la base de datos principal y una siguiente copia de seguridad del registro, y que ambas se restauren en la instancia de servidor reflejado mediante WITH NORECOVERY. Además, antes de que pueda iniciar la creación de reflejo, si se realizan copias de seguridad del registro, adicionales posteriores a la copia de seguridad del registro requerida, también debe aplicar manualmente cada copia de seguridad del registro adicional (siempre usando WITH NORECOVERY). Tras aplicar la copia de seguridad del registro más reciente, puede iniciar la creación de reflejo.

# Conmutación de Funciones durante una Sesión de Creación de Reflejo de la Base de Datos

En el contexto de una sesión de creación de reflejo de la base de datos, las funciones principal y reflejo suelen ser intercambiables en un proceso conocido como **conmutación de funciones**. En la conmutación de funciones, el servidor reflejado actúa como el asociado de conmutación por error para el servidor principal al asumir la función principal, y al recuperar su copia de la base de datos y ponerla en conexión como la nueva base de datos principal. El servidor principal anterior, cuando esté disponible, asumirá la función reflejo y su base de datos se convertirá en la nueva base de datos reflejada. Potencialmente, las funciones pueden conmutarse como respuesta a varios errores o con fines administrativos.

Hay tres tipos de conmutación de funciones:

- Conmutación por error automática
- Conmutación por error manual
- Servicio forzado (con posible pérdida de datos).





La compatibilidad con cada forma depende del modo operativo de la sesión.

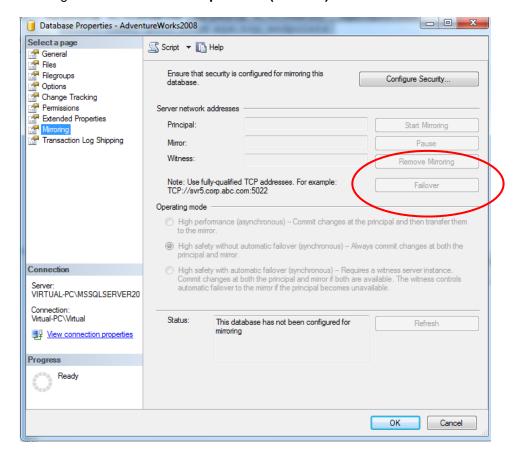
#### Conmutación por Error Manual

El modo de alta seguridad admite la conmutación por error manual. Al sincronizar la base de datos, su propietario puede iniciar una conmutación por error manual.

La conmutación por error manual se proporciona con fines administrativos.

Para realizar manualmente una conmutación por error de la creación de reflejo de la base de datos (SQL Management Studio):

- Conéctese a la instancia del servidor principal y, en el panel Explorador de objetos, haga clic en el nombre del servidor para expandir el árbol de servidores.
- Expanda Bases de datos y seleccione la base de datos de la que se va a realizar la conmutación por error.
- Haga clic con el botón secundario en la base de datos, seleccione Tareas y, a continuación, haga clic en Reflejo.
- Se abre la página Creación de reflejo del cuadro de diálogo Propiedades de la base de datos.
- Haga clic en Conmutación por error (Failover).



Para realizar una conmutación por error manual en una sesión de creación de reflejo de base de datos (Transac-SQL):

- Conéctese al servidor principal.
- Establezca el contexto de la base de datos en la base de datos master:
- Escriba la siguiente instrucción en el servidor principal:





#### ALTER DATABASE nombre basedatos SET PARTNER FAILOVER

donde database\_name es la base de datos reflejada.

Esto inicia una transición inmediata del servidor reflejado hacia la función principal. En el principal antiguo, los clientes se desconectan de la base de datos y las transacciones en curso se revierten.

**Nota:** Las transacciones que se han preparado mediante el Coordinador de Transacciones Distribuidas de Microsoft pero que aún no están confirmadas en el momento de la conmutación por error se consideran anuladas tras la conmutación por error de la base de datos.

Aparece un cuadro de confirmación. Si confirma que desea realizar una conmutación por error en la base de datos reflejada, la operación continúa y las funciones de servidor principal y reflejado se intercambian. La base de datos reflejada se convierte en la base de datos principal, y la base de datos principal se convierte en la reflejada.

#### Conmutación por Error Automática

En presencia de un **testigo**, el modo de alta seguridad admite la conmutación por error automática. La conmutación por error automática sólo se produce con la pérdida del servidor principal cuando el testigo y el servidor reflejado siguen conectados entre sí y la base de datos ya está sincronizada.

#### Forzar Servicio (con posible pérdida de datos)

Se permite forzar el servicio en modo de alta seguridad si no hay ningún testigo definido y en modo de alto rendimiento. Al perderse el servidor principal, el propietario de la base de datos puede hacer que ésta esté disponible forzando el servicio en el servidor reflejado (con posible pérdida de datos).

Para forzar el servicio en una sesión de creación de reflejo de la base de datos:

- Conéctese al servidor reflejado.
- Escriba la instrucción siguiente:

ALTER DATABASE <database\_name> SET PARTNER FORCE SERVICE ALLOW DATA LOSS

El servidor reflejado pasa inmediatamente al servidor principal y la creación del reflejo se suspende.

En la siguiente tabla se resumen las formas de conmutación por error admitidas en cada uno de los modos operativos.

	Alto rendimiento	Modo de alta seguridad sin ningún testigo	Modo de alta seguridad con un testigo
Conmutación por error automática	No	No	Sí
Conmutación por error manual	No	Sí	Sí
Servicio forzado	Sí	Sí	No

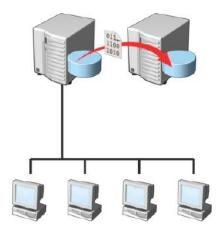


#### d

# 4- Trasvase de Registros (Log Shipping)

#### Definición

El trasvase de registros permite enviar automáticamente copias de seguridad del registro de transacciones desde una base de datos primaria de una instancia del servidor primario a una o varias bases de datos secundarias en instancias independientes del servidor secundario. Las copias de seguridad del registro de transacciones se aplican a cada una de las bases de datos secundarias de forma individual. En una tercera instancia de servidor opcional, denominado servidor de supervisión, se registra el historial y el estado de las operaciones de copia de seguridad y restauración y, opcionalmente, se activan alertas si estas operaciones no se producen según lo programado.



#### Operaciones del Trasvase de Registros

El trasvase de registros consta de tres operaciones:

- Realizar una copia de seguridad del registro de transacciones en la instancia del servidor primario.
- Copiar el archivo de registro de transacciones en la instancia del servidor secundario.
- Restaurar la copia de seguridad de registros en la instancia del servidor secundario.

El registro se puede trasvasar a varias instancias del servidor secundario. En ese caso, las operaciones 2 y 3 se repiten para cada instancia del servidor secundario.

En una configuración de trasvase de registros no se realiza automáticamente la conmutación por error del servidor primario al servidor secundario. Si la base de datos primaria deja de estar disponible, cualquiera de las bases de datos secundarias se puede poner en conexión manualmente.

Puede utilizar una base de datos secundaria para la generación de informes.

#### Configurar el Trasvase de Registros

El trasvase de registros puede configurarse utilizando SQL Server Management Studio o ejecutando manualmente una serie de procedimientos almacenados. Para la configuración del trasvase de registros es necesario realizar los siguientes pasos básicos:

- Elegir el servidor primario, el servidor secundario y un servidor de supervisión opcional.
- Crear un recurso compartido de archivos para las copias de seguridad del registro de transacciones, preferiblemente en un servidor tolerante a errores que no forme parte de la configuración del trasvase de registros. Para maximizar la disponibilidad del servidor primario, Microsoft recomienda ubicar el recurso compartido de copia de seguridad en un equipo host separado.



- Elegir una programación de copia de seguridad para la base de datos primaria.
- Crear una carpeta para cada servidor secundario en la que se copiarán los archivos de la copia de seguridad del registro de transacciones. Estas carpetas suelen residir en los servidores secundarios.
- Configurar una o más bases de datos secundarias.
- Opcionalmente, configurar un servidor de supervisión.

Al configurar un servidor secundario para el trasvase de registros, puede elegir las siguientes opciones en el cuadro de diálogo Configuración de base de datos secundaria del trasvase de registros de Management Studio para configurar la base de datos secundaria:

- Crear automáticamente una copia de seguridad en la base de datos primaria y restaurarla en el servidor secundario y, si es necesario, crear una base de datos secundaria.
- Restaurar una copia de seguridad existente de la base de datos primaria en el servidor secundario y, si es necesario, crear la base de datos secundaria.

También puede inicializar la base de datos secundaria restaurando manualmente una copia de seguridad de la base de datos.

**Importante:** La herramienta de trasvase de registros de Management Studio está diseñada sólo para hacer frente a casos simples de copia de seguridad y restauración. Para los casos más complejos, como las bases de datos con muchos archivos u opciones que no son las predeterminadas, debe realizar una copia de seguridad y restaurar toda la base de datos manualmente. Por lo general, utilice la copia de seguridad y restauración manuales para los casos que requieran un comando BACKUP o RESTORE complejo. Una vez que la base de datos secundaria se ha restaurado, utilice la herramienta de trasvase de registros de Management Studio para finalizar la configuración del trasvase de registros.

Cuando configure el servidor primario para el trasvase de registros, puede especificar la frecuencia con la que se crean las copias de seguridad del registro de transacciones en el servidor primario. Si el volumen de transacciones es alto, puede ser útil hacer una copia de seguridad frecuente del registro de transacciones para minimizar la pérdida potencial de datos.

Cómo habilitar el trasvase de registros (SQL Server Management Studio):

- Haga clic con el botón secundario en la base de datos que desea usar como base de datos primaria en la configuración del trasvase de registros y, a continuación, haga clic en **Propiedades**.
- En Seleccionar una página, haga clic en Trasvase de registro de transacciones.
- Active la casilla de verificación Habilitar ésta como base de datos primaria en una configuración de trasvase de registros.
- En Copias de seguridad de registros de transacciones, haga clic en Configuración de copia de seguridad.
- En el cuadro Ruta de red a esta carpeta de copia de seguridad, escriba la ruta de acceso de red al recurso compartido que creó para la carpeta de copias de seguridad de los registros de transacciones.
- Si la carpeta de copias de seguridad se encuentra en el servidor primario, escriba la ruta de acceso local a la carpeta de copias de seguridad en el cuadro Si la carpeta de copia de seguridad está ubicada en el servidor primario, escriba una ruta local a la carpeta (si la carpeta de copias de seguridad no está situada en el servidor primario, puede dejar este cuadro vacío).

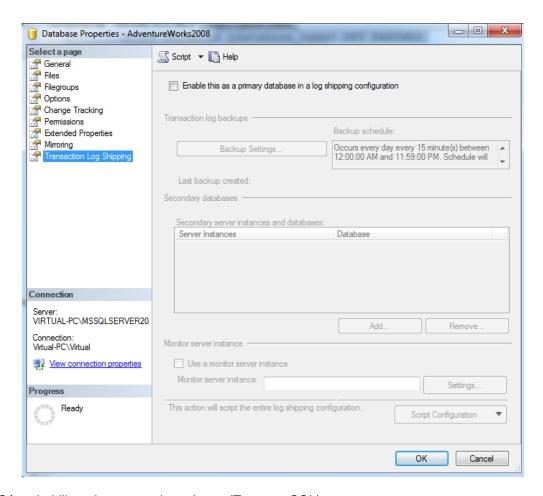
**Importante:** Si la cuenta de servicio de SQL Server en el servidor primario se ejecuta bajo una cuenta del sistema local, debe crear la carpeta de copias de seguridad en el servidor primario y especificar una ruta de acceso local a esa carpeta.



- Configure los parámetros Eliminar archivos con más de y Mostrar una alerta si no se produce una copia de seguridad tras.
- Tenga presente la programación de copia de seguridad que aparece en el cuadro Programación bajo Trabajo de copia de seguridad. Si desea personalizar la programación de su instalación, a continuación, haga clic en Programar y ajuste la programación del Agente SQL Server según sus necesidades.
- Haga clic en Aceptar.
- En Instancias de servidores secundarios y bases de datos, haga clic en Agregar.
- Haga clic en Conectar y conéctese a la instancia de SQL Server que desea utilizar como servidor secundario.
- En el cuadro Base de datos secundaria, elija una base de datos de la lista o escriba el nombre de la base de datos que desea crear.
- En la ficha **Inicializar base de datos secundaria**, elija la opción que desea utilizar para inicializar la base de datos secundaria.
  - **Nota:** Si elige que Management Studio inicialice la base de datos secundaria desde la copia de seguridad de una base de datos, los archivos de datos creados en el servidor secundario tendrán los mismos nombres que los del servidor primario, y se creará una estructura de directorios idéntica, incluida la letra de la unidad.
- En la ficha Copiar archivos, en el cuadro Carpeta de destino de los archivos copiados, escriba la ruta de acceso de la carpeta en la que deben copiarse las copias de seguridad de los registros de transacciones. Esta carpeta normalmente está situada en el servidor secundario.
- Tenga presente la programación de copia que aparece en el cuadro Programación bajo Trabajo de copia. Si desea personalizar la programación de su instalación, haga clic en Programar y, a continuación, ajuste la programación del Agente SQL Server según sus necesidades. Esta programación debe aproximarse a la programación de las copias de seguridad.
- En la ficha Restaurar, en Estado de la base de datos al restaurar copias de seguridad, elija la opción Modo sin recuperación o Modo de espera.
- Si elige la opción Modo de espera, seleccione si desea desconectar a los usuarios de la base de datos secundaria mientras se realiza la operación de restauración.
- Si desea retrasar el proceso de restauración en el servidor secundario, elija un tiempo de retraso en Retrasar la restauración de las copias de seguridad al menos.
- Elija un umbral de alerta en Mostrar una alerta si no se produce una restauración tras.
- Tenga presente la programación de la restauración que aparece en el cuadro Programación bajo Trabajo de restauración. Si desea personalizar la programación de su instalación, haga clic en Programar y, a continuación, ajuste la programación del Agente SQL Server según sus necesidades. Esta programación debe aproximarse a la programación de las copias de seguridad.
- Haga clic en Aceptar.
- En Instancia del servidor de supervisión, active la casilla de verificación Utilizar una instancia del servidor de supervisión y, a continuación, haga clic en Configuración. Importante: Para supervisar esta configuración de trasvase de registros, debe agregar ahora el servidor de supervisión. Para agregar un servidor de supervisión más adelante, deberá quitar esta configuración de trasvase de registros y reemplazarla por una configuración nueva que incluya un servidor de supervisión.
- Haga clic en Conectar y conéctese a la instancia de SQL Server que desea utilizar como servidor de supervisión.
- En **Supervisar conexiones**, elija el método de conexión que utilizarán los trabajos de copia de seguridad, copia y restauración para conectarse al servidor de supervisión.
- En **Retención de historial**, elija el período de tiempo que desea retener un registro del historial de trasvase de registros.



- Haga clic en Aceptar.
- En el cuadro de diálogo Propiedades de la base de datos, haga clic en Aceptar para comenzar el proceso de configuración.



Cómo habilitar el trasvase de registros (Transact-SQL):

- Inicialice la base de datos secundaria restaurando una copia de seguridad completa de la base de datos primaria en el servidor secundario.
- En el servidor primario, ejecute sp\_add\_log\_shipping\_primary\_database para agregar una base de datos primaria. El procedimiento almacenado devuelve el ld. de trabajo de la copia de seguridad y el ld. principal.
- En el servidor primario, ejecute sp\_add\_jobschedule para agregar un esquema para el trabajo de copia de seguridad.
- En el servidor de supervisión, ejecute sp\_add\_log\_shipping\_alert\_job para agregar el trabajo de alerta.
- En el servidor primario, habilite el trabajo de copia de seguridad.
- En el servidor secundario, ejecute sp\_add\_log\_shipping\_secondary\_primary proporcionando los detalles del servidor y la base de datos primarios. Este procedimiento almacenado devuelve el ld. secundario y los ld. de trabajo de copia y restauración.
- En el servidor secundario, ejecute **sp\_add\_jobschedule** para establecer el esquema para los trabajos de copia y restauración.



- En el servidor secundario, ejecute **sp\_add\_log\_shipping\_secondary\_database** para agregar una base de datos secundaria.
- En el servidor primario, ejecute **sp\_add\_log\_shipping\_primary\_secondary** para agregar la información necesaria acerca de la nueva base de datos secundaria al servidor primario.
- En el servidor secundario, habilite los trabajos de copia y restauración.

#### Cambiar las funciones entre el servidor primario y secundario

Después de haber realizado la conmutación por error al servidor secundario, puede configurar la base de datos secundaria para que actúe como base de datos primaria. De este modo, podrá intercambiar la base de datos primaria y la secundaria cuando sea necesario.

La primera vez que desee conmutar por error a una base de datos secundaria para convertirla en su nueva base de datos primaria, debe realizar una serie de pasos. Una vez realizados, podrá intercambiar fácilmente las funciones entre la base de datos primaria y la base de datos secundaria.

Realice manualmente la conmutación por error de la base de datos primaria a la secundaria. Asegúrese de realizar una copia de seguridad del registro de transacciones activo en su servidor primario mediante **NORECOVERY**.

Deshabilite el trabajo de copia de seguridad de trasvase de registros en el servidor primario, así como los trabajos de copia y restauración en el servidor secundario original.

En la base de datos secundaria (la que desea convertir en principal), configure el trasvase de registros mediante SQL Server Management Studio. Siga estos pasos:

- Utilice el mismo recurso compartido para crear copias de seguridad que el utilizado para el servidor primario original.
- Cuando agregue la base de datos secundaria, en el cuadro de diálogo Configuración de base de datos secundaria, escriba el nombre de la base de datos primaria original en el cuadro Base de datos secundaria.
- En el cuadro de diálogo Configuración de base de datos secundaria, seleccione
   No, la base de datos secundaria está inicializada.