PRIMEROS PASOS CON SQL SERVER

Introducción

SQL Server es un gestor de base de datos relacionales que utiliza la arquitectura cliente servidor. Es un entorno de trabajo sobre Windows NT, siendo sus principales armas la rapidez, la integración con el sistema operativo y una gran facilidad de administración del sistema.

Breve historia sobre SQL Server.

El primer Sistemas Gestores de Bases de Datos (SGBD), SQL Server fue creado por Microsoft junto con Sybase en 1988, diseñado para una plataforma OS/2.

Luego Microsoft creó una nueva versión de su SGBD ligada al sistema operativo Windows NT, por lo que aparecieron en 1993 Windows NT 3.1 y su gestor asociado, SQL Server 4.2 para NT.

Posteriormente debido a que las necesidades de las empresas estaban en sintonía con la distribución de los datos en diversos servidores, Microsoft lanzó en 1995 la versión 6.0 del SQL Server, su primer SGBD Relacional diseñado específicamente para un proceso distribuido cliente/servidor, el cual era mucho más sencillo de administrar, incorporaba herramientas visuales a tal efecto.

En 1996 apareció la versión 6.5 de SQL Server, que mejoró el rendimiento, añadiendo nuevos campos de trabajo, tales como Internet y facilitando el trabajo de muchos ámbitos.

Características de SQL Server y su integración con Windows NT.

Una de las principales cualidades de SQL Server 6.5 es aprovechar al máximo las posibilidades del sistema operativo, característica que hace que los rendimientos que obtiene SQL Server en Windows NT sean muy superiores a sus competidores.

- Multithreading: Permite utilizar diversos threads para realizar las tareas más fácilmente.
- Multiproceso simétrico (SMP): Permite aumentar el rendimiento mediante el uso de procesadores adicionales. SQL Server usa un único proceso y lo divide en diversos threads en conjunción con NT, con lo que el sistema permite dividir y balancear las tareas.
- Independencia del soporte de red: Pueden utilizarse diversos protocolos de red: TCP/IP, IPX/SPX, etc.
- Unificación de validación de accesos: SQL Server puede utilizar como propias las cuentas de usuarios de Windows NT y sus contraseñas, eliminando duplicaciones y simplificando el mantenimiento.

- Integración de control: Es posible utilizar el visor de sucesos y el monitor de rendimiento de Windows NT para monitorizar el funcionamiento de SQL Server.

Distribución de datos

SQL Server proporciona los mecanismos para distribuir los datos esparcidos a lo largo de la corporación o de la empresa, para reducir los riesgos de pérdida de datos y mejorar el rendimiento acercando la información a los usuarios que la necesiten.

Gestión centralizada de servidores

Uno de los aspectos más interesantes y sorprendentes de SQL Server en un ambiente corporativo, radica de la posibilidad de ubicar de manera distribuida diversos servidores en la empresa y gestionarlos de manera centralizada. SQL Server proporciona un interfaz visual de administración que permite realizar la tarea sobre los servidores de manera remota utilizando mecanismos propios del interfaz orientado a objetos de Windows 95 y Windows NT.

Elementos constitutivos de SQL Server

Base de datos: Es una colección de datos; en SQL se almacenan en dispositivos. Puede encontrarse en un solo dispositivo o distribuidos de manera aleatoria a lo largo de los medios de almacenamiento.

SQL Server al momento de la instalación crea las bases de datos del sistema *master, model, tempdb, pubs y musdb*.

Dispositivos: Un dispositivo (device) es un fichero en el que se almacenan las bases de datos. Dicho fichero es patrimonio y responsabilidad del sistema, en el sentido de que el usuario no lo utiliza sino a través de un gestor.

En el momento de instalación el sistema crea los dispositivos MASTER, MSDBDATA y MSDBLOG, en los que se almacenan las bases de datos instaladas inicialmente.

Ubicación de las tablas de sistema en los dispositivos.

BASE DE DATOS DISPOSITIVOS

master MASTER

model MASTER

tempdb MASTER

pubs MASTER

msdb(datos) MSDBDATA

msdb(long) MDBLOG

BASE DE DATOS DEL SISTEMA

Master: Contiene el catálogo, es decir una especie de diccionario de datos que controla la funcionalidad de SQL Server. Por la dependencia que de ella tiene el sistema, la base de datos master resulta extremadamente importante, y es preciso mantenerla en buen estado, cuidando de disponer siempre de una copia de seguridad de los datos que contiene.

- Las cuentas de usuarios, locales y remotos.
- Las bases de datos existentes.
- Los servidores con los que puede interactuar.
- Los procesos de datos que están llevándose a cabo en cada momento y los bloqueos.
- Los recursos del sistema para almacenamiento de datos.
- Las variables de configuración.

Tablas de sistema de la base de datos master

Tabla Descripción

syscharsets Contiene una fila por cada juego de caracteres y ordenación

sysconfigures Una fila por cada opción de configuración de usuario.

syscurconfigs Más datos de configuración.

sysdatabases Contiene una fila por cada base de datos existente.

sysdevices Una fila por cada dispositivo definido.

syslanguages Una fila pro cada lenguaje instalado.

syslocks Información sobre los bloqueos activos.

syslogins Almacena las cuentas de usuarios activas.

symessages Contiene todos los mensajes de error.

sysprocesses Información sobre los procesos SQL Server.

sysremotelogins Contiene una fila para cada usuario remoto que tiene acceso a los procedimientos almacenados de SQL Server.

sysservers Contiene una fila para cada servidor remoto que tiene acceso a los procedimientos almacenados de SQL Server.

sysusages Información del almacenamiento del disco.

Msdb: Contiene información de la planificación de tareas, es decir, de aquellas labores

Que deben llevarse a cabo de manera programada. Es utilizada por SQL Server Executive.

ADMINISTRACION DE DATOS EN SQL SERVER

Enterprise Manager: Es una herramienta que permite la administración por medio de un intefaz gráfico.

Permite: Configuración de SQL server, iniciar interrumpir, reanudar y finalizar la ejecución de diversos servicios de SQL Server, ejecución y análisis de consultas, gestión de dispositivos de BD, gestión de objetos (tablas, índices, procedimientos, vistas), gestión de accesos, gestión de alertas, generación de SQL scripts, copias de seguridad, especificar calendario de tareas, monitorización de la actividad de usuarios (optimiza el rendimiento de sistemas delimita cuellos de botella, gestiona procesos de replicación (entre distintos servidores, asigna permisos, etc.

Como Registrar el Servidor: Especifícar nombre del servidor, seleccionar el modo de seguridad para acceder al servidor (trusted o standard), seleccionar el grado de servidor en el que estará incluido, registrar.????

Creación de grupos de servidores: Permite acomodarlos a la estructura del entorno en el que se usen.

Conexión a un servidor registrado: Es necesario que SQL Server esté ejecutándose en la máquina la

conexión se realiza presionando + en el nodo del servidor.

Arquitectura de almacenamiento de datos: Usa intensivamente RAM y disco. Se integra totalmente con el sistema operativo (por lo que puede aprovecharse de características de almacenamiento de Windows NT Ej la implementación de sistemas RAID para protección de BD críticas contra fallos en los discos.

Memoria y SQL Server 6.5: Cuando un sistema Win. NT ejecuta SQL Server el servidor necesitará las siguientes porciones de memoria:

Kernel: 2mb de memoria para el ejecutable.

Memoria config. por el usuario: 2mb (para los parámetros de configuración que el usuario puede modificar)

Cache de datos: Almacena tablas y los índices.

Cache de procedimientos: Almacena los procedimientos almacenados.

Tempdb: Base de datos de objetos temporales (en RAM o en disco)

Almacenamiento físico de los datos: Es independiente del almacenamiento lógico.

Estructura de datos: El almacenamiento se realiza en páginas, extensiones y unidad de almacenamiento.

Páginas: Datos, índices e información del catálogo se almacenan en pág. de 2kb c/u tiene un identificador numérico para referenciarla. Cada pág. está vinculada a un objeto solamente y tiene una cabecera que detalla a que objeto pertenece, puteros a pág. anteriores y a pág. posteriores en el almacenamiento y ubicación física.

Se almacenan por separado de otros datos el texto e imagen. Pág.de datos: Tienen una cabecera de 32 bytes, y el almacenamiento máximo puede ser de 2016 bytes, las páginas se vinculan en forma de lista doblemente enlazada. Una fila no puede almacenarse en más de una pág. Cada fila tiene un nº para la identificación o información de los datos que almacena. Pág. especiales: Las imágenes o bloques de texto que necesitan mucho almacenamiento se denominan BLOBS (Binary Large Objects) Para permitir crear columnas que almacenen estos datos SQL S. separa éstas pág. y las vincula a pág. que contienen las filas de la tabla.

Extensiones: Las pág. contiguas se agrupan de ocho en cho en extents (esta es la unidad minima de almacenamiento en una BD por lo que se asocia a un objeto solamente (y todas las pág. se usan para almacenar datos del objeto).

Unidad de almacenamiento: Contienen 32 extents (256 pág) 512 Kb (se las llama allocation unit) La 1º`pág. se llama allocation page y tiene información para control de la ubicacción de cada pág. en la unidad de almacenamiento como vínculos de cada extents con los objetos de las BD. Tras la cabecera se almacenan 32 estructuras de 16 bytes cada una, una para cada extent. Estas estructuras tambien tienen información de la asignación del extent a los objetos.

Indices: Se almacenan como un objeto más de la BD. sirven seguna modelo orientado a la pág. y no a registros. Se almacenan en pág. separadas dentro de la misma BD.

Existen dos modos de creación:

- Server clustered indexes: Indices que reflejan la ordenación física de los datos (al generarse reordenan la tabla).
- Non clustered: Indices de ordenación lógica (no modifican la ubicación física).

Cuando una tabla no tiene índice clustered los datos se añaden secuencialmente en la última pág. de datos asignada a dicha tabla. Esta forma de almacenamiento es sencilla pero poco eficiente para acceder a la información.

Estructura de los índices

Arboles balanceados: Las búsquedas se organizan sigueindo una ramificación por árbol, cada rama se conecta a otros dos o a un nodo terminal u hoja.

Indices con ordenación: Al crear un índice de este tipo las pág. de datos se ordenan logicamente siguiendo la clave posteriormente se ordenan fisicamente según el mismo criterio. Luego el sistema genera pág. que contengan punteros a las pág. físicas haciendo referencia a todos los registros.

Indices sin ordenación: No ordenan los datos de la tabla, sino que crean punteros que permitiran acceder rápidamente a los datos. Un conjunto de pá. Que contienen una fila para cada uno de los registros de la tabla (estas filas contienen información del mismo de pág. de datos en el que se almacena el registro que posee la clave).

Protección ante fallos de Hard (RAID): Redundant array of indexpensive disks. Se divide la información se pone en varios discos y se repite (al poseer varios dispositivos cada uno de ellos controlados de forma independiente y distribuidos los datos es posible hacer simultáneamente una lectura − escritura con la consiguiente mejora del rendimiento.

RAID 0 (disk stripping): Los datos se dividen en bloques y se distribuyen en los discos de un array de discos este nivel no aporta tolerancia a fallos aunque mejora el rendimiento al repartir lectura − escritura.

RAID 1 (disk mirroring): Constituye una copia identica y redundante de un disco en otro tiene un excelente tolerancia a fallos pero es poco eficiente;desperdicia almacenamiento y hace más lerda las operaciones de L/E

RAID 2: La estructura física de duplicación es igual a la de RAID 0 y añade redundancia con un código corrector de errores (que ocupa bastante espacio en disco por lo que no es eficiente)

RAID 3: Tiene mejoras de rendimiento. Usa el método del nivel 2, pero la información de paridad puede concentrarse en un dispositivo(que puede usarse para recuperar datos)

RAID 4: divide los datos en blques mayores que los niveles 3 y 4 y no es eficiente.

RAID 5 (disk stripping con paridad): Obtiene mayor rendimiento que el mirroring.

Uso del RAID: Los más adecuados son 0. 1 y 5. El 0 ofrece buen rendimiento L/E y si se coloca el transaction log en un dispositivo mirrored, da un buen rendimiento con alta seguridad. El 2 es muy efectivo para evitar pérdidas de información y el rendimiento bueno si los datos de E/S no son muy grandes (es el más adecuado).

El RAID 5 es bueno en fiabilidad y rendimiento en procesos de lectura pero no adecuado para sistemas que acceden mucho a disco para escribir.

DISPOSITIVO: Es un fichero que reserva espacio para almacenamiento de BD, transaction log, y copias de seguridad. El espacio para estos datos se reserva de antemano. Un dispositivo puede almacenar diversas BD yu a su vez una BD puede dividirse entre varios dispositivos. Antes de crear una BD es necesario crear el dispositivo en el que podrá almacenarse. Existen dispositivos de BD, y dispositivos de copias de seguridad.

Dispositivos de BD: se crean para almacenar BD y transaction log. Deben ubcarse en disco duro. Cuando se instala SQL Server se crean 3 dispositivos de BD:

master.dat: almacena la BD model, tempdb, pubs ;

msbdata.dat: Almacena información para la BD msdb para información de gestión de tareas temporizadas.

Msdblog..dat: Almacena información necesaria para la BD msdblog.

Dispositivos por defecto: SQL S. mantiene una lista de ellos para ser usados para albergar una BD si en la creación no se especifíca un dispositivo en el que ubicarlo cuando uno de estos dispositivos agota su tamaño el gestor ubica las BD siguientes en el siguiente en la lista (ordenando alfabéticamente)

Creación de dispositivos mediante la sentencia DISK INIT: Esta sentencia solo la puede ejecutar el administrador de sistema.

DISK INIT

NAME = 'DatosPrubeba' (dispositivo)

PHISNAME = 'D:\SQL 6.0\DATA\DatosPrueba.dat' (disp.físico, no debe existir aneriormente)

VDEVNO = 3 (identificación de las pág. de almacenamiento)

SIZE = 2506 (tamaño del dispositivo en bloques de 2kb)

Con esto se crea un fichero en disco, se añade una fila a la tabla sysdevice de la BD master (que especifíca las características del dispositivo; nombre mirror asociado, su tamaño y las pág. de uso).

Información de los disp. Instalados: Para obtenerla se puede acceder a Enterprise Manager, en la ventana server manager en el node device, o más detalles de las características de cada disp. Usando el procedim. Almacenado de sistema: sp_helpdevice [nombre del dispositivo].

Modificación de un dispositivo de BD existente: Se puede expandir o reducir mediante Enterprise M.

Accediendo al cuadro de diálogo Edit Database Devices (en ventana Server Manager pulsar boton del del mouse sobre el disp.) o mediante la orden:

DISK RESIZE NAME = 'nombre lógico'

SIZE = 'nuevo tamaño ' (bloques de 2k)

Eliminación de dispositivos de BD: Se denomina dropping y elimina la entrada referente al dispositivo en la tabla sysdevices de manera que el dispositivo borrado puede ser reutilizado (el fichero físico deberá borrarse con el administrador de archivos de Win NT) Existen dos formas: Mediante Enterprise M. Pulsar el boton der del mouse (estando sobre el dispositivo) y elegir delete, o mediante el procedimiento almacenado de sistema sp_dropdevice [nombre lógico del dispositivo]

Dispositivos de copias de seguridad: Almacenan backups de BD o transaction log pueden crearse como fricheros en discos, cintas, disquettes, o para integración de otras aplicaciones en names piped????

Creación de un backup device: Mediante Enterprise M. (Se abre cuadro de diálogo tools/ database

backup−restore, pestaña backup; aquí aparece una lista de los dispositivos de volcado de información que existen si deseamos crear uno nuevo picamos en new) o mediante el procedimiento almacenado de sistema:

sp−addumpdevice {'disk'| 'disquette" | 'tape'},

'nombre lógico' , nombre físico

[,{{cntrltype [,noskip | skip [,capacidad]]} | {@devstatus = {noskip | skip}}}]

Mirroring: Sirve para evitar la perdida de datos (por fallos de disp. Error de operación o averia del disp.físico) Se crea un segundo dispositivo que se halla sincronizado de contenidos con el disp. A respladar. Puede llevarse a cabo usando técnicas de hard o soft (la hard es más eficiente pero más costos ej: sistema RAID) con la creación de una copia de seguridad (deberíamos respaldar al menos el dispositivo master). El mirroring se lleva a cabo al nivel del dispositivo y no al nivel de la BD. Para respaldar una BD hay que respaldar todos los dispositivos en los que se halla almacenada. El mirror duplica la información.

Se puede realizar por medio de Enterprise M. Pulsando mirroring y creando el archivo. Mir o usando la orden transact SQL:

DISK MIRROR

NAME= 'nombre lógico'

MIRROR = 'nombre físico'

[WRITES = {serial | no serial} (solo por compatibilidad)

Esta orden crea e inicaliza el dispositivo (crea y reserva el espacio del archivo . mir asociado, copia los datos del dispositivo a respaldar al dispositivo).

Eliminación del Respaldo: Lo hace el administrador del sistema con:

DISK UNMIRROR

NAME = 'nombre logico del disp.respaldado'

[,SIDE = {'primary'| 'secondary'}]

[,MODE = {retain | remove}]

Secondary es la opcion por defecto de SIDE y primary el dispositivo respaldado es reemplazado por su mirror. La opción retain conserva las entradas de la tabla sysdevice.

Sustitución permanente de un dispositivo por su mirror

DISK UNMIRROR

NAME ='DatosPrueba'

SIDE = 'Primary'

MODE = remove

Eliminación temporal del disp. Mirror.

DISK UNMIRROR

NAME = 'DatosPrueba'

MODE = retain

Recuperación de un respaldo existente: A veces SQL elimina un respaldo automáticamente a causa de fallo de escritura en un dispositivo (el respaldado o el mirror) una vez recuperada la información será necesario reanudar los vínculos de respaldo. Por otra parte si hemos eliminado el respaldo con disk unmirror con mode = retain se recupera el respaldo con:

DISK REMIRROR

NAME = 'nombre lógico'

Segmentos: Mejoran el rendimiento colocando tablas e índices a los que se accede frecuentemente en una ubicación separada. Un segmento es la representación de una localización física que permite ubicar objetos de manera controlada por el administrador en lugar de permitir que SQL Server esoja el lugar en el que colocará los datos. (cada BD en SQL Server puede tener hasta 32 segmentos) Para definirlos se usa el procedim. Almacenado sp−addsegment segmento, dispositivo.

Para añadir espacio de almacenamiento se usa sp_extendsegment segmento, dispositivo

Ubicación de Objetos en segmentos:

Ej CREATE TABLE nombre de la tabla

CREATE INDEX nombre del índice

ON nombre del segmento

ON nombre del índice

ON nombre del segmento.

Eliminación de segmentos: sp_dropsegment nombre del segmento.

La tabla Syssegments: Tiene información de los segmentos de cada BD contiene una fila para cada segmento. Se indica: nº de cada segmento, nombre lógico, un bit para índicar si es definido por defecto o no.

Ej (una consulta a esta tabla para la BD pubs)

SELECT *

FROM syssegments

Mostraría: **segment name status**

0 system 0

1 default 1

2 logsegment 0

BASES DE DATOS: Para SQL S. es una entidad lógica en la que se puede crear y almacenar objetos (triggers, vistas, y otros elementos) no puede existir ningun objeto que no se halle en una BD.

Donde Ubicar una BD: La BD debe estar asociada al menos a un dispositivo de BD (sino se la coloca en uno o más dispositivos especificados por defecto)

Relación de Transacciones: Cada BD tiene asociada una relación de transacciones (transaction log) que es una zona de almacenamiento en la que SQL S. mantiene una lista de transacciones que se han llevado a cabo en la BD, SQL S anota en transaction log cada operación osbre la BD antes de reflejarlas en la misma.

La BD temporal tempdb: Ubicación en que SQL S. vuelca los datos temporalmente cuando es necesario (cuando se hacen consultas que producen una tabla muy extensa que deba almacenarse durante la realización de la consulta) los usuarios pueden crear un objeto temporal.

El nombre de las tablas temporales se identifican por el primer carácter que es #. Es conveniente que el tamaño sea un 25 % de la BD más grande, esta BD también puede ubicarse en la RAM.

Tamaño de una BD: Es la suma del espacio de almacenamiento de datos (datos, datos del usuario y tablas del sistema) más el tamaño del transaction log. SQL S. usa 3 unidades de tamaño en el almacenamiento: página (2kb), página extendida (8 pág.contínuas), unidad de almacenamiento (256 pág. =0.5 Mb.)

La primer pág. se llama pág. de ubicación y contiene una matriz que especifica el uso de pág. cuando se crea la BD se especifícan los dispositivos y la porción de los mismos que se destinará a esta BD lo que definirá el tamaño.

Limites del Sistema para creación de BD: El nº max. De BD es 32.767, tamaño máx. de una BD es de un TB. (que puede estar distribuido en 32 disp. Como máx.), tamaño min y por defecto: es igual al de la BD model inicialmente de 2 MB)

Documentación de la creación: Es conveniente, para reconstruir la BD o si se quiere repetir el proceso en otros servidor) se hace creando y almacenando en un fichero las sentencias que se usan y ejecutarlas de manera automática en su momento. Antes de crear una BD es conveniente hacer un checklist de la información necesaria.

Permisos de Creación de BD: Igual que modificar solo lo puede hacer el administrador de sistema u otorgar permiso a usuarios para hacerlo

(ej. GRAN CREATE DATABESE TO nombre de usuario)

El transaction log: Almacena las modificaciones de la BD hechas con INSERT, UPDATE, DELETE, a través de sentencias transact SQL se escriben en el transaction log antes de ejecutarse , tiene el objeto de mantener la integridad de los datos.

Transacciones: Son un conjunto de operaciones que se llevan a cabo de manera completa o se descartan (asegurando así la coherencia e integridad de datos)

Inicio y final de una transacción: Cada sentencia transact SQL es una transaccion el inicio se da por la sentencia BEGIN TRANSACTION. Especificadas las sentencias de las unidades de trabajo se puede validar o

descartar. En la validación (se aceptan las modificaciones) se hace con COMMIT TRANSACTION, y para descartarla usa la sentencia ROLL BACK TRANSACTION.

La tabla syslogs: Cada BD inluida master, model tempdb tiene su transaction log constituido como una tabla denominada syslogs (formada por dos columnas por actitid (binary), y op(timyint) los datos almacenados por esta tabla solo son interpretablas por SQL S.

El proceso de escritura write−Thead: En esta disciplina de escritura, las modificaciones sobre la BD no son escritas directamente sobre la misma, sino que se escriben primero en el transaction log y no se traspasan a la ubicación física de los datos hasta que no sxe produce un COMMIT.

La cahe de datos: Las operaciones de la transacción, antes de ser validadas se almacenan temporalmente en una estructura de datos intermedia, la cache (implementada por SQL S sobre la RAM) que minimiza el n° de escrituras en disco, facilitando de este modo que el rendimiento sea mayor.

Creación de una BD para recup. De una copia de seguridad: Cada BD se respalda con una copia de seguridad, se almacenan los datos pero no su estructura para recuperar si la BD original se pero sin datos y utilizar la orden LOAD DATABASE para rellenar dicha estructura con los datos almacenados en el backup.

Creación mediante Enterprise M.: Seleccionar el modo database, presionar boton derecho del mouse y elegir del menú contextual new database (elegir nombre y dispositivo en el que residirá la BD)

Creación mediante CREATE DATABASE:

CREATE DATABASE BaseDatosPrueba

ON DatosPrueba = 2

LOG ON LogPrueba = 1

CREATE DATABASE nombre de la BD.

[ON {DEFAULT | DISP.DE LA BD }[TAMAÑO]]

[LOG ON disp [=tamaño]]

[FOR LOAD]

Si la BD ya existe y hubiese sufrido deterioro y deseamos recuperar de una copia de seguridad agregar FOR LOAD

Modificación de la BD: Ampliación del Log (mediante Enterprise M.) en ventana server manager posicionarse en database y en menu contextual elegir edit seleccionar la pestaña database y pulsando el boton expand aparecerá un cuadro que permite aumentar el tamaño de los datos del log

Expansion mediante ALTER DATABASE: Permite ampliar la BD y sus logs no reducirlos

ALTER DATABASE BaseDatosPrueba

ON DatosPrueba = 5

LOG ON LogPrueba = 2

Modificación de opciones en la BD con sp_dboption (sobre la BD master)

Eliminación de la BD: solo lo puede hacer el administrador del sistema o el propietario. Se eliminan fisicamente las BD del dispositivo físico en el que esta almacenada y los objetos que contiene. No pueden eliminarse master, model tempdb y las que estan abiertas o replicadas. Se puede eliminar mediante DROP DATABASE. O mediante el procedimiento sp_dbremove nombre de la BD

Que hacer para poder eliminar una BD corrupta: En ocasiones no podremos eliminar una BD por que se ha corrompido la solución puede ser usar la orden DBCC DBREPAIR

DBCC NEWALLOC: examina la estructura de almacenamiento físico de los datos de una BD la salida es información de los extents y pages para cada objeto de la BD.

DBCC CHECKDB: Balida los vinculos entre punteros a datos y las pág. De datos físicamente ubicadas en el servidor (de esta forma se verifican si los punteros apuntan a la ubicación fisica de los datos.

Ej. DBCC CHECDB (nombre BD, [noindex])

DBCC CHECKTABLE: Es similar a DBCC CHECKDB, pero solo opera con tablas de la BD actual, DBCC CHECKTABLE (nombre de la tabla [noindex | identificador del índice]

BCC CHECKCATALOG: Recorre la tabla syscolumns y asegura que los tipos almacenados en ellas son validos (que se hallen en la tabla systypes, también asegura cada uno de los objetos de la tabla sysobjects DBCC CHECKCATALOG (nombre BD)

DBCC SHRINKDB: Permite reducir el tamaño de un BD.

Ej. DBCC SHRINKDB (BD [, nuevo tamaño [,masteroverride]]). El tamaño se especifíca en cant de pag. De 2k. La opción MASTEROVERRIDE se usa para permitir la reducción de la BD Master

DBCC SQLPERF ({IOSTATS |LRUSTATS | NETSTATS | RASTATS | [CLEAR]}

Permite controlar el rendimiento del sistema

IOSTATS: Esta opción proporciona una información de la actividad de E/S que se lleva a cabo en SQL S. (referencia al n° y tipo de lecturas y escrituras en disco las transacciones procesadas, la escritura en transaction log etc)

LRUSTATS: Informa como esta usándose la cache, incluye entre otra información tasa de aciertos en la cache, el tamaño y uso de la misma.

NETSTATS: Valora el uso y rendimiento que se obtiene de las transacciones de datos a través de la red (es muy importante en entornos cliente /servidor)

THREADS: Presenta información del uso que está haciendo SQL S. De la multitarea de Win NT mostrando los threads que se hallan activos.

LOGSPACE: Muestra el espacio que aún resta disponible para el transaction log en tdoas las BD del sistema.

DBCC MEMUSAGE: Detalla la configuración de memoria del servidor, la información incluye uso de la memoria y de la cache mostrando que objetivo han sido más frecuentemente almacenados en la misma.

Objetos de Base de Datos. Estructura de almacenamiento y manipulacion.

Un sistema de base de datos esta orientado a almacenar, procesar y responder las solicitudes de un conjunto de datos.

Las consultas se realizan generamente mediante consultas SQL, aunque SQL Server, tambien permite la obtencion de datos atraves de estructuras que facilitan la obtencion de informacion, como el caso de vistas y procedimientos almacenados.

Tablas

La tabla en SQL Server es la unidad basica de almacenamiento de informacion. Existen sentencias en SQL para la manipulacion de las mismas.

Creacion de Tablas

Las tablas en SQL Server se pueden crear mediante el Enterprise Mananger o mediante sentencias Transact SQL (Lenguaje de SQL Server).

Creacion mediante ordenes

Se utiliza la sentencia CREATE TABLE.

CREATE TABLE [base de datos.[propietario].]nombre de table (columna tipo de datos [not null | null] IDENTITY [(semilla, incremento)] [restriccion][[,]{siguiente columna | siguiente restriccion}...])[ON nombre segmento]

Nombre de tabla

Contiene 3 partes: nombre de base de datos, identificador de usuario y nombre de tabla. Las dos primeras son opcionales.

La primera es la base de datos a la que pertenece, si se omite es la current database.

El creador es generamente el que se incluye en la segunda parte que puede no coincidir, si se omite es el creador.

El propietario puede autorizar a los demas usuarios a usar la tabla.

Tipos de datos definidos por el usuario

Ademas de los predefinidos, SQL Server da la posibilidad de crear tipos de datos. Estos tipos de datos son como los predefinidos con alguna caracteristica adicional.

Para crearlos se utiliza el PA sp_addtype.

sp_addtype nombre del tipo, tipo predefinido, null

sp_addtype nombre, char(20), null

Para eliminar el tipo de datos se usa el PA sp_droptype.

spdroptype nombre del tipo

Valores nulos

La especificacion null permite que un campo pueda almacenarce sin dato.

CREATE TABLE alumnos (nombre char(20), edad int null)

INSERT INTO alumnos VALUES(`pepe')

Columnas con valores autoincrementales

Si se especifica IDENTITY, indica al gestor que cada nueva fila se debera incrementar el valor de esta columna. Se puede tener una semilla inicial que se utiliza como valor incial.

CREATE TABLE Alumnos (Nombre char(20), id int IDENTITY(100,1))

Restricciones

En SQL Server se llaman *constraint*, se utilizan para especificar condiciones de integridad referencial. Estas restricciones incluyen la definicion de claves, limitaciones del valor a utilizar y valores por defecto.

CONTRAINT nombre de restriccion(lista de columnas)

Los tipos de restricciones pueden ser:

PRIMARY KEY

Especifica que la columna sera clave primaria.

CREATE TABLE Alumnos(Nombre char(20)) CONTRAINT Rest_Nombre PRIMARY KEY CLUSTERED(Nombre))

FOREIGN KEY

La columna sera clave ajena que se vincula con otra, que debe ser primaria, siguiendo el modelo relacional.

Foreign y Primary Key , conforman lo que SQL Server denomina DRI (Declarative Referential Integrity)

CONSTRAINT nombre de restriccion FOREIGN KEY (columnas) REFERENCES tabla vinculada (columnas)

UNIQUE

Indica que los valores de una columna deberan ser unicos. La columna podria no ser clave primaria.

CONSTRAINT nombre de restriccion UNIQUE CLUSTERED/NONCLUSTERED (Columnas)

La sentencia crea un indice para esta columna automaticamente.

CHECK

Esta restriccion limita los valores admisibles en un campo mas alla de los especificado por el tipo de dato.

CHECK nombre de restriccion (Columna IN (Lista de valores))

DEFAULT

Permite indicar valores por defecto a una columna en caso de omitir un dato en momento de agregar un registro.

DEFAULT nombre restriccion valor FOR Columna

CREATE TABLE alumnos (nombre char(20),edad int, DEFAULT edad_def 20 FOR edad)

Modificacion de tablas

Es posible añadir nuevas columnas o condiciones a una tabla existente mediante ALTER TABLE.

ALTER TABLE [base de datos.[propietario].]nombre tabla [WITH NOCHECK]

[ADD{ Columna propiedades [restriccion [restriccion]]] [DROP [CONSTRAINT] restriccion]

A**ñadiendo columnas y restricciones**

Es posible añadir nuevas columnas y restricciones a la tabla existente, especificado la clausula ADD.

Eliminacion de tablas

DROP TABLE [[Base de datos.]propietario.]nombre de tabla [, [[Base de datos.]propietario.]nombre de tabla]

Vistas

Una vista esta constituida por una sentencia SELECT constituyendo un objeto dentro de SQL Server. El resutado es un conjunto de registro que se pueden tratar como una tabla sin serlo.

Utilidad de las vistas

Proteger confidencialidad: Si en una tabla hay datos que no se deben mostrar a un usuario, puede contruir una vista con las columnas necesarias.

Facilidad de utilizacion: El administrador puede crear complejas consultas para que el comun de los usuarios la use.

Independencia de datos: Los cambios de implementacion de las tablas pueden motivar al cambio de nombres de las columnas, y debe reformular ciertos prog., para evitarlo puede crear una vista de acuerdo a sus necesidades.

Consulta y actualizacion

Si se actualiza un vista las modificaciones debes reflejarse en la tabla de la que se deriva. Esto no siempre es posible.

No se podra actualizar cuando las vistas contengas:

Vistas con JOIN

Vistas con funciones de grupo

Vistas con GROUP BY

Vistas con DISTINCT

Creacion de vistas

Para Transact SQL se utiliza CREATE VIEW.

CREATE VIEW [propietario.]vista [(col1[,col2]...)] [WITH ENCRYPTION] AS subselect [WITH CHECK OPTION]

Escriptado

La clausula WITH ENCRIPTION encripta las filas de las tablas *syscomments* que contiene el textode definicion de la vista en el catalogo.

Eliminacion de Vistas

Para eliminar una vista se utiliza DROP.

DROP VIEW vista

Indices

Utilizando indices en una tabla se logra realizar busquedas mas eficientemente.

Creacion de indices

La creacion de indices se lleva a cabo mediante la orden CREATE INDEX

CREATE [UNIQUE] [CLUSTERED|NONCLUSTERED] INDEX nombre de indice ON [base.] propietario.]tabla (columna[,columna]...) [WITH [FILLFACTOR = x] [[,] IGNORE_DUP_KEY] [[,]{SORTED DATA | SORTED_DATA_REORG}] [[,] {IGNORE_DUP_ROW | ALLOW_DUP_ROW}]][ON segment_name]

Indice unico

Mediante la clausula UNIQUE se crea un indice que no permita valores duplicados.

Tipo de indice

Puede especificarse con CLUSTERED/NONCLUSTED.

Un indice *clustered* crea un objeto en el orden fisico de las filas igual a su orden logico, de modo que cambia el orden de la tabla. Es mas rapido.

Un indice *nonclustered* no produce reorganizacion fisica, sino solo logica.

Actualizacion de estadisticas

Cuando SQL Server crea un indice crea tambien un conjunto de estadisticas que utiliza para optimizar las consultas.

UPDATE STATISTICS [[base de datos.]propietario.]tabla [indice]

Reglas

Es otro metodo para limitar los valores admisibles a una columna.

Creacion de reglas

La orden para esto es:

CREATE RULE nombre regla AS expresion condicional

La expresion condicional sera cualquiera de las que puedan utilizarse para particion horizontal el la clausula WHERE, pero cualquier parametro que exista en la misma debera expresarse perecedido del carácter @.

CREATE RULE edadesadmisibles AS @edad in (20,21,22)

Vinculacion de reglas(binding)

Para que una regla se pueda utilizar se debe vincular a una columna.

sp_bindrule regla tabla.columna | tipo [,futureonly]

Este procedimiento vincula una regla con una columna de una tabla o con un usuario. El parametro futureonly provoca que no se apliquen reglas a las columnas existentes.

sp_bindrule edadesadmisibles alumnos.edad

Solo se pueden aplicar reglas a un tipo de usuario o a una columna.

Desvinculacion de reglas

Si decidimos desvincular una regla se debe usar el PA sp_unbindrule

sp_unbindrule tabla.columna | tipo [, futureonly]

Valores por defecto(Default)

Los valores por defectos se aplican cuando se defina una tabla mediante CREATE TABLE.

Sirve para darle valores a una columna en el caso de omitirlo durante una insercion.

Creacion de valores por defecto

La orden que permite crear un valor por defecto es CREATE DEFAULT.

CREATE DEFAULT nombre del valor por defecto AS valor constante

El valor constante sera el valor que debera tomar la columna en el caso de omitir su valor.

CREATE DEFAULT cero AS 0

Vinculacion de valores por defecto(binding)

Para que un valor por defecto se pueda utilizar se debe vincular a una columna.

sp_binddefault valor por defecto tabla.columna | tipo [,futureonly]

Este procedimiento vincula un valor por defecto con una columna de una tabla o con un usuario. El parametro futureonly provoca que no se apliquen el valor por defecto a las columnas existentes.

sp_ binddefault ceros alumnos.notas

Solo se pueden aplicar un unico valor por defecto a un tipo de usuario o a una columna.

Desvinculacion de valores por defecto

Si decidimos desvincular un valor por defecto se debe usar el PA sp_unbindefautl

sp_ unbindefautl tabla.columna | tipo [, futureonly]

Procedimientos almacenados y triggers.

Los procedimientos almacenandos son una serie de colecciones de sentencias Trasact SQL que se hallas precompiladas y optimizadas para efectuar tareas determinadas.

Un tipo especial es el triggers, su traduccion podria ser desecadenadores o disparadores, prestan utilidad durante la modificacion de los datos.

Los procedimientos almacenados son funciones similares a las de C, es decir, mediante un identificador, puede recibir y devolver valores.

Caracteristicas basicas

Aceptan parametros.

Devuelven valores de retorno.

Pueden anidarse una dentro de otro (hasta 16)

Se pueden hacer referencia a otros objetos, exepto a las tablas temporales que hayan sido creado por el primero.

Tambien incoporan mecanismos de seguridad. Permite que un usuario no tenga acceso directo a las tablas pero si a los procedimientos almacenados y con estos si tener acceso a las mismas.

Tipos de Procedimientos

Permantentes

Son los determinados por los usuarios y estan almacendos.

Temporales

Son solo accesibles en la sesion actual del usuario.

Triggers

Es un procedimiento almacenado qe se ejecuta como respuesta a una llamada explicita al mismo.

Los triggers se utilizan entre otras cosas para establecer las reglas de integridad referncial.

Acciones que motivan la ejecucion de un trigger

Los triggers se ponen en funcionamientos cuando se modican los datos, ya se actualizacion, insercion o borrado, existiendo un trigger para cada situacion.

Un ejemplo es que cuando se agrega una fila se deben verificar el valor de cierto campos, validandolos contra otras tablas u valores. O para el borrado en cascada. O actualizacion en cascada.

Si se eliminacen o actualizacen multiple filas o solo se llamara una ves la ejecucion.

Sirven para:

Preservar la integridad referencial. En el caso que dos o mas valores se hallen vinculados por sus claves no resulta posible modificar la primaria sin la ajena. Utilizando triggers controlamos este proceso, forzamos a actualizar los ajenos.

Establecimiento de condiciones complejas. Permiten validar multiples columnas.

Respuesta a cambios de estado de tablas. Permiten examinar el estado de la tabla antes y despues de la actualizacion.

Creacion de Procedimientos almacenados

La orden que se usa es CREATE PROCEDURE.

CREATE PROCEDURE [propietario.]nombre de procedimiento[;numero][(lista de parametros)]

[{FOR REPLICATION} | {WITH RECOMPILE}

[{WITH} | [,]} ENCRYPTION}}

AS sentencias SQL

Nombre del procedimiento

Sera el identificador del mismo, el propietario es opcional y no podra exceder los 20 caracteres.

Se pueden renombrar con sp_rename.

Sp_rename nombreantiguo, nombrenuevo

Parametros

Los procedimientos almacenados pueden recibir parametros. Puede tener una maximo de 255 parametros.

El tipo image no puede ser usado de parametro.

Son valores constantes.

Se puede especificar valores por defecto.

Recompile

Cuando se necesita compilar un procedimiento inmediatamente se debe usar esta opcion ya que de caso contrario la compilacion se realizara cuando se efecute por primera vez. Otra alternativa es utilizar el PA sp_recompile

Replication: Procedimientos de duplicacion.

Es posible crear procedimientos almacenados para casos especiales como los de replicacion, que solo son ejecutables desde el servidor publicador.

Encription: Procedimientos de descripcion encriptada.

Este parametro se utiliza para negar la posibilidad que un usuario pueda observar el PA ya que el mismo se guarda encriptado inpidiendo su consulta visual.

Sentencias SQL del procedimiento almacenado.

El cuerpo del Procedimiento estara integrado por un conjunto de sentencias SQL que realizaran la tarea esperada y estas se espeficicaran despues de la clausula AS.

La maxima cantidad de caracteres es de 65.025.

CREATE PROCEDURE infousuarios AS SELECT * FROM sysusers

Sentencias admisibles

En gral puede incluir cualquier tipo de sentencias pero hay restricciones a tener en cuenta.

No se pueden usar las sentencias de creacion:

- CREATE VIEW
- CREATE TRIGGER
- CREATE DEFAULT
- CREATE PROCEDURE
- CREATE RULE

Las sentencias de borrado estan permitidas.

Se permita tambien la creacion de tablas temporales.

Eliminacion de procedimientos almacenados.

La sentencia DROP_PROCEDURE es la encargada de este tema.

Ejecucion de procedimientos almacenados.

La sentencia utilizada para este fin es EXECUTE.

La sentencia EXECUTE.

Se utilizar tanto para lanzar procedimientos como para ejecutar cadenas de caracteres que alli se localizan.

EXEC[ute] {[@valor de retorno =

=]{[[[nombreproc[;num] | @variablenombre}

[[@parametro =] {valor | @variable [OUTPUT]]

[,[@parametro=] {valor | @variable [OUTPUT]}]...]

[WITH RECOMPILE]]

Valor de retorno

Como ya dijimos un PA puede devolver una valor de retorno que se almacenara en una variable @valor de retorno

Paso de Parametros

La sintaxis para pasao un parametro al procedimiento es:

@nombre de parametro = valor

Tambien esta la posibilidad de hacerlo en base al orden de ingreso

omitiendo el nombre del parametro.

Valores por defecto

Es tambien posible en la definicion del procedimiento almacendo especificar el valor por defecto que tomara una variable.

Parametros de retorno

Para los parametros creados como retorno, utilizando OUTPUT, se bede proporcionar la variable que almacene dicho valor.

Ejecucion de procedimientos

El formato utilizado para ejecutar un PA es:

EXECUTE infoautores ringer, anne

Informacion retornada por un procedimiento almacenado

Los PA pueden devolver valores de dos tipos:

- Mediante el retorno de una valor.
- Mediante parametros de retorno.

Valor de estado (status value)

Al igual que cualquier lenguajes las funciones retornan un valor que nos permite testear como opero.

Gralmente estos valores esta prefijados de antemano.

Valores de estado prodefinidos

SQL Server predefine ciertos valores y cada uno posee su significado. El 0 es una ejecucion exitosa, el resto esta comprendido entre −1 y −99 de los cuales entre el −1 y −14 tiene significado.

Valores de estado definidos por el usuario

La sentencia RETURN es la encargada de devolver el valor. Dicho valor no puede estar comprendido entre 0 y −99.

CREATE PROCEDURE aprobado @nombrealumno varchar(20)

AS

IF (SELECT nota FROM alumnos WHERE nombre =

= @nombrealumno) >5

RETURN 100

ELSE

RETURN 200

Retorno de parametros

En el momento de la definicion del procedimiento, se puede especificar que alguno, o todos los paramentros sean especificados como de retorno.

Paso de parametros por referencia

Cuando se utiliza la opcion OUTPUT, en la clausula CREATE PROCEDURE y EXECUTE, la modificacion que se produzcan dentro del procedimiento se reflejaran en las variables al final de la ejecucion del mismo

Paso de parametros por valor

Si no se utiliza OUTPUT, estaremos realizando paso de parametros por valor, al pasar un parametro a un procedimiento almacenado se crea una copia del argumento, que sera utilizada localmente en el procedimiento.

CREATE PROCEDURE suma @sum1 int, @sum2 int, @res int OUTPUT AS

SELECT @res = @sum1 + @sum2

RETURN 0

Llamado :

DECLARE @resultado int

DECLARE @retorno int

EXECUTE @retorno = suma 2, 2, @resultado OUTPUT

Procedimientos almacenados que brindan informacion del sistema

Tanto la definicion de los objetos como los PA se almacenan en las tablas de sistema. Para poder obtener informacion de ellos SQL Server provee de una serie de PA:

sp_help: Proporcina informacion de un determinado PA.

sp_depends: Devuelve la lista de todos los objetos relacionado con este.

Creacion de triggers

Aspectos basicos

Son procedimientos almacenados que se ejecutan como respuesta de la modificacion de una tabla.

Su utilidad basica es de mantener la integridad de los datos.

La sentencia encargada de su creacion es CREATE TRIGGER, con una sintaxis similar a los otros objetos.

Creacion de triggers mediante la sentencia CREATE TRIGGER

CREATE TRIGGER [propietario.] nombretrigger ON [propietario.] nombretabla FOR {INSERT, UPDATE, DELETE}

[WITH ENCRYPTION]

AS sentenciasSQL

O

CREATE TRIGGER [propietario.] nombretrigger ON [propietario.] nombretabla FOR {INSERT, UPDATE}

[WITH ENCRYPTION]

AS IF UPDATE nombrecolumna

[{AND | OR} UPDATE (nombrecolumna)...]

sentenciasSQL

Acciones que desatan la ejecucion del trigger

Cuando sobre una tabla se ejecuta una sentencia

UPDATE/INSERT/DELETE, seleccionada en FOR{UPDATE INSERT DELETE}

Recursividad

No se permite recursividad de ejecucion en un trigger.

TRUNCATE TABLE

Cuando se produce un TRUNCATE TABLE no se ejecutan triggers.

Sentencias SQL del trigger

Al igual que un PA admite todo tipo de sentencias SQL eseptuando las sentencias de creacion como:

- CREATE VIEW
- CREATE TRIGGER
- CREATE DEFAULT
- CREATE PROCEDURE
- CREATE RULE

y otras mas como

- CREATE DATABASE
- CREATE INDEX

Tampoco se puede utilizar sentencias de borrado de tablas u elementos

DROP y de modificacion de estructuras como:

- ALTER DATABASE
- ALTER TABLE

No se puede utilizar tampoco

- TRUNCATE TABLE
- GRANT
- REVOKE

CREATE TRIGGER triggerprueba

ON dbo.authors

FOR INSERT

AS

Print `Nuevo autor'

Utilizacion de trigger para validar actualizaciones de tablas

Una de las principales utilidades es mantener las tablas coherentes, ya que se pueden validar los datos, tanto para añadir como para actualizar.

Las tablas Inserted y Deleted

Cuando se producen cambios sobre las tablas ya sean insercion, borrado o actualizacion, SQL Server ejecuta el trigger y ademas dispone de dos tablas temporales inserted y deleted con los datos antes de la actulizacion o borrado. Estas tablas se las puede maniputar con SELECT.

Insercion o borrado multiple

Cuando se producen insercion o borrado multiple, en trigger se ejecuta una sola vez para todas las columnas y en las tablas temporales se encuentran todos los registros afectados.

Transacciones

Con los trigger tambien se pueden descartar transacciones mediante la sentencia ROLLBACK TRANSACTION.

Administración de la Seguridad en el acceso a los datos

Debido al avance las comunicaciones y las tecnologías de red, es fundamental para un ADMBD el saber de forma controlar , que no es necesariamente lo mismo que limitar el acceso a la información contenida en una base de datos.

En un ambiente cliente/servidor es tan necesario el correcto diseño de la forma de almacenar la información, como la identificación de los usuarios y la asignación de permisos a dichos clientes para el acceso a diferentes áreas de la base de datos que le provean la información necesaria para el desarrollo de su tarea.

Por lo antes expuesto es necesario el Sistema de Base de Datos nos provea de las herramientas necesarias para realizar este control, ya que la seguridad de los datos almacenados en la misma son de vital importancia para el ente al que prestan servicio.

SQL define una serie de figuras cuya identificación ayudará al administrador en la tarea de mantener un sistema seguro, los usuarios del servidor, los usuarios de las bases de datos y los permisos a cada objeto de SQL Server.

Características del Sistema de Seguridad del SQL Server.

El mismo esta compuesto de varias capas, por decirlo de otra manera el usuario para poder acceder a los datos debe saber abrir varias puertas. Un ejemplo del mismo se detalla en la figura siguiente

Seguridad del Sistema Operativo.

El usuario en primer lugar debe sortear la barrera que impone el sistema operativo que se este usando,

Seguridad de Servidor − Identificación en SQL Server

Una vez que el usuario ingreso al sistema operativo deberá sortear el acceso a la utilización de SQL sea cual fuere la tarea a realizar, (consultar, agregar o modificar datos) o también si desea utilizar Enterprise Manager

para administrar el Sistema. Es posible sortear este control si se considera que con la clave de acceso al Sistema Operativo es suficiente.

Seguridad de la Base de Datos.

Las bases de datos disponen de mecanismos para la identificación de usuarios y controlar cuales de los usuarios que tienen acceso al servidor tendrán también acceso a los datos de dicha base. Este tercer proceso de verificación permite el acceso a los datos.

Seguridad de Objeto. Permisos de utilización.

Una vez que se accedió a este nivel se debe sortear también el ultimo escalón del sistema de seguridad. Pues también deberá poseer autorización para poder ejecutar sentencias SQL sobre los objetos.

El sistema posee gran flexibilidad a la hora de proteger los datos. Podemos a cada momento determinar quien puede y que puede realizar determinado usuario y sobre un determinado objeto o también solo columnas individuales de un objeto. Los permisos para realizar operaciones pueden concederse o revocarse colectiva o individualmente.

NIVELES DE SEGURIDAD

Existen cuatro categorías a saber .

a − Adminstrador

Es quién tiene acceso sin restricciones y puede realizar cualquier tipo de tareas con los datos, dar y quitar permisos etc.

b − Propietarios de Bases de Datos (DBO)

Es el usuario que ha creado de la base de datos o a quien se le asigno esta propiedad. Tiene acceso a todos los objetos de la base de datos y puede también asignar permisos sobre dicha base.

c − Propietarios de Objetos de Base de Datos (DBOO)

Es el usuario que ha creado dicho objeto en la Base de Datos puede realizar cualquier tarea sobre ese objeto (Select,Update.). Es decir es propietario de ese objeto cabe aclarar que la propiedad de un objeto no se puede modificar, ni siquiera por el Administrador de Sistema, se debe borrar el objeto y crear uno nuevo si se quiere cambiar la propiedad.

d − Usuarios

Son el resto de los integrantes del ente que a los que se le pueden otorgar permisos para operar la Base de Datos.

Figuras de identificación

Para ingresar a SQL primero se asigna un login id que es el nombre que permite a un usuario acceder a SQL Server . Debe ser único para cada usuario al que queramos dar acceso. Por ejemplo sa es el login id para el administrador. Pero añadir un nuevo login id no le proporciona acceso a la base de datos sin solamente a la base de datos por defecto que se denomina master..

Para poder acceder a una base de datos específica debe el usuario haber sido declarado en la misma con un identificador de usuario llamado username . Cada usuario declarado tiene sus propios permisos y perfiles de trabajo. Por lo tanto cada login id debe tener asociado un nombre de usuario, es decir que para cada login id debe existir solo un nombre de usuario o username, en cambio para cada username pueden existir varios login i, esta configuración se llama figura. Es lo que se denomina alias.

Modos de Seguridad

Existen tres modos de seguridad que determinan cómo se crean y mantienen los login id

Seguridad Standard.

En este modo SQL Server es el responsable de mantener y manejar las cuentas de acceso al servidor, de modo que cada vez que se quiera acceder al mismo se debe proporcionar un nombre de usuario asociado con una palabra clave que es almacenada previamente una tabla del sistema. Estos no debe coincidir con el acceso al sistema base.

Seguridad Integrada

En este modo se integran los mecanismos de acceso al sistema con los de SQL. Para el caso de Windows NT se asocian las tablas de ingreso al sistema con SQL por lo tanto todo aquel que tenga una cuenta en Windows NT tendrá automaticamente ingreso al servidor de SQL.

Seguridad Mixta

Es una combinación de las anteriores. Los usuarios que tengan una relación de confianza con el dominio en el que se encuentre el servidor y utilicen una conexión aprovechando esta relación tendrán seguridad integrada. Mientras que los que no posean esta relación utilizaran la seguridad estándar.

Altas de Login ID la tabla SysLogins

Cada login id esta asociado con una clave o password que es la que utiliza el sistema para comprobar la autenticidad del usuario que desea ingresar. No es impresindible pero si aconsejable asignar una password por cada login dado de alta.

La tabla syslogins.

Cuando se da de alta un nuevo usuario el sistema añade un registro a la tabla de sistema syslogins que contiene la siguiente información

− Suid Número único que identifica al usuario

− dbname Nombre de la base de datos por defecto a la que accede

− name Nombre del login id

− password Contraseña del usuario − que aparece encriptada

− Languaje Leguaje por defecto del usuario

− Status Reservado del sistema

− Accdate Reservado del sistema

− totcpu Reservado del Sistema

− tatio Reservado del Sistema

− spacelimit Reservado del Sistema

− timelimit Reservado del Sistema

−resultlimit Reservado del Sistema

El valor identifcativo que utiliza el sistema es el Suid y name que es a titulo solo descriptivo y lo utiliza el sistema para poder acceder al Suid correspondiente.

Contraseñas : Este el uno de los medios que proporciona el sistema para poder controlar el acceso al sistema y se utiliza general mente el el tipo de seguridad standard. Puede contener caracteres alfanuméricos y caracteres especiales su longitud no es restringida solo que no puede tener más de treinta caracteres.

La misma se puede cambiar periódicamente un usuario solo puede cambiar su propia contraseña con el comando

sp_password contraseña antigüa , contraseña nueva [login id]

A diferencia del administrador que puede cambiar la de todos los usuarios, es por eso que se coloca al fina el login id, es decir que si la contraseña no coincide con el usuario que esta tratando de cambiar la misma el sistema no la realiza y lanza un mensaje de error.

Usuarios por defecto

Al instalar el sistema se generan automaticamente, (en caso de no el sistema no participe en casos de replicación) de los usuarios sa y probe y en caso que este realizando replicaciones se generan dos adicionales repl_publisher y repl_subcriber

El administrador sa

El usuario que ingresa con esta password tiene todos los privilegios que su nombre indica es el administrador del sistema y el propietario de las bases de datos propias del sistema (master model tempdb msdb), Además sa es visto por el sistema como el propietario alternativo de todas las bases de datos que se creen el el sistema

Probe

Este identificador de acceso se utiliza solo para monitorear el rendimiento del servidor . Ningún usuario debe acceder con esta password.

rep_publisher

Este identificador se utiliza en el caso de replicaciones y es quién hace accesible a otros servidores copias actualizadas de la bases de datos presentes en el servidor local. Es decir que permite que los procesos que intervienen en la replicación accedan al servidor para poder llevar a cabo su tarea. El login id de este usuario no debe ser modificado y mucho menos utilizarlo para el acceso de usuario convencionales.

rep_suscriber

Se utilizan como el servidor al que se conectarán los servidores suscritos en un proceso de duplicación de datos . Permite que los procesos que intervienen en la suscripción accedan a el para ejecutar los procedimientos que sean necesarios.

Creación del login id

Esta tarea puede realizarse a traves de ordenes o bien con Enterprise Manager

Añadir con Enterprise Manager

Solamente se utiliza el botón Manage Login de la bara de Herramientas de Enterprise Manager y al abrirse el cuadro de diálogo se iningrese el nombre y la contraseña del usuario .

Login ids mediante órdenes

Puede añadir un nuevo usuario mediante la siguiente orden

sp_addlogin usuario [, contraseña`, base defecto[, lenguaje]]]

No es necesario especificar todos los datos solo usuario y contraseña

Modificación y Eliminación de Logins mediate ordenes

Solo indicaremos las órdenes ya que el uso de Enterprise Manager es muy sencillo y solo debemos seguir las indicaciones.

sp_defaultdb nombre de usuario, base de datos

Este procedimiento almacenado falcilita al sistema el nombre por default de la base de datos a que ingresara el usuario al conectarse. Como ya dijimos si no se especifica ingresa a la base de datos master. Como ya dijimos el usuario solo puede hacer esto para las bases de las cuales ellos sean propietarios.

sp_defaultlanguaje nombre de usuario. [lenguaje]

De esta forma se puede asignar un leguaje para que el usuario trabaje, si no se especifica el usuario trabajara en el lenguaje nativo del sistema SQL.

sp_droplogin nombre de usuario

Se utiliza para eleminar un usuario y los que hace es borrar de la tabla syslogin los datos correspondientes a ese usuario .

Este comando solo puede ser realizado por el administrador del sistema.

Aclaración : No puede borrarse ningún usuario que sea propietario de alguno de los objetos del sistema. Es decir que primero se deben eleminar los objetos de los cuales son propietarios y luego dar de baja dicho usuario.

Lo antes expuesto solo se refiere al ingreso a SQL Server y a la tabla de logins.

ACCEDIENDO A LAS BASES DE DATOS.

Como ya dijimos ahora que se ha ingresado a SQL debemos tratar de instrumentar el siguiente nivel de seguridad, y este es restringir el acceso a las diferentes bases de datos y a los datos que ellas contienen para esto se define una figura del sistema de seguridad llamado username, es decir un usuario de base de datos.

Como dijimos un usuario es un identificador que en una determinada base de datos , identifica a un login (persona) un grupo de logins (entidad) que desean acceder a los datos almacenados.

Relación username / login id

Cuando queremos que un cierto usuario de SQL acceda a los datos de una base de datos debemos dar de alta en dicha base un username asociado a su login id. por ello es impresindible cuando se da de alta un usuario indicar a que login id corresponde.

Como ya un username puede tener varios logins y para esto se utiliza la figura del Alias.

Un alias es un nombre de usuario que tiene bajo su tutela varios logins y se almacenan en la tabla de la base de datos denominada sysalternates y permiten cuando se crea un login asignarle un nombre de usuario ya existente . A su vez los usernames se ubican el grupos

lo que permite asignar determinado grupo de permisos a varios usuarios a la vez.

La tabla sysusers

Esta tabla que es análoga a la tabla syslogins para los logins id pues contiene la información relativa a los usernames y los grupos . Cada nuevo usuario y grupo supone una nueva fila en dicha tabla. Pero al contrario de la tabla syslogins que existe solo una para todo el servidor en el caso de la tabla sysusers debe existir una por cada base de datos existente.

Sus componentes son

- Suid Login id asociado al usuario de la base de datos Es el mismo que aparece en la tabla syslogins
- Uid Identificador unico del usuario en la base de datos.

El propietario de una base de datos tiene un uid igual a 1

- Gid Identificado de un grupo de usuarios.

Cuando el identificador numérico de grupo gid sea igual al identificador del usuario uid se entiende que la fila pertenece a un grupo.

- Name Nombre del usuario o grupo
- Environ Resevado por el sistema

suid uid gid name environ

−2 0 0 public (null)

−1 2 0 guest (null)

1 1 0 dbo (null)

El propietario por tanto de la base de datos es dbo ya que tiene el suid y uid iguales a 1(En todas la bases de datos el propietario tiene el uid igua a 1)

Si queremos que varios usuario accedan a la base con los privilegios del dbo y ya que el usuario dbo solo admite un solo login, lo que se hace es crear un alias denominado dbo y de esa manero asignar varios login a dicho alias.

El usuario Guest

Este usuario no se crea por defecto sino que debe ser añadido para que cualquier usuario que se conecte a SQL y no tenga un usuario asignado ingrese con este usuario. Ahora bien este usuario guest debe tener permisos limitados. Este usario siempre tiene un uid igual a 2

Y contiene los privilegios del grupo public.

Altas − Bajas y Modificaciones de Usuarios

Todo el trabajo relacionado con el mantenimiento de los usuarios puede realizarse a travez de la interface gráfica denominada Enterprise Manager o bien a travez de procedimientos almacenados. Nos referiremos en este caso solo a los procedimientos pues la interface antes mencionada tiene sus ayudas en linea.

Los comandos son los siguientes

− sp_adduser login id [, username [, nombre del grupo]

Como puede observarse en la sintaxis el único parámetro obligatorio es el login id al que el nuevo usuario estará asociado. Si se omite el username SQL asume que el nombre es el mismo que el de el login id. Otro dato que se puede agregar es el Grupo al que pertenecerá el usuario , si se omitiese SQL asume que pertenece al grupo Public.

Tanto el grupo como el login id deben existir de antemano . Caso contrario el procedimiento dara un mensaje de error.

− sp_dopuser nombre de usuario

Este procedimiento facilita al sistema del nombre del usuario de la base de datos actual que deseamos eliminar. No se puede eliminar al propietario de la base de datos.

Para obtener información sobre los usuarios existentes en una base de datos se utiliza el procedimiento sp_helpuser [nombre de usuario]. Si omitimos el nombre del usuario la información que nos brinda es la referida a todos los usuario de la base de datos activa.

ALIAS

Recordemos que los alias se utilizan para otorgar acceso a los logins que no tienen definido un usuario en la tabla sysusers sino aprovechado otro username previamente definido en dicha tabla. Por ejemplo el usuario dbo que existe en todas las bases de datos y es el propietario de de dicha base. Por lo tanto podemos asignar al login id un alias que haga referencia a dicho usuario y al ingresar tomará los privilegios del usuario dbo. Como podrá verse esta técnica es usada para asignar privilegios de propietarios a mas de un usuario de la base de datos.

Los alias se graban en la tabla sysalternates que contiene información sobre los alias definidos y existe una

por cada base de datos. Cada vez que se define un alias se agrega una fila a dicha tabla la información que contiene es la siguiente

- Suid Username al que se desea crear un alias
- Altsuid Login id del nuevo usuario.

La metodología seguida al ingresar es la siguiente cuando un usuario trata de acceder a una base de datos SQL busca en la tabla sysusers de la misma para encontrar en la misma una equivalencia entre el login id y un usuario de la base de datos . Si esta búsqueda no tiene éxito el sistema examina la tabal sysaltenates en busca de un alias.

Uno de los procedimientos para crear alias el el siguiente

Sp_addalias login id, username

GRUPOS

Como ya vimos dijimos es posible agrupar usuarios para otorgarle determinados permisos, por lo tanto un grupo esta formado por un conjunto de usuarios en una base de datos. Esto facilita la tarea del administrador de seguridad pues puede otorgar conjuntamente privilegios a un grupo de usuarios. El procedimiento que se sigue es el siguiente primero se crea el grupo independientemente de los usuarios que lo vayan a integrar se le asignan los privilegios y luego se van incorporando al mismo los usuario que se crea conveniente.

No existe relación entre los grupos de Windows NT y los de SQL Server.

Se debe tener en cuenta que un usuario puede pertenecer solo a un solo grupo. De todas maneras el grupo por defecto que existe en todas la bases de datos public es quien contiene a todos los usuarios del sistema.

Como en los casos siguientes se puede agregar grupos a travez de Enterprise Manager o bien con los procedimientos incorporados.

- Sp_addgroup nombre del grupo

La ejecución del este procedimiento agrega una fila a la tabla sysusers de la base de datos activa en la que el Suid y Uid son iguales .

- Sp_helpgroup [nombre del grupo]

Este procedimiento funciona de la misma forma que el sp_helpuser .

PERMISOS DE ACCESO A OBJETOS

Este es el ultimo eslabón el la cadena de seguridad, pues aunque el usuario halla llegado hasta este punto, lo que podemos realizar en este nivel es definir que sentencias y sobre que objetos podrá realizar trabajos el usuario.

Existen dos tipos de permisos : De objeto y De sentencia.

Los permisos de Objeto permiten utilizar y modificar los objetos existentes. Es decir añadir o eliminar filas, como así también realizar una consulta.

Los permisos de Sentencia permiten ejecutarlas, sin limitar este permiso al un objeto en particular. De hecho

estos permisos se refieren a las sentencias del creación y eliminación de objetos.

Por defecto solo el propietario de la base (DBO) y el Administrador del Sistema tienen potestad para realizar creaciones de tablas, modificación de estructuras y/o bajas en la base de datos activa. Además de otorgar permisos de utilización para los objetos creados.

Cada vez que se otorga un permiso se agrega una fila a la tabla sysprotects presenten en todas las bases de datos y que contiene la siguiente información

. id ID del objeto para el que se otorga permiso

. Uid ID del usuario o grupo al que se otorga permiso

. Action Permos que se otorga Pueden ser

- References
- Select
- Insert
- Delete
- Update
- Create table
- Create database
- Create wiew
- Create procedure
- Execute
- Dump Database
- Create default
- Dump Transaction
- Create Rule

− Protect type: Tipo de protección ante la posibilidad de que el permiso pueda trasmitirse

− Columns: Indica las columnas a las que se asigna el permiso

− Grantor: Identificador del usuario que ejecutó la sentencia Grant y revoke que motivó que se añadiese la fila a la tabla.

Permisos a Objetos

Permisos para tablas

- Select
- Update
- Delete
- Insert
- DRI (Declarative Referencial Integrity)

Permisos para columnas

- Select
- Update

Permisos para vistas

- Select
- Update
- Delete
- Insert

Permisos para Procedimientos

- Execute

Asignación de Permisos con Enterprise Manger

Pasos

− Acceder a Server Manager y seleccionar una base de datos

− Pulsar el Botón del menú Object/Permissions . Aparecerá un cuadro de dialogo multipágina con dos secciones By Object y By Users .

La página By Object permite gestionar los permisos para un objeto (tabla vista procedimiente)

La página By Users permite seleccionar a un usuario o grupo de usuarios y gentionar los permisos a objetos para ese conjunto de usuarios.

Otra forma de otorgar permisos a Objetos es mediante las ordenes GRAN y REVOKE por ejemplo

GRANT {ALL | lista de permisos}

ON { objeto[lista de columnas]

TO { Lista de grupos y/o usuarios}

[¨WITH GRANT opción]

. ALL (Mediante esta opción se trasmiten todos los permisos del usuario que ejecuta la orden a quien se la trasmite)

. Lista de permisos (Trasmite solo los permisos que indiquemos)

. Objetos (podrá contener el nombre de una tabla . vista, vista o procedimiento)

. Lista de columnas (Indica las columnas a las que se aplicara el permiso separadas por comas

. WITH GRANT (Esta opción permite al usuario que recibe los permisos a su vez trasmitirlos a otros)

Para quitar permisos

REVOKE { ALL | Lista de permisos}

ON {objeto [lista de columnas]

FROM { lista de usuario o grupos}

[CASCADE]

La opción CASCADE retira todos los permisos que hayan sido dados con la opcion WITH GRANT.

Permisos a Sentencias

Permiten controlar quién puede crear y eliminar objetos de una base de datos. Solo el SA y el DBO pueden administrarlos. Algunas sentencias son

. CREATE DATABASE (solo puede ser concedida por el sa y solo a usuarios incluidos en la base de datos master.)

. CREATE DEFAULT (permite espeficar un valor por defecto para una columna de una tabla)

. CREATE PROCEDURE (para crear un procedimiento almacenado)

. CREATE TABLE (para crear una tabla)

. CREATE VIEW (permite crear una vista)

. DUMP DATABASE (vuelca el contenido de la base de datos a un dispositivo de respaldo)

. DUMP TRANSACTION (vuelca el contenido de la relación de transacciones a un dispositivo de respaldo)

Para agregar o cambiar los permisos de sentencias a travez de Enterprise Manager se realiza lo siguiente

. Acceder a Server Manager y seleccionar una base de datos .

. Pulsar el boton derecho y seleccionar Edit Database

. Cuando aparezca el cuadro multipágina seleccionar permissions

. Modificar los permisos de manera deseada

Mediante ordenes

GRANT [ALL | LISTA SENTENCIAS}

TO [PUBLIC | LISTA DE USUARIOS O GRUPOS}

Para revocarlos

REVOKE { ALL | LISTA DE SENTENCIAS}

FROM {PUBLIC | LISTA DE USUARIOS O GRUPOS}

ADMINISTRACION DE LA SEGURIDAD

Como ya dijimos anteriormente existen tres tipo de seguridad

- Seguridad standard − Es aquella que se basa únicamente en las posibilidades que brinda SQL serves
- Seguridad integrada − Es aquella que utiliza los procedimientos de seguridad de WINDOWS NT conjuntamente con las de SQL
- Seguridad Mixta − Es aquella que utiliza las dos definidas anteriormente

Seguridad Integrada.

Como dijimos este tipo utiliza la integración entre SQL y WINDOS NT . L a idea es que el usuario que se conecte a WINDOWS NT tenga acceso a SQL sin tener que colocar un nuvo login id y una password adicional ,

Para que un servidor pueda configurarse según este modo de seguridad es preciso que las conexiones que se establezcan con el sean trusted. Decimos que una conexión es trusted cuando se realiza desde un sistema que implementa un sistema de validación de usuarios y contraseñas que pueda tener una correspondencia con Windows NT.

Habida cuenta de la menor rigidez del sistema de Seguridad , se hace necesario establecer con cuidado los mecanismos de administración de seguridad que permitan controlar el acceso .

De alguna manera la idea es aprovechar las características de seguridad de WINDOWS NT

Que son superiores a las de SQL.

ESTRATEGIAS DE SEGURIDAD

SEGURIDAD BASADA EN GRUPOS

Cuando existen diversos usuarios que precisen permisos similares, es conveniente basar a la seguridad en los grupos de usuarios. Más que en usuarios individuales . Pues esto reducirá la cantidad de sentencias GRANT y REVOKE necesarias

Par lo cual debemos tener en cuenta que

Todos los usuarios pertenecen al grupo public

Un usuario puede pertenecer solo a un grupo (ademas del grupo public)

Cuando un usuario es incluido en un grupo deja de aparecer como parte del grupo public , pero sigue formando parte de el.

UTILIZACION DE VISTAS

Las vistas permiten limitar el acceso y modificación del los usuarios a los datos de una tabla. Ya que como sabemos una vista es una tabla virtual en la que solo se deja ver la informacion que el usuario necesita a travez del filtrado de las columnas y filas, se puede tener un control más preciso puede o no hacer el usuario asignando permisos solo a vistas.

UTILIZACION DE PROCEDIMIENTOS ALMACENADOS

El uso de procedimientos para acceder y modificar datos es que los usuarios sólo necesitan tener el permiso EXECUTE para ejecutarlo, no precisando que los usuario tengan accesos a las tablas y vistas que conforman el procedimiento.

TRIGGERS PARA CONTROLAR CAMBIOS EN LOS DATOS.

Los triggers son sentencias SQL que se ejecutan de forma automática cuando una tabla se ve modificada mediante cláusulas INSERT, UPDATE O DELETE al ser ejecutados do manera automática

GESTION DE LA SEGURIDAD

Existen básicamente dos herramientas para gestionar y administrar la seguridad en SQL Server

La primera de ellas es a travez de Enterprise Manager para configurar todas las características atinentes al modo de seguridad del servidor.

El segundo es una herramienta que se instala con SQL Server 6.5 denominado SQL Security Manager que es la herramienta que permite aunar la seguridad entre WINDOWS NT y SQL Server.

A TRAVEZ DE ENTERPRISE MANAGER

Esta herramienta que nos proporciona SQL podemos definir todos los aspectos de la gestión de seguridad referida al sistema.

Ingresando a la parte de configuración puede elegirse el modo de seguridad con el que se desea trabajar en la subventana Login Security Mode , en este mismo cuadro puede seleccionarse

Default Login Se utiliza en seguridad integrada. El nombre de usuario que se utilizará por defecto al acceder a SQL, si el nombre del usuario que accede en seguridad integrada no aparece en la lista de los definidos en SQL. Recordemos que estos identificadores apararecerán en la tabla syslogins. La especificación de este usuario es imprecindible para que puedan acceder usuarios no explícitamente identificados en SQL.

Default domain Análogo al anterior pero para el dominio de red . Este dominio será el que utilizará para buscar el nombre del usuario que intenta acceder si este no especifica explícitamente en el momento de la conexión.

En el modo de seguridad integrada es posible además solicitar que WINDOWS NT mantenga un registro de los intentos , exitosos o no de acceder al servidor . Es lo que se denomina audit level . Se puede especificar que se registren los intentos con éxito , infructuosos o ambos. Este registro aparecerá en el visor de sucesos de WINDOS NT. Es decir en event log

En WINDOWS NT el identificador de un usuario es la suma de su dominio y el nombre de usuario cuando se ingresa explícitamente son separados por \ , este carácter no es utilizable en SQL por lo que deberá traducirse a otro identificador, Es por esto que existen los servicios mapping donde podemos indicar la conversión de \ por _ para que sean reconocidos por SQL −

INTEGRACION DE USUARIOS Y GRUPOS DE WINDOWS NT

Y SQL SERVER

Cómo dijimos anteriormente cuando la seguridad esta configurada en los modos integrada y mixta la validación de el acceso de los usuarios a SQL , es confiada a WINDOWS NT. En estos entornos es fundamental definir una correspondencia entre los usuario WINDOWS NT y los de SQL. El procedimiento habitual es el siguiente.

- Crear usuarios y grupos WINDOWS NT, a los que se les dará acceso a SQL. Para ello utilizaremos el

Administrador de Usuarios de Windows NT.
- Proporcionar a esos usuarios el acceso a SQL, mediante el SQL Security Manager.

La forma más habitual de llevar a cabo este proceso es crear dos grupos locales diferentes de usuarios denominados

SQLusers Con privilegios de usuarios en Windows NT

SQLadmins Con privilegios de administrador del sistema

Una vez creados estos grupos agregar los usuarios que vayan a tener determinadas características en cada uno de los grupos antes mencionados.

Es recomendable utilizar nombres para los usuarios NT que sean identificadores válidos para SQL.

FIGURAS DE ADMINISTRACION : USER Y SA

En SQL Security Manager solo existen dos grupos de usuarios desde el punto de vista de los permisos que poseen estos son los usuarios (users) y los administradores (sa).

Los usuarios pueden administrase sólo en estos dos grupos, de tal manera que existen dos vistas diferenciadas : las vistas de los usuarios con privilegios user y la de los usuarios con privilegios sa

UTILIZACION DE SQL SECURITY MANAGER

Como dijimos anteriormente esta aplicación sirve a los efectos de permitir la fácil unión entre los usuarios NT y los de SQL su creación y modificación .

Una vez que hemos iniciando SQL Security Manager se nos presentará un dialogo que nos solicitará el nombre del servidor que deseamos administrar en sus aspectos de seguridad.

Una vez que ingresamos el nombre del servidor su login y password y si son correctos se oprime un botón denominado connect se presentara la pantalla base de SQL Security Manager donde aparecen los dos grupos USERS y SQL USER y en la barra de tareas las dos opciones de configuración como privilegios de usuarios o de administrador.

Cabe destacar que esas distinciones pueden hacerse exclusivamente con grupos completos , No con usuarios individuales.

Al hacer clic en WIEW/USER PRIVILEGE y luego en el menu SECURITY/GRANT NEW se despliega una pantalla con todos los grupos Windows NT, se selecciona el grupo al que se quiere otorgar permisos en SQL . Se puede crear username para cada integrante del grupo, agregar bases de datos por defecto etc.

Con este utilitario se puede dar privilegios quitar o modificarlos de manera muy sencilla solo siguiendo las instrucciones de los menus de ayuda que como todo producto Microsoft tiene incorporado.

DUPLICACION DE DATOS ENTRE SERVIDORES

La duplicación de datos entre servidores, conocida en SQL Server como replication, es la capacidad para duplicar una fuente de datos en diferentes servidores de manera fiable. La duplicación consiste en distribuir datos de solo lectura a otros servidores que pueden estar muy alejados, y actualizarlos con periodicidad. A este proceso de actualización se lo llama sincronización

Distribución y duplicación

Cuando procedemos a distribuir los datos, cada servidor del sistema contiene una porción de la información, con lo que la suma de la acción coordinada de todos los servidores proporciona un sistema gestor de bases de datos único.

En la duplicación, la información se centraliza en un único servidor que proporciona el resto una copia de sus datos, ocupándose de mantener dicha copia actualizada.

El modelo de sincronización

¿Cómo hacer que los datos que el servidor remoto tendrá en todo momento a su disposición sean los mismos que se almacenan en la fuente de datos en todo momento?. El proceso de duplicación de datos en SQL Server utiliza un modelo de actualización que Microsoft llama en tiempo lo suficientemente real, es decir, con unos períodos de respuesta a los cambios, lo suficientemente pequeños como para que los datos en la copia remota resulten siempre válidos. Este modelo es denominado de consistencia no estricta para la sincronización de los datos entre la fuente y el destino.

El modelo de consistencia estricta garantiza que la copia de los datos contiene permanentemente la misma información que el original, pero requiere conexiones de alta velocidad entre los servidores que intervienen en la duplicación.

El modelo de consistencia no estricta permite que exista un intervalo de tiempo no nulo entre el momento en el que se producen modificaciones en los datos originales y el instante en el que dichos cambios se reflejan en la copia. Los requerimientos de comunicaciones son mucho menores.

Duplicación de datos en SQL Server

Publicar y suscribirse

Los servidores publican los datos a duplicar y los servidores destino de dichos duplicados se suscriben a estas publicaciones.

Publicaciones y artículos

El artículo puede contener una o varias tablas, bases de datos completas, vistas o porciones horizontales o verticales de información.

A su vez, una publicación puede constar de uno o varios artículos. Podríamos decir que la publicación se enmarca en el nivel de la base de datos, mientras que los artículos se orientan a la tabla.

Transaction log

Cuando una transacción que se haya aplicado a la base de datos fuente se marca como susceptible de duplicación, se almacena en el *transaction log*. En el momento de la actualización, el sistema leerá el *log* y los cambios a los que se haya visto sometida la fuente de datos se reflejarán en los servidores que reciban la copia.

Tablas no duplicables: Las tablas de sistema en la base de datos master

Las tablas que carezcan de clave primaria.

Modelos de duplicación

Figuras en el modelo de duplicación de datos

Publicador: Contiene los datos fuente. Mantiene las bases de datos y las hace disponibles a los suscriptores.

Suscriptor: Recibe los datos duplicados. Deberá ocuparse de establecer y mantener una conexión con el publicador.

Distribuidor: El trabajo de este servidor e transmitir los datos desde los publicadores a los suscriptores.

En muchas ocasiones el publicador realiza también las tareas de distribuidor.

Modos de duplicación

Publicador único

Publicador único / distribuidor

Publicadores y suscriptores múltiples

El proceso de duplicación de datos

Componentes del proceso

Proceso lector del transaction log

Este proceso detecta en la relación de transacciones de las bases de datos publicadas aquellas transacciones que afectan a datos publicados y las ubica en la base de datos de distribución.

Servidor y base de datos de distribución

El servidor de distribución y la base de datos del mismo nombre sirven de puente entre el publicador y los suscriptores. Si se utiliza un distribuidor remoto, se alivia el trabajo del servidor que publica y por consiguiente mejora el rendimiento.

Proceso de sincronización

Este proceso asegura que la base de datos publicada y las bases de datos suscritas estén sincronizadas antes de comenzar el proceso de duplicación.

El proceso de sincronización inicial

Cuando se crea una publicación debe llevarse a cabo un proceso de sincronización antes de que puedan comenzar a duplicarse los datos. Este proceso garantiza que los suscriptores tengan preparadas las estructuras de datos de almacenamiento para la recepción de los datos. Es lo que llamaremos sincronización inicial.

La sincronización inicial asegura que en el suscriptor y publicador cada uno de los objetos que forman parte de la duplicación se hallen en el mismo estado ya sea tanto la estructura de las tablas como los propios datos que albergan.

Sincronización inicial automática

El proceso se lleva a cabo a través de tareas automáticas que radican en la base de datos de distribución, a intervalos regulares y solo sobre aquellos suscriptores que lo hayan solicitado desde el último proceso de sincronización.

Sincronización inicial manual

Requiere que un usuario sincronice las bases de datos y que notifique a SQL Server que dicho proceso se ha completado. El proceso en sí es llevado a cabo por el publicador igual a lo expuesto anteriormente. La diferencia estriba en cómo esas copias son transferidas a los suscriptores. En este caso será el usuario el que procederá a la transferencia de la información almacenada en los ficheros hacia los suscriptores.

Este modo es útil cuando se manejan grandes tablas o líneas de comunicación lentas.

Sin sincronía

En este modo se asume que los artículos en la fuente están ya en sincronía con los artículos en el destino. No se recomienda en sistemas con usuarios no avanzados.

Instalando el servicio de duplicación

Planificación de la duplicación de datos

Ciertos requerimientos deben cumplirse antes de poder proceder a la duplicación:

- Los servidores implicados deben poseer las relaciones de confianza adecuadas entre ellos
- Deben haber sido instalados los Drivers ODBC (Open Database Conectivity) de 32 bits.
- Si el servidor es publicador o servidor de distribución Windows NT deberá disponer al menos de 32 MB de memoria, con 16 MB asignados a SQL Server.
- La base de datos publicada debe tener espacio suficiente para el *Transaction log.*
- Todas las tablas que se deseen publicar deben tener claves primarias.

Creación de base de datos de distribución

Este es el primer paso necesario para proceder a la duplicación. El proceso de creación puede realizarce desde *Enterprise Manager* y consta de los siguientes pasos:

- Registrar el servidor en el que se creará la base de datos de distribución.
- Seleccionar el servidor en el que se desea instalar la base de datos de distribución.
- Crear el dispositivo para la base de datos y el dispositivo para el *transaction log*.
- Seleccionar la orden de menú *Server / Replication Configuración*.
- Seleccionar la opción *Install Publishing*. Aparecerá el cuadro de diálogo *Install Replication Publishing*.
- Seleccionar *Install New Local Distribution Database*.
- Seleccionar los dispositivos para los datos y el *transaction log* de la base de datos de distribución.
- Introducir un valor para el tamaño de ambos objetos. El tamaño mínimo recomendado es de 30 MB para los datos y 15 MB para el *transaction log.*

Publicación de una base de datos completa

Para hacerlo deberán seguirse los siguientes pasos en *Enterprise Manager*:

- Seleccionar el servidor publicador de la lista de los que estemos administrando en la ventana *Server*

Manager.

- Seleccionar el menú *Manage / Replication* y la opción *Publications*. Aparecerá el cuadro de diálogo *Manage Publications*.
- En el cuadro de diálogo aparecerán aquellas bases de datos que hayan sido activadas para la publicación en la configuración del publicador que vimos anteriormente.
- Para crear una nueva publicación, seleccionar una de esas bases de datos y pulsar el botón *New*. Un nuevo cuadro de diálogo (*Edit Publications*) permitirá introducir el nombre de la publicación a crear.
- Elegir el modo de frecuencia de duplicación: basado en transacciones o según el refresco periódico de las tablas que la integran.
- Elegir la publicación de todas las tablas, es decir, de la base de datos completa. Esto motiva la creación automática de un artículo para cada una de las tablas que contenga la base de datos.
- Una vez creada la publicación aparecerá en la ventana *Server Manager*.

Modo de sincronización

El cuadro de diálogo *Edit Publications* permite especificar el modo de sincronización. Se refiere a la manera en cómo se crean los archivos de información en el momento de crear la publicación. Existen dos modos:

− *Native Format*: Formato optimizado para usar entre servidores SQL Server.

− *Character Format*: Formato no optimizado para uso con sistemas de otro tipo.

Seguridad

El cuadro de diálogo *Edit Publications* permite también especificar las opciones de seguridad de la publicación. Es posible restringir o permitir el acceso de cada posible servidor suscriptor.

SISTEMA SQL BASE OBJETOS

DE DE

OPERATIVO SERVER DATOS DATOS

Publicador

Suscriptor

Suscriptor

Suscriptor

Publicador

Distribuidor

Suscriptor

Suscriptor

Suscriptor

Publicador

Publicador

Suscriptor

Suscriptor

Suscriptor