

www.softdownload.com.ar
SQL Server 7

OBJETIVO

El objetivo de esta investigación, es estudiar en forma concreta una aplicación diseñada especialmente para operar dentro del ambiente de las redes de computadoras, tal como lo es Microsoft SQL Server 7.0; con el fin de poder conocer su arquitectura, las plataformas en las cuales es capaz de operar, sus métodos de instalación, los procedimientos necesarios para trabajar en él y los elementos por los cuales se encuentra constituida dicha aplicación.

INTRODUCCIÓN

SQL Server es un sistema administrador para Bases de Datos relacionales basadas en la arquitectura Cliente / Servidor (RDBMS) que usa Transact-SQL para mandar peticiones entre un cliente y el SQL Server.

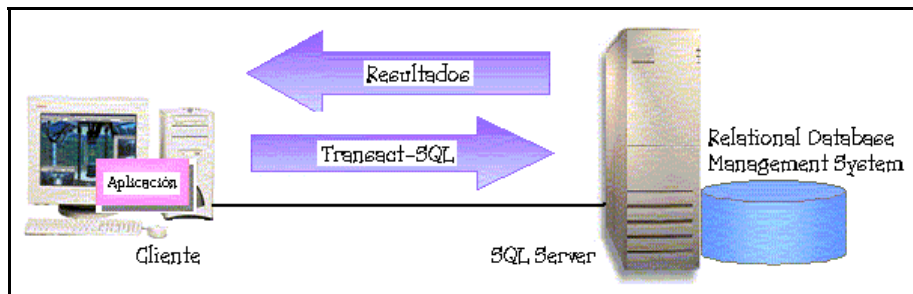


Figura 1

ARQUITECTURA CLIENTE / SERVIDOR:

SQL Server usa la arquitectura Cliente / Servidor para separar la carga de trabajo en tareas que corran en computadoras tipo Servidor y tareas que corran en computadoras tipo Cliente:

- ☐ El Cliente es responsable de la parte lógica y de presentar la información al usuario. Generalmente, el cliente corre en una o más computadoras Cliente, aunque también puede correr en una computadora Servidor con SQL Server.
- ☐ SQL Server administra Bases de Datos y distribuye los recursos disponibles del servidor (tales como memoria, operaciones de disco, etc) entre las múltiples peticiones.

La arquitectura Cliente /Servidor permite desarrollar aplicaciones para realizar en una variedad de ambientes.

SISTEMA ADMINISTRADOR PARA BASES DE DATOS RELACIONALES (RDBMS):

El RDBMS es responsable de:

- ☐ Mantener las relaciones entre la información y la Base de Datos.
- ☐ Asegurarse de que la información es almacenada correctamente, es decir, que las reglas que definen las relaciones ente los datos no sean violadas.
- ☐ Recuperar toda la información en un punto conocido en caso de que el sistema falle.

TRANSACT - SQL:

Éste es una versión de SQL (Structured Query Lenguaje) usado como lenguaje de programación para SQL Server. SQL es un conjunto de comandos que permite especificar la información que se desea restaurar o modificar. Con Transact – SQL se puede tener acceso a la información, realizar búsquedas, actualizar y administrar sistemas de Bases de Datos Relacionales.

PLATAFORMAS PARA SQL

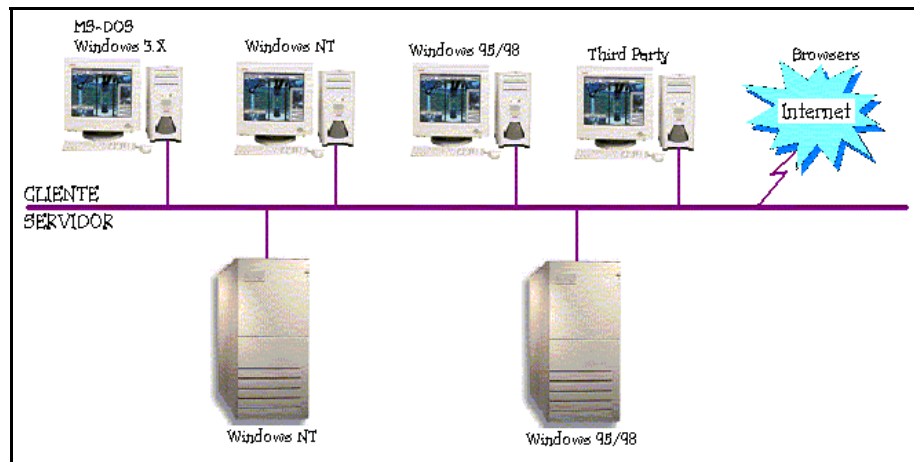


Figura 2

Los componentes Cliente y Servidor de SQL Server corren en los Sistemas Operativos mostrados en la siguiente tabla:

PLATAFORMA	COMPONENTE SERVER	COMPONENTE CLIENTE
Microsoft Win 95/98	Si	Si
Microsoft Windows NT Workstation 4.0 y posteriores	Si	Si
Microsoft Windows NT Server 4.0 y	Si	Si

posteriores		
Microsoft Windows NT Server Enterprise Edition 4.0 y posteriores	Si	Si
Windows 3.X	No	Si
MS-DOS	No	Si
Third party	No	Si (Unix, apple Macintosh)
Internet browsers	No	Si

Tabla 1.

INTEGRACIÓN DE SQL CON MICROSOFT WINDOWS NT

SQL se encuentra totalmente integrado con Windows NT y toma ventaja de muchas de sus características:

SEGURIDAD:

SQL Server está integrado con el sistema de seguridad de Windows NT. Esta integración permite acceder tanto a Windows NT como a SQL Server con el mismo *user name* y *password*. Además SQL Server una las características de encriptación que Windows NT para la seguridad en red. SQL Server está provisto de su propia seguridad para clientes *no-Microsoft*.

SOPORTE MULTIPROCESADOR:

SQL Server soporta las capacidades de multiprocesamiento simétrico (SMP) de Windows NT. SQL Server automáticamente toma ventaja de cualquier procesador adicional que sea agregado al Servidor.

SERVICIOS DE WINDOWS NT:

SQL Server corre como un servicio dentro de Windows NT, permitiendo operarlo remotamente.

MICROSOFT CLUSTER SERVER:

Es un componente de Windows NT Enterprise Edition. Soporta la conexión de dos servidores, o nudos, en un cluster para aumentar las habilidades y tener un mejor manejo de la información y las aplicaciones. SQL Server trabaja en conjunto con el Cluster Server para intercambiar papeles automáticamente en caso de que el nodo primario falle.

INTEGRACIÓN DE SQL CON MICROSOFT BACK OFFICE

SQL Server es capaz de funcionar con los productos Microsoft Back Office. Back Office es un grupo de aplicaciones para servidor que trabajan juntos para ayudar a construir *business-solutions*.

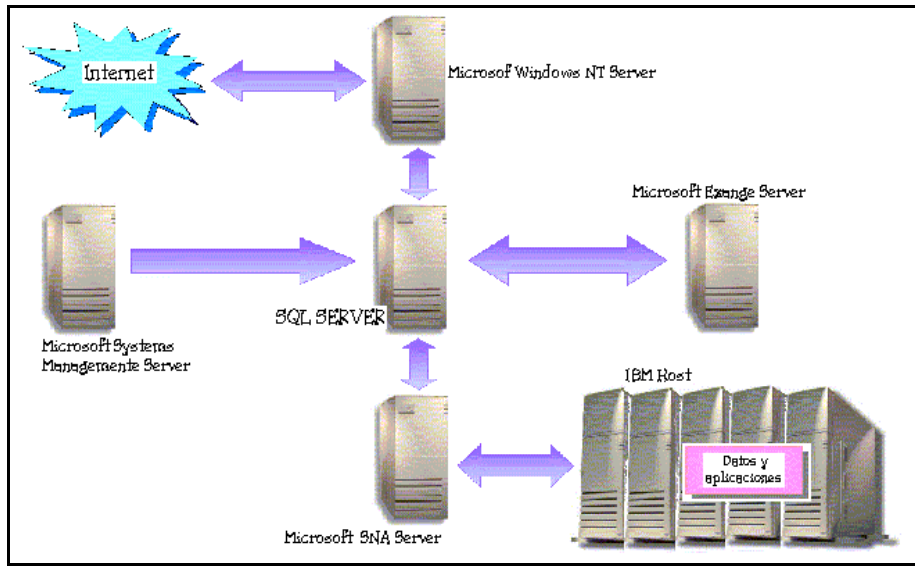


Figura 3.

La siguiente tabla describe algunas aplicaciones de Back Office que trabajan con SQL Server:

APLICACIÓN BACK OFFICE	DESCRIPCIÓN
Microsoft Windows NT Server	Permite que SQL Server se comunice con clientes de Internet
Microsoft Exchange Server	Permite que SQL Server envíe e-mails usando el servidor de Exchange u otro MAPI (Messaging Application Programming Interface).
Microsoft SNA Server	Enlaza ambientes IBM corriendo el protocolo SNA (Systems Network Architecture) con redes PC-based
Microsoft Systems Management Server	Administra el software y el hardware, usa SQL para almacenar sus bases de datos, de las cuales tiene inventarios.

Tabla 2.

SERVICIOS DE SQL SERVER

Los servicios de SQL Server incluyen MSSQLServer, SQLServerAgent, Microsoft Distributed Transaction Coordinator (MSDTC), y Microsoft Search. Aunque estos servicios de SQL generalmente corren en Windows NT, también pueden correr como aplicaciones.

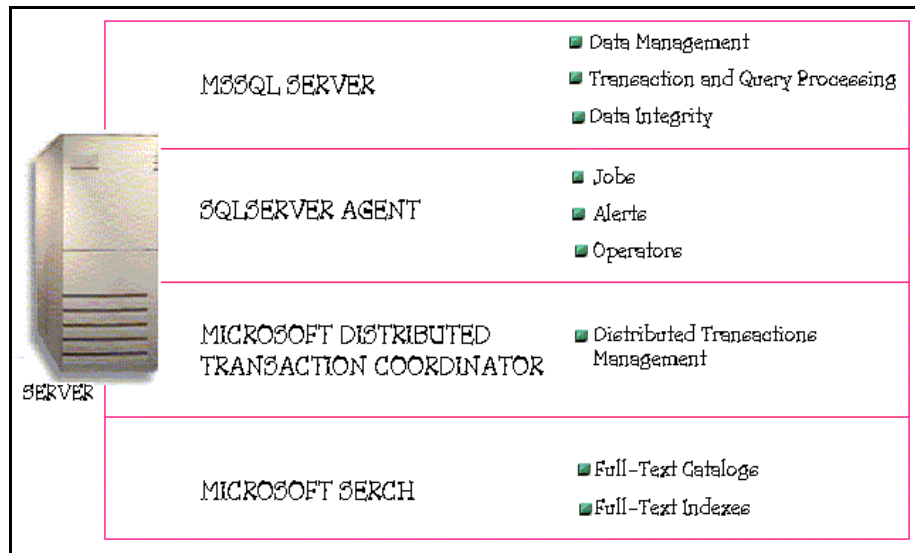


Figura 4.

SERVICIO MSSQLServer:

Este servicio es el motor de la Base de Datos. Este es el componente que procesa todas las declaraciones de Transact-SQL y administra todos los archivos que definen a la Base de Datos dentro del Servidor. Sus características son:

- Asignar los recursos de la computadora a múltiples usuarios simultáneos.
- Previene problemas lógicos, tales como sincronización de peticiones de usuarios que desean actualizar la misma información al mismo tiempo.
- Garantiza la integridad y consistencia de datos.

SERVICIO SQLServerAgent:

Este es un servicio que trabaja conjuntamente con SQL Server para crear y administrar tareas locales o externas; letras y operadores.

SERVICIO MICROSOFT DISTRIBUTED TRANSACTION COORDIRATOR:

MSDTC permite a los clientes incluir muchos tipo de datos en una transacción. Coordina la correcta realización de las transacciones distribuidas para asegurar que todas las actualizaciones en todos los servidores son permanentes; o en caso de errores, que las modificaciones son canceladas.

SERVICIO MICROSOFT SEARCH:

Este servicio es un motor de full-text que corre como un servicio de Windows NT. El soporte Full Text involucra la habilidad de emitir queries hacia los datos y la creación y mantenimiento de índices que facilitan dichos queries.

SOFTWARE DE SQL SERVER

SQL Server incluye una variedad de software para administrar y mantener al servidor, encontrando ayuda acerca de temas específicos, diseñando y creando Bases de Datos y buscando información.

SQL SERVER ENTERPRISE MANAGER SNAP-IN:

SQL Server está provisto de un cliente administrativo, que es el SQL Server Enterprise Manager, el cual es una Consola de Administración de Microsoft (MMC) de tipo Snap-in. MMC es una interfase de usuario

compartida para administración de servidor usada por Back Office. Esta consola compartida, provee un ambiente consistente para administración de herramientas.

HERAMIENTAS Y ASISTENTES PARA ADMINISTRACIÓN DE SQL SERVER:

Sql Server provee un número de herramientas administrativas y asistentes que atienden aspectos particulares de SQL Server. La siguiente tabla describe las herramientas y asistentes de SQL Server:

HERRAMIENTA GRÁFICA	APLICACIÓN
Configuración Cliente de SQL Server	Utilidad para administrar la configuración cliente para componentes de comunicación
Monitor de Funcionamiento de SQL Server	Archivo usado para integrar SQL Server con El Monitor de Funcionamiento de Windows NT, para informar las estadísticas más recientes de actividad
SQL Server Profiler	Utilidad para capturar el record continuo de la actividad del servidor
Analizador de Queries de SQL Server	Herramienta gráfica de Queries usada para analizar el plan de un query, visualizar información estadística, y administrar varios queries en diferentes ventanas al mismo tiempo.

Tabla 3.

ARQUITECTURA DE SQL SERVER

COMUNICACIÓN:

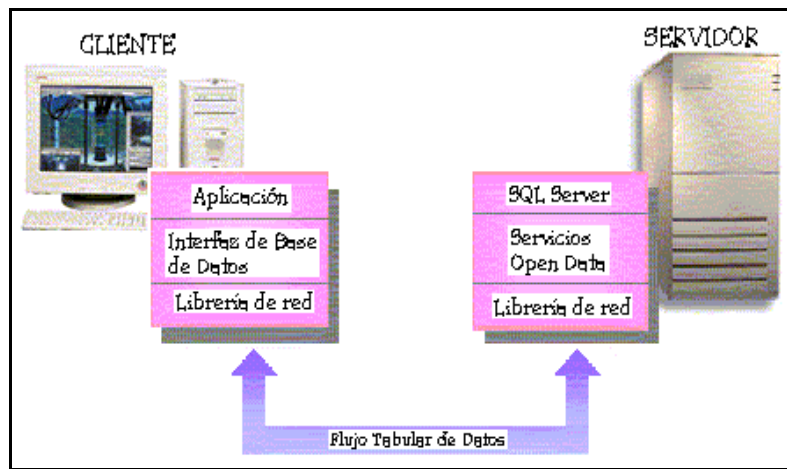


Figura 5.

SQL Server usa una arquitectura de comunicación por capas para aislar aplicaciones internas de red y protocolos. Esta arquitectura permite desplegar la misma aplicación en diferentes ambientes de red. Los componentes en la arquitectura de comunicación incluyen:

- ❏ **APLICACIÓN:** Una aplicación es desarrollada usando una aplicación de interfaz de programación para Base de Datos (API). La aplicación no tiene conocimiento de los protocolos internos de red usados para la comunicación con SQL Server.
- ❏ **INTERFAZ DE LA BASE DE DATOS:** Esta es una interfaz usada por una aplicación para mandar peticiones a SQL Server y procesar los resultados devueltos por SQL Server.

- ❏ **LIBRERÍA DE RED:** Este es un componente de Software de comunicación que empaqueta las peticiones de la Base de Datos y los resultados para transmitirlos por medio del protocolo de red apropiado. Una librería de Red, también conocida como Net-Library, debe ser instalada tanto en el cliente como en el servidor. Tanto Clientes como Servidores pueden usar más de una Net-Library al mismo tiempo, pero deben usar una Librería de Red común para comunicarse satisfactoriamente. SQL Server soporta protocolos de red tales como TCP/IP, Novell, IPX/SPX, Banyan VINES/IP, Named Pipes, y Apple Talk ADSP.
- ❏ **TABULAR DATA STREAM: (TDS)** Es un protocolo por niveles de aplicación usado para la comunicación entre un Cliente y SQL Server. Los paquetes TDS son encapsulados en los paquetes de red hechos por la protocol stack usada por las Net-Libraries.
- ❏ **SERVICIOS OPEN DATA:** Este es un componente de SQL Server que se encarga de las conexiones de red, pasando las peticiones del cliente al SQL Server para procesar y regresar cualquier resultado a los Clientes. Open Data escucha automáticamente en todas las Net-Libraries que están instaladas en el servidor.

DESARROLLO DE APLICACIONES:

Los usuarios accesan al SQL Server a través de una aplicación que está escrita con una interfaz de objetos de datos o con una API. SQL Server soporta interfaces comunes y APIs nativos de bajo nivel.

INTEFACES DE PROGRAMACIÓN DE APLICACIONES:

Una Base de Datos API define como escribir una aplicación para conectar una Base de Datos y pasar comandos a la Base de Datos. SQL Server provee soporte nativo para dos clases principales de Bases de Datos API, lo cual define la interfaz de objetos de datos que se puede usar. Las Bases de Datos API se usan para tener mayor control sobre el comportamiento y desarrollo de las aplicaciones.

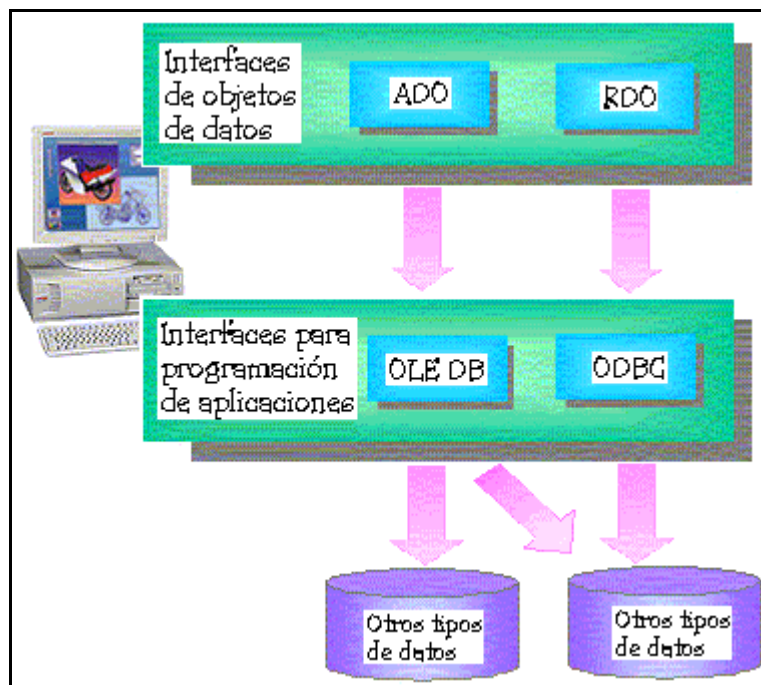


Figura 6.

- ❏ **OLE DB:** Esta es una interfaz de acceso a datos basada en el COM (Component Object Model). Soporta aplicaciones escritas usando OLE DB o Interfaces de Objetos de Datos basadas en OLE DB. Puede acceder a la información en SQL Server, otras Bases de Datos relacionales y otras fuentes de datos.
- ❏ **OPEN DATABASE CONNECTIVITY: (ODBC)** Es una interfaz por capas. Accesa directamente al protocolo SQL Server TDS y soporta aplicaciones o componentes que estén escritos usando ODBC o

interfaces basadas en ODBC. Puede acceder a los datos en SQL Server, y otras Bases de Datos relacionales, pero generalmente no puede ser usado para acceder otras fuentes de datos.

DATA OBJECT INTERFACES:

En general, estas interfaces son más fáciles de usar que las Bases de Datos API pero pueden no tener tanta funcionalidad como un API.

- ☞ ACTIVE X DATA OBJECTS: (ADO) Encapsula la OLE DB API en un modelo simplificado de objetos que reduce el desarrollo de aplicaciones y los costos de mantenimiento. ADO puede ser usado a partir de Microsoft Visual Basic, Visual Basic para Aplicaciones, Active Server Pages (ASP) y el Scripting Object Model de Microsoft Internet Explorer.
- ☞ REMOTE DATA OBJECTS: (RDO) Mapea y encapsula al ODBC API. RDO puede ser usado desde Visual Basic y Visual Basic para aplicaciones.

ADMINISTRACIÓN:

SQL Server provee una variedad de herramientas de administración para minimizar y automatizar las tareas administrativas rutinarias. Las declaraciones de Transact-SQL son el mecanismo interno usado para administrar SQL Server.

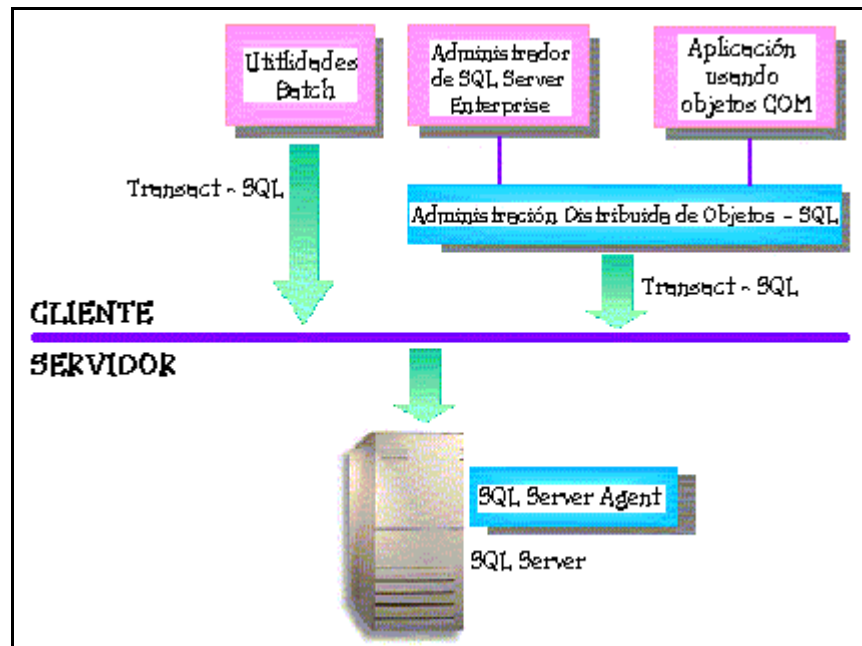


Figura 7.

ADMINISTRACIÓN DE SQL SERVER:

SQL Server puede ser administrado usando:

- ☞ Utilidades Batch incluidas en SQL Server, tales como OSQL o BCP.
- ☞ Herramientas de administración gráfica incluidas en SQL Server.
- ☞ Aplicaciones COM-compatibles: tal como Visual Basic.

ADMINISTRACIÓN DISTRIBUÍDA DE OBJETOS SQL:

(SQL-DMO) Es una colección de objetos de administración basados en COM, usados por SQL Server. SQL-DMO oculta los detalles de las operaciones Transact-SQL y es apropiado para escribir scripts de administración para SQL Server. Las herramientas de administración incluidas en SQL Server están escritas usando SQL-DMO.

SQL SERVER AGENT:

Es un servicio que trabaja en conjunto con SQL Server para desempeñar las siguientes tareas administrativas:

- ☐ Administración de Alertas: Las alertas brindan información acerca del estado de un proceso, tal como cuando un trabajo está completo o cuando ocurre un error. El agente de SQL Server monitorea la aplicación de Windows NT y genera alertas.
- ☐ Notificación: El agente de SQL Server puede enviar e-mails, o iniciar otra aplicación cuando ocurre una alerta, por ejemplo, se puede programar una alerta para que ocurra cuando una Base de Datos o cuando una transacción está casi completa o cuando un respaldo de la Base de Datos ha terminado exitosamente.
- ☐ Ejecución de Tareas: El agente de SQL Server incluye un motor de creación y planeación de tareas. Las tareas pueden ser simples operaciones de un solo paso, o pueden ser tareas complejas de varios pasos que requieren planeación. También se pueden crear pasos de las tareas con Transact-SQL, lenguajes script, o comandos del Sistema Operativo.
- ☐ Administración de Réplicas: La replicación es el proceso de copiar datos o transacciones de un SQL Server a otro. El agente de SQL Server es responsable de sincronizar los datos entre los servidores, monitorear los datos para buscar cambios y replicar la información en otros servidores.

SEGURIDAD EN SQL SERVER

SQL Server valida a los usuarios con 2 niveles de seguridad; autenticación del login y validación de permisos en la Base de Datos de cuentas de usuarios y de roles. La autenticación identifica al usuario que está usando una cuenta y verifica sólo la habilidad de conectarse con SQL Server. El usuario debe tener permiso para acceder a las Bases de Datos en el Servidor. Esto se cumple para asignar permisos específicos para la Base de Datos, para las cuentas de usuario y los roles. Los permisos controlan las actividades que el usuario tiene permitido realizar en la Base de Datos del SQL Server.

AUTENTICACIÓN DEL LOGIN:

Un usuario debe tener una cuenta para conectarse al SQL Server. Este reconoce 2 mecanismos de autenticación: Autenticación de SQL Server y de Windows NT. Cada uno tiene un diferente tipo de cuenta.

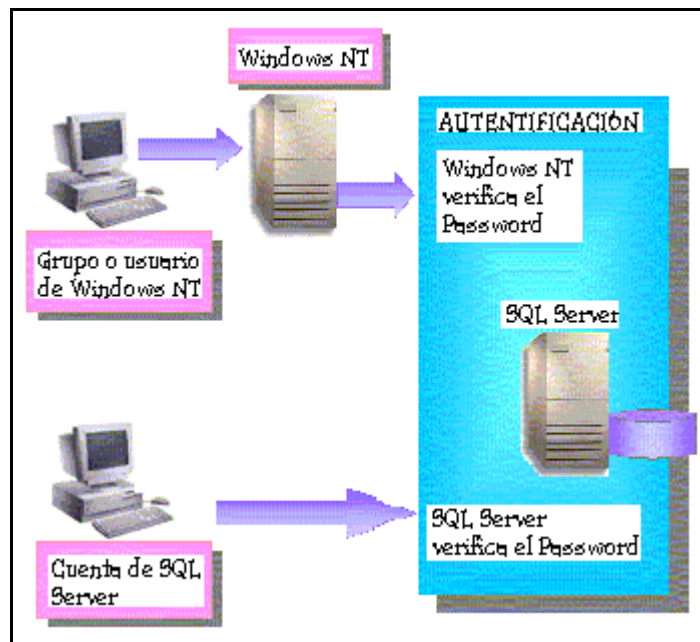


Figura 8.

AUTENTIFICACIÓN DE SQL SERVER:

Cuando se usa, un administrador del Sistema de SQL Server, define una cuenta y un password WQL Server. Los usuarios deben suministrar tanto el login como el password cuando se conectan al SQL Server.

AUTENTIFICACIÓN DE WINDOWS NT:

Cuando se usa, el usuario no necesita de una cuenta de SQL Server, para conectarse. Un administrador del sistema debe definir, ya sea cuentas de Windows NT o grupos de Windows NT como cuentas válidas de SQL Server.

MODO DE AUTENTIFICACIÓN:

Cuando SQL Server está corriendo en Windows NT, un sistema administrador puede especificar que está corriendo en uno de 2 modos de autenticación:

- ☐ Modo de autenticación de Windows NT: Sólo está autorizada la autenticación de Windows NT. Los usuarios no pueden usar cuentas de SQL Server.
- ☐ Modo mixto: Cuando se usa este modo de autenticación, los usuarios se pueden conectar a SQL Server con la autenticación de Windows NT o con la de SQL Server.

CUENTAS DE USUARIO Y ROLES EN UNA BASE DE DATOS:

Después de que los usuarios han sido autenticados, y se les ha permitido conectarse al SQL Server, deben tener cuentas en la Base de Datos. Las cuentas de usuario y los roles, identifican permisos para ejecutar tareas.

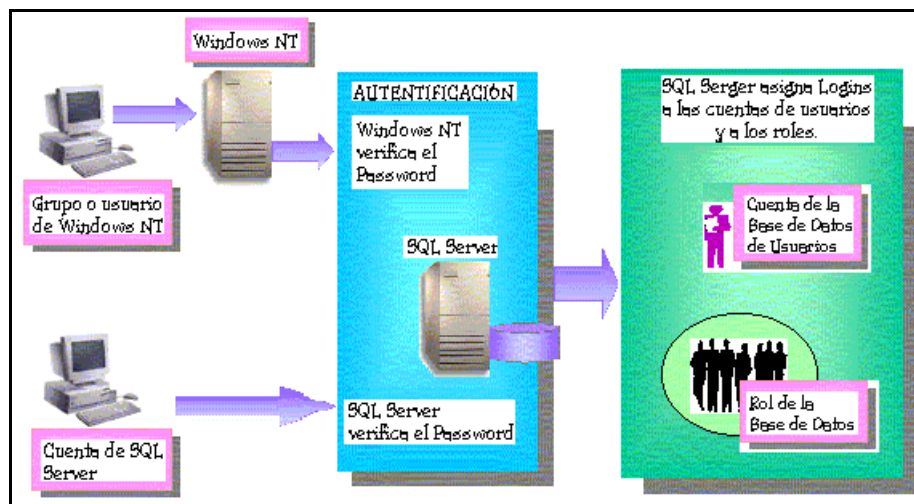


Figura 9.

CUENTAS DE USUARIOS DE LA BASE DE DATOS:

Las cuentas de usuario utilizadas para aplicar permisos de seguridad son las de usuarios, o grupos de Windows NT o las de SQL Server. Las cuentas de usuario son específicas para cada Base de Datos.

ROLES:

Permiten reunir a los usuarios en una sola unidad a la cual se le pueden aplicar permisos. SQL Server contiene roles de servidor y de Base de Datos predefinidos, para tareas administrativas comunes, de manera que pueden asignársele determinados permisos administrativos a un usuario en particular. También se

pueden crear roles de Base de Datos definidos por el usuario. En SQL Server, los usuarios pueden pertenecer a varios roles:

- ❏ Roles fijos del Servidor: Proveen agrupamientos con privilegios administrativos a nivel del Servidor. Son administrados independientemente de las Bases de Datos de usuarios a nivel servidor.
- ❏ Roles fijos de la Base de Datos: Proveen agrupamientos con privilegios administrativos a nivel de Base de Datos.
- ❏ Roles de usuarios definidos en la Base de Datos: También se pueden crear roles para Base de Datos, para representar un trabajo desarrollado por un grupo de empleados dentro de una organización. No es necesario asignar y quitar permisos a cada persona. En función de que cambia un rol, se pueden cambiar fácilmente los permisos del rol y hacer que los cambios se apliquen automáticamente a todos los miembros del rol.

VALIDACIÓN DE PERMISOS:

Dentro de cada Base de Datos, se asignan permisos a las cuentas de usuarios y a los roles para permitir o limitar ciertas acciones. SQL Server acepta comandos después de que un usuario ha accedido a la Base de datos.

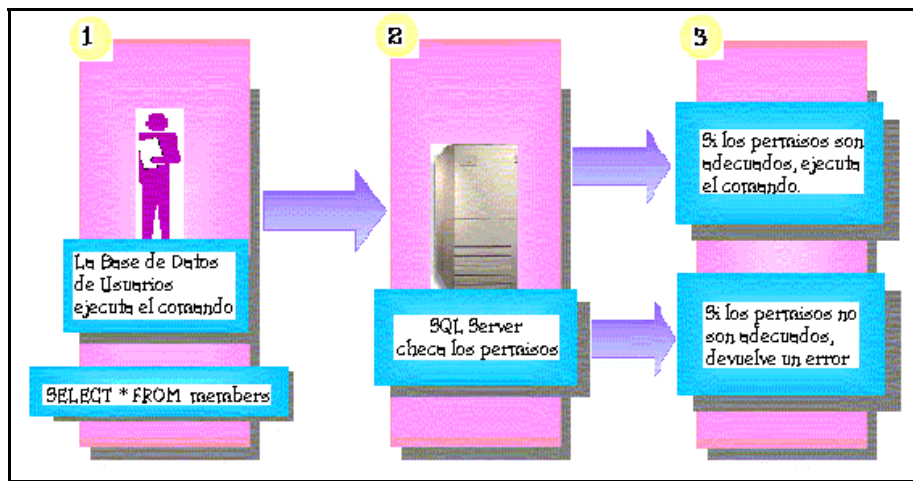


Figura 10.

SQL Server realiza los siguientes pasos cuando valida permisos:

1. Cuando el usuario realiza una acción, tal como ejecutar un comando de Transact-SQL o elegir una opción de un menú, los comandos de Transact SQL son enviadas al SQL Server.
2. Cuando SQL Server recibe un comando de Transact –SQL, chequea que el usuario tenga permiso de ejecutar dicha instrucción.
3. Después, SQL realiza cualquiera de las siguientes acciones:
 - a) Si el usuario no tiene los permisos adecuados, SQL Server devuelve un error.
 - b) Si el usuario tiene los permisos adecuados, SQL Server realiza la acción.

BASES DE DATOS EN SQL SERVER

Cada SQL Server tiene dos tipos de Bases de datos: Bases de Datos del Sistema y Bases de Datos del usuario. Las Bases de Datos del sistema almacenan información acerca de SQL Server como un total. SQL Server usa la Base de Datos del sistema para operar y administrar al sistema. Las Bases de Datos de usuarios, son Bases de Datos creadas por los usuarios. Una copia del SQL Server puede administra una o más Bases de datos de usuario.

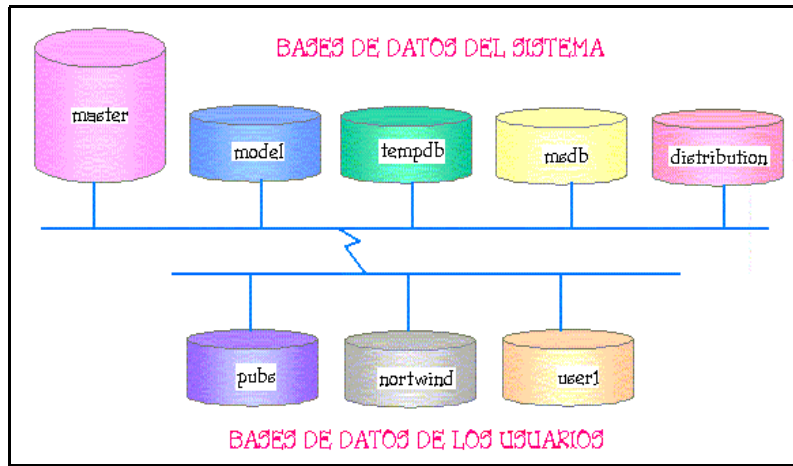


Figura 11.

BASES DE DATOS DE SISTEMA Y DE USUARIO:

Cuando SQL Server es instalado, el setup crea 4 bases de datos de sistema y 2 de usuario, de ejemplo. La Base de Datos de distribución es instalada cuando se configura SQL Server para actividades de replicación.

OBJETOS DE LA BASE DE DATOS:

Una Base de Datos, es una colección de datos, tablas y otros objetos. Los objetos de la Base de Datos ayudan a estructurar los datos y definir mecanismos para la integridad de datos.

INSTALANDO SQL SERVER

REQUERIMIENTOS MÍNIMOS DE HARDWARE:

SQL Server 7.0 requiere el siguiente hardware como mínimo:

- ☛ Computadora: DEC Alpha AXP y sistemas compatibles, Intel o compatibles (Pentium 166 MHz o superior, Pentium PRO, o Pentium II).
 - Memoria: 32 MB de RAM.
- ☛ Unidad de Disco: Un CD-ROM, más un disco duro con al menos 80 MB de espacio libre en disco para la instalación mínima.

La siguiente tabla muestra la cantidad mínima de espacio disponible en disco que requieren las diferentes instalaciones:

OPCIÓN DE INSTALACIÓN	ESPACIO EN DISCO
Completa	210 MB
Típica	185 MB
Herramientas de administración	90 MB
Mínima	80 MB

OPCIONES DE INSTALACIÓN:

El usuario puede elegir entre tres opciones de instalación: típica, mínima y personalizada. Una instalación típica instala los archivos binarios de SQL Server en el directorio Mssql7. La opción típica, instala los dispositivos de datos en el directorio Mssql\Data, y utiliza los llamados Pipes y Sockets escuchando en el puerto 1433.

Para cambiar estas configuraciones, se debe seleccionar la instalación personalizada. Si la instalación de SQL Server detecta que SQL Server 6.X está instalado en la computadora, la opción de actualización se presentará en un cuadro de diálogo. La siguiente lista muestra qué componentes se instalan o no con cada opción de instalación:

TÍPICA:

- Named Pipes, TCP/IP, y las Multi-Protocol network libraries
- ISO Character Set (1252)
- Dictionary order, case-insensitive sort order
- SQL Server Books Online
- Dirige la instalación al directorio Mssql7

MÍNIMA: (no instala)

- SQL Server Enterprise Manager
- SQL Server Profiler
- SQL Server Query Analyzer
- Version Upgrade Wizard
- Client Diagnostic Utilities
- SQL Server Books Online
- Replication objects
- MS DTC Client Support
- Development files
- Sample files
- Server Debug symbols

PERSONALIZADA:

- Ofrece elegir entre distintos protocolos, tal como: Named Pipes, TCP/IP, y Multi-Protocol que se encuentran seleccionados por default; además de NWLink IPX/SPX, AppleTalk ADSP, y Banyan VINES que también se encuentran disponibles
- Permite seleccionar el sort order. Tiene una estricta compatibilidad con 1.x y alterna selecciones de diccionario
- Provee opciones de herramientas de administración, pero siempre instala BCP, ISQL, OSQL, ODBC, y DB-Library.
- Ofrece elegir si se desea correr los SQL Server Books Online desde el disco duro o desde el CD.

Después de que los componentes ha sido seleccionados, el programa de instalación tiene información suficiente para continuar. El Setup informa al usuario que tiene suficiente información e inicia el proceso. El proceso de copiar archivos, mueve todos los archivos requeridos a la carpeta de instalación seleccionada y a

los directorios de Windows. Después, el setup detiene el MSSQL y al servicio SQL Executive si se tiene una versión previa instalada.

El siguiente paso es instalar los paquetes que son requeridos por componentes de soporte adicionales. Estos consisten en: Microsoft Data Access Components, Microsoft Management Console, MSDTC, HTML Help viewer y DLT Tape driver. La selección de paquetes está basada en las selecciones del usuario para la instalación.

Después de que los valores de registro han sido modificados, el sistema es actualizado para incluir el nuevo Mssql7, y el servicio de SQL Server inicia. Cuando el servicio de SQL Server está funcionando, el Setup inicia el Cnfgsvr.exe para configurar las configuraciones iniciales de SQL Server.

Después de que todos estos pasos se han llevado a cabo, pasa lo siguiente:

- ❏ Los Windows NT Performance Monitor entries son agregados al registro.
- ❏ La replicación es instalada.
- ❏ Se crean los grupos de programas y los íconos.
- ❏ Se actualiza el archivo Setup.iss en el directorio Windows .
- ❏ Aparece un cuadro de diálogo indicando que ha terminado la instalación.

ARCHIVOS DE INFORMACIÓN CREADOS:

Durante la instalación, se generan los siguientes archivos de información, para ayudar a localizar cualquier problema que ocurra.

- ❏ Windows\Sqstp.log
- ❏ C:\Mssql7\Log>Errorlog
- ❏ C:\Mssql7\Install\Cnfgsvr.out

INSTALACIÓN REMOTA:

La primera pantalla de instalación de SQL Server da la opción de realizar una instalación remota, pero los prerequisites deben estar previamente instalados en la computadora remota.

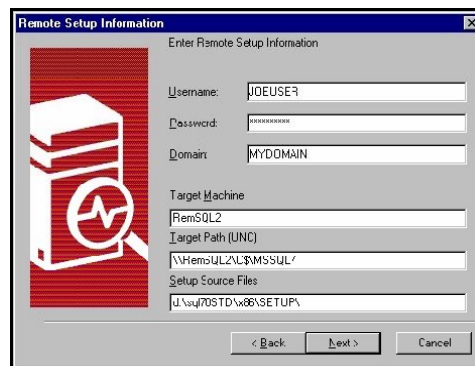


Figura 12.

INSTALACIÓN AUTOMÁTICA:

Para iniciar una instalación automática, primero se debe generar un archivo ".iss". Se puede crear este archivo iniciando la instalación de SQL Server con la opción `-r` y seguir la instalación interactuando con las opciones correctas para su sistema. Una vez que la instalación ha terminado exitosamente se tendrá el

archivo Instalar.iss en el directorio de Windows. Se puede copiar o mover este archivo a la ubicación que se desee. En instalaciones subsecuentes se podrá iniciar la instalación de SQL y especificar el archivo ".iss" como entrada, usando la opción de instalación -f1.

SI LA INSTALACIÓN NO TERMINÓ EXITOSAMENTE:

Si falló la instalación de SQL Server 7.0, hay varios archivos que pueden ayudar a determinar qué falló. El primer archivo es Sqlstp.log en el directorio de Windows. El archivo Sqlstp.log da información detallada de lo que hace la instalación. Revisando este archivo se dará una idea de lo que ocurrió durante la instalación.

Si el proceso de instalación falló en la parte de configuración, se debe revisar tanto los archivos de error en el directorio MSSQL7\Log y Cnfgsvr.out en el directorio MSSQL7\Install. La instalación de SQL Server ejecuta una aplicación llamada Cnfgsvr.exe para configurar SQL Server. Esta aplicación inicia SQL Server, se conecta a él y ejecuta los primeros comandos de instalación.

Cualquier error encontrado durante este proceso es escrito en el archivo Cnfgsvr.out. Cuando SQL Server inicia, genera un registro (log) de error que contiene los errores que SQL Server puede encontrar. Este archivo, llamado errorlog, se encuentra en el directorio

DESINSTALACIÓN DE SQL SERVER 7.0:

Para desinstalar SQL Server 7.0, use cualquiera de las siguientes opciones:

1. En el menú de Inicio, seleccionar Programas, seleccione Microsoft SQL Server 7.0, y seleccionar Desinstalar SQL Server 7.0.
2. Usar Agregar/Quitar programas en el Panel de Control para eliminar SQL Server 7.0.
3. Ejecutar un guión de desinstalación.

DESINSTALACIÓN AUTOMÁTICA:

Cuando SQL Server 7.0 se ha instalado satisfactoriamente, un archivo de desinstalación llamado Uninst.isu, es creado. Este archivo se localiza en el directorio especificado para los archivos de programa. Para iniciar una desinstalación automática, se corre el archivo UnInstallShield, Isuninst.exe, y se selecciona el archivo guión de desinstalación.

POR QUÉ SQL SERVER 7.0 NO SE INSTALA EN UNA COMPUTADORA QUE TENGA UN CHIP CYRIX:

Versiones anteriores del chip Cyrix no soportan el juego completo de instrucciones del chip Pentium. SQL Server 7.0 hace uso de algunas de esas instrucciones por lo que el programa de instalación detecta dicho chip y se niega a instalar el programa.

LIMITACIONES DE INSTALAR SQL SERVER 7.0 DESKTOP EDITION EN UN EQUIPO CON WINDOWS 95 O WINDOWS 98

Las siguientes características no están disponibles en SQL Server 7.0 Desktop si se ejecuta en un equipo con Windows 95 o Windows 98:

- ☐ Conexiones entrantes PIPE
- ☐ Autenticación Windows NT
- ☐ I/O Asíncrono
- ☐ Publicación por Transacción
- ☐ Clustering

- ☐ Búsqueda de texto completo
- ☐ Detección automática de archivos Unicode

CONFIGURANDO SQL SERVER

CONFIGURACIONES DE MEMORIA RECOMENDADAS PARA SQL SERVER PARA WINDOWS NT:

Microsoft SQL Server permite el uso de hasta 2,048 MB de memoria virtual. Este artículo describe la cantidad de memoria que debe asignar a SQL Server en distintas configuraciones de memoria.

Windows NT otorga a cada aplicación para Windows de 32-bits, una dirección de espacio virtual de 4-gigabytes (GB), de la cuál, los 2 GB de la parte baja es privada por proceso y disponible para el uso de la aplicación. La parte alta (2 GB) se reserva para uso del sistema.

El espacio de 4-GB se mapea a la dirección física de memoria por el Administrador de Memoria Virtual de Windows NT (Windows NT Virtual Memory Manager, VMM). La memoria física disponible puede ser de hasta 4 GB, dependiendo de la plataforma de soporte de hardware.

Una aplicación Windows de 32-bits tal como SQL Server solamente percibe direcciones virtuales o lógicas, no físicas. La cantidad de memoria física que una aplicación usa en un momento dado (el conjunto de trabajo) se determina por la cantidad de memoria física disponible y el VMM. La aplicación no puede controlar directamente la residencia en memoria.

Los sistemas de direcciones virtuales, como Windows NT permiten un mejor rendimiento de la memoria física, tal que la proporción de memoria virtual contra la física excede 1:1. Como resultado, programas más grandes pueden ser ejecutados en computadoras con una gran diversidad de configuraciones de memoria física. Sin embargo, en la mayoría de los casos, al usar una cantidad significativamente mayor de memoria virtual, que la suma de la combinación de elementos de trabajo de todos los procesos, resultará en un desempeño bajo.

Por lo tanto, configurar SQL Server para más memoria virtual que la cantidad de memoria física disponible, resultará en un desempeño bajo.

También se deben considerar los requerimientos de memoria del sistema operativo Windows NT, unos 12 MB aproximadamente, con algunas variaciones, dependiendo de las demandas posteriores de la aplicación. Ya que los parámetros de SQL Server se configuran hacia delante, estas demandas posteriores pueden ir en aumento conforme Windows NT requiera más memoria residente para soportar elementos adicionales como tablas de páginas, etc.

Esto resulta en una cantidad variable de memoria que podrá ser usada por SQL Server dependiendo de la configuración de memoria de la computadora. La tabla que sigue, muestra un estimado general de configuraciones de memoria y asume que se cuenta con un servidor dedicado para base de datos. Si la computadora se comparte entre varios usuarios (tal como un servidor de archivos, servidor de base de datos, y/o estaciones clientes), menor cantidad de memoria se deberá asignar a SQL Server y más se deberá dejar para el sistema operativo y otros usos.

Recuerde que estos valores solo son estimados, y se presentan para darle una idea aproximada de la ubicación de memoria de SQL Server sobre diferentes estados de memoria. Para más información, usted podrá usar las características de monitoreo de Windows NT (Performance Monitor) para determinar el comportamiento de memoria de sus sistema. Una buena fuente de información es el Volumen 3 de Windows NT Resource Kit, "Optimizing Windows NT," por Russ Blake, [ISBN 1-55615-619-7], quien dedica cerca de 600 páginas a varios aspectos de monitoreo y optimización de Windows NT y Aplicaciones Windows de 32-bits.

MEMORIA DE LA	MEMORIA APROX.
---------------	----------------

COMPUTADORA	PARA SQL SERVER
16 MB	4 MB
24 MB	8 MB
32 MB	16 MB
48 MB	28 MB
64 MB	40 MB
128 MB	100 MB
256 MB	216 MB
512 MB	464 MB
1 GB	950 MB
1.5 GB	950 MB
2 GB	1500 MB

Debido a que Windows NT asigna recursos adicionales para cada *thread spawned* (por ejemplo, se asigna 1 MB por cada *thread*), SQL Server rara vez requerirá ser configurado para usar más de 1500 MB, aun en sistemas con 2 GB o más de memoria física. Los intentos de hacerlo pueden causar un comportamiento impredecible cuando toda la memoria en los 2GB de espacio virtuales del procesador se haya utilizado.

En sistemas configurados adecuadamente para ejecutar SQL Server Enterprise Edition, dónde el espacio de memoria virtual disponible se expande a 3 GB, más memoria puede ser configurada para SQL Server. Se debe consultar la documentación de SQL Server Enterprise Edition para más guías en la configuración de memoria de estos sistemas.

La cantidad mínima de memoria para SQL Server en un procesador Intel es de 16 megabytes (MB). SQL Server para plataformas RISC requerirá de más memoria debido a la cantidad promedio de baja densidad de las instrucciones de la computadora.

Sin embargo, considerando en general al software, hardware, aplicaciones e inversión de personal en los sistemas cliente/servidor, agregar más memoria es generalmente una sabia decisión, y por comparación una inversión económica. Muchas instalaciones aseguran que 32 MB es un buen inicio, y no es poco común que se configuren los servidores con 128 MB o incluso más memoria, la cual asignan para usos en beneficio de los usuarios.

El punto en el que la memoria deja de proporcionar beneficios generales, depende completamente de cada situación, y es determinada principalmente por la ubicación o referencia de los accesos de la base de datos. El punto importante que se debe recordar es que los incrementos de memoria que son relativamente pequeños, tan solo un porcentaje del total de la memoria, rara vez aportan un beneficio significativo. Dos cosas controlan esta situación: SQL Server usa memoria principal extra como buffer de caché; y la mayoría de los estudios de estadísticas de caché indican que se presenta una curva ligeramente plana después de varios *megabytes*.

Es por esta razón, que en un equipo de 32 MB, si se otorga a SQL Server una memoria de 14 MB, 16 MB, o 18 MB, difícilmente habrá una diferencia significativa en su desempeño. Por el contrario, intentar "saturar" Windows NT con excesiva memoria para SQL Server podría resultar en un bajo desempeño debido al excesivo mapeo.

Se deberá agregar memoria física al equipo en cantidades significativas antes de asignarlas a SQL Server. Que resulte o no provechoso agregar más memoria al equipo deberá ser estudiado con anticipación.

La forma más sencilla de determinar lo anterior es usando el Monitor de Desempeño de Windows NT (Performance Monitor) para conocer el porcentaje de mapeo de SQL Server mientras se ejecuta con una carga normal de trabajo. Si este promedio es relativamente alto (más de 90 por ciento), el agregar más memoria no será redituable. Ya que esta memoria adicional se usará probablemente para realizar un caché a los datos de SQL, y por lo mismo, aumentaría el promedio de mapeo. En este caso, el promedio es alto y por lo mismo será bajo el nivel de optimización máxima.

Si el promedio es relativamente menor a 90, el adicionar memoria puede mejorar el promedio y por lo tanto el desempeño, si la localidad de referencia es tal, que puede ser "fraccionada" (bracketed) en cantidades de memoria económica y técnicamente factibles.

CONFIGURACIÓN ÓPTIMA DE SQL SERVER EN RELACIÓN CON SMS:

Microsoft System Management Server (SMS) proporciona un método de gestión centralizado de hardware y software para redes corporativas.

Es un producto muy útil que proporciona un sistema integrado para mantener el inventario del hardware, software, configuraciones de ordenadores de la red, distribución e instalación de software, gestión de aplicaciones de red y monitorización de tráfico de datos en la red. Microsoft SMS incorpora Microsoft SQL Server como sistema de gestión de base de datos back-end. SMS usa SQL Server para almacenar la base de datos de inventario. SMS recolecta y mantiene el inventario hardware y software de toda la empresa. Esta información de inventario es almacenada en una base de datos SQL Server. Existirá una base de datos de inventario por cada Primary Site de SMS que haya en la jerarquía de SMS que forme la red, si bien la base de datos SQL Server puede residir en el mismo ordenador en el que reside el site de SMS o en un ordenador distinto, dedicado de forma exclusiva a mantener la base de datos SQL Server.

Después de esta breve descripción e introducción de la interrelación entre SMS y SQL Server, pasemos a analizar las siguientes configuraciones /parámetros en SQL Server que afectan al trabajo de SMS en cualquier Primary o Central site. Microsoft SMS requiere que diversas opciones de configuración de SQL Server sean fijadas correctamente para que las prestaciones sean óptimas. A continuación, se resumen las opciones de configuración recomendadas para la ejecución de la base de datos de SMS en SQL Server.

SORT ORDER:

SMS usará para ejecutar las consultas y ordenar los datos el mismo "sort order" y "character set" que SQL Server.

SQL LOGIN ID:

Se necesitará tener un SQL Login ID para SMS al instalar un site como Primary o Central site. Este Login ID se usa durante el programa de instalación de SMS, así como para acceder a la base de datos en el servidor SQL Server una vez que SMS esté instalado y en operación. En muchos casos el Login ID será "sa", porque en general, el administrador de SMS será también el administrador de SQL Server, aunque esto no es absolutamente necesario.

SITE DATABASE DEVICES:

Microsoft SMS requiere que cada Primary site tenga su propia base de datos, y el "transaction log" debe residir en su propio device. Los devices de la base de datos del site y la propia base de datos se pueden crear de dos formas:

1. El programa de instalación de SMS puede crear los devices para la base de datos y el "transaction log". Puede crear también la propia base de datos, siempre y cuando en el propio servidor del site de SMS esté SQL Server instalado. Para poder hacer esto, el Login ID de SMS (en la base de datos) debe tener privilegios de administrador en SQL Server.
2. Si SQL Server está en un servidor remoto (distinto del servidor en el que reside el site SMS), necesitaremos crear los devices para las bases de datos en el servidor SQL Server ANTES de la ejecución del programa de instalación de SMS en el site, el cual creará la base de datos del site en los devices ya existentes de antemano. En este caso, el Login ID de SMS (en la base de datos del site) debe tener los permisos Create Database, Dump Database y Dump Transaction en la base de datos Master. Esto posibilita al programa de instalación de SMS para la creación y mantenimiento de la base de datos del site. Sin embargo, SMS borrará todos los objetos si una base de datos existe ya en dichos devices. SMS requiere una base de datos y el correspondiente "transaction log" para su propio uso. Cualquier dato existente se borrará antes de la creación de la base de datos del site en los devices especificados de SQL Server.

USO DE LA BASE DE DATOS TEMPDB:

El tamaño de la base de datos Tempdb, depende del número de ordenadores-clientes de SMS que tenga un site particular y todos sus sites hijos, para los cuales se coleccionará y almacenará el inventario en SQL Server. Un tamaño grande de Tempdb mejorará las prestaciones para consultas que contengan orden de clasificación.

En general, si hay 1.000 ordenadores-clientes en un site de SMS, se recomienda un tamaño de 5-10 MB. El tamaño por defecto de Tempdb es 2 MB y reside en el Master device. Es mejor alterar el tamaño de Tempdb en otros devices, más que incrementar su tamaño en el propio Master device.

Si un site utiliza SMSVIEWS de forma continua, el tamaño de Tempdb debería ser incrementado para facilitar el procesamiento de las consultas y vistas de forma apropiada. Microsoft NO recomienda ubicar la Tempdb en RAM en un servidor SQL Server que sea además site server de SMS. En SQL Server 6.5 se pueden cambiar las opciones de configuración de SQL Server usando el interface de usuario del SQL Enterprise Manager, haciendo clic en "SQL Server Configure" del menú "Server". A continuación, escoger la ficha "Configuration". En SQL Server 6.5 se pueden cambiar las opciones de configuración usando el procedimiento almacenado SP_CONFIGURE.

USER CONNECTIONS:

SQL Server debería tener al menos 5 user connections configuradas de forma separada para su uso por SMS. Sin embargo, en la práctica, es mejor tener al menos de 10 a 15 user connections configuradas para el uso exclusivo por Microsoft SMS.

Es importante fijar las "user connections" apropiadamente. Cada "user connection" ocupa 40 KB de RAM, por tanto este valor viene determinado por la cantidad de memoria dedicada a SQL Server y por el número de conexiones concurrentes requeridas.

Cada site server de SMS que reporta los datos de inventario a un servidor SQL Server requiere al menos 10 conexiones. Cada logon server para el/los site/s server requiere al menos una conexión adicional. Además, cada instancia en ejecución del programa Administrator de SMS y del SQL Enterprise Manager requieren al menos una conexión más.

MEMORIA:

El parámetro óptimo depende de cuanta RAM esté instalada en el servidor SQL Server y de qué otras aplicaciones estén en ejecución en dicho servidor. En un servidor dedicado para SQL Server, con 32 MB de memoria física RAM, podemos configurar 16 MB para uso por SQL Server.

Esto posibilitaría que Microsoft Windows NT Server tuviera suficiente memoria para la ejecución de sus propios procesos y evitaría la paginación a disco duro.

Es importante fijar la memoria para SQL Server de forma apropiada, es decir, fijar la cantidad de RAM dedicada a SQL Server. Este parámetro depende de la cantidad de RAM física que tenga el servidor y del uso y requerimientos de prestaciones de SQL Server.

La memoria está designada en bloques de 2 KB. Por ejemplo, para un servidor dedicado a SQL Server con 128 MB de RAM, podemos fijar la memoria para SQL Server a 64 MB (32.768 bloques de 2-KB). Sin embargo, en un servidor con SQL Server y un site de SMS con 128 MB de RAM, podemos dedicar sólo para SQL Server 40 MB (20.480 bloques de 2-KB).

OPEN OBJECTS:

Para SMS, los "objetos abiertos" en SQL Server deberían estar configurados a 5.000-10.000. Normalmente, se fijan los "objetos abiertos" a 5.000-7.000, dependiendo del tamaño del site y de los sites hijos bajo el site central. El valor por defecto de "open objects" de SQL Server es 500, que no es adecuado ni siquiera para un pequeño servidor con SQL Server que sea también site de SMS.

Los síntomas de que el parámetro "open objects" está demasiado bajo en un servidor SQL Server son las bajas prestaciones de SMS o SQL Server, una acumulación (backlog) de ficheros deltamifs o .mif en la estructura de directorios de SMS, o retrasos en el inventario, la distribución de paquetes y el procesamiento de MIFs de estado de jobs.

LOCKS:

Sólo para SMS, la configuración por defecto de 5.000 bloqueos en SQL Server debería ser suficiente. Sin embargo, si el servidor tiene otras bases de datos activas, este parámetro debería ser apropiadamente ajustado.

SYNCHRONIZE TIME:

Si SQL Server está en un servidor remoto (distinto del servidor en el que reside el site SMS), ambos servidores (SMS site server y SQL Server) se deberían sincronizar con la hora actual del site server SMS. En Microsoft Windows NT Server debemos usar el comando NET TIME para realizar esta sincronización.

ACTUALIZACIÓN:

Hay varios aspectos a considerar cuando se trate de actualizar SMS y SQL Server a sus respectivas nuevas versiones. A modo de resumen:

1. Microsoft SMS 1.0 es compatible con servidores SQL Server 4.21a.
2. Microsoft SMS 1.1 es compatible con servidores SQL Server 4.21a, 6.0 y 6.5.
3. Microsoft SMS 1.2 es compatible con servidores SQL Server 6.0 y 6.5.

En la actualización el orden es importante. Hay diferencia entre si se actualiza primero SMS o SQL Server.

En el caso de SMS 1.0 y SQL Server 4.21a, los sites de SMS se deberían actualizar primero a SMS 1.1 y, posteriormente, SQL Server debería ser la versión 6.x. Esto se debe a que SQL Server 6.x es incompatible con SMS 1.0. Después, SQL Server 6.0 se puede actualizar a la versión 6.5 sin ningún problema, puesto que los site servers de SMS ya estarán todos ejecutando SMS 1.1.

Para el caso de una actualización de SMS 1.1 a SMS 1.2, el primer paso sería actualizar SQL Server de la versión anterior (4.21a) a la versión SQL Server 6.x, y en segundo lugar pasaríamos a la actualización de SMS de la versión 1.1 a la versión 1.2.

NETWORK SUPPORT:

El soporte de red "Named Pipes" es un requerimiento que SMS usa para comunicarse con la base de datos que SMS mantiene en SQL Server.

Podemos cambiar el soporte de red de SQL Server ejecutando el programa de instalación de SQL Server, seleccionando la opción "Change Network Support" y escogiendo "Named Pipes" como red instalada.

OPCIONES RECOMENDADAS PARA LAS BASES DE DATOS TEMPDB Y SMS:

Opciones activadas para la base de datos Tempdb:

- Select Into/ Bulk Copy
- Truncate Log on Checkpoint

Opciones desactivadas para la base de datos Tempdb:

- Columns Null by Default
- No CheckPoint on Recovery
- Single User
- DBO Use Only
- Read Only

Opciones activadas para la base de datos SMS:

- Truncate Log on CheckPoint (si se realiza un procedimiento planificado de backup o dump diario de SQL Server esto no es necesario)

Opciones desactivadas para la base de datos SMS:

- Select Into/ Bulk Copy
- Columns Null by Default
- No CheckPoint on Recovery

Single User
DBO Use Only
Read Only

En SQL Server 6.5 se pueden cambiar las opciones de una base de datos usando el interface de usuario del "SQL Enterprise Manager" y haciendo clic en "Databases" del menú "Manage". A continuación, hacer doble-clic en la base de datos a editar y escoger la ficha "Options". También es posible hacer doble-clic en el nombre de la base de datos en la ventana del "Server Manager".

En SQL Server 6.5 se pueden cambiar las opciones de una base de datos usando el procedimiento almacenado SP_DBOPTION.

CUÁNDO USAR TEMPDB EN RAM:

Microsoft SQL Server proporciona una poderosa función llamada "tempdb en RAM." Esta función permite a la base de datos temporal tempdb, que se utiliza para espacio de trabajo al ordenar datos y crear tablas temporales en algunas operaciones ligadas entre sí, y convertirse en memoria residente únicamente. En algunas situaciones específicas, ésto puede ofrecer una ventaja en el desempeño. Sin embargo, si tempdb en RAM se usa inapropiadamente, puede consumir memoria que debería ser usada para sistema de caché de SQL Server y ésto puede mermar su desempeño.

En la mayoría de los casos, el RAM disponible es mejor utilizado como caché de información, más que como locación para tempdb. La información en tempdb se almacenará a sí misma mediante el algoritmo LRU del sistema caché de SQL.

Ésto es análogo a la decision de usar un disco RAM contra usar el programa caché smartdrive en una estación de trabajo de Microsoft Windows. En este caso, el RAM utilizado para el disco RAM no está disponible para smartdrive, y puede usarse solamente para objetos asignados específicamente en el disco RAM. En algunos casos donde su conocimiento del ambiente de la aplicación es tal que sabe que la mayoría de los accesos van a unos pocos archivos, y que si son lo suficientemente pequeños para ajustarse en el disco RAM, y los accesos restantes al disco tienen una referencia de locación muy pobre que ninguna cantidad factible de caché proporcionará un buen índice de aciertos, entonces el disco RAM será superior a smartdrive. Sin embargo, en la mayoría de los casos smartdrive será superior, ya que almacena todos los accesos (no sólo aquellos localizados en el disco RAM).

Similarmente, el uso de tempdb en RAM puede acelerar las operaciones de tempdb pero agotará la memoria disponible para el caché de SQL, lo que puede disminuir el índice de aciertos de la memoria caché. La memoria usada para tempdb en RAM es localizada separadamente de la reserva vista en sp_configure "memoria", y el servidor debe ser configurado apropiadamente. Por ejemplo, si utiliza 10MB para tempdb en RAM, el parámetro "memoria" de sp_configure de SQL NT debe reducirse en 10MB para liberar memoria para esta operación. En contraste, si se da toda la memoria disponible a SQL Server (contrario a configurar memoria aparte para tempdb en RAM) puede incrementarse el índice de aciertos de caché. El sistema caché de SQL puede almacenar todas las operaciones I/O, incluyendo tempdb.

Debido a la disponibilidad limitada de RAM en muchas máquinas, ésto restringe el tamaño disponible de tempdb cuando se usa en RAM. Si los requerimientos imprevistos de crecimiento de tempdb se llegan a dar, ésto podría convertirse en un problema. No es conveniente tener a tempdb parcialmente en RAM y parcialmente en disco. Tampoco es conveniente excederse de la memoria física disponible cuando se usa tempdb en RAM. Aún si ésto funcionara, las referencias de tempdb serían copiadas al disco, eliminando cualquier beneficio posible. Consulte la "Guía para configuración de SQL NT" para configurar tempdb en RAM.

Si usar el RAM disponible para el sistema de caché de SQL es generalmente mejor que usar una buena parte de tempdb en RAM, ¿habrán algunos casos cuando ésto no sea verdad? Sí, si todas las siguientes condiciones aplican, usar tempdb en RAM puede ser conveniente:

1. Tiene una cantidad considerable de sistema RAM. Ésto normalmente equivale a más de 64 MB, donde cantidades como 128 MB ó más son más comunes.
2. Sus aplicaciones tienen una localidad de referencia tal que el índice de aciertos de caché de SQL NT es deficiente, aún con suficiente memoria caché disponible. Éste índice de aciertos puede ser monitoreado con el Monitor de desempeño (Performance Monitor) como el objeto "SQLServer", y el contador como "Índice de aciertos de memoria caché " (Cache Hit Ratio).
3. Sus aplicaciones hacen muchas operaciones en tempdb. En vez de adivinar si esta condición aplica, se puede monitorear la operación usando sp_lock para observar la actividad lock en tempdb mientras se ejecutan las búsquedas. También, puede hacer lo siguiente, o algo similar:

```
SELECT SUM(DPAGES) FROM TEMPDB..SYSINDEXES
```

4. Ya sea interactivamente ó desde un archivo de lotes (batch file) sin fin para monitorear el consumo espacio de tempdb.
5. Las operaciones en tempdb se compactan de tal manera que se ajustarán en tempdb gracias a la configuración de RAM.

Si se decide por colocar a tempdb en RAM, es mejor verificar objetivamente el beneficio de desempeñar esta operación. Seleccione una búsqueda que tipifique las operaciones más frecuentes en tempdb. Ejecute ésto varias veces, poniendo atención al tiempo de ejecución. Entonces vuelva a configurar tempdb en RAM, ejecute las mismas búsquedas y notará la diferencia. Si la mejora obtenida no es muy significativa, probablemente sea mejor regresar RAM al sistema de caché de SQL.

Colocar tempdb en RAM es seguro y no afectará la integridad ó recuperabilidad de la base de datos. Ésto se debe a que tempdb sólo se usa para operaciones intermedias, y se vuelve a crear totalmente cada vez que el servidor se arranca.

Tempdb en RAM es una herramienta importante de desempeño disponible para casos donde el análisis demuestra que es benéfico. En algunos casos puede proporcionar una mejora significativa en el desempeño, pero no debe dársele un uso indiscriminado

TRABAJANDO CON SQL SERVER

DISEÑO DE UNA APLICACIÓN PARA SQL SERVER:

La planeación del diseño de una Base de Datos requiere del conocimiento de las funciones del usuario que se desean modelar, y los conceptos de la Base de Datos y características que se usan para representar dichas funciones. Antes de diseñar una aplicación para SQL Server es importante pasar tiempo diseñando una Base de Datos que refleje exactamente las funciones realizadas por el usuario. Una Base de Datos bien diseñada requiere cambios mínimos y generalmente se desarrolla con mayor eficiencia. La arquitectura que se elija, afectará la forma en que se desarrolle, administre y visualice la aplicación de Software.

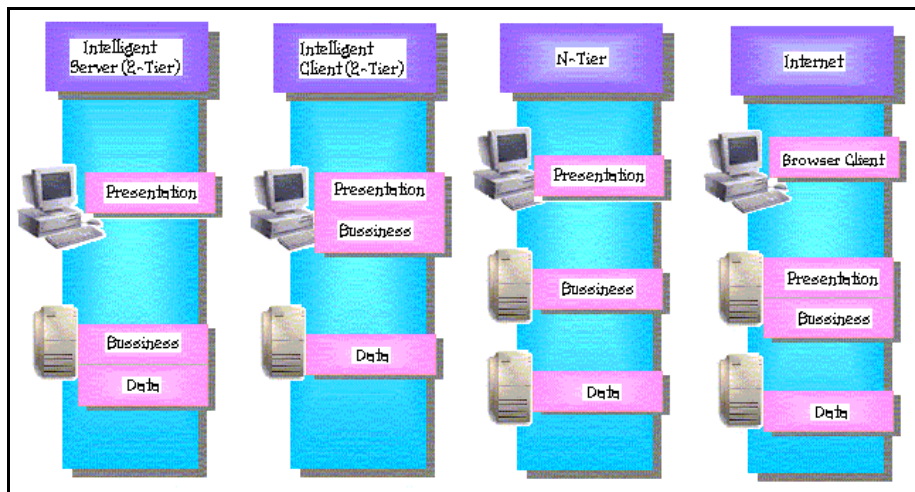


Figura 13.

ARQUITECTURA DE SOFTWARE:

Se puede elegir de entre muchas arquitecturas de aplicación para implementar aplicaciones cliente/servidor. Sin embargo elegir un enfoque de aplicación por capas permite flexibilidad y elegir entre opciones de administración. Las aplicaciones de Software se pueden dividir entre capas lógicas, las cuales pueden residir físicamente en uno o más servidores.

DISEÑO ARQUITECTÓNICO:

Las opciones típicas para visualizar una aplicación son:

- ❏ INTELLIGENT SERVER (2-TIER): La mayor parte del proceso ocurre en el servidor con los servicios de presentación realizados en el Cliente. En muchas instancias, la gran mayoría de la lógica de los servicios es implementada en la Base de Datos. Este diseño es útil cuando los clientes no tienen los suficientes recursos para procesar esta lógica. Sin embargo, el servidor puede volverse un *cuello de botella* porque los servicios de Base de Datos y los de aplicación compiten por los mismos recursos de Hardware. Un ejemplo de este diseño son las aplicaciones asociadas diseñadas para un punto de vista de una Base de Datos céntrica.
- ❏ INTELLIGENT CLIENT (2-TIER): La mayor parte del proceso ocurre en el cliente, con los servicios de datos realizados en el Servidor. Este diseño es ampliamente usado. Sin embargo el tráfico en la red puede ser pesado y alargar las transacciones, lo que puede afectar la ejecución. Un ejemplo de este diseño son las aplicaciones desarrolladas para pequeñas empresas con productos tales como Microsoft Access.
- ❏ N-TIER: el proceso es dividido entre un servidor de Base de Datos, un Servidor de Aplicación y clientes. Este enfoque separa los servicios lógicos de los de datos, y se pueden agregar fácilmente más servidores de aplicación o de Base de Datos, según se requiera. Sin embargo, el potencial de complejidad aumenta, y este enfoque puede ser más lento para pequeñas aplicaciones. Las aplicaciones de empresa multienlazada sin ejemplo de este diseño.
- ❏ INTERNET: El proceso es dividido en 3 capas, con los servicios de presentación y los de aplicación residen en el Servidor Web, y los clientes usan simples browsers. Cualquier cliente que tenga un browser puede ser soportado, y el Software no necesita estar en el cliente. Un ejemplo de este diseño es un sitio Web que usa muchos servidores Web para administrar las conexiones de los clientes, y una base de Datos de SQL Server que atiende peticiones de datos.

IMPLEMENTACIÓN DE UNA BASE DE DATOS EN SQL SERVER:

Implementar una Base de Datos en SQL Server significa planear, crear y mantener un número de componentes interrelacionados. La naturaleza y complejidad de una aplicación de Base de Datos, así como el proceso de planearla puede variar enormemente. Por ejemplo, una Base de Datos puede ser relativamente

simple, diseñada para ser usada por una sola persona, o puede ser grande y compleja, diseñada para atender todas las transacciones de cientos o miles de clientes.

En cuanto al tamaño y complejidad de la Base de Datos, generalmente la implementación de una Base de Datos involucra:

1. Diseñar la Base de Datos de manera que la aplicación optimice el uso de Hardware y permita crecimiento futuro, identificar y modelar objetos de la Base de Datos y aplicaciones de lógica, y especificar tipos de información para cada objeto y tipo de relación.
2. Crear la Base de Datos y los objetos, incluyendo tablas, mecanismos de integridad de datos, entrada de datos y objetos, índices y seguridad.
3. Probar la aplicación y la base de Datos. Cuando se diseña una Base de Datos, se desea asegurar que la Base de Datos realiza las funciones importantes en forma rápida y correcta.
4. Planear el funcionamiento, lo que incluye analizar la carga de trabajo y recomendar una configuración óptima para la Base de Datos de SQL Server.
5. Administrar la aplicación, lo que incluye configurar a los clientes y servidores, monitorear el funcionamiento del server, administrar tareas, alertas y operadores, administrar seguridad y procedimiento de backup de la Base de Datos.

ADMINISTRACIÓN DE UNA BASE DE DATOS DE SQL SERVER:

Abarca 3 puntos importantes:

1. Instalar y configurar SQL Server y establecer la seguridad de red.
2. construir las Bases de Datos: incluye asignar espacio en disco para la Base de Datos y la conexión, transferir datos de y hacia la Base de Datos, definir e implementar la seguridad de la base de Datos y crear trabajos automatizados para ciertas tareas.
3. Administrar actividades entrantes, como la importación y exportación de datos, respaldar y restaurar la base de Datos y la conexión, y monitorear la Base de Datos. Una tarea opcional es automatizar algunas de estas tareas administrativas recurrentes.

CÓMO CONVERTIR UNA BASE DE DATOS DE ACCESS A SQL SERVER:

La forma más fácil de convertir una base de datos a SQL Server es usar el asistente Upsizing Wizard. El Upsizing Wizard:

- Preserva la estructura de la base de datos incluyendo los datos, índices, valores por defecto, etc.
- Automáticamente convierte las reglas de validación y valores por defecto de Access a los equivalentes apropiados de SQL Server.
- Mantiene la relación entre tablas y la integridad de las referencias después de la conversión.

MÁS INFORMACIÓN:

Para ejecutar el Upsizing Wizard desde Access 2000, haga clic en el menú Tools (Herramientas), señale Database Utilities (Utilerías de base de datos) y haga clic en Upsizing Wizard.

Para ejecutar el Upsizing Wizard desde Access 97, debe primero descargar las herramientas del siguiente sitio:

<http://www.microsoft.com/accessdev/prodinfo/aut97dat.htm>

Si tiene una versión anterior de Microsoft Access, ya sea puede:

- Primero actualizar su versión de Access ya sea a Access 97 o Access 2000 y entonces ejecutar el Upsizing Wizard.

- ❏ Utilizar Data Transformation Services (Servicios de transformación de datos, DTS) de SQL Server para importar datos desde la base de datos de Access a la base de datos de SQL Server.

ACCESS 2000:

Si está usando Access 2000, puede usar lo siguiente:

- ❏ Desde el menú File (Archivos), señale Nuevo (Nuevo) y luego seleccione New Project from Existing Database (Nuevo proyecto desde la base de datos existente).

NOTA: Esta opción crea un proyecto de Microsoft Access (ADP), que automáticamente usa el Microsoft Data Engine (Motor de datos de Microsoft, MSDE) o SQL Server al final del proceso con un archivo ADP al inicio del proceso.

ALGUNOS TIPS PARA TRABAJAR CON SQL SERVER:

- ❏ Si se tiene un servidor que tiene instalado SQL Server 6.X, se puede instalar SQL Server 7.0, pero no se podrá ejecutar simultáneamente SQL Server 6.x y SQL Server 7.0. La instalación de SQL Server agrega una versión con un interruptor, que cambia entre SQL Server 6.x y SQL Server 7.0. Si está instalando SQL Server 7.0 junto con SQL Server versión 6.x en la misma computadora, no se debe instalar SQL Server 7.0 en el mismo directorio que SQL Server 6.x.
- ❏ No se necesita Microsoft Internet Explorer 4.01 Service Pack 1 para instalar sólo las herramientas de conectividad cliente. Si sólo desea instalar las herramientas de conectividad cliente, no se necesita de Internet Explorer 4.01 Service Pack 1. Sin embargo, si se pretende instalar las herramientas de administración o SQL Server 7.0 Books Online, se necesitará Internet Explorer 4.01 Service Pack 1.
- ❏ Es posible administrar bases de datos de SQL Server 6.5 desde SQL Server Enterprise Manager 7.0, si previamente se han instalado las herramientas de SQL Server 6.5 en la computadora cliente y ésta ha sido actualizada a la versión 7.0. Cuando intenta conectarse a SQL Server 6.5 usando SQL Server Enterprise Manager 7.0, abrirá la versión 6.5 de SQL Enterprise Manager.
- ❏ Actualmente no es posible instalar SQL Server 7.0 en un servidor ejecutando Windows NT 4.0 Terminal Server Edition, pero se está estudiando para posiblemente incluirlo en un futuro Service Pack de SQL Server 7.0.

CONCLUSIONES

PROS Y CONTRAS DE SQL SERVER 7.0

LOS PROS:

SQL Server 7.0 está plagado de nuevas características. Vamos a repasar algunas de las más significativas. Asignación Dinámica de Recursos. La asignación dinámica de recursos del SQL Server 7.0 es una característica muy útil. La asignación dinámica de recursos permite la escalabilidad del uso del disco y memoria para acomodarse a las necesidades de la base de datos en cada momento. Esta flexibilidad permite un mejor rendimiento y simplifica la administración del software. La eliminación de dispositivos también es una ventaja añadida.

El Soporte 9x para Windows. El soporte para la plataforma Win9x aumenta significativamente la base de aplicaciones posibles para el SQL Server 7.0. Al usarlo con la replicación distribuida de fusión del SQL Server 7.0, el soporte Win9x permite que las empresas con sucursales pequeños que incluyen solo unos pocos sistemas Win9x en cada oficina remota aprovechen de las aplicaciones del Servidor SQL a través de la empresa entera.

El Analizador Gráfico de Consultas. El programa ISQL/w del Servidor SQL 6.5 es una herramienta útil y a menudo necesaria para construir y ejecutar las sentencias interactivas de SQL. El nuevo Analizador de Sentencias del SQL Server 7.0 representa un paso adelante dentro de este programa. No solo se puede construir unos procedimientos guardados y ejecutar unas consultas interactivas, sino que también se puede enseñar gráficamente los pasos que el procesador de consultas usa para ejecutar la consulta.

Los Servicios OLAP del Servidor SQL de Microsoft. Después de toda la incertidumbre acerca de si Microsoft iba a añadir un servidor OLAP a SQL Server, o si por el contrario iba a ofrecerlo por separado, disponer por fin de los Servicios OLAP para SQL Server es casi como recibir un producto gratis. Con la inclusión de los Servicios OLAP como parte del Servidor SQL, Microsoft ha abierto el mercado del data warehousing, data mart, y el soporte a tomas de decisión a muchas empresas pequeñas o medianas que no habrían pensado en usar este tipo de herramienta dados sus elevados costes.

Los Servicios de Transformación de Datos (DTS). La nueva característica DTS del SQL Server 7.0 es una poderosa herramienta y muy flexible. Aunque Microsoft la ha diseñado pensando en facilitar el almacenamiento de datos, la utilidad del producto no acaba allí. DTS simplifica la importación y la exportación de datos entre dos bases de datos compatibles con OLE DB. DTS también genera scripts Visual Basic (VBScript) que se puede ejecutar desde el WSH (Windows Scripting Host) u otros entornos COM (Component Object Model).

Las funciones del Enterprise Manager (EM). Además de implementar el SQL Server Enterprise Manager como un snap-in del MMC (Microsoft Management Console), Microsoft ha mejorado sus funciones y ha incorporado de nuevas. La característica que nos más nos ha llamado la atención es la posibilidad de mirar los contenidos de una tabla directamente desde el EM. Otra función muy útil es la posibilidad de cambiar directamente los tipos de datos de las tablas existentes.

LOS CONTRAS:

Y aunque el SQL Server 7.0 tenga muchas ventajas, también tiene varias desventajas. Aquí tiene algunas áreas en las cuales debe mejorar en próximas versiones...

La instalación y operación requiere del Internet Explorer (IE) 4.0. Le guste o no, la interfaz del navegador de Web sigue siendo cada vez más habitual, y su uso es lo último en desarrollo de interfaces. Podemos entender por qué Microsoft quiere usarlo con el Servidor SQL, ya que también es un produce de la compañía. Sin embargo, no tenemos ninguna utilidad para un navegador de Web en nuestro servidor de la base de datos, y su instalación es un problema que posiblemente, a más de uno le gustaría evitar.

La migración requiere un reinicio de la base de datos. El reinicio de todos los datos en una base de datos es un trabajo serio que invita a la potencial pérdida de datos. Cuanto más grande sea la base de datos, más onerosa será esta obligación. Sin embargo, después de mirar las herramientas de migración del SQL Server 7.0, es obvio que Microsoft se ha planteado esta operación como algo muy serio.

Ausencia de integridad referencial declarativa en cascada (DRI). La ausencia de una integridad referencial en cascada podría ser la desventaja más grande del Servidor SQL en comparación con las otras bases de datos dentro del mercado NT. Incluso Access ofrece soporte de este estilo. Se pueden utilizar triggers para compensar esta desventaja, aunque en otras bases de datos esta técnica no es necesaria, así que no es lógico que deba utilizar para trabajar con SQL Server 7.0. Al considerar las otras nuevas características de SQL Server 7.0, es una pena que ésta no este incluida.

BLOGRAFÍA

1. Implementing a database on Microsoft SQL Server 7.0.
Workbook
Microsoft Training and Certification
2. <http://www.microsoft.com>
3. <http://www.microsoft.com/latam/soporte>
4. <http://windowsnt.about.com>
5. <http://support.microsoft.com/support/sql/70faq.asp>

Autora : Gretel Flores

Bajado desde : www.softdownload.com.ar

Año 2001