UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA AUDITORA EN SERVIDORES SQL SERVER SYBASE

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A N

BAÑOS LIRA VERÓNICA
MORÁN CASTILLO SILVIA

DIRECTOR

ING. ARMANDO VEGA ALVARADO

COODIRECTOR

ING. ALBERTO GONZÁLEZ GUIZAR

México, D.F. Marzo del 2006

Agradecimientos

La felicidad existe para aquellos que lloran, aquellos que les duele, aquellos que han buscado, aquellos que han tropezado; porque solamente ellos pueden apreciar la importancia de las personas que han tocado sus vidas.

A mi papá Juan que con sus sabios consejos me ayudó a comprender que en la vida esta permitido caerse pero lo que no esta permitido es no levantarse y que no hay mejor manera de superarse que con trabajo, esfuerzo, sacrificio y mucha dedicación.

A mi Mamá Carmela porque fuiste un gran ejemplo, por tu fortaleza, tu dedicación, tu lucha continua para sacar a una familia adelante tu sola.
Ya no estas aquí físicamente, pero espero que desde donde estés te sientas muy orgullosa de mí.

A mis hermanos:
Maru porque has sido mi compañera de peleas, por tu forma de ser tan bondadosa y tierna, por tu apoyo cuando más lo necesitaba, por ser mi amiga.
Gustavo por tu amor, por tu apoyo y por creer en mí.

A mi hijo Diego porque es mi razón de ser, por cambiar el olor de mis mañanas, sonreírme todos los días y porque que cuando me abrazas y me dices que me amas me ciento invisible, gracias por hacerme saber que soy importante para ti, porque te observo me hace sentir fuerte el verte crecer y gracias también por enseñarme lo valioso que es cada momento. Espero que siempre te sientas orgulloso de mí.

A Israel: por aparecer en mi vida y confrontarme, por enseñarme y aprender conmigo, por tu amor y tu presencia, gracias mi amor.

A mi Director de Tesis: por su asesoramiento, su paciencia, por su amistad y estímulo para seguir creciendo intelectualmente.

A la Mat. Yolanda Valencia Aguilar por su apoyo y acceder a la realización de esta tesis y gracias también por dejarme ser parte de este equipo de trabajo.

A mi jefa Ing. Graciela Valencia por darme el tiempo, la facilidad para poder realizar esta tesis, por su tolerancia, por darme la oportunidad de desarrollarme profesionalmente y sobre todo por sus consejos que me han ayudado a crecer.

Nuevamente agradezco a Yolanda y Graciela por ser un ejemplo a seguir, por ser mujeres profesionistas y enseñarme que la maternidad no es un impedimento para poder alcanzar metas.

A mis suegros por su gran apoyo, por dejarme ser parte de su familia, por cuidar de mi hijo lo más valioso que tengo en la vida, para que yo me pudiera continuar la escuela; de verdad mil gracias.

A todos mis maestros de la facultad por su tiempo y conocimientos; por haberme preparado profesionalmente para enfrentar los retos que se me presenten en la vida.

A mis amigos por hacer inolvidable el paso por la universidad, por estar conmigo cuando los necesito muy especialmente a Iliana, Arely, Jessica, Viky, Carina, Julio y Juan.

A mi Amiga y compañera de tesis por tolerarme, por su paciencia y por compartir conmigo todo su conocimiento.

A mis amigos y compañeros trabajo muy en especial a:
Graciela, Minerva, Diana, Mauricio, Emilio, Sergio, José Carlos, Daniel, Soledad, Dora, Jorge, Guen, Patricia, Armando y Blanca.
Muchas gracias por su interés, por dejarme aprender de cada uno de ustedes y sobre todo por su amistad.

A la Universidad Nacional Autónoma de México por que no solo me esta dando armas para desarrollarme profesionalmente, sino que también me enseñó:

1.- Que puedo cambiar muchísimo y ni siquiera notarlo.
2.- Que puedo querer a muchas personas de diferentes formas.
3.- Que los niños de la universidad también avientan papelitos.
4.- Que normalmente conoces ahí el amor de tu vida, al cual siempre recordarás.
5.- Que copiar puede ser sinónimo de verificar o corregir.
6.- Que se puede estar en una fiesta la noche anterior al examen final.
7.- Que si llevas una chamarra que te cubra completamente del frío, todos te preguntarán: ¿Dónde cayo la nevada?
8.- Que existen materias que requieren mas tiempo que todas las clases juntas.
9.- Que puedo saber todo y reprobar el examen.
10.- Que puedo no saber nada y sacar una buena calificación.
11.- Que la mayor parte de mi educación la obtendré fuera del salón de calases.
12.- Que la clase es un lugar para visitar cuando hay un examen.
13.- Que es posible que estés solo aun cuando estas rodeado de mil personas.
14.- Que tus amigos tomarán caminos diferentes, pero que nunca se olvidarán de la amistad.
15.- Que si se pudiera volver a la universidad, para volver a estar con mis amigos y revivir tantas cosas Bellas.
16.-Que cada reloj del campus tiene diferentes horas.
17.-Que los amigos son quienes hacen de este, un lugar valioso e importante,
18.- Que valió la pena salirse de clase para apoyar a los PUMAS en nuestro estadio olímpico universitario.
19.- Que la UNAM fue y seguirá siendo nuestra máxima casa de estudios.

Porque al pasar unos años después no se recordara muy bien lo
Que se aprendió en clase, pero siempre recordaras todo aquello que viviste con tus
Amigos... con quienes estudiaste, te desesperaste, lloraste, gritaste y festejaste, con
Quienes compartiste todo tu tiempo.

"Quien logro hacer un amigo en la universidad, logro hacer un amigo para toda la vida "

¡¡Gooooya!!, ¡¡¡Gooooya,!!!

Cachun, cachun ra ra, cachun cachun ra ra,

¡¡¡¡¡Goooooooya!!!!!!

¡¡¡¡¡¡¡¡¡¡UNIVERSIDAD!!!!!!!!!!

Verónica Baños Lira

Agradecimientos

Cuando inicié este trabajo de tesis pensé que sería un punto culminante en mi trayectoria académica, pero ahora que terminó me doy cuenta que es sólo un eslabón más en la cadena de mi vida, y en esta 'mi vida' ya son muchos los eslabones entrelazados.

Quiero dedicar esta tesis a las personas más importantes que han estado siempre a mi lado:

A mi mami:

Por que con todo el amor que me brindas, has llenado los momentos más difíciles, con los que he aprendido a levantarme cada vez más fuerte, enfrentando el presente con amor y alegría. Por la presión ejercida para terminar esta tesis, por repetirme todos los días "Y la tesis para cuando", "Ya titúlate".

A mi papito:

Por que con tu ejemplo me has enseñado a alcanzar mis metas, a ser constante y así luchar contra la adversidad. Es importante saber que cuando levanto la mirada siempre estás ahí, dándome la mano para ayudarme.

A ambos, con mucho cariño por todos los esfuerzos y sacrificios que realizaron para sacarme adelante, por sus horas de trabajo, por enseñarme a vivir, porque cuando los he necesitado siempre están ahí sabiendo como animarme y enseñándome el camino para seguir creciendo. Ustedes son mi inspiración para seguir luchando y trabajando. Gracias mamá y papá, me enorgullece tener unos padres como ustedes, siempre dispuestos a escuchar y apoyarme, los quiero mucho.

A mi hermana Mine, por la paciencia, el apoyo y su ayuda incondicional que me han motivado a superarme constantemente y a Juguete por ser tan latoso, por brindarme siempre tu compañía, cariño y la alegría que le das a mi vida todos los días.

A mis abuelitos, a todos mis tíos y primos, que siempre me han apoyado (perdón que no los incluya en una lista pero son tantos que no quisiera que nadie me falte). Gracias por los buenos y malos momentos, por aguantarme y por escucharme.

A mi director de tesis, Ing. Armando Vega, cualquier palabra de agradecimiento no reflejará cuan profunda es mi gratitud por toda tu invaluable ayuda y apoyo para la realización de este trabajo, ya que no sólo he recibido consejos de tu parte, sino que me has brindado tu amistad y confianza.

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería, que me brindaron la oportunidad y los medios para prepararme.

A toda la gente que me ha apoyado directa o indirectamente en este trabajo, principalmente a la Mat. Yolanda Valencia Aguilar por permitirme pertenecer a su equipo de trabajo, a mis compañeros del Departamento de Servidores Unix y Sybase: Armando y Guendaviani, del departamento de Soporte de Sistemas: la Ing. Graciela Valencia por todo su apoyo, a Mauricio, Diana, Sergio, José Carlos, Emilio, Dany, además a Paty, Rita y a todos los demás que conforman la SSRE muchas gracias.

Por último quiero dar las gracias a todos aquellos que me han devuelto una sonrisa, a todos aquellos que me ofrecieron su apoyo en tiempos difíciles, a todos los que han puesto de su parte para que el trajín diario sea más llevadero y muy en especial a la vida por todo lo que me ha dado.....

Silvia Morán Castillo.

ÍNDICE TEMÁTICO

Prólogo

Hoy en día se considera a la información de una empresa como uno de los activos más valiosos, por lo que la seguridad de la misma es muy importante.

Dicha información frecuentemente se encuentra almacenada en sistemas conocidos como Sistemas Manejadores de Bases de Datos (DBMS). Dada su importancia en las organizaciones, los DBMS deben proporcionar información veraz y exacta de los datos que ellos almacenan, por lo que es vital mantener un estricto control de las actividades que se llevan a cabo en estos sistemas.

La presente tesis tiene como objetivo, presentar los conocimientos y herramientas básicas para diseñar e implementar un protocolo de seguridad para la auditoría de los servidores de bases de datos en servidores SQL Server Sybase, y con base en éste, implementar una herramienta auditora para detectar puntos vulnerables dentro del mismo, que ponga en riesgo la seguridad de su información.

La primera parte contiene los conceptos teóricos en los que se basa la implantación del protocolo de seguridad para la auditoría de servidores de bases de datos:

Capítulo 1. En este capítulo se presenta una breve historia acerca de las bases de datos. Se define el marco teórico de las bases de datos y sus componentes, las funciones que debe desempeñar un administrador de base de datos las cuales son importantes para el desarrollo e implementación de las mismas.

Capítulo 2. Describe los aspectos teóricos de la seguridad en base de datos. Se proporciona la definición de seguridad en base de datos y sus elementos, así como las razones por las cuales es un tema que debe tomarse en cuenta actualmente como un elemento indispensable de la cultura computacional.

La segunda parte contiene la exposición, desarrollo y la presentación final de la herramienta auditora:

Capítulo 3. Describe un esquema de seguridad en el DBMS Sybase. Proporciona los métodos para la administración del servidor, cuentas de login, creación de usuarios dentro de las bases de datos, otorgar y revocar permisos, la implementación de la auditoría, etc.

Se explican los procedimientos, herramientas y lenguajes a utilizar para la construcción de nuestra herramienta auditora.

Capítulo 4. Describe la herramienta que se ha desarrollado como parte de esta tesis, para cubrir necesidades específicas de seguridad en servidores SQL Server Sybase.

Por otro lado se muestran los distintos reportes generados por la herramienta auditora, los cuales permiten verificar la consistencia de las bases de datos así como su integridad con respecto a la información contenida.

Conclusiones. Conclusiones de la tesis

Glosario de Términos. Este glosario servirá como apoyo para el lector.

Anexos. El primer anexo contiene una lista de los Procedimientos Almacenados de la versión 12.5 de Sybase, traducidos en español. El segundo anexo contiene información acerca de los valores en evento y en la columna extrainfo de la tabla sysaudits_01. En el tercer anexo se tiene un listado de algunas rutinas DBLibrary (traducidas en español). Finalmente el último anexo contiene información acerca de las tablas de sistema del Adaptive Server Enterprise 12.5 de Sybase, que consiste en una breve descripción de cada tabla y sus respectivas columnas.

CAPITULO 1. ANTECEDENTES

1.1 Reseña Histórica

Se dice que los sistemas de bases de datos tienen sus raíces en el proyecto estadounidense Apolo de mandar al hombre a la luna, en los años sesenta. En aquella época, no había ningún sistema que permitiera gestionar la inmensa cantidad de información que requería el proyecto. La primera empresa encargada del proyecto, NAA (North American Aviation), desarrolló un software denominado GUAM (General Update Access Method) que estaba basado en el concepto de que varias piezas pequeñas se unen para formar una pieza más grande, y así sucesivamente hasta que el producto final está ensamblado. Esta estructura, que tiene la forma de un árbol, es lo que se denomina una estructura jerárquica. A mediados de los sesenta, IBM se unió a NAA para desarrollar GUAM en lo que ahora se conoce como IMS (Information Management System). El motivo por el cual IBM restringió IMS al manejo de jerarquías de registros fue el de permitir el uso de dispositivos de almacenamiento serie, más exactamente las cintas magnéticas, ya que era un requisito del mercado por aquella época.

A mitad de los sesenta, se desarrolló IDS (Integrated Data Store), de General Electric. Este trabajo fue dirigido por uno de los pioneros en los sistemas de bases de datos, Charles Bachmann. IDS era un nuevo tipo de sistema de bases de datos conocido como sistema de red, que produjo un gran efecto sobre los sistemas de información de aquella generación. El sistema de red se desarrolló, en parte, para satisfacer la necesidad de representar relaciones entre datos más complejos que las que se podían modelar con los sistemas jerárquicos, y, en parte, para imponer un estándar de bases de datos. Para ayudar a establecer dicho estándar, CODASYL (Conference on Data Systems Languages), formado por representantes del gobierno de EEUU y representantes del mundo empresarial, formaron un grupo denominado DBTG (Data Base Task Group), cuyo objetivo era definir unas especificaciones estándar que permitieran la creación de bases de datos y el manejo de los datos. El DBTG presentó su informe final en 1971 y aunque éste no fue formalmente aceptado por ANSI (American National Standards Institute), muchos sistemas se desarrollaron siguiendo la propuesta del DBTG. Estos sistemas son los que se conocen como sistemas de red, o sistemas CODASYL o DBTG.

Los sistemas jerárquicos y de red constituyen la primera generación de los SGBD. Pero estos sistemas presentaron algunos inconvenientes:

- Era necesario escribir complejos programas de aplicación para responder a cualquier tipo de consulta de datos, por simple que ésta sea.

- La independencia de datos era mínima.

- No tenían un fundamento teórico.

En 1970 Codd, de los laboratorios de investigación de IBM, escribió un artículo presentando el modelo relacional. En este artículo, presentaba también los inconvenientes de los sistemas previos, el jerárquico y el de red. Entonces, se comenzaron a desarrollar muchos sistemas relacionales, apareciendo los primeros a finales de los setenta y principios de los ochenta. Uno de los primeros es System R, de IBM, que se desarrolló para probar la funcionalidad del modelo relacional, proporcionando una implementación de sus estructuras de datos y sus operaciones. Esto condujo a dos grandes desarrollos:

- El desarrollo de un lenguaje de consultas estructurado denominado SQL, que se ha convertido en el lenguaje estándar de los sistemas relacionales.

- La producción de varios SGBD relacionales durante los años ochenta, como DB2 y SLQ/DS de IBM, y ORACLE de ORACLE Corporation.

Hoy en día, existen cientos de SGBD relacionales, tanto para microordenadores como para sistemas multiusuario, aunque muchos no son completamente fieles al modelo relacional.

Otros sistemas relacionales multiusuario son INGRES de Computer Associates, Informix de Informix Software Inc. y Sybase de Sybase Inc. Ejemplos de sistemas relacionales de microordenadores son Paradox y dBase IV de Borland, Access de Microsoft, FoxPro y R:base de Microrim.

Los SGBD relacionales constituyen la segunda generación de los SGBD. Sin embargo, el modelo relacional también tiene sus fallos, siendo uno de ellos su limitada capacidad al modelar los datos. Se ha hecho mucha investigación desde entonces tratando de resolver este problema. En 1976, Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de bases de datos. En 1979, Codd intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T (1979) y más recientemente RM/V2 (1990). Los intentos de proporcionar un modelo de datos que represente al mundo real de un modo más fiel han dado lugar a los modelos de datos semánticos.

Como respuesta a la creciente complejidad de las aplicaciones que requieren bases de datos, han surgido dos nuevos modelos: el modelo de datos orientado a objetos y el modelo relacional extendido. Sin embargo, a diferencia de los modelos que los preceden, la composición de estos modelos no está clara. Esta evolución representa la tercera generación de los SGBD.

1.2 Concepto de Base de Datos

¿Qué es una base de datos?

- "Es un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo".[1]

- "Se designa una colección de datos que es administrada por un sistema de administración de base de datos".[2]

Definición

Una **base de datos** es un conjunto de datos relacionados entre sí. Por **datos** entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito. La definición anterior es muy general, ya que la acepción común del término *base de datos* suele ser más restringida. Una base de datos tiene las siguientes propiedades implícitas:

- Una base de datos representa algún aspecto del mundo real, en ocasiones llamado minimundo o universo de discurso. Las modificaciones del minimundo se reflejan en la base de datos.

- Una base de datos es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos.

- Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

De esta forma, se define Base de Datos como "Colección o depósito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, únicas para cada tipo de datos, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos".

[1] De Miguel, Adoración y Piattini, Mario Gerardo, ***Concepción y Diseño de Base de Datos***, Segunda Edición, Addison-Wesley Iberoamericana, México 1999, p. 43.
[2] Ullman, Jeffrey D. y Widon, Jennifer, ***Introducción a los Sistemas de Base de Datos***, Prentice-Hall Hispanoamericana, S.A., México 1999, p. 1.

Las bases de datos pueden ser de cualquier tamaño y tener diversos grados de complejidad. La generación y mantenimiento de las bases de datos pueden ser manuales o mecánicos. Las bases de datos computarizadas se pueden crear y mantener con un grupo de programas de aplicación escritos específicamente para esa tarea, o bien mediante un sistema de gestión de bases de datos.

1.3 Que es un Sistema de Base de Datos

Un sistema de bases de datos es un sistema computarizado de información para el manejo de datos por medio de paquetes de software llamados sistemas de manejo de bases de datos (DBMS). Los cuatro componentes principales de un sistema de bases de datos son la información, el hardware, el software DBMS y los usuarios. Ver figura 1.

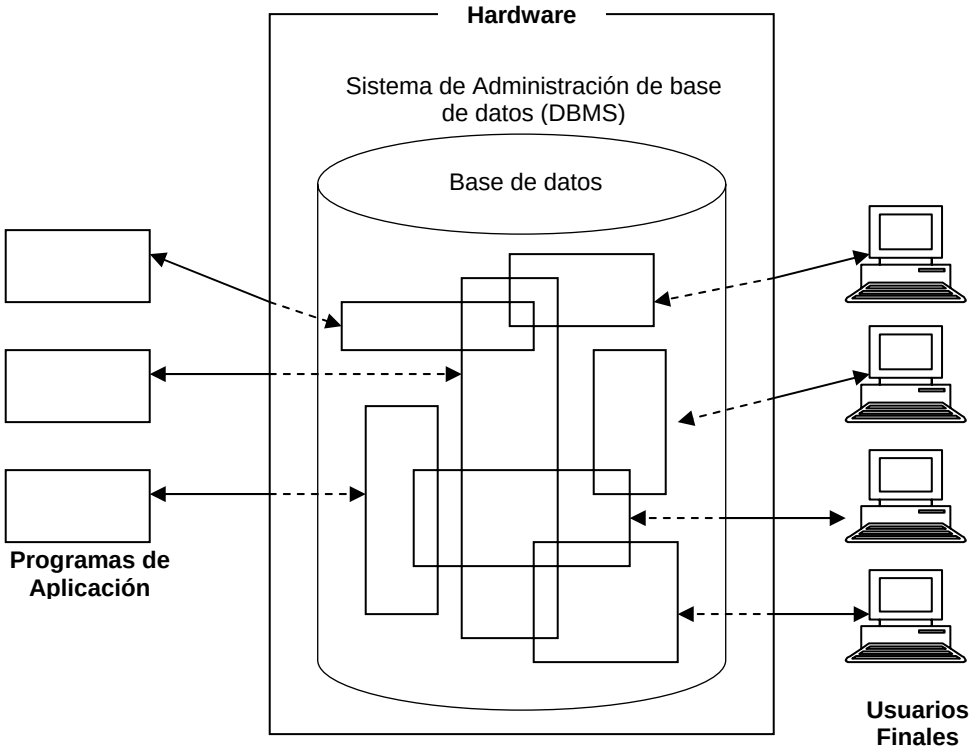


Figura 1. Imagen simplificada de un sistema de base de datos

Hardware.

Los componentes del hardware del sistema son:

- Los volúmenes de almacenamiento secundario - por lo regular discos magnéticos de cabeza móvil - donde se conservan los datos almacenados, junto con los dispositivos de E/S asociados (unidades de disco, etc...), controladores de dispositivos, canales de E/S y demás.

- El procesador o procesadores y la memoria principal asociada que hacen posible la ejecución de los programas del sistema de bases de datos.

Software.

Entre la base de datos física misma (los datos y como están almacenados) y los usuarios del sistema existe un nivel de programas, el manejador de base de datos o, en la mayoría de los casos, el sistema de administración de base de datos (SMBD, database management system).

Usuarios.

Podemos definir a los usuarios como toda persona que tenga todo tipo de contacto con el sistema de base de datos desde que este se diseña, elabora, termina y se usa.

Hay 4 tipos diferentes de usuarios, diferenciados por la forma en que esperan interactuar con el sistema.

- Programadores de aplicaciones: Los profesionales en computación interaccionan con el sistema por medio de llamadas en DML, las cuales están incorporadas en un programa escrito en un lenguaje principal (Cobol, Pascal, etc...). Estos son programas de aplicación. Las llamadas en DML están precedidas de un carácter especial de forma que se pueda generar el código apropiado. Un preprocesador especial, llamado precompilador de DML, convierte las sentencias en DML a llamadas normales en el lenguaje de programación. El programa resultante se ejecuta entonces por el compilador del lenguaje de programación, el cual genera el código objeto apropiado.

- Usuarios sofisticados: Los usuarios sofisticados interactúan con el sistema sin escribir programas. En cambio escriben sus preguntas en un lenguaje de consultas de base de datos. Cada consulta se somete a un procesador de consultas cuya función es tomar una sentencia en y descomponerla en instrucciones que entienda el gestor de bases de datos.

- Usuarios especializados: Usuarios sofisticados que escriben aplicaciones de bases de datos que no encajan en el marco tradicional de procesamiento de datos; sistemas de diseño ayudados por computadoras, sistemas expertos y basados en conocimiento, etc.

- Usuarios ingenuos: Los usuarios no sofisticados interactúan con el sistema invocando a uno de los programas de aplicación permanentes que se han escrito anteriormente en el sistema de base de datos, podemos mencionar al usuario ingenuo como el usuario final que utiliza el sistema de base de datos sin saber nada del diseño interno del mismo por ejemplo: un cajero.

Información.

Es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario; para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe guardar lógicamente en archivos. En general, la información en la base de datos estará integrada y además será compartida. Integrada significa que la base de datos puede considerarse como una unificación de varios archivos de datos, por lo demás distintos, y que elimina del todo o en parte cualquier redundancia entre ellos.

Compartida significa que los elementos individuales de información en la base de datos pueden compartirse entre varios usuarios distintos, en el sentido de que todos ellos pueden tener acceso al mismo elemento de información (y diferentes usuarios pueden utilizarlo para propósitos diferentes). Esta capacidad de compartir (en forma simultánea o no) se desprende en parte de la integración de la base de datos.

Objetivos de los sistemas de Bases de Datos.

Los objetivos principales de un sistema de base de datos es disminuir los siguientes aspectos:

- **Redundancia e inconsistencia de datos.**

Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información sea más de una vez en un dispositivo de almacenamiento. Esto aumenta los costos de almacenamiento y acceso a los datos, además de que puede originar la inconsistencia de los datos -es decir diversas copias de un mismo dato no concuerdan entre sí -, por ejemplo: que se actualiza la dirección de un cliente en un archivo y que en otros archivos permanezca la anterior.

- **Dificultad para tener acceso a los datos**.

Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos. Supóngase un banco, y que uno de los

gerentes necesita averiguar los nombres de todos los clientes que viven dentro del código postal 78733 de la ciudad. El gerente pide al departamento de procesamiento de datos que genere la lista correspondiente. Puesto que esta situación no fue prevista en el diseño del sistema, no existe ninguna aplicación de consulta que permita este tipo de solicitud, esto ocasiona una deficiencia del sistema.

- **Aislamiento de los datos.**

Puesto que los datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.

- **Anomalías del acceso concurrente.**

Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente. En un entorno así la interacción de actualizaciones concurrentes puede dar por resultado datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.

- **Problemas de seguridad.**

La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la autentificación y protección de los datos. En un banco por ejemplo, el personal de nóminas sólo necesita ver la parte de la base de datos que tiene información acerca de los distintos empleados del banco y no a otro tipo de información.

- **Problemas de integridad.**

Los valores de datos almacenados en la base de datos deben satisfacer cierto tipo de restricciones de consistencia. Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación.

1.4 Arquitectura de los Sistemas de Bases de Datos

Un sistema de gestión de bases de datos (SGBD o DBMS ‘Database Management System’) consiste en una colección de datos interrelacionados y un conjunto de programas que permiten a los usuarios acceder y modificar dichos datos. La colección de datos se denomina base de datos El primer objetivo de un SMBD es

proporcionar un entorno que sea tanto práctico como eficiente de usar en la recuperación y el almacenamiento de la información de la base de datos. Otro de los objetivos principales de un SMBD es proporcionar al usuario una visión abstracta de la información, es decir, el sistema oculta detalles como los relativos a la forma de almacenar y mantener los datos, de tal forma que para que el sistema sea útil la información ha de recuperarse de forma eficiente.

La búsqueda de la eficiencia conduce al diseño de estructuras complejas para usuarios sin conocimientos de computación, para lo cual esta complejidad ha de estar oculta. Para poder lograr lo anterior es necesario definir los distintos niveles de abstracción de una base de datos, lo que constituirá el marco necesario para identificar las diferentes funciones que han de cumplir estos sistemas.

El estándar ANSI/SPARC.

El objetivo principal de la arquitectura ANSI/SPARC es definir un SMBD con el máximo grado de independencia, separando las aplicaciones de usuario y la base de datos física. Para ello se utilizan tres niveles de abstracción conocidos como interno, conceptual y externo.

1. El **nivel interno** (también conocido con el nivel físico) es el más cercano al almacenamiento físico; es decir, es una representación a bajo nivel de la base de datos en la que se define la forma en la que los datos se almacenan físicamente en la máquina. Se definen características como los dispositivos en donde se almacenan los datos, el espacio que se reserva, las estrategias de acceso, la creación de ficheros de índices, etc. Es dependiente de la máquina en que se vaya a instalar la base de datos, del sistema operativo que exista, etc.

2. El **nivel conceptual** (también conocido como el nivel lógico) tiene un esquema conceptual, que describe la estructura de los datos que van a ser almacenados en la base de datos. El esquema conceptual esconde los detalles del almacenamiento físico y se concentra en describir entidades, tipos de datos, relaciones, operaciones de usuario y restricciones.

3. El **nivel externo** (también conocido como el nivel lógico de usuario) es el más próximo a los usuarios; es decir, el que tiene que ver con la forma en que los usuarios individuales ven los datos. Cada esquema externo describe la parte de la base de datos en la que está interesado un grupo de usuarios en particular y esconde el resto de la base de datos para esos usuarios. La información se manipula sin saber cómo está almacenada internamente (nivel interno) ni su organización (nivel conceptual).

El Nivel Externo

El nivel externo es el más cercano a los usuarios, es decir, es el que se ocupa de la forma en que los usuarios perciben los datos. El nivel externo es del usuario individual. Estos usuarios pueden ser o bien programadores de aplicaciones o usuarios finales con conocimientos muy variables de informática. El administrador de la base de datos es un caso especial (también debe interesarse por los demás niveles de la arquitectura).

Cada usuario dispone de un lenguaje:

En el caso del programador de aplicaciones, dicho lenguaje será o bien un lenguaje de programación convencional, o bien un lenguaje de cuarta generación (4GL) específico para el sistema en cuestión.

Para el usuario final será o bien un lenguaje de consulta, o algún lenguaje de aplicación especial, quizá manejado mediante formas o menús, adaptado a los requerimientos de ese usuario y apoyado por algún programa de aplicación en línea (cuya función es servir a un usuario final que tiene acceso a la base de datos desde una terminal en línea).

El aspecto importante de todos estos lenguajes es que deben incluir un sublenguaje de datos, es decir, un subconjunto del lenguaje total que se ocupe de manera específica de los objetos y operaciones de la base de datos. Se dice que el sublenguaje de datos (DSL ‘data sublanguage’) está embebido (o inmerso) dentro del lenguaje anfitrión correspondiente. Este último se encarga de varios aspectos no relacionados con la base de datos, como por ejemplo variables locales (temporales), operaciones de cálculo, lógica condicional, etc. Un sistema dado puede permitir el empleo de varios lenguajes anfitriones y varios sublenguajes de datos. Un sublenguaje de datos en particular cuyo uso es posible en casi todos los sistemas relacionales actuales es el lenguaje SQL.

En principio, cualquier sublenguaje de datos es en realidad una combinación de por lo menos dos lenguajes subordinados: un lenguaje de definición de datos (DDL ‘data definition language’), con el cual es posible definir o declarar los objetos de la base de datos, y un lenguaje de manipulación de datos (DML, ‘data manipulation language’) con el que es posible manipular o procesar dichos objetos.

Como ya se ha dicho, al usuario individual (en general), sólo le interesará una porción de la base de datos total; por añadidura, la forma como ese usuario percibe dicha porción casi siempre será un tanto abstracta comparada con el almacenamiento físico de los datos. El término ANSI/SPARC para la vista individual de un usuario es vista externa. Así, una vista externa es el contenido de la base de datos tal como lo percibe algún usuario determinado (es decir, para ese usuario la vista externa es la base de datos). Por ejemplo, un usuario del

departamento de personal podría contemplar la base de datos como un conjunto de ocurrencias de registros de departamento unido a un conjunto de ocurrencias de registros de proveedor y de parte vistas por los usuarios del departamento de compras).

Toda vista externa se define mediante un esquema externo, que consiste básicamente en definiciones de cada uno de los diversos tipos de registros externos en esa vista externa. El esquema externo se escribe con la porción DDL del sublenguaje de datos del usuario (por ello se le denomina a ese DDL en ocasiones como DDL externo). Por ejemplo, el tipo de registro externo de empleado puede definirse como un campo de número de empleado de seis caracteres unido a un campo de salario de cinco dígitos, etc. Además, debe haber una definición de la correspondencia entre el esquema externo y el esquema conceptual subyacente.

El Nivel Conceptual

El nivel conceptual es un nivel de mediación entre el nivel interno y externo. La vista conceptual es una representación de toda la información contenida en la base de datos, también (como en el caso de una vista externa) en una forma un tanto abstracta si se compara con el almacenamiento físico de los datos.

Además, puede ser muy diferente de la forma como percibe los datos cualquier usuario individual. A grandes rasgos, la vista conceptual debe ser un panorama de los datos “tal como son”, y no como por fuerza los perciben los usuarios debido a las limitaciones del lenguaje o el equipo específicos utilizados, por ejemplo.

La vista conceptual se compone de varias ocurrencias de varios tipos de registro conceptual. Por ejemplo, puede estar formada por un conjunto de ocurrencias de registros de departamento unido a un conjunto de ocurrencias de registro de empleado y a un conjunto de ocurrencias de registros de proveedor y a un conjunto de ocurrencias de registros de parte... Un registro conceptual no es por necesidad idéntico a un registro externo, por un lado, ni a un registro almacenado, por el otro.

La vista conceptual se define mediante un esquema conceptual, el cual incluye definiciones de cada uno de los tipos de registro conceptual. El esquema conceptual se escribe utilizando otro lenguaje de definición de datos, el DDL conceptual. Si ha de lograrse la independencia de los datos, esas definiciones en DDL conceptual no deberán implicar consideraciones de estructura de almacenamiento o de técnica de acceso. Si el esquema conceptual se hace en verdad independiente de los datos de esta manera, entonces los esquemas externos, definidos en términos del esquema conceptual, serán por fuerza también independientes de los datos.

Así pues, la vista conceptual es una vista del contenido total de la base de datos, y el esquema conceptual es una definición de esa vista. No obstante, sería engañoso sugerir que el esquema conceptual es sólo un conjunto de definiciones similar a las sencillas definiciones de registros encontradas por ejemplo en un programa en Cobol. Es de esperar que las definiciones en el esquema conceptual incluyan muchas características más, como son las verificaciones de seguridad y de integridad. Algunos expertos podrían llegar a sugerir que el objetivo primordial del esquema conceptual es describir la empresa en su totalidad (no sólo los datos en sí, sino también la forma como se utilizan: cómo fluyen de un punto a otro dentro de la empresa, qué se hace con ellos en cada punto, qué controles de auditoría o de otro tipo deben aplicarse en cada punto, etc.

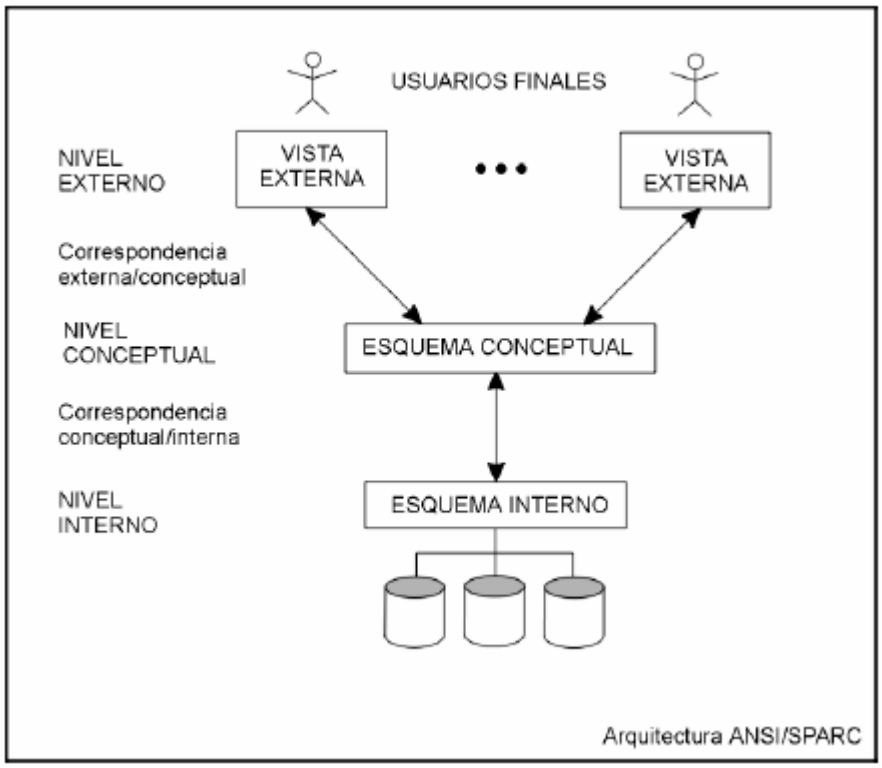
Debe hacerse hincapié en que en ningún sistema actual es posible mantener realmente un nivel conceptual que se aproxime siquiera a ese grado de complejidad; en casi todos los sistemas existentes el esquema conceptual no es mucho más que una simple unión de todos los esquemas externos individuales, con la posible adición de algunas verificaciones sencillas de integridad y seguridad. Con todo, parece evidente que los sistemas del futuro llegarán a mantener niveles conceptuales mucho más complejos.

El Nivel Interno

El tercer nivel de la arquitectura es el nivel interno. La vista interna es una representación de bajo nivel de toda la base de datos; se compone de varias ocurrencias de varios tipos de registro interno. Este último término es el que utiliza ANSI/SPARC para referirse a la construcción que hemos estado llamando registro almacenado. La vista interna, por tanto, todavía está a un paso del nivel físico, ya que no maneja registros físicos (llamados también páginas o bloques), ni otras consideraciones específicas de los dispositivos como son los tamaños de cilindros o de pistas.

La vista interna se define mediante el esquema interno, el cual no sólo define los diversos tipos de registros almacenados sino también especifica que índices hay, cómo se representan los campos almacenados, en qué secuencia física se encuentran los registros almacenados, etc. El esquema interno se escribe con otro lenguaje más de definición de datos, el DDL interno.

En algunas situaciones excepcionales podría permitirse a los programas de aplicación operar directamente en el nivel interno en vez de hacerlo en el nivel externo. Esta práctica no es recomendable ya que representa un riesgo para la seguridad (ya que pasan por alto las verificaciones de seguridad) y para la integridad (hace lo mismo), y el programa será en extremo dependiente de los datos; sin embargo, en ciertos casos puede ser la única forma de obtener la función o desempeño deseados, del mismo modo como el usuario de un lenguaje de programación de alto nivel puede verse obligado en ocasiones a descender al lenguaje ensamblador para satisfacer ciertos objetivos.



1.5 Definición del Sistema Manejador de Base de Datos (SMDB).

Los sistemas de bases de datos se diseñan para almacenar grandes volúmenes de información, la gestión de los datos implica entonces la definición de estructuras para el almacenamiento de la información y la provisión de mecanismos para la manipulación de éstos. Además deben proporcionar mecanismos de seguridad de los datos que protejan al sistema frente a caídas o a intentos de acceso de personas no autorizadas. Si los datos están compartidos por varios usuarios, el sistema debe asegurar la consistencia de los datos evitando posibles resultados anómalos.

Para plasmar los tres niveles en el enfoque o modelo de datos, es necesaria una aplicación que actúe de interfaz entre el usuario, los modelos y el sistema físico. Esta es la función que desempeñan los SMDB, y que pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario.

Las principales funciones que debe cumplir un SMDB son (ver figura 1.3):

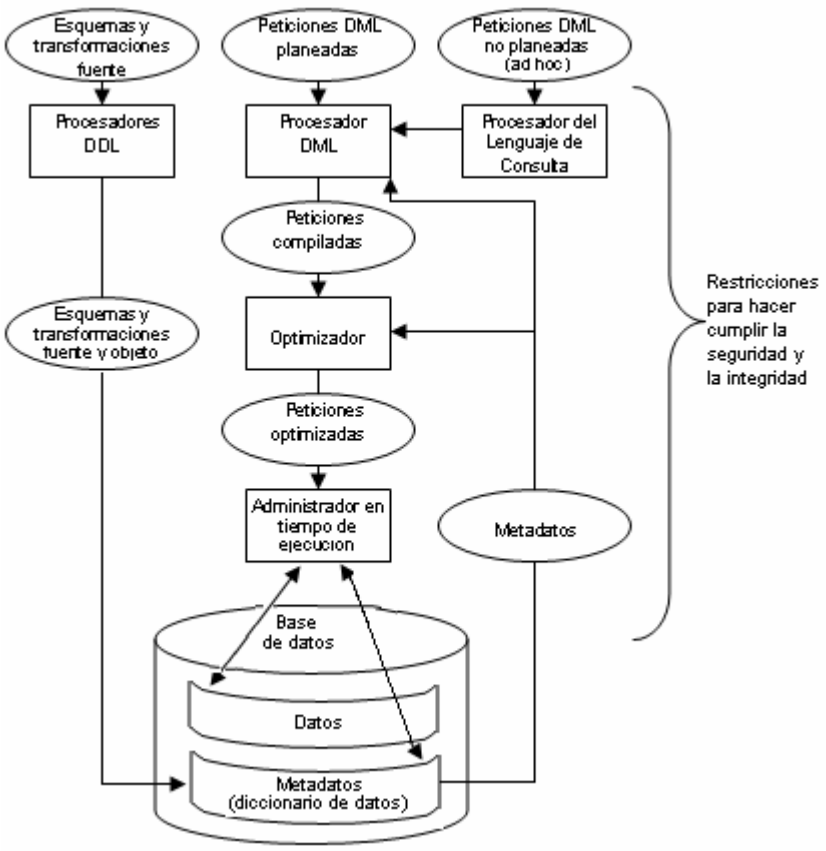


Figura 1.3 Funciones y componentes principales del SMDB

- *Definición de Datos:*

  El DBMS debe ser capaz de aceptar definiciones de datos (esquemas externos, el esquema conceptual, el esquema interno y todas las transformaciones respectivas) en la forma fuente y convertirlas a la forma objeto correspondiente. En otras palabras, el DBMS debe incluir entre sus componentes un procesador DDL, o compilador DDL, para cada uno de los diversos DDLs (lenguajes de definición de datos). El DBMS también debe "entender" las definiciones DDL.

  Lenguaje de definición de datos (DDL: *Data Definition Language*). Sencillo lenguaje artificial para definir y describir los objetos de la base de datos, su estructura, relaciones y restricciones. En la práctica puede consistir en un subconjunto de instrucciones de otro lenguaje informático. Aparte suele poseer dos subconjuntos de instrucciones:

- Lenguaje de definición del almacenamiento de los datos (DSDL: *Data Storage Definition Language*): permite especificar características físicas de la base de datos (volúmenes y archivos donde van a ser almacenados los datos, etc).

- Lenguaje de control de datos (DCL: Data Control Language): encargado del control y seguridad de los datos (privilegios y modos de acceso, etc).

- *Manipulación de Datos*

  El DBMS debe ser capaz de manejar peticiones para recuperar, actualizar o eliminar datos existentes en la base de datos o agregar nuevos datos a ésta. En otras palabras, el DBMS debe incluir un componente procesador DML o compilador DML para tratar con el DML (lenguaje de manipulación de datos).

  Lenguaje de manipulación de datos (DML: *Data Manipulation Language*). Lenguaje artificial de cierta complejidad que permite el manejo y procesamiento del contenido de la base de datos. En la práctica puede consistir en un subconjunto de instrucciones de otro lenguaje informático. Las aplicaciones que trabajan sobre la base de datos se programan en un lenguaje de programación (C, Cobol) insertando en el código fuente sentencias del DML. Al utilizar un DML se deben especificar los datos que serán afectados por las sentencias del lenguaje. Un DML puede tener o no procedimientos, según sea necesario especificar además cómo deben obtenerse esos datos. Los DML con procedimientos tienen sentencias de control de flujo como bucles o condicionales. Los DML sin procedimientos son conocidos también como declarativos.

- En general, las peticiones DML pueden ser “planeadas “ o “no planeadas”:

  a. Una petición **planeada** es aquella cuya necesidad fue prevista antes del momento de ejecutar la petición. Probablemente el DBA habrá afinado el diseño físico de la base de datos de tal forma que garantice un buen desempeño para las peticiones planeadas.

  b. En contraste, una petición **no planeada** es una consulta *ad hoc*; es decir, una petición para la que no se previó por adelantado su necesidad, sino que en vez de ello, surgió sin pensarlo. El diseño físico de la base de datos podría o no ser el adecuado para la petición específica en consideración.

- *Optimización y ejecución*

  Las peticiones DML, planeadas o no planeadas, deber ser procesadas por el componente **optimizador**, cuya finalidad es determinar una forma eficiente de implementar la petición. Las peticiones optimizadas se ejecutan entonces bajo el control del **administrador en tiempo de ejecución.**

- *Seguridad e integridad de los datos*

  El SMDB debe vigilar las peticiones del usuario y rechazar todo intento de violar las restricciones de seguridad y de integridad definidas por el DBA. Estas tareas pueden realizarse durante el tiempo de compilación, de ejecución o entre ambos.

- *Recuperación de datos y concurrencia*

  El SMDB –o más probablemente, algún otro componente de software relacionado, denominado comúnmente **administrador de transacciones** o **monitor de procesamiento de transacciones** (monitor PT)– debe imponer ciertos controles de recuperación y concurrencia.

- *Diccionario de datos*

  El SMDB debe proporcionar una función de **diccionario de datos**. Este diccionario puede ser visto como una base de datos por derecho propio (aunque una base de datos del sistema más que como una base de datos del usuario). El diccionario contiene "datos acerca de los datos" (en ocasiones llamados *metadatos* o *descriptores*); es decir, *definiciones* de otros objetos del sistema, en lugar de simples "datos en bruto". En particular, todos los diversos esquemas y transformaciones (externos, conceptuales, etcétera) y todas las diversas restricciones de seguridad y de integridad, serán almacenadas en el diccionario, tanto en forma fuente como objeto. Un diccionario extenso incluirá además mucha información adicional; mostrará por ejemplo qué programas utilizan qué partes de la base de datos, qué usuarios necesitan qué informes, etcétera. El diccionario podrá incluso –y de hecho, debería – estar integrado dentro de la base de datos que define, e incluir por lo tanto su propia definición.

- Rendimiento

  Sobra decir que el SMDB debe realizar todas las tareas antes identificadas de la manera más eficiente posible.

El uso real de un SMBD puede realizarse de forma única o combinada utilizando directamente el DDL y el DML o bien a través de una interfaz gráfica o basada en menús.

En un SMBD, los datos se pueden crear, borrar o cambiar en una base de datos integrada. El término integrada se refiere a la capacidad del SMBD de relacionar lógicamente un registro con otro. El usuario tiene acceso directo mediante instrucciones en el teclado. Un SMBD permite entonces:

1. **Independencia de los Datos:** La independencia de los datos es un objetivo primordial de los sistemas de bases de datos. Esta independencia puede definirse como la inmunidad de las aplicaciones ante los cambios en la estructura de almacenamiento y en la técnica de acceso, lo cual implica que las aplicaciones en cuestión no dependen de una estructura de almacenamiento o una técnica de acceso.

Todos los datos necesarios pueden ser almacenados en una base general. Si hay que hacer cualquier cambio a los datos pueden efectuarse sin necesidad de cambiar los programas que accedan datos. Esto es posible porque el SMBD proporciona dos aspectos de los datos. La visión física de una base de datos, se relaciona con la localización actual de los datos en el dispositivo de almacenamiento. La visión lógica representa los registros.

2. **Eliminación de la redundancia e incremento de la integridad de los datos:** Todos los datos relacionados se almacenan en un lugar, si un elemento de los datos debe ser cambiado sólo tiene que hacerse en un lugar.

3. **Datos integrados, a partir de otros archivos:** Un usuario puede recabar datos de cierto número de archivos de una base de datos y aplicar esos datos combinados, a reportes u otras aplicaciones, creando relaciones entre los registros. Realza la flexibilidad.

4. **Mayor seguridad, a través del manejo de acceso de datos:** La capacidad para negar el acceso a usuarios no autorizados, a datos restringidos, mejora enormemente la seguridad de los datos y pone a salvo la integridad.

5. **Normalización de reportes y consultas:** Un SMBD permite a un usuario realizar reportes normalizados. Esto permite que el usuario formule preguntas breves.

Los componentes de procesamiento de consultas incluyen:

- Compilador DML.
- Precompilador DML.
- Intérprete DDL.
- Motor de Evaluación de Consultas.

Los componentes de gestión de almacenamiento proporcionan la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y envío de consultas al sistema.

El gestor de almacenamiento incluye:

- Gestor de Transacciones.
- Gestor de Archivos.
- Gestor de Memoria Intermedia.

Además, se necesitan varias estructuras de datos como parte de la implementación física del sistema:

- **Archivos de Datos:** Almacenan la base de datos.
- **Diccionario de Datos:** Almacena metadatos (datos acerca de los datos).
- **Índices:** Proporcionan acceso rápido a elementos de datos que tienen valores particulares.
- **Datos estadísticos:** Almacenan información estadística sobre los datos en la base de datos. El procesador de consultas utiliza esta información para seleccionar las formas eficientes para ejecutar una consulta.

1.6 Administrador de Bases de Datos (DBA)

El Administrador de bases de datos (DBA: Database Administrator) es la persona o equipo de personas profesionales responsables del control y manejo del sistema de base de datos, generalmente tiene (n) experiencia en DBMS, diseño de bases de datos, sistemas operativos, comunicación de datos, hardware y programación.

Funciones del Administrador (DBA)

La primera tarea importante del administrador del servidor de bases de datos es resolver las diferencias entre varias funciones de la organización, con el fin de desarrollar una estructura conceptual y, más tarde, lógica del modelo de base de datos para la empresa.

Entre algunas de las tareas de DBA, se encuentran las siguientes:

- Definir el esquema conceptual

  Es trabajo del administrador de datos decidir exactamente qué información contendrá la base de datos; en otras palabras, identificar las entidades de interés para la empresa e identificar la información que hay que registrar acerca de dichas entidades. Por lo regular a este proceso se le conoce como diseño lógico –en ocasiones *conceptual* – de la base de datos.
  Una vez que el administrador decidió el contenido de la base de datos a un nivel abstracto, entonces el DBA creará el esquema conceptual correspondiente, utilizando el DLL conceptual. El DBMS usará la forma objeto (compilada) de ese esquema para responder a las peticiones de acceso. La forma fuente (sin compilar) actuará como documento de referencia para los usuarios del sistema.

- Definir el esquema interno

  El DBA también debe decidir la forma en que van a ser representados los datos en la base de datos almacenada. A este proceso se le conoce comúnmente como diseño físico de la base de datos. Una vez realizado el diseño físico, el DBA deberá crear la definición de la estructura de almacenamiento correspondiente (es decir, el esquema interno), utilizando el DDL interno. Además, también deberá definir la transformación conceptual/interna asociada.

- Establecer un enlace con los usuarios

  Es asunto del DBA enlazarse con los usuarios para asegurar que los datos necesarios estén disponibles y para escribir los esquemas externos necesarios, utilizando el DDL externo aplicable. También es necesario definir las transformaciones externas/conceptual correspondientes.

  Otros aspectos de la función de enlace con los usuarios incluyen la asesoría sobre el diseño de aplicaciones; una capacitación técnica; ayuda en la determinación y resolución de problemas, etc.

- Definir las restricciones de seguridad y de integridad

Las restricciones de seguridad y de integridad pueden ser vistas como parte del esquema conceptual.

- Definir las políticas de respaldos y recuperación

  Una vez que una empresa se compromete con un sistema de base de datos, se vuelve drásticamente dependiente del funcionamiento exitoso de dicho sistema. En el caso de que se produzca un daño en cualquier parte de la base de datos –ocasionado, por ejemplo, por un error humano o por una falla en el hardware o en el sistema operativo– resulta esencial poder reparar los datos afectados con el mínimo de demora y con tan poco efecto como sea posible sobre el resto del sistema. El DBA debe definir e implementar un esquema apropiado de control de daños que comprenda la descarga o "vaciado" periódico de la base de datos en un dispositivo de almacenamiento de respaldo y la recarga de la base de datos cuando sea necesario, a partir del vaciado más reciente.

- Supervisar el rendimiento y responder a los requerimientos cambiantes

  El DBA es el responsable de organizar el sistema de tal manera que se obtenga el rendimiento "ideal para la empresa" y de hacer los ajustes apropiados –es decir, **afinar** – conforme las necesidades cambien.

1.7 Métodos de Organización de las Bases de Datos

Una característica fundamental del enfoque de bases de datos es que proporciona cierto nivel de abstracción de los datos al ocultar detalles de almacenamiento que la mayoría de los usuarios no necesitan conocer.

Los modelos de datos son el principal instrumento para ofrecer dicha abstracción.

Un modelo de datos es un conjunto de conceptos que pueden servir para describir la estructura de la base de datos. Es decir, un modelo de datos no es más que una colección de herramientas conceptuales que se utilizan para describir los datos, las relaciones existentes entre ellos, la semántica asociada a los mismos y las restricciones de consistencia.

Los modelos de datos se dividen en 3 grupos:

A. Modelos Lógicos basados en Registros
B. Modelos Lógicos basados en Objetos
C. Modelos Físicos de Datos

Definición de Modelo de Datos

"Es un conjunto de conceptos, reglas y convenciones que nos permiten describir los datos del universo del discurso, constituyendo una herramienta que facilita la interpretación de nuestro universo del discurso y su representación en forma de datos en nuestro sistema de información."[3]

Modelo de datos y lenguaje de datos.− Son modelos en los que se basan los lenguajes de datos.

A. Modelos Lógicos Basados en Registros

Los tres modelos de datos más ampliamente aceptados son:

- Modelo Relacional
- Modelo de Red
- Modelo Jerárquico

Modelo Relacional

En este modelo se representan los datos y las relaciones entre éstos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

Existen dos formas de representar las relaciones entre las entidades en este modelo; pero para ello necesitamos definir que es una llave primaria: Es un atributo el cual definimos como atributo principal, es una forma única de identificar a una entidad. Las formas de representar las relaciones en este modelo son:

- ❖ Haciendo una tabla que contenga cada una de las llaves primarias de las entidades involucradas en la relación.

Este modelo se volverá a ver mas adelante.

Modelo de Red

Este modelo representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias.

[3] Elmasri, Ramez y Navathe, Shamkant, op. cit., p. 22.

Modelo Jerárquico

Es similar al modelo de red en cuanto a las relaciones y datos, ya que éstos se representan por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.

B. Modelos Lógicos Basados en Objetos.

Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

Modelo Entidad-Relación.

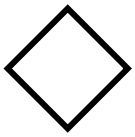
Denominado por sus siglas como: E-R; el modelo E-R se basa en una percepción del mundo real, la cual está formada por objetos básicos llamados entidades y las relaciones entre estos objetos así como las características de estos objetos llamados atributos.

- **Tangibles:** Son todos aquellos objetos físicos que podemos ver, tocar o sentir.

- **Intangibles:** Todos aquellos eventos u objetos conceptuales que no podemos ver, aún sabiendo que existen, por ejemplo: la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.

Para la representación de un modelo E-R gráficamente, se emplean símbolos, los cuales son: Diagramas Entidad-Relación (ER).

La estructura lógica general de una base de datos puede expresarse en forma gráfica por medio de un diagrama E-R, que se integra con los siguientes componentes:

Símbolo	Representa
	**Rectángulos:** Representan conjuntos de entidades.
	**Elipses:** Representan atributos.
	**Rombos:** Representan conjuntos de relaciones.
	Líneas: Enlazan atributos a entidades y entidades a relaciones.

Estructura

Entidades y conjuntos de entidades

"Entidad representa un objeto o concepto del mundo real".[4]

Las entidades pueden ser identificadas unívocamente en una situación a modelar. Así una entidad corresponde a una categorización de objetos de la situación a modelar. También un tipo entidad puede ser visto como una agregación de atributos. **Conjunto de entidades**: Es un grupo de entidades del mismo tipo.

Relaciones y conjuntos de relaciones.

Una relación es una asociación entre varias entidades. En términos de abstracción un tipo de relación corresponde a una agregación de uno o más tipos de entidades. Las relaciones no tienen existencia propia, ya que dependen de entidades. Un **conjunto de relaciones** es un grupo de relaciones del mismo tipo.

Se puede decir entonces que es posible tener más de un conjunto relación entre dos conjuntos entidades.

Un conjunto asociación puede ser n-ario, es decir, entre n conjuntos de entidades.

[4] Ibídem, p. 23.

Atributo, valor y conjunto de valores.

Un atributo puede ser definido formalmente como una función que transforma un conjunto de entidades o relaciones.

Se puede decir que los atributos son interpretaciones de conjunto de valores en un contexto de conjunto E-R.

Por lo que una entidad relación puede ser representada por un conjunto de atributos.

1.8 Bases de datos Relacionales

Una Base de datos Relacional, es un conjunto de datos interrelacionados con dependencia física y lógica, consistentes, íntegros y con redundancia (repetición de datos) controlada; almacenados de tal forma que pueden ser compartidos por usuarios y programas de aplicación sin conocer los detalles de la estructura del almacenamiento; los datos son almacenados en tablas de filas y columnas.

Las bases de datos relacionales se caracterizan fundamentalmente porque distribuyen la información en varias tablas, en lugar de condensarla en una sola logrando de esta manera reducir tiempo y trabajo, puesto que no se introducen los datos repetidamente.

Hay otras dos ventajas que no son menos importantes:

- Al no duplicar información se ahorra, no solo trabajo, sino también espacio, sin llegar nunca a la tacañería, es importante tener siempre presente la necesidad de ahorrar el máximo espacio posible.

- La facilidad para realizar el mantenimiento de los datos. Por ejemplo en el caso de que un cliente cambie de domicilio, sólo se tendrá que actualizar un registro de la tabla de los clientes.

Además las bases de datos relacionales son el tipo de bases de datos actualmente más difundido, debido a dos razones fundamentales:

1. ofrecen sistemas simples y eficaces para representar y manipular los datos y
2. se basan en un modelo, el relacional, con sólidas bases teóricas

El modelo relacional fue propuesto por E.F. Codd en un ya famoso artículo de 1970. Gracias a su coherencia y facilidad de uso, el modelo se ha convertido en el

más usado para la producción de DBMS (Sistemas Manejadores de Bases de Datos).

Las bases de datos relacionales están constituidas por una tabla o más tablas que contienen la información ordenada de una forma organizada y cumplen las siguientes leyes básicas:

- Generalmente, contendrán muchas tablas.
- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no están determinados.
- Para cada campo existe un conjunto de valores posible.

El modelo relacional representa los datos y las relaciones entre ellos, mediante una colección de tablas, cada una de las cuales tiene cierto número de columnas con nombres únicos y filas que representan cada ocurrencia de la tabla.

1.8.1 Elementos Básicos

Tablas

Una tabla es conocida como una relación, es una abstracción del almacenamiento físico de los datos y tiene las siguientes propiedades:

- Cada columna contiene valores relativos al mismo atributo y cada valor de una columna de la tabla debe ser simple, es decir un solo valor.

- Cada columna tiene un nombre distinto (nombre del atributo), y el orden de las columnas no es importante.

- Cada renglón es distinto, es decir un renglón no puede duplicarse en otro para un grupo de columnas seleccionadas como llave.

- Cada atributo no llave debe depender de la llave de la relación y no de ningún otro o llave.

Tupla

Es un conjunto de valores que componen un renglón de la relación. Es equivalente a la instancia de un registro.

Grado de una Tupla

Número de atributos que tiene una tupla (n de una n-tupla).

Dominio

Conjunto de todos los valores posibles para un atributo.

Cardinalidad

Es el número de tuplas de una relación (filas de la tabla).

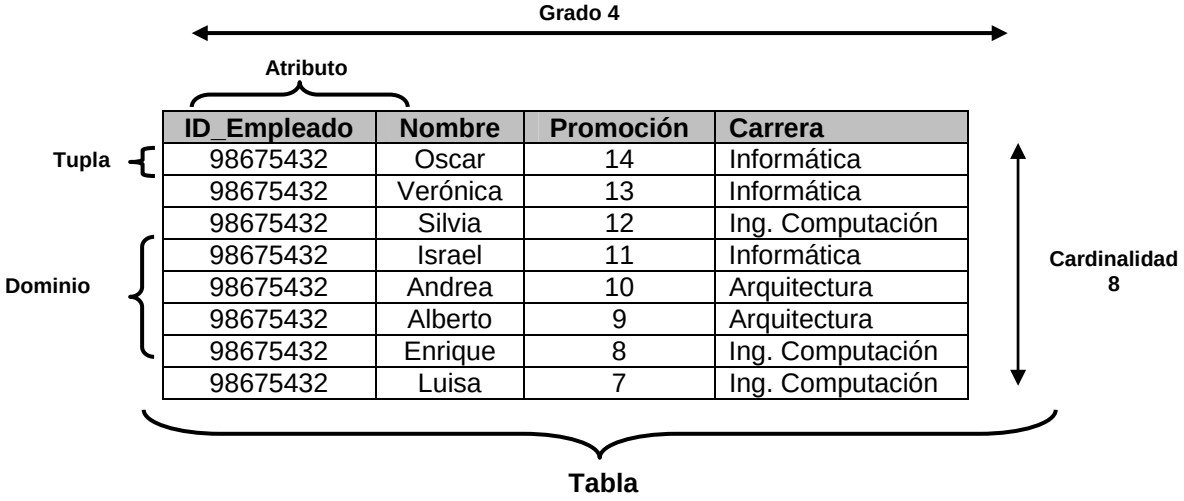
Atributos

Un atributo de una relación o de una tabla corresponde a una columna de la tabla. Los atributos están desordenados y se referencian por nombres y no por la posición que ocupan. Esto significa que no se puede, por ejemplo, hacer referencia al tercer atributo de una relación. Todos los valores de los atributos son atómicos y una relación que satisfaga esta condición se llama relación normalizada. Un atributo extrae sus valores desde un dominio simple.

Formalmente, un atributo es una función que se define entre un Dominio y un determinado tipo de Entidad de la base de datos. Dicha función asocia una ocurrencia de Tipo de Entidad con un determinado elemento del dominio.

Estructura de datos relacional

TABLA DE EMPLEADOS



ID_Empleado	Nombre	Promoción	Carrera
98675432	Oscar	14	Informática
98675432	Verónica	13	Informática
98675432	Silvia	12	Ing. Computación
98675432	Israel	11	Informática
98675432	Andrea	10	Arquitectura
98675432	Alberto	9	Arquitectura
98675432	Enrique	8	Ing. Computación
98675432	Luisa	7	Ing. Computación

Término Relacional	Equivalentes
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Numero de filas
Atributo	Columna o campo
Grado	Número de columnas
Dominio	Fondo de valores legales

1.9 Relaciones

En las bases de datos relacionales, con diferencia a las simples, la información se distribuye en varias tablas que están interconectadas entre sí. La manera en la que se interconectan las distintas tablas de una base de datos relacional da lugar a que el comportamiento del conjunto sea distinto en cada caso.

1.9.1 Propiedades de las relaciones

Las relaciones poseen ciertas propiedades, todas ellas consecuencias inmediatas de la definición de relación, y todas ellas muy importantes. Las propiedades son como sigue. Dentro de cualquier relación dada:

- No existen tuplas duplicadas;
- Las tuplas están en desorden, de arriba hacia abajo;
- Los atributos están en desorden, de izquierda a derecha;
- Cada tupla contiene un valor para cada atributo.

1.9.2 Tipos de Relaciones

Relación de uno a muchos

La relación de uno a muchos es la más común, en una relación de éste tipo, cada registro de la tabla A (que se denomina tabla principal, tabla primaria o tabla padre) puede tener más de un registro en la tabla B (más de una correspondencia); pero cada registro de la tabla B (que se denomina tabla relacionada, tabla secundaria o tabla hijo) sólo puede tener un registro de la tabla A.

Ejemplo:

Pensemos en una base de datos que almacena información sobre cinematografía. La primera tabla (tabla A) contendrá los datos de los directores de cine y la segunda tabla (tabla B) contendrá datos sobre las películas.

Suponiendo que no hay películas codirigidas, se establece una relación de uno (director) a muchas (películas), o puesto que cada director produce varias películas, pero cada película sólo tiene un director.

Relación de muchos a muchos

En una relación de este tipo, cada registro de la tabla A puede tener más de un registro en la tabla B y cada registro de la tabla B puede tener más de un registro en la tabla A.

Ejemplo:

Siguiendo el ejemplo de cinematografía y considerando la tabla de películas y una de actores y actrices.

Cada registro de la tabla de películas puede tener más de una correspondencia en la tabla de actores lo cual significaría, como es normal, que en una película trabaja más de un actor. A su vez cada actor puede tener más de una correspondencia con la tabla de películas.

Relación uno a uno

En una relación del tipo uno a uno, cada registro de la tabla A solamente se relaciona como máximo con un registro de la tabla B y viceversa, cada registro de la tabla B solamente tiene como máximo un registro de la tabla A. Es importante señalar que este tipo de relación es el menos frecuente.

Ejemplo:

Se tiene una base de datos que contiene la información de los empleados de una empresa y que está compuesta por dos tablas, la primera contiene la información pública del empleado (nombre, apellidos, etc.) y la segunda la información privada (nómina).

Se establece una relación de uno a uno, puesto que cada empleado sólo aparece en cada nómina una vez y cada registro de la nómina pertenece a un sólo empleado.

1.10 Llaves

Cada fila de una relación (tabla) debe ser identificada por un valor único al que llamamos ***llave***. A veces la llave puede ser establecida por una única columna y en otros casos la llave estará compuesta por varias.

La llave debe cumplir estos requisitos:

- **Identificación unívoca**. En cada fila de la tabla la llave debe identificarla de forma unívoca.
- **No redundancia**. No se debe descartar ningún atributo de la llave para identificar la fila.

1.10.1 Tipos de Llaves

Superllave

Es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades.

Llave Primaria

Atributo que identifica de manera única a un registro. Es decir, no debe haber tuplas que tengan el mismo valor en todos los valores de la llave, por lo tanto con sólo conocer el valor de K para una tupla será suficiente para identificarlo de manera única.

Una clave o llave primaria es indicada gráficamente en el modelo E-R con una línea debajo del nombre del atributo.

Una clave (primaria, candidata y superclave) es una propiedad del conjunto de entidades más que de las entidades individuales. Cualesquiera dos entidades en el conjunto no pueden tener el mismo valor en sus atributos clave al mismo tiempo. La designación de una clave representa una ligadura en el desarrollo del mundo real que se modela.

Llave Candidata

Atributo o conjunto de atributos que podrían servir como llaves primarias.

Puede haber varias llaves candidatas para distinguir una misma entidad. Se elegirá como llave candidata aquel atributo que posea un dominio en el que se tenga valores únicos. Si esto no es posible, entonces usaremos como llave

candidata la combinación de varios atributos, de manera que esta combinación sí sea única.

Llave secundaria

Todas aquellas llaves candidatas que no se eligieron como llave primaria. Son llaves que tienen todas las características para ser primarias, pero que por alguna razón no fueron consideradas como tales, ya que hubo otra (s) que cumplió mejor con este objetivo.

Llave extranjera o foránea

Es la llave primaria que es la llave primaria de otra entidad. Las llaves foráneas o extranjeras son la materialización de las asociaciones entre las entidades, son llaves que son compartidas por dos tablas para lograr una relación entre ellas.

1.11 Reglas de Integridad

Una vez definida la estructura de datos del modelo relacional, pasamos a estudiar las reglas de integridad que los datos almacenados en dicha estructura deben cumplir para garantizar que son correctos.

Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. A este tipo de restricciones se les denomina *restricciones de dominios*. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados o instancias (las reglas se deben cumplir todo el tiempo). Estas reglas son la *regla de integridad de entidades* y la *regla de integridad referencial*. Antes de definirlas, es preciso conocer el concepto de *nulo*.

1.11.1 Nulos

Cuando en una tupla un atributo es desconocido, se dice que es *nulo*. Un nulo no representa el valor cero ni la cadena vacía, éstos son valores que tienen significado. El nulo implica ausencia de información, bien porque al insertar la tupla se desconocía el valor del atributo, o bien porque para dicha tupla el atributo no tiene sentido.

Ya que los nulos no son valores, deben tratarse de modo diferente, lo que causa problemas de implementación. De hecho, no todos los SGBD relacionales soportan los nulos.

1.11.2 Regla de integridad de entidades

La primera regla de integridad se aplica a las claves primarias de las relaciones base: *ninguno de los atributos que componen la clave primaria puede ser nulo.*

Por definición, una clave primaria es un identificador irreducible que se utiliza para identificar de modo único las tuplas. Que es irreducible significa que ningún subconjunto de la clave primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se contradice la irreducibilidad.

Nótese que esta regla sólo se aplica a las relaciones base y a las claves primarias, no a las claves alternativas.

1.11.3 Regla de integridad referencial

La segunda regla de integridad se aplica a las claves ajenas: *si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos*.

La regla de integridad referencial se enmarca en términos de estados de la base de datos: indica lo que es un estado ilegal, pero no dice cómo puede evitarse. La cuestión es ¿qué hacer si estando en un estado legal, llega una petición para realizar una operación que conduce a un estado ilegal? Existen dos opciones: *rechazar* la operación, o bien *aceptar* la operación y realizar operaciones adicionales compensatorias que conduzcan a un estado legal.

Por lo tanto, para cada clave ajena de la base de datos habrá que contestar a tres preguntas:

- *Regla de los nulos:* ¿Tiene sentido que la clave ajena acepte nulos?
- *Regla de borrado:* ¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?
    - *Restringir:* no se permite borrar la tupla referenciada.
    - *Propagar:* se borra la tupla referenciada y se propaga el borrado a las tuplas que la referencian mediante la clave ajena.
    - *Anular:* se borra la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).

- *Regla de modificación:* ¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?
    - *Restringir:* no se permite modificar el valor de la clave primaria de la tupla referenciada.
    - *Propagar:* se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que la referencian mediante la clave ajena.
    - *Anular:* se modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).

1.11.4 Reglas de negocio

Además de las dos reglas de integridad anteriores, los usuarios o los administradores de la base de datos pueden imponer ciertas restricciones específicas sobre los datos, denominadas *reglas de negocio*.

Por ejemplo, si en una oficina de la empresa inmobiliaria sólo puede haber hasta veinte empleados, el SGBD debe dar la posibilidad al usuario de definir una regla al respecto y debe hacerla respetar. En este caso, no debería permitir dar de alta un empleado en una oficina que ya tiene los veinte permitidos.

Hoy en día aún existen SGBD relacionales que no permiten definir este tipo de restricciones ni las hacen respetar.

1.12 Reglas de Codd

Para que una base de datos sea considerada verdaderamente relacional Ted Codd, en su artículo de 1985 en Computerworld, presentó doce reglas que una base de datos debe cumplir para considerarse enteramente relacional. Desde entonces se han convertido en una definición semioficial de una base de datos relacional. Las reglas se derivan del trabajo teórico de Codd sobre el modelo relacional, y representan realmente más un objetivo ideal que una definición de una base de datos relacional. Ningún DBMS relacional actualmente disponible satisface totalmente las reglas de Codd.

Regla 0

Cualquier DBMS que proclame ser relacional deberá manejar, completamente las bases de datos por medio de sus capacidades relacionales.

Regla 1 (The information rule)

La regla de información. Toda la información de una base de datos relacional está representada explícitamente a nivel lógico y exactamente de un modo, mediante valores en tablas.

Regla 2 (Guaranteed access rule)

Regla de acceso garantizado. Todos y cada uno de los datos (valor atómico) de una base de datos relacional se garantiza que sean lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave primaria y nombre de columna.

Regla 3 (Systematic treatement of null values)

Tratamiento sistemático de valores nulos. Los valores nulos (distinto de la cadena de caracteres vacías o de una cada de caracteres en blanco y distinta del cero o de cualquier otro número) se soportan en los DBMS completamente relacionales para representar la falta de información y la información inaplicable de un modo sistemático e independiente del tipo de datos.

Regla 4 (Dynamic on line catalog based don the relational model)

Catálogo en línea dinámico basado en el modelo relacional. La descripción de la base de datos se representa a nivel lógico, así como los datos ordinarios, de modo que los usuarios autorizados puedan aplicar a su interrogación el mismo lenguaje relacional que aplican a los datos regulares.

Regla 5 (Comprehensive data sublanguage)

Regla de sublenguaje completo de datos. Un sistema relacional puede soportar varios lenguajes y varios modos de uso de Terminal (por ejemplo, el modo de rellenar con blancos). Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante alguna sintaxis bien definida, como cadenas de caracteres, y que sea completa en cuanto al soporte de todos los puntos siguientes:

- Definición de datos.
- Definición de vista.
- Manipulación de datos (interactiva y por programa).
- Restricciones de integridad (manejo).
- Autorización.
- Fronteras de transacciones (inicio y fin de una transacción).

Regla 6 (View updating rule)

Regla de actualización de vista. Todas las vistas que sean teóricas actualizables son también actualizables por el sistema.

Regla 7 (High level Insert, update and delete)

Inserción, actualización y supresión de alto nivel. La capacidad de manejar una relación de base de datos o una relación derivada como un único operando se aplica no solamente a la recuperación de datos, sino también a la inserción, actualización y supresión de los datos.

Regla 8 (Physical data independence)

Independencia física de los datos. Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cualquiera que sean los cambios efectuados ya sea a las representaciones de almacenamiento o a los métodos de acceso.

Regla 9 (Logical data independence)

Independencia lógica de los datos. Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cuando se efectúen sobre las tablas de base cambios preservadores de la información de cualquier tipo que teóricamente permita alteraciones.

Regla 10 (Integrity independence)

Independencia de integridad. Las restricciones de integridad específicas para una base de datos relacional particular deben ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación.

Regla 11 (Distribution independence)

Independencia de distribución. Un DBMS relacional tiene independencia de distribución.

Regla 12 (Nonsubversion rule)

Regla de no subversión. Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro cada vez), ese bajo nivel no puede ser utilizado para subvertir o suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).

1.13 Normalización

La normalización es el proceso de simplificar la relación entre los campos de un registro. Por medio de la normalización un conjunto de datos en un registro se reemplaza por varios registros que son más simples y predecibles y, por lo tanto, más manejables.

La normalización se lleva a cabo por cuatro razones:

1. Estructurar los datos de forma que se puedan representar las relaciones pertinentes entre los datos.

2. Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultas y reportes.

3. Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.

4. Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan nuevas aplicaciones.

En términos más sencillos la normalización trata de simplificar el diseño de una base de datos, esto a través de la búsqueda de la mejor estructuración que pueda utilizarse con las entidades involucradas en ella.

Pasos de la Normalización:

- Descomponer todos los grupos de datos en registros bidimensionales.

- Eliminar todas las relaciones en la que los datos no dependan completamente de la llave primaria del registro.

- Eliminar todas las relaciones que contengan dependencias transitivas.

- La teoría de normalización tiene como fundamento el concepto de formas normales; se dice que una relación está en una determinada forma normal si satisface un conjunto de restricciones.

Las reglas de normalización fueron definidas por Codd (1970). El punto fundamental en el proceso es que dada una relación que posee ciertas propiedades indeseables, las reglas de normalización permiten reconocer tales casos y muestran cómo esa relación puede ser descompuesta en una forma más deseable.

La Normalización se define como un método de diseño ascendente basado en el concepto de formas normales, así se dice que una relación está en una forma normal particular si cumple con un conjunto de restricciones.

A medida que se incrementan las formas normales se incrementa el número de restricciones que debe cumplir esa relación.[5] La normalización se encarga de obtener los datos agrupados en distintas tablas siguiendo una serie de pasos, de tal manera que los datos obtenidos tienen una estructura óptima para su implementación, gestión y explotación desde distintas aplicaciones futuras. Una de las ventajas principales que se obtiene al realizar la normalización es que la información no estará duplicada innecesariamente dentro de las estructuras: habrá mínima redundancia. Al modelar una base de datos, desearemos evitar puntos que crean confusión, duplicación de la información y por ende, un mal funcionamiento y exploración de la información. Entre las propiedades indeseables en un diseño de bases de datos tenemos:

- Redundancia en la información.
- Incapacidad de representar cierta información.
- Registrar información que no sea identificable.

1.13.1 Formas de Normalización

Son las técnicas para prevenir las anomalías en las tablas. Dependiendo de su estructura, una tabla puede estar en primera forma normal, segunda forma normal o en cualquier otra. El Objetivo es obtener la forma normal mayor posible. La teoría de normalización consiste en obtener esquemas relacionales que cumplan determinadas condiciones y se centra en las determinadas formas normales.

Se dice que un esquema de relación está en una determinada forma normal si satisface un conjunto determinado de restricciones.

En la figura 1.4 se muestra la relación entre las formas normales.

[5] Hansen, Gary y Hansen, James, ***Diseño y Administración de Base de Datos***, Tercera Edición, Prentice-Hall Hispanoamericana, México 2000, p. 12.

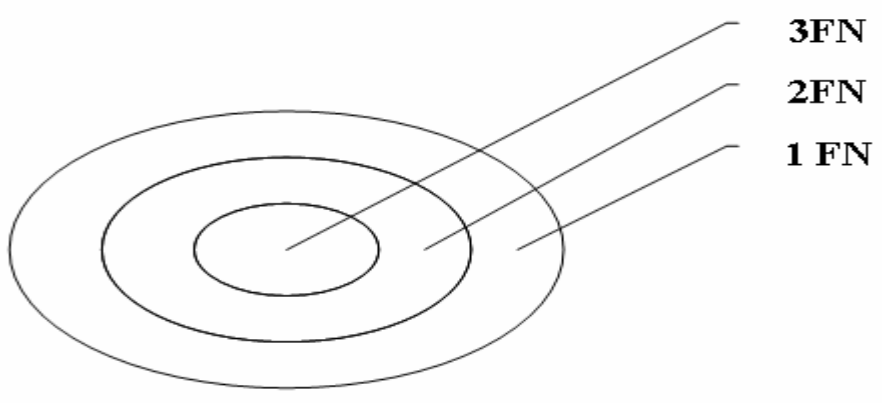


Figura 1.4 Relación entre las formas normales

Primera Forma Normal

Abreviada como 1FN, una relación R se encuentra en 1FN si y solo sí por cada renglón columna contiene valores atómicos, se considera que una relación se encuentra en la primera forma normal cuando cumple lo siguiente:

- Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un sólo valor por cada celda.

- Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.

- Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.

- Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

Por lo general la mayoría de las relaciones cumplen con estas características, así que podemos decir que la mayoría de las relaciones se encuentran en la primera forma normal.

Una relación está en primera forma normal (1FN) si y sólo si todos los dominios son atómicos. Un dominio es atómico si los elementos del dominio son indivisibles. Es decir, no tenemos grupos de repetición o un conjunto de valores asociados repetidos asociados a una misma tupla.

Segunda Forma Normal

Abreviada como 2FN, Para definir la segunda forma normal requerimos saber qué es una dependencia funcional: Consiste en edificar qué atributos dependen de otro(s) atributo(s). Como se ve en la Figura 1.5

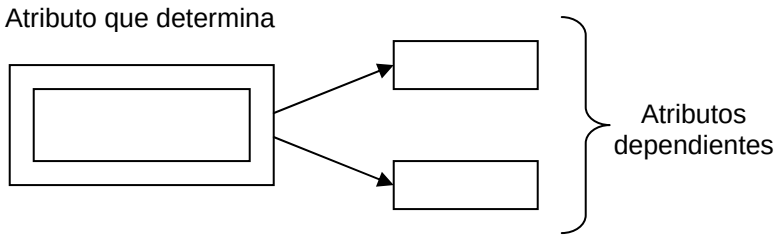


Figura 1.5 Segunda Forma Normal

Una relación R está en 2FN sí y sólo sí está en 1FN y los atributos no primos dependen funcionalmente de la llave primaria.

Una relación se encuentra en segunda forma normal, cuando cumple con las reglas de la primera forma normal y todos sus atributos que no son claves (llaves) dependen por completo de la clave. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, está en segunda forma normal. La segunda forma normal se representa por dependencias funcionales.

Tercera Forma Normal

Abreviada como 3FN. Una relación R está en 3FN sí y sólo sí esta en 2FN y todos sus atributos no primos dependen no transitivamente de la llave primaria. Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación está en tercera forma normal si está en segunda forma normal y no existen dependencias transitivas entre los atributos, nos referimos a dependencias transitivas cuando existe más de una forma de llegar a referencias a un atributo de una relación.

CAPITULO 2. PROTOCOLO DE AUDITORÍA EN BASES DE DATOS

Introducción

Para muchas organizaciones, la información y la tecnología que la soporta, representan los activos más valiosos de la empresa. Ya que la supervivencia de la misma puede depender de una correcta administración de la seguridad que garantice la confidencialidad, integridad y disponibilidad de la información. Esta situación hace que las empresas estén invirtiendo una parte de su presupuesto en la seguridad y protección de su información.

En la actualidad, cualquier aplicación independientemente del lenguaje en el que esté programado, consulta, modifica e introduce nuevos datos en una base de datos y por tal motivo es el punto donde confluyen todas las aplicaciones. Por esta razón las bases de datos se han convertido en el principal objetivo para el robo o alteración de su contenido.

Dicha información frecuentemente se encuentra almacenada en sistemas conocidos como Sistemas Manejadores de Bases de Datos (DBMS). Dada su importancia en las organizaciones, los DBMS deben proporcionar información veraz y exacta de los datos que ellos almacenan, por lo que es vital mantener un estricto control de las actividades que se llevan a cabo en estos sistemas.

En los ambientes de bases de datos, uno de los objetivos básicos de la administración de bases de datos consiste en la protección de la información que contiene. Cualquier administrador de base de datos debe conocer qué procedimientos utilizar para controlar los accesos no autorizados, los accesos de usuarios a información para la cual no tienen privilegios, así como el borrado o modificación de la información privilegiada.

Por esta razón la protección de bases de datos es un aspecto serio que se debe considerar explícitamente, no como aspecto aislado, pero sí como un elemento presente en todas las etapas del ciclo de vida de la construcción de la base de datos.

2.1 Definición de Seguridad

Una de las definiciones más generales de seguridad es la siguiente:

"Un sistema de cómputo es seguro si se puede confiar en que él y su *software* se comportarán como se espera que lo hagan, y que la información almacenada en él se mantendrá inalterada y accesible durante tanto tiempo como su dueño lo desee."[1]

Esta definición reconoce la amplitud y subjetividad del concepto. De acuerdo a la idea popular de seguridad en cómputo, su único objetivo es impedir que la información caiga en manos no autorizadas, es decir, la confidencialidad de los datos. Aunque este aspecto de la seguridad es muy importante, no lo es todo. Se considera la existencia de tres tipos principales de seguridad en cómputo[2]: confidencialidad, integridad y disponibilidad.

2.2 Seguridad en Base de Datos

La seguridad de la información en las bases de datos incluye cuatro principales aspectos: integridad, autenticidad, confidencialidad y disponibilidad.

Seguridad significa el prevenir y/o detectar la difamación de información. En general seguridad se refiere a la protección de los datos en diferentes ambientes, tanto en ambientes militares como en ambientes comerciales.

Otro término importante en el estudio de la seguridad de la información es el de la privacidad de los datos que se refiere a la información individual de cada individuo, grupo o institución para determinar cuándo o qué información le concierne para su propio propósito, y ésta a su vez puede ser almacenada o liberada para otras personas o entidades.

Privacidad se refiere a los datos de las personas que están protegidos por las leyes o reglas dependiendo del país donde se encuentre.

En los ambientes comerciales se encuentra información secreta estrictamente, acompañada por sistemas de seguridad (llamados policías de seguridad) que aseguran que la información denominada como crítica por la organización tendrá una alta seguridad en donde los empleados no tendrán acceso a información que no les corresponde.

- La información en un sistema informático reside, fundamentalmente, en bases de datos cuya seguridad es, por consiguiente, de vital importancia.

[1] O'Reilly & Associates. Practical Unix Security. Pág. 4
[2] O'Reilly & Associates. Practical Unix Security. Pág. 9

- Uno de los filtros de seguridad de la base de datos lo constituye, evidentemente el sistema operativo.

- No obstante, la seguridad que debe proporcionar la base de datos tiene algunas características diferenciadas:

    - ❖ Hay muchos más objetos a proteger.
    - ❖ El promedio de tiempo de vida de los objetos es mayor.
    - ❖ La granularidad del control es mayor.
    - ❖ Los objetos son estructuras lógicas complejas.
    - ❖ La seguridad está relacionada con la semántica de los datos, no con sus características físicas,

Dada la complejidad de los problemas anteriores, es el propio Sistema de Gestión de Bases de Datos (SGBD) el que proporciona la seguridad de éstas.

- Un SGBD debe mantener los cuatro criterios básicos:

    - ❖ Confidencialidad.
    - ❖ Integridad.
    - ❖ Disponibilidad.
    - ❖ Autenticidad

La seguridad en bases de datos se implementa mediante mecanismos de:

- ❖ Identificación y autentificación.
- ❖ Control de acceso a los objetos (datos y recursos).
- ❖ Registro de Auditoría.
- ❖ Protección criptográfica de alguno de los datos.

- Las posibles vulnerabilidades en la bases de datos son:

    - ❖ Los Ataques.

- La prevención frente a los ataques pasa por mecanismos de identificación, autentificación y control de acceso.

2.3 Objetivos de la seguridad de la información

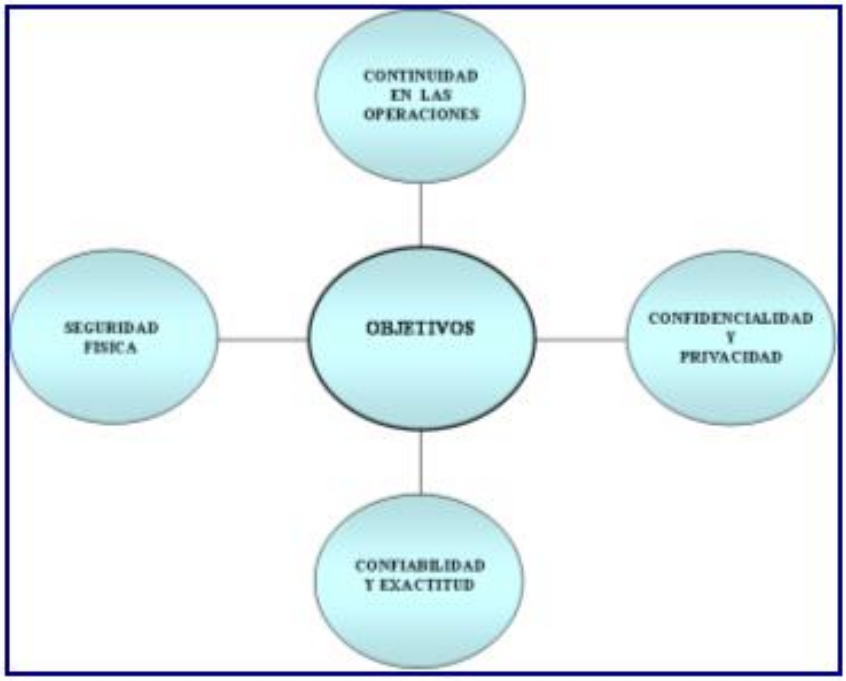
Los objetivos de la Seguridad de la información están orientados a proteger la organización contra amenazas que atenten contra:

1. La continuidad de las operaciones.

2. La confidencialidad y privacidad de la información manejada.

3. La confiabilidad y exactitud el sistema

4. La seguridad física.



OBJETIVOS DE LA SEGURIDAD

En relación a la seguridad misma de la información, estas medidas han de tenerse en cuenta para evitar la pérdida o modificación de los datos, información o software inclusive.

La seguridad de la información tiene dos aspectos. El primero consiste en negar el acceso a los datos a aquellas personas que no tengan derecho a ellos, al cual también se le puede llamar protección de la privacidad, si se trata de datos personales, y mantenimiento de la seguridad en el caso de datos institucionales.

Un segundo aspecto de la protección es garantizar el acceso a todos los datos importantes a las personas que ejercen adecuadamente su privilegio de acceso, las cuales tienen la responsabilidad de proteger los datos que se les ha confiado.

En general, la protección de los datos requiere ejercer un control sobre la lectura, escritura y empleo de esa información. Para obtener mayor eficiencia en la protección se debe tener siempre presente la protección de los datos, el mantenimiento de la privacidad y la seguridad del secreto.

El secreto se logra cuando no existe acceso a todos los datos sin autorización. La privacidad adecuada puede lograrse cuando los datos que puedan obtenerse no pueden enlazarse a individuos específicos o no pueden utilizarse para imputar hechos acerca de ellos.

Por otro lado, es importante incorporar dispositivos de seguridad durante el diseño del sistema en vez de añadirlos después. Los diseñadores de sistemas deben entender que las medidas de seguridad han llegado a ser criterios de diseño tan importantes como otras posibilidades funcionales, así como el incremento de costos que significa agregar funciones, después de desarrollado un Sistema de Información.

2.4 Importancia de la seguridad de la información

Como ya se ha mencionado la seguridad de la información tiene como principal función preservar, las características de confidencialidad, integridad y disponibilidad.

Implementando un conjunto adecuado de controles, que abarquen políticas, prácticas, procedimientos, estructuras organizacionales y funciones de software.

De aquí su importancia ya que debido a que la confidencialidad, integridad y disponibilidad de la información pueden ser esenciales para mantener la ventaja competitiva, la rentabilidad, el cumplimiento de las leyes y la imagen comercial de la organización. Además considera que, las organizaciones y sus redes y sistemas de información, se enfrentan en forma creciente con amenazas relativas a la seguridad, de diversos orígenes, incluyendo el fraude asistido por computadora, espionaje, sabotaje, vandalismo, incendio o inundación. Daños tales como los ataques mediante virus informáticos, "hacking" y denegación de servicio se han vuelto más comunes, ambiciosos y crecientemente sofisticados. Además, hoy en día, la dependencia de las organizaciones respecto de los sistemas y servicios de información denota que ellas son más vulnerables a las amenazas concernientes a seguridad. La interconexión de las redes públicas y privadas y el uso compartido de los recursos de información incrementa la dificultad de lograr el control de los accesos. También, la tendencia hacia el procesamiento distribuido ha debilitado la eficacia del control técnico centralizado.
Muchos sistemas de información no han sido diseñados para ser seguros. La seguridad que puede lograrse por medios técnicos es limitada y debe ser respaldada por una gestión y procedimientos adecuados. La identificación de los controles que deben implementarse requiere una cuidadosa planificación y

atención a todos los detalles. La administración de la seguridad de la información, exige, como mínimo, la participación de todos los empleados de la organización.

2.5 Servicios de Seguridad

Los servicios de seguridad de la información son los servicios de nivel básico que son utilizados para combatir los ataques definidos en el siguiente subtema. Cada uno de los cuatro servicios de seguridad combate ataques específicos, es importante que no sean confundidos con mecanismos de seguridad, ya que dichos mecanismos son la implementación de los servicios de seguridad.

Confidencialidad

La información debe estar disponible solamente para aquellos usuarios autorizados a usarla. Es prevenir, detectar, impedir el descubrimiento de información. En general la Confidencialidad se refiere a la protección de datos implicados en entornos altamente protegidos, como entornos militares, comerciales, etc. Privacidad se refiere a información sobre individuos. En la mayoría de los países la Privacidad está protegida por las leyes.

Integridad

La información no se puede falsear. Los datos recibidos (o recuperados) son los mismos que fueron enviados (o almacenados), etc.

Es prevenir, detectar, impedir la modificación inadecuada de información. Por ejemplo en un entorno militar, el mando responsable de un misil no debe ser modificado inadecuadamente. En un entorno comercial, la integridad de los datos es especialmente relevante, puesto que el éxito de una organización depende de lo correctas que son las operaciones que se llevan a cabo y la coherencia en los datos.

Tenemos los siguientes tipos de integridad:

a) Integridad semántica: Respeto en todo momento de las reglas de integridad definidas en la base de datos.

b) Integridad Operacional: Garantizar la consistencia de la base de datos con respecto al uso concurrente de la misma.

Disponibilidad

Es quién puede acceder a la información y cuándo hacerlo. La falta de accesibilidad produce una denegación de servicio, que es uno de los ataques más frecuentes en Internet.

La disponibilidad es prevenir, detectar, impedir la denegación inadecuada del acceso a servicios ofrecidos por el sistema.

El servicio de disponibilidad mantiene la utilidad de la información, permite a los usuarios tener acceso a los sistemas de cómputo, a la información de los sistemas y a las aplicaciones que realizan operaciones sobre la información, así como también, permite que los sistemas de comunicaciones transmitan información entre ubicaciones o sistemas de cómputo.

La importancia relacionada con los mecanismos de recuperación de la base de datos ante caídas del sistema asegura la disponibilidad de la información.

Autenticidad

Se asegura el origen y el destino de la información. La autenticación de los usuarios autorizados evita que los usuarios no autorizados obtengan el acceso a los sistemas de información. El uso de mecanismos de autenticación también puede evitar que los usuarios autorizados tengan acceso a la información que no les es permitido consultar.
En la actualidad, las contraseñas siguen siendo el principal mecanismo de autenticación para el acceso interno del sistema.

Los siguientes puntos son recomendados para la utilización de contraseñas.

- Longitud de la contraseña.
- Frecuencia de cambio de las contraseñas.
- Historial de contraseñas.
- Contenido de las contraseñas.

Ahora definiremos algunos más, que no son menos importantes ya que son manejados por algunos autores, los cuales son:

- No repudio: cualquier entidad que envía o recibe datos no puede alegar desconocer el hecho.

- Consistencia: asegurar que el sistema se comporta como se supone que debe hacerlo con los usuarios autorizados.

- Aislamiento: impedir que personas no autorizadas entren en el sistema.

- Auditoría: capacidad de determinar qué acciones o procesos se han llevado a cabo en el sistema y quién y cuándo las han llevado acabo.

- Prevención: los usuarios deben saber que sus actividades quedan registradas.

- Información: posibilidad de detectar comportamientos sospechosos.

2.6 Requerimientos de seguridad y forma de establecerlos

Es esencial que la organización identifique sus requerimientos de seguridad. Los tres recursos principales que plantea para lograrlo son:

a. Evaluar los riesgos que enfrenta la organización. Mediante la evaluación de riesgos se identifican las amenazas a los activos, se evalúan las vulnerabilidades y probabilidades de ocurrencia, y se estima el impacto potencial.

b. Los requisitos legales, normativos, reglamentarios y contractuales que deben cumplir la organización, sus socios comerciales, los contratistas y los prestadores de servicios.

c. El conjunto específico de principios, objetivos y requisitos para el procesamiento de la información, que ha desarrollado la organización para respaldar sus operaciones.

2.7 Administración de Riesgos

Como vimos anteriormente las definiciones de los servicios de seguridad de la información nos beneficiarán para conocer los detalles de cómo se deben de emplear dentro de una organización, esto dependerá de un análisis de riesgo. Por lo tanto es importante conocer todo lo que involucra la administración de riesgos.

Activos

Los activos son aquellos componentes de la organización (tangible e intangible) que son parte del patrimonio de la misma y necesitan ser resguardados.

Se pueden así estructurar en 5 categorías:

1. El entorno del Sistema de Información

2. El Sistema de Información
3. La propia Información.
4. Las Funcionalidades de la Organización
5. Otros Activos

La falla de un Activo de una categoría puede generar cadenas de fallas en otras categorías.

Así por ejemplo, fallas en Activos del Entorno (1) provocarían otras fallas en el Sistema de Información (2); éstos inciden en fallas de la Información (3), que soporta las Funcionalidades de la Organización (4) y ésas condicionan los otros Activos (5).

Una frase común a la cual se suele recurrir es que "Una cadena se rompe por el eslabón más débil" lo mismo ocurre en materia de seguridad no importa que la seguridad para un activo sea alta si para otro ésta es débil.

Metodología

Las técnicas de valoración del riesgo se basan en la identificación del mismo y en la asignación de un valor cualitativo y cuantitativo, para su ponderación respectiva.

Vulnerabilidad

Una vulnerabilidad se define como la "ocurrencia real de materialización de una Amenaza sobre un Activo"[3]

La Vulnerabilidad es una propiedad de la relación entre un activo y una amenaza. También es considerada una vía de ataque potencial. Está caracterizada por la dificultad que el nivel de capacidad técnica que se requiera para explotarla. El resultado de la explotación también debería ser tomada en cuenta. Ya que la consecuencia de una vulnerabilidad es que hace sistemas más propensos (débiles) de ser atacado por una amenaza o que un ataque tenga una mayor probabilidad de tener éxito.

Amenaza

Una amenaza se puede definir como: una acción o evento que puede violar la seguridad de un entorno de sistemas de información. [4]

[3] Maiwald, Eric, ***Fundamentos de Seguridad de Redes***, Segunda Edición, McGraw-Hill, México 2005, p. 144.

[4] Ibídem, p.145.

Una amenaza contra los datos es una persona hostil que de manera casual o usando alguna técnica especial (usualmente denominamos como fuerza bruta, hace uso de técnicas especiales para poder acceder a los datos), para difamar o modificar los datos manejados por un sistema.

Tipos de amenazas para una Base de Datos

Los datos se dividen en 2 clases:

- Datos
- Metadatos

Las amenazas afectan a ambas clases de datos pero con diferente impacto.

Amenazas:

- Accidentales: Fallos del software/hardware y errores humanos.
- Intencionadas: Intrusos y abusos de usuarios autorizados.

Existen tres componentes de amenazas:

a. Objetivo: El aspecto de la seguridad que puede ser atacado.

b. Agentes: Las personas u organizaciones que originan las amenazas

c. Eventos: El tipo de acción que representa la amenaza

a. Objetivos

Cuando la motivación es revelar la información sin autorización a individuos u organizaciones, la confidencialidad es el blanco. Cuando la amenaza implica modificar la información, el objetivo es la integridad. El atacante busca modificar ya sea su propia información o la de otras personas (por ejemplo: alterar la base de datos para poner en duda la exactitud de los datos).

La disponibilidad cuando se ejecuta un ataque de denegación de servicios. Tales ataques pueden enfocar en la disponibilidad de la información, de las aplicaciones, de los sistemas o de la infraestructura.

b. Agentes

Son las personas que pretenden dañar a una organización. Para ser creíbles de una amenaza, debe tener tres características.

- Acceso. La capacidad que tiene un agente para llevar a cabo el objetivo (contraseñas, password de acceso, etc.). Un agente debe tener acceso al sistema, a la red, a las instalaciones o a la información que desea.

- Conocimiento. El nivel y tipo de información que tiene un agente acerca del objetivo. El conocimiento que puede ser útil para un agente incluye lo siguiente:

    - ❖ Identificación de usuarios.
    - ❖ Contraseñas.
    - ❖ Ubicación de archivos.
    - ❖ Procedimiento de acceso físico.
    - ❖ Nombres de empleados.
    - ❖ Número telefónico de acceso.
    - ❖ Dirección de red.
    - ❖ Procedimientos de seguridad.

- Motivación. Las razones que puede tener un agente para presentar una amenaza hacia el objetivo. Un agente requiere de una motivación para actuar en contra del objetivo. Probablemente por las siguientes causas:

    - ❖ Reto
    - ❖ Codicia
    - ❖ Intento malintencionado

  Agentes a considerar

    - ❖ Los exempleados
    - ❖ Los hackers
    - ❖ Rivales comerciales
    - ❖ Terroristas
    - ❖ Delincuentes
    - ❖ Público en general
    - ❖ Las compañías que prestan servicio a la organización
    - ❖ Los clientes
    - ❖ Los invitados
    - ❖ Los desastres

c. Eventos

Son la manera o forma en la que un agente de amenazas puede ocasionar el daño a una organización.[5]

- ❖ Abuso del acceso autorizado a la información, a los sistemas o a los sitios
- ❖ Alteración malintencionada de la información
- ❖ Alteración accidental de la información
- ❖ Acceso no autorizado a la información, a los sistemas o a los sitios
- ❖ Destrucción malintencionada de la información, de los sistemas o de los sitios
- ❖ Destrucción accidental de la información, de los sistemas o de los sitios
- ❖ Interferencia física malintencionada con los sistemas o las operaciones
- ❖ Interferencia física accidental con los sistemas o con las operaciones
- ❖ Eventos físicos naturales que pueden interferir con los sistemas o con las operaciones
- ❖ Introducción de software malintencionado (de manera intencional) a los sistemas
- ❖ Interrupción de las comunicaciones internas o externas
- ❖ Escucha furtiva pasiva de comunicaciones internas o externas
- ❖ Robo de hardware o software

2.8 Análisis de Riesgo

La seguridad informática tiene como objetivo el mantenimiento de la confidencialidad, integridad, autenticidad y disponibilidad de los sistemas informáticos, así como definir e implantar las defensas necesarias para eliminar o reducir sus posibles consecuencias.

Sus componentes:

- Sistema de información.
- Amenaza
- Vulnerabilidad
- Impacto
- Riesgo
- Control o defensa

Algunos de estos componentes ya han sido explicados, sin embargo a continuación daremos la siguientes definiciones que harán que el tema sea mejor comprendido.

[5] Ídem

La defensa o control es todo medio físico o lógico empleado para mitigar un riesgo.

Un riesgo es el potencial de lo que puede ser perdido y necesita protección y la probabilidad de que éste produzca un impacto en la organización.

$$\text{Riesgo} = \text{Amenaza} + \text{Vulnerabilidad}$$

El riesgo es la combinación de las amenazas con la vulnerabilidad. Las amenazas sin vulnerabilidades no representan un riesgo. Del mismo modo, las vulnerabilidades sin amenazas no plantean riesgo alguno. En el mundo real no existe ninguna de estas dos condiciones.[6]

Por tanto, la evaluación del riesgo es un intento de identificar la probabilidad de que ocurra un evento perjudicial. El riesgo puede definirse cualitativamente en estos tres niveles:

- Bajo
- Medio
- Alto

La identificación del riesgo es sencilla. Todo lo que se tiene que hacer es identificar tanto las vulnerabilidades como las amenazas. La identificación de los riesgos que puede tener una organización debe ser adaptada a sus características.

2.9 Problemas de Seguridad en las Bases de Datos

Como lo hemos visto en los ambientes de bases de datos, las diferentes aplicaciones y usuarios de una organización acceden a los datos a través del DBMS. Uno de los principales problemas a los que se enfrenta un DBMS y debe resolver a través de una buena administración es el controlar que no haya datos duplicados, inconsistencia de los datos, y el manejo de amenazas de posibles usuarios no autorizados al acceso de los datos.

Una amenaza contra los datos la podemos definir de la siguiente manera: como una persona hostil que de manera casual o usando alguna técnica especial (usualmente denominamos como fuerza bruta, hace uso de técnicas especiales para poder acceder a los datos), para difamar o modificar los datos manejados por un sistema.

[6] Ibídem, p. 149.

Las violaciones a la seguridad de las bases de datos consisten en, la lectura, difamación modificaciones y borrado de los datos, por personas no autorizadas.

Las consecuencias de las violaciones de los datos están agrupadas dentro de 3 categorías:

- Difamación de los datos: Esto es causado por lectura de datos de manera intencionalmente así como casualmente por usuarios no autorizados, incluyendo en esta categoría las violaciones de claves secretas de datos autorizados únicamente a ciertas personas.

- Impropia modificación de los datos: Esto envuelve todas las violaciones a la integridad de los datos a través del manejo impropio de los datos o modificaciones de los mismos.

- Mal funcionamiento del servicio: Envuelve todas aquellas acciones que niegan el servicio a los usuarios al acceso de los datos o uso de los recursos.

Las amenazas contra la seguridad de los datos están clasificadas dentro de los siguientes factores en los que puede ocurrir de manera accidental o intencional.

- Desastres naturales o accidentales: tales como temblores, inundaciones o fuego. Estos accidentes pueden dañar los sistemas de hardware como los de almacenamiento de los datos, estos accidentes siempre causan violaciones de integridad y negación del servicio.

- Errores o problemas en hardware o software: Esto puede permitir la incorrecta aplicación de policías de seguridad y por consiguiente el acceso fácil para la lectura de datos no autorizados y modificación de los mismos o la negación del servicio a las personas autorizadas al uso y manejo de los datos.

- Errores humanos: violaciones al acceso del sistema no intencionadas como al dar una clave de acceso incorrecta o un mal uso de las aplicaciones, las consecuencias son las mismas a las anteriores causando problemas en el sistema o falta de integridad de los datos.

Factores intencionales denotan explícita y determinadamente fraude que causa problemas en los accesos de los datos, las violaciones a los sistemas envuelven dos tipos de usuarios los cuales son:

- Usuarios autorizados: Aquellos quienes abusan de sus privilegios y autorizaciones para hacer de los datos lo que a ellos mejor le convenga, como es el vender información dar accesos a ciertos datos o eliminar información.

- Agentes hostiles: Este tipo de personas realizan acciones de vandalismo hacia el software y/o hardware, leyendo, modificando datos privados. En ambos casos "legal" o "ilegal", el uso que estas personas le dan a los datos puede ser para sus propósitos fraudulentos. Usualmente las personas hostiles (denominadas también hackers) suelen usar cierto tipo de técnicas para hacerse de información tales como: virus, caballos de troya y trampas de puerta, éstas son algunas de las técnicas que las personas hostiles usan para hacerse de la información.

Estos son solo algunos de los problemas que se pueden presentar para la base de datos, sin embargo como lo mencionamos anteriormente al realizar el análisis de riesgo, determinaremos cuales son sus principales problemas.

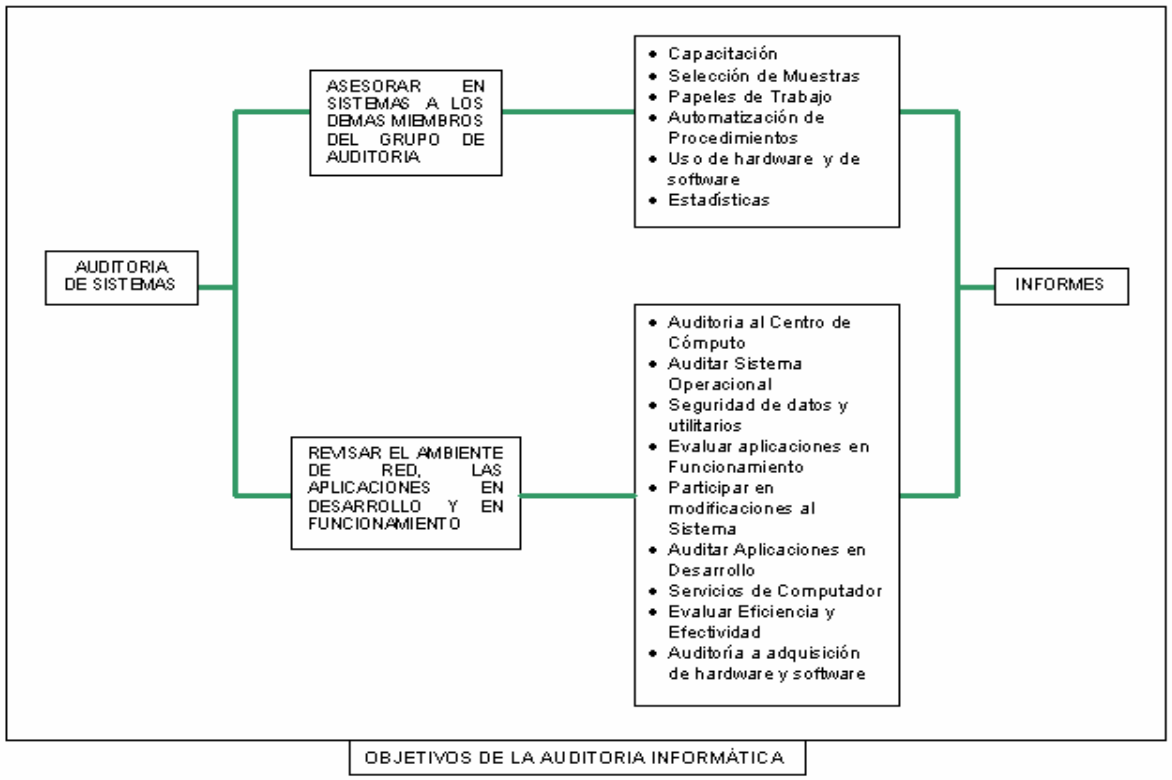
2.10 Definición de Auditoría Informática

Es la revisión y la evaluación de los controles, sistemas y procedimientos de informática; de los equipos de cómputo, su utilización, eficiencia y seguridad; de la organización que participan en el procesamiento de la información, a fin de que por medio del señalamiento de cursos alternativos se logre una utilización más eficiente y segura de la información que servirá para una adecuada toma de decisiones.

La auditoría en informática deberá comprender no sólo la evaluación de los equipos de cómputo de un sistema o procedimiento específico, sino que además habrá de evaluar los sistemas de información en general desde sus entradas, procedimientos, controles, archivos, seguridad y obtención de información.

La auditoría informática es de vital importancia para el buen desempeño de los sistemas de información, ya que proporciona los controles necesarios para que los sistemas sean confiables y con un buen nivel de seguridad.

Podemos expresarlo gráficamente así:



2.11 Norma de Seguridad ISO/17799

¿Qué es?

En toda organización que haga uso de las tecnologías de información se recomienda implementar buenas prácticas de seguridad, pues en muchas ocasiones el no seguir un proceso de implementación adecuado como el que establece el ISO 17799 puede generar huecos, en ese sentido, aumenta la posibilidad de riesgos en la información.

Este estándar internacional de alto nivel para la administración de la seguridad de la información, fue publicado por la ISO (International Organization for Standardization) en diciembre de 2000 con el objeto de desarrollar un marco de seguridad sobre el cual trabajen las organizaciones.

El ISO 17799, al definirse como una guía en la implementación del sistema de administración de la seguridad de la información, se orienta a preservar los siguientes principios de la seguridad informática:

Confidencialidad. Asegurar que únicamente personal autorizado tenga acceso a la información.

Integridad. Garantizar que la información no será alterada, eliminada o destruida por entidades no autorizadas.

Disponibilidad. Asegurar que los usuarios autorizados tendrán acceso a la información cuando la requieran.

Estos principios en la protección de los activos de información constituyen las normas básicas deseables en cualquier organización, sean instituciones de gobierno, educativas e investigación; no obstante, dependiendo de la naturaleza y metas de las organizaciones, éstas mostrarán especial énfasis en algún dominio o área del estándar ISO 17799.

El objetivo de la seguridad de los datos es asegurar la continuidad de las operaciones de la organización, reducir al mínimo los daños causados por una contingencia, así como optimizar la inversión en tecnologías de seguridad.
Como todo buen estándar, el ISO 17799 da la pauta en la definición sobre cuáles metodologías, normas o estándares técnicos pueden ser aplicados en el sistema de administración de la seguridad de la información, se puede entender que estos estándares son auxiliares y serán aplicados en algún momento al implementar el mismo.

La aplicación de un marco de referencia de seguridad basado en el ISO 17799 proporciona beneficios a toda organización que lo implemente, al garantizar la existencia de una serie de procesos que permiten evaluar, mantener y administrar la seguridad de la información.

Las políticas, estándares locales y los procedimientos se encuentran adaptados a las necesidades de la organización debido a que el proceso mismo de su elaboración integra mecanismos de control y por último, la certificación permite a las organizaciones demostrar el estado de la seguridad de la información.

Breve historia

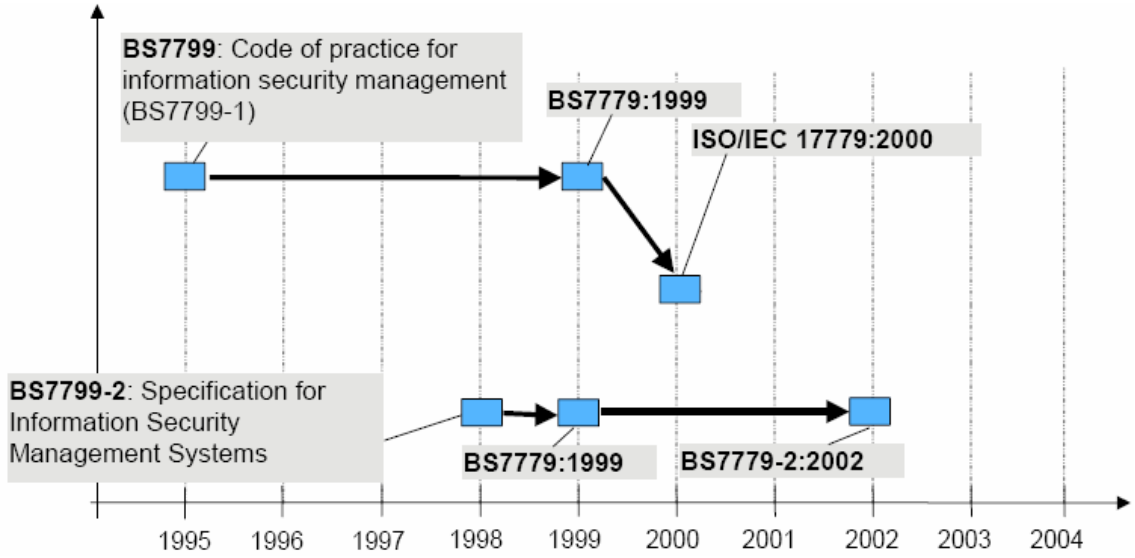
Es importante entender los principios y objetivos que dan vida al ISO 17799, así como los beneficios que cualquier organización, incluyendo las instituciones públicas, privadas y ambientes educativos pueden adquirir al implementarlo en sus prácticas de seguridad de la información.

El estándar de seguridad de la información ISO 17799, descendiente del BS 7799 – Information Security Management Standard – de la BSI (British Standard Institute) que publicó su primera versión en Inglaterra en 1995, con actualizaciones realizadas en 1998 y 1999, consiste de dos partes:

- **Parte 1.** Código de prácticas.

- **Parte 2.** Especificaciones del sistema de administración de seguridad de la información.

Por la necesidad generalizada de contar con un estándar de carácter internacional que permitiera reconocer o validar el marco de referencia de seguridad aplicado por las organizaciones, se elaboró el estándar ISO17799:2000, basado principalmente en la primera parte del BS 7799 conocida como Código de Prácticas (BS 7799 Part 1: Code of Practice).



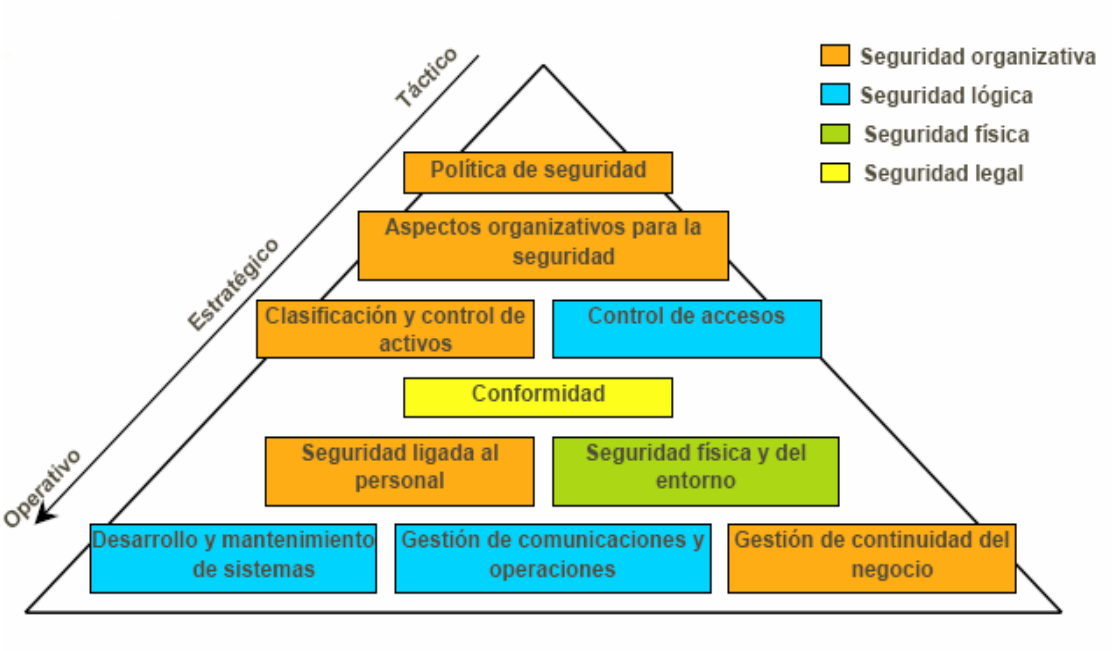
Los controles del ISO 17799

El éxito de la implementación del estándar de seguridad ISO 17799 requiere de una serie de procedimientos donde, inicialmente, el análisis de riesgos identificará los activos de la información y las amenazas a las cuales se encuentra expuesta.

La norma ISO 17799 establece diez dominios de control que cubren por completo la Gestión de la Seguridad de la Información:

1. Políticas de seguridad.
2. Seguridad organizacional.
3. Clasificación y control de activos.
4. Seguridad del personal.
5. Seguridad física y de entorno.
6. Comunicaciones y administración de operaciones.
7. Control de acceso.
8. Desarrollo de sistemas y mantenimiento.
9. Continuidad de las operaciones de la organización.
10. Requerimientos legales.

Gráficamente los dominios de control se clasifican y se ordenan de la siguiente manera:



A continuación, se describirán cada uno de los diez dominios de control.

POLITICAS DE SEGURIDAD.

- Dirigir y dar soporte a la gestión de la seguridad de la información.

- La alta dirección debe definir una **política** que refleje las líneas directrices de la organización en materia de seguridad, aprobarla y publicitarla de la forma adecuada a todo el personal implicado en la seguridad de la información.

- La política se constituye en la base de todo el sistema de seguridad de la información.

- La alta dirección debe **apoyar visiblemente** la seguridad de la información en la compañía.

SEGURIDAD ORGANIZACIONAL

- Gestionar la seguridad de la información dentro de la organización.
- Mantener la seguridad de los recursos de tratamiento de la información y de los activos de información de la organización que son accedidos por terceros.
- Mantener la seguridad de la información cuando la responsabilidad de su tratamiento se ha dado otra organización.

- Debe diseñarse una estructura organizativa dentro de la compañía que defina las **responsabilidades** que en materia de seguridad tiene cada usuario o área de trabajo relacionada con los sistemas de información de cualquier forma.

- Dicha estructura debe poseer un enfoque **multidisciplinar**: los problemas de seguridad no son exclusivamente técnicos.

CLASIFICACION Y CONTROL DE ACTIVOS

- Mantener una protección adecuada sobre los activos de la organización.
- Asegurar un nivel de protección adecuado a los activos de información.

- Debe definirse una **clasificación** de los activos relacionados con los sistemas de información, manteniendo un **inventario** actualizado que registre estos datos, y proporcionando a cada activo el nivel de **protección** adecuado a su criticidad en la organización.

SEGURIDAD DEL PERSONAL

- Reducir los riesgos de errores humanos, robos, fraudes o mal uso de las instalaciones y los servicios.
- Asegurar que los usuarios son conscientes de las amenazas y riesgos en el ámbito de la seguridad de la información, y que están preparados para sostener la política de seguridad de la organización en el curso normal de su trabajo.
- Minimizar los daños provocados por incidencias de seguridad y por el mal funcionamiento, controlándolos y aprendiendo de ellos.

Seguridad del personal. Contrario a lo que uno se puede imaginar, no se orienta a la seguridad del personal desde la óptica de protección civil, sino a proporcionar controles a las acciones del personal que opera con los activos de información.

- Las implicaciones del **factor humano** en la seguridad de la información son muy elevadas.
- Todo el personal, tanto **interno** como **externo** a la organización, debe conocer tanto las líneas generales de la política de seguridad corporativa, como las implicaciones de su trabajo en el mantenimiento de la seguridad global.
- Diferentes relaciones con los sistemas de información: operador, administrador, guardia de seguridad, personal de servicios, etc.
- Procesos de **notificación de incidencias** claros, ágiles y conocidos por todos.

SEGURIDAD FISICA Y DE ENTORNO

- Evitar accesos no autorizados, daños e interferencias contra los locales y la información de la organización.
- Evitar pérdidas, daños o comprometer los activos así como la interrupción de las actividades de la organización.
- Prevenir las exposiciones a riesgo o robos de información y de recursos de tratamiento de información.

- Identificar los perímetros de seguridad, de forma que se puedan establecer controles en el manejo de equipos, transferencia de información y control de los accesos a las distintas áreas con base en el tipo de seguridad establecida.

COMUNICACIONES Y CONTROL DE OPERACIONES

- Asegurar la operación correcta y segura de los recursos de tratamiento de información.
- Minimizar el riesgo de fallos en los sistemas.
- Proteger la integridad del software y de la información.
- Mantener la integridad y la disponibilidad de los servicios de tratamiento de información y comunicación.
- Asegurar la salvaguarda de la información en las redes y la protección de su infraestructura de apoyo.
- Evitar daños a los activos e interrupciones de actividades de la organización.
- Prevenir la pérdida, modificación o mal uso de la información intercambiada entre organizaciones.

- Se debe garantizar la seguridad de las **comunicaciones** y de la **operación** de los sistemas críticos para el negocio.

CONTROL DE ACCESO

- Controlar los accesos a la información.
- Evitar accesos no autorizados a los sistemas de información.
- Evitar el acceso de usuarios no autorizados.
- Protección de los servicios en red.
- Evitar accesos no autorizados a ordenadores.
- Evitar el acceso no autorizado a la información contenida en los sistemas.
- Detectar actividades no autorizadas.
- Garantizar la seguridad de la información cuando se usan dispositivos de informática móvil.

- Habilitar los mecanismos que permitan monitorear el acceso a los activos de información, que incluyen los procedimientos de administración de usuarios, definición de responsabilidades o perfiles de seguridad y el control de acceso a las aplicaciones.

DESARROLLO DE SISTEMAS Y MANTENIMIENTO

- Asegurar que la seguridad está incluida dentro de los sistemas de información.
- Evitar pérdidas, modificaciones o mal uso de los datos de usuario en las aplicaciones.
- Proteger la confidencialidad, autenticidad e integridad de la información.
- Asegurar que los proyectos de Tecnología de la Información y las actividades complementarias son llevadas a cabo de una forma segura.
- Mantener la seguridad del software y la información de la aplicación del sistema.

- Debe contemplarse la seguridad de la información en **todas las etapas** del ciclo de vida del software en una organización: especificación de requisitos, desarrollo, explotación, mantenimiento...

CONTINUIDAD DE LAS OPERACIONES DE LA ORGANIZACIÓN

- Reaccionar a la interrupción de actividades del negocio y proteger sus procesos críticos frente grandes fallos o desastres.

- Todas las situaciones que puedan provocar la **interrupción** de las actividades del negocio deben ser **prevenidas** y **contrarrestadas** mediante los planes de contingencia adecuados.

- Los **planes de contingencia** deben ser probados y revisados periódicamente.

- Se deben definir **equipos de recuperación** ante contingencias, en los que se identifiquen claramente las funciones y responsabilidades de cada miembro en caso de desastre.

REQUERIMIENTOS LEGALES

- Evitar el incumplimiento de cualquier ley, estatuto, regulación u obligación contractual y de cualquier requerimiento de seguridad.

- Garantizar la alineación de los sistemas con la política de seguridad de la organización y con la normativa derivada de la misma.

- Maximizar la efectividad y minimizar la interferencia de o desde el proceso de auditoría de sistemas.

- La organización establecerá los requerimientos de seguridad que deben cumplir todos sus proveedores, socios y usuarios; éstos se encontrarán formalizados en los contratos o convenios.

- Cada una de las áreas establece una serie de controles que serán seleccionados dependiendo de los resultados obtenidos en el análisis de riesgos, además, existen controles obligatorios para toda organización, como es el de las políticas de seguridad cuyo número dependerá más de la organización que del estándar, el cual no establece este nivel de detalle.

2.12 Definición de FRAP (Facilitated Risk Analysis Process)

Es una metodología para realizar un Proceso de Análisis de Riesgos de Forma Fácil (FRAP: Facilitated Risk Análisis Process por sus siglas en inglés).
Este proceso permitirá a cualquier organización implementar técnicas de administración de riesgo. FRAP puede ser utilizado por los profesionales en seguridad de la información, administradores de proyecto, seguridad física, etc.

FRAP utiliza metodologías cualitativas formales del análisis del riesgo para determinar las soluciones rentables para los temas, las aplicaciones o los sistemas específicos.

Como Trabaja FRAP

- Requiere un nivel de vulnerabilidad
- Requiere del impacto desfavorable al negocio

FRAP nos sirve para identificar y priorizar los riegos, determina las acciones que deberán ser seguidas en función de los niveles de riesgo.

Concepto de vulnerabilidad

- Debilidad o hueco de seguridad.
- Indica que el activo es susceptible a recibir un daño a través de un ataque.
- Los ataques pueden ser intencionados o accidentales.

Los tipos de vulnerabilidad son:

- Alto
- Medio
- Bajo

Concepto de impacto

- Es la “materialización” de un riesgo.
- Una medida del grado de daño o cambio.

Posibles Impactos a la Institución

- Revelación de información confidencial.
- Pérdidas materiales por eventos catastróficos.
- Interrupción de los procesos críticos y no críticos de la institución.

- Insatisfacción e incumplimiento de terceras partes.
- Incumplimientos de requerimientos legales.

Concepto de Riesgo

- Evento Potencial.
- Puede tener un impacto negativo.
- El impacto es sobre:
  - Misión de la Institución.
  - Los objetivos de la Institución.

La materialización de un riesgo, puede resultar en:

- Acceso no autorizado.
- Revelación no autorizada de información.
- Observar y monitorear transacciones.
- Copiar sin autorización.
- Acceder a información confidencial.

Concepto de control

- Es una medida para
  - detectar,
  - reducir o
  - recuperarse de un impacto

- Medidas y procedimientos establecidos con objeto de alcanzar los objetivos de una institución.

Riesgo de Integridad

Se refiere a la pérdida o deficiencia en la autorización, totalidad o exactitud de la información de la organización.

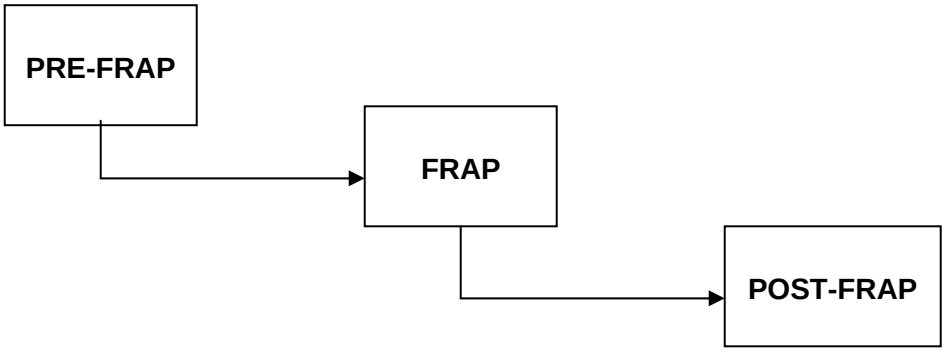
Riesgo de Confidencialidad

El riesgo de que las personas no autorizadas tengan acceso a información confidencial o que sea negado al personal que si cuenta con dicha autorización.

Riesgo de Disponibilidad

El riesgo de que la información no esté disponible en el momento en que se requiere.

2.13 Etapas del FRAP



Pre – FRAP

Descripción de la etapa:

Es una etapa clave para el éxito del proyecto. Son definidos, desarrollados y afinados todos los requerimientos de la Sesión FRAP.

Es necesaria la participación de:

- Business Managers – Administradores del Negocio
- Project development leaders – Líderes de proyecto
- Facilitator – Facilitador

Las etapas son:

- PRE-FRAP Meeting (reunión)
- FRAP Team (equipo)
- FRAP Facilitator (facilitador)

Realizar una reunión de trabajo (Pre-FRAP meeting) y definir los siguientes 5 componentes clave:

1. Scope statement – Declaración del alcance
2. Visual model – Modelo Visual

3. Establish the FRAP team – Establecimiento del Equipo FRAP
4. Meeting mechanics – Mecánica de la reunión
5. Agreement definitions – Acuerdo de definiciones

Pre-FRAP meeting y los 5 componentes clave:

1. Scope statement – Declaración del Alcance
    - Es desarrollado por los gerentes del negocio y los líderes de proyecto.
    - Define en palabras, **que exactamente se va a revisar**.

2. Visual model – Modelo Visual
    - Es un diagrama que ilustre el proceso a revisar.
    - El modelo es utilizado durante la Sesión de FRAP para familiarizar el Equipo FRAP con el proceso de principio a fin.

3. Establish the FRAP team – Establecimiento del Equipo FRAP
    - Un equipo típico de FRAP tiene entre 7 y 15 miembros.
    - Incluye representantes de las áreas de negocio.
    - Incluye también representantes de las áreas de soporte al negocio.

4. Meeting mechanics – Mecánica de la reunión
    - Es la definición de la logística de la sesión.
    - Incluye, definir sala, ubicación, material (gente, proyectores, acetatos, rotafolios, café y donas).

5. Agreement definitions – Acuerdo de definiciones
    - La uniformidad de términos es un requisito
    - Con esto se unifican los conceptos y hace que las divergencias sean menores.

Durante la sesión Pre-FRAP es importante discutir el proceso para priorizar los riesgos.

Existen dos escuelas para hacer esto:

1. Hacer que el Equipo FRAP revise todos los riesgos como si no existieran controles en sitio.
    - Esta alternativa permitirá pensar en el control "ideal"
    - Esto permitirá al equipo FRAP hacer un análisis entre "cómo deben ser" los controles y "cómo son", demostrando las vulnerabilidades
2. Evaluar los riesgos considerando los controles en sitio
    - La palabra clave es "evaluar".

FRAP Team

Durante la Pre-FRAP Meeting, los gerentes de negocio y líderes de proyecto determinarán:

- Quiénes serán parte del equipo FRAP.
- El número de participantes (entre 7 y 15)
- Es recomendable incluir a las siguientes áreas en el proceso de FRAP

❖ Dueños funcionales
❖ Usuarios del sistema
❖ Administradores del sistema
❖ Analistas del sistema
❖ Programadores del sistema
❖ Programadores de aplicaciones
❖ Administradores de bases de datos
❖ Seguridad informática
❖ Seguridad física
❖ Telecomunicaciones
❖ Administradores de Red
❖ Proveedores de Servicios
❖ Auditoria Legal y Recursos Humanos (si es apropiado)

FRAP Facilitator

- El facilitador de la Sesión debe cubrir con un número de skills especiales.
- Estos skills pueden ser mejorados con entrenamiento especial.

Los skills requeridos incluyen habilidades para:

- Listen – Escuchar.
  - Tener la habilidad de ser sensible a los comportamientos verbales y no verbales de los asistentes.
  - Ser capaz de parafrasear respuestas del punto bajo revisión
  - Tener la habilidad de aclarar respuestas.
- Lead – Liderear.
  - Iniciar la sesión y motivar a los participantes a abrir discusiones enfocadas en un tópico a la vez.
- Reflect – Reflejar.
  - Repetir ideas con palabras frescas y con nuevo énfasis.
- Summarize – Resumir.
  - Ser capaz de resumir temas e ideas.
- Confront – Confrontar.
  - Ser capaz de retroalimentar opiniones.
  - Reaccionar honestamente a las entradas de información.
  - Ser capaz de tomar comentarios severos o ásperos y convertirlos en declaraciones positivas.
- Support – Soportar
  - Crear un clima de confianza y aceptación.
- Crisis Intervention – Intervenir en Crisis.
  - Ayudar a ampliar la visión de las personas con opciones y alternativas.
  - Reforzar puntos de acción que pueden resolver conflictos o crisis.
- Center – Centrar.
  - Ayudar al equipo a aceptar otros puntos de vista.
  - Generar confianza para una participación de todos.

- Solve Problem – Resolver Problemas.
  - Obtener información relevante en relación al manejo de los temas.
  - Ayudar al equipo a establecer objetivos de control específicos.
- Change Behavior – Cambio de comportamiento.
  - Identificar aquellos participantes que en apariencia no son parte del proceso.
  - Lograr que tengan una participación activa.

Reglas básicas que el Facilitador tendrá que observar:

1. Observar cuidadosamente todo lo que los participantes hagan y digan.
2. Reconocer todas las entradas y alentar la participación.
3. Observar las respuestas no verbales.
4. Nunca leer; escuchar y lograr el involucramiento del grupo.
5. Nunca perder el aspecto de la objetividad.
6. Ser neutral (o siempre aparentar ser neutral).
7. Aprender a esperar hostilidad, pero nunca ser hostil.
8. Evitar ser “la autoridad experta”. El rol del facilitador es escuchar, preguntar, alentar el proceso y ofrecer alternativas.
9. Adherirse a los tiempos y ser puntual.
10. Utilice intermedios (breaks) para liberar una discusión.
11. Esta allí para servir al equipo FRAP.
12. Terminar el proceso si el grupo es indolente y difícil de controlar.

Reglas básicas que deberán seguirse en la sesión:

- Todos participan
- Todos tienen roles identificados.
- Todos se apegarán a una agenda.
- Todas las ideas tienen el mismo valor.
- Escuchar otros puntos de vista.

- Todos los puntos son registrados.
- Registrar y postergar los puntos fuera del alcance.
- Fijar la idea antes de discutirla.
- Asegurar que todas las ideas fueron registradas por el apuntador.
- Una conversación a la vez.
- Una persona enojada a la vez.
- Aplicar la regla de los 3 minutos:
  Todas las discusiones deberán ser concluidas en un marco de tiempo acordado.

SER:

  - PUNTUAL
  - JUSTO
  - AGRADABLE
  - CREATIVO

- Diviértanse – Have Fun!

FRAP Sesión

La Sesión FRAP esta dividida en cuatro secciones:

- Logística
- Lluvia de ideas
- Priorizando los riesgos
- Identificando Controles

FRAP Session. Logística

- El equipo FRAP se presentará por sí mismo.
- Indicará Nombre, Título, Departamento y Teléfono (todo esto deberá se registrado).
- Los roles del equipo FRAP deben ser identificados y discutidos.

- Típicamente existen cinco roles:

    - Dueño (Owner)
    - Líder de Proyecto (Project lead)
    - Facilitador (Facilitator)
    - Apuntador(Scribe)
    - Equipo FRAP (Teams members)

- El equipo FRAP deberá proporcionar una idea clara del proceso en el cual forman parte.

- Debe también exponer la declaración del alcance (Scope Statement).

- Y algún miembro del equipo que sea de tecnología deberá describir el proceso bajo revisión (Visual Model).

- Finalmente, deben revisarse las definiciones (Agreement Definitions) por todo el equipo FRAP.

FRAP Sesión. Lluvia de Ideas (Brainstorming process)

- Una vez entendido el proceso del negocio, las definiciones clave y el alcance, el siguiente paso es la lluvia de ideas.

- Por cada elemento bajo revisión, se tendrán que identificar los riesgos que puedan impactar en la integridad, confidencialidad y disponibilidad.

- En este proceso el facilitador deberá proporcionar la definición y algunos ejemplos de riesgos.

Definición y ejemplos de riesgos:

Ejemplos de Riesgos (lista de referencia)

- ❖ Riesgos a la confidencialidad:
  - Acceso sin autorización
  - Revelación sin autorización
  - Observar o monitorear transacciones
  - Copiar sin autorización
  - Escuchar en la red (packet sniffing)
  - Terceros con acceso a información confidencial
- ❖ Definición:

  Confidencialidad: Que la información no haya experimentado un acceso no autorizado o deseable.

- El equipo FRAP tendrá unos minutos para escribir los riesgos relacionados con ellos.
- El facilitador obtendrá del equipo FRAP un riesgo por cada miembro.
- El proceso de brainstorming continúa hasta que cada elemento haya sido revisado.

. . .

- En este momento es recomendable hacer un receso (coffe break).

. . .

- Cuando sea reactivada la sesión, los miembros del equipo deberán revisar los riesgos que han definido.
- Esto los llevará a identificar riesgos duplicados y en algunos casos, a renombrar los ya definidos.
- Lo anterior dará como resultado una lista de riesgos y se tendrá lista la información para la siguiente etapa.

FRAP Sesión. Priorizando los Riesgos (Prioritizing the Risk)

- El equipo deberá ser instruido y concentrado en priorizar los riesgos.
- Esto será realizado mediante la determinación del nivel de vulnerabilidad y el impacto al negocio, si el riesgo ocurre.
- Las siguientes definiciones debieron ser definidas en el Agreement Definitions y presentadas al equipo en la introducción logística.

- Vulnerabilidad Alta.

    - Es una debilidad sustancial que existe en los sistemas o rutinas operacionales.

    - Son debilidades en la seguridad que representan un riesgo elevado, y que de ser explotadas pueden generar una interrupción de los servicios, o bien, proporcionar un acceso sin restricciones, o con muy pocas de ellas, a personal no autorizado.

- Vulnerabilidad Media.

    - Existen algunas debilidades.

    - Son debilidades en la seguridad que por si mismas no constituyen un nivel de riesgo significativo, sin embargo, al encontrarse de forma conjunta originan la posibilidad de accesos no autorizados, e inclusive interrupciones en el servicio.

- Vulnerabilidad Baja.

    - Los sistemas han sido bien construidos y operan correctamente.

    - Son debilidades en la seguridad que al ser explotadas proporcionan información confidencial, como datos de componentes tecnológicos y de usuarios, dicha información es utilizada para conocer las configuraciones y tecnología existente.

- Impacto Severo (alto)

    - Puede poner a la empresa fuera de su negocio.

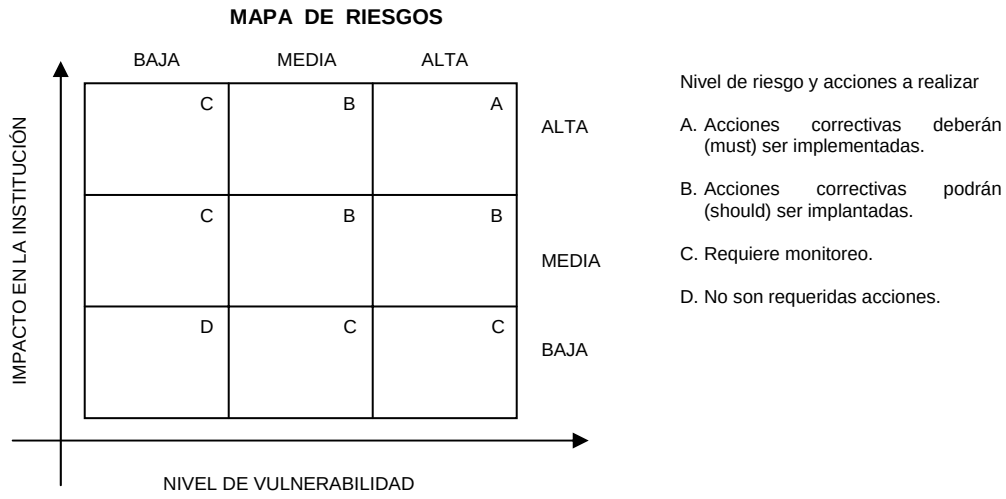
- Impacto Significativo (medio)

    - Puede causar daño y costo significativo, pero la empresa puede sobrevivir.

- Impacto Menor (bajo)

    - Es un impacto operacional que tiene que ser administrado como parte de las operaciones cotidianas.

- Una vez finalizada la lista de riesgos y entendidas las definiciones, el facilitador puede ayudarse del siguiente modelo:



- Para priorizar los riesgos existen diferentes técnicas, las tres más populares son:

    1. El facilitador toma riesgo por riesgo y el equipo FRAP discute para lograr un consenso.

    2. El facilitador revisa tres o cuatro riesgos para asegurar que el equipo tiene una idea correcta de cómo el proceso trabajará.

Cada miembro del equipo tiene un marcador de color y marcará sobre una tarjeta blanca la prioridad, si no hay opinión, mostrará su tarjeta en blanco.

Finalmente, el facilitador revisará la dispersión en las respuestas, y si considera pertinente discutirá el tema con el equipo FRAP para asegurarse de la respuesta más correcta.

3. La tercera técnica, puede ser utilizada si el facilitador da a cada miembro del equipo diez puntos.

   Cada miembro tendrá permitido votar por los diez mayores riesgos.

- El resultado final de la priorización de riesgos puede ser el siguiente:

#	Risk	Priority
1	La información puede ser accesada por personal que no debe tener acceso.	B
2	Versiones no claras o existentes de información	B
3	La base de datos puede ser corrupta por fallas de hardware o software	D
4	Los datos pueden ser corruptos por una transacción no completada	C
5	Falla que reporte puntos de integridad	A
6	No notificación de problemas de integridad	A
7	Información utilizada en un contexto	B
8	Acceso de terceros a información confidencial	A

- La etapa final del proceso FRAP es identificar controles para los riesgos identificados.

- Para esto es posible utilizar una lista de controles previamente definidos, que deberán ser distribuidos en el equipo FRAP.

- Los controles son el punto de partida, sin embargo estos podrán ser modificados y adicionados según sea necesario.

- A continuación se muestra una lista de 26 controles desarrollados por facilitadores FRAP.

FRAP Sesión. Controles Identificados

#	Control	Descripción del Control
1	Backup	Los requerimientos de respaldo deben ser determinados al proveedor de servicios, incluyendo los requerimientos de notificación electrónica que la realización de respaldos ha sido completada y debe ser enviada al administrador de las aplicaciones. El proveedor de servicios tiene que requerir las pruebas de los procedimientos de respaldo.
2	Planta de recuperación	Desarrollar, documentar y probar los procedimientos de recuperación diseñados para asegurar que las aplicaciones e información pueden ser recuperadas, usando los respaldos creados, en caso de suceder una perdida.
3	Control de Acceso	Implementar mecanismos de control de acceso que prevenga un acceso no autorizado a la información. Estos mecanismos deben de incluir la capacidad de detectar, registrar y reportar intentos de violación de a seguridad en esta información.
4	Control de Accesos	Origen da acceso: Implementar mecanismos para limitar el acceso a información confidencial especificada en rutas de la red o en localidades físicas.
5	Control de Acceso	Implementar mecanismos de autenticación de usuarios (como firewall, control de dial-in, identificadores seguros) para limitar el acceso al personal autorizado.
6	Control de Acceso	Implementar mecanismos de encriptación (datos, end-to-end) para prevenir un acceso no autorizado y proteger la integridad y confidencialidad de la información.
7	Control de aplicaciones	Diseño e implementación de control en aplicaciones (entrada de datos, verificación de edición, validación de requerimientos de campo, identificadores de alarma, capacidades de expiración de password, checksum) para asegurar la integridad, confidencialidad y disponibilidad de la información de la aplicaciones.
8	Pruebas de aceptación	Desarrollar procedimiento de prueba para ser aplicados durante el desarrollo y modificaciones de las aplicaciones existentes las cuales incluyan la participación y aceptación del usuario.
9	Administración de cambios	Apegarse a los procedimientos de administración diseñados para facilitar el aprovechamiento de la estructura de las modificaciones de “emergencia” se deben de incluir en este proceso.

10	Antivirus	1. Asegura que el administrador de red instale es software corporativo estándar de antivirus en todas las computadoras. 2. Capacitación y concientización en las técnicas de prevención de los virus deben ser incorporadas en el programa IP de la organización
11	Política	Desarrollar políticas y procedimientos para limitar el acceso y privilegios de opresiones para que éstos sean solo los necesarios para el negocio.
12	Entrenamiento	El entrenamiento del usuario deberá incluir instrucciones y documentación en el uso adecuado de la aplicación. La importancia del mantenimiento de la confidencialidad de las cuentas de usuarios, passwords la confidencialidad y naturaleza competitiva de la información debe ser acentuada.
13	Auditoría/ Monitoreo	Implementar mecanismos para el monitoreo, reporte e identificación de actividades de auditoria como revisión de requerimientos independientes, incluyendo revisiones periódicas del identificador de usuarios (user-id) para asegurar y verificar las necesidades del negocio.
14	Respaldos	Control en las operaciones: Capacitación para la realización de respaldos en los sistemas administrativos debe de ser provista conjuntamente con la rotación de funciones para asegurar la suficiencia del programa de entrenamiento.
15	Entrenamiento	Control en las operaciones: Los desarrolladores de aplicaciones deben de proveer la documentación, guías, y el soporte al staff de operación (proveedor de servicios) con la implementación de mecanismos para asegurar que la transferencia de información entre las aplicaciones sea de manera segura.
16	Control de Acceso	Control en las operaciones: Mecanismos para proteger las bases de datos contra accesos no autorizados y modificaciones realizadas fuera de la aplicación, pueden ser determinadas e implementadas.
17	Departamento de interfases	Control en las operaciones: Sistemas que puedan alimentar información podrían identificar y comunicarse con el proveedor de servicios para enfatizar el impacto de la funcionalidad si estas aplicaciones de alimentación no están disponibles.
18	Mantenimiento	Control en operaciones: Requerimientos en tiempo para el mantenimiento podrían seguirse y solicitar para el ajuste mientras sea comunicado al administrador si la garantía expira.

19	Entrenamiento	Control de usuarios: Implementar un programa para los usuarios (evaluación del desempeño del usuario) diseñado para alentar el cumplimiento con las políticas y procedimientos en lugar de asegurar la apropiada utilización de la aplicación.
20	Acuerdos de nivel de servicios	Adquirir acuerdos de nivel de servicios para establecer niveles esperados con los clientes y de esta forma asegurarse del soporte en las operaciones.
21	Mantenimiento	Adquirir mantenimiento que sustituya los acuerdos para facilitar la continuidad en el estado de operación de la aplicación.
22	Seguridad Física	En comparación con las facilidades de administración, las facilidades de implementación de los controles de seguridad física designados para proteger la información, software y hardware son requeridos para el sistema.
23	Administración de Soporte	Requerimientos de administración del soporte para asegurar la cooperación y coordinación de las unidades de negocio para facilitar y preparar el terreno para la transición de la aplicación.
24	Propiedad	Controles sobre la propiedad
25	Estrategias correctivas	El equipo de desarrollo deberá de implementar estrategias correctivas como trabajos de reprocesamiento, aplicaciones lógicas para la revisión, etc.
26	Administración de cambios	Controles para la migración a producción, como búsquedas y remoción de procesos para asegurar que los datos estén limpios.

Los controles pueden ser identificados de dos formas:

- El facilitador debe ir a los riesgos con alta prioridad y hacer que el equipo diga en voz alta el número de control que piensan mitigará el riesgo.

- El facilitador puede trabajar en los primeros tres o cuatro riesgos y después permitir que el equipo, por cada riesgo, anote en un tablero el número de control sugerido.

- El equipo necesita entender que lo seleccionado es ahora lo que deberá implantar.

- La Sesión FRAP ha terminado con los siguientes entregables:

    - Riesgos Identificados

- Riesgos Priorizados
- Controles Identificados

Post – FRAP

- Finalmente, el gerente del negocio, líder de proyecto y facilitador deberán trabajar en completar el plan de acción para la administración del riesgo.
- La etapa Post-FRAP tiene cinco entregables:
  - ❖ Hoja de referencias cruzadas
  - ❖ Identificación de controles existentes
  - ❖ Identificación y selección de nuevos controles para los riesgos nuevos.
  - ❖ Reporte Final
  - ❖ La hoja de referencias cruzadas (Cross-reference sheet) es una de las tareas que mas consumen tiempo del facilitador y del apuntador (Scribe).
- Este documento toma cada control e identificar todos los riesgos que pueden ser impactados por ese simple control.
- Esta actividad llega a tardar hasta dos días por su nivel de análisis y complejidad.

CAPITULO 3. IMPLEMENTACION DE LA METODOLOGIA

3.1 Niveles de Acceso a un Servidor Sybase

Dentro del Sistema Manejador de Base de Datos, podemos encontrar un sistema multicapas, mediante el cual el usuario final puede acceder a los datos dentro del Servidor:

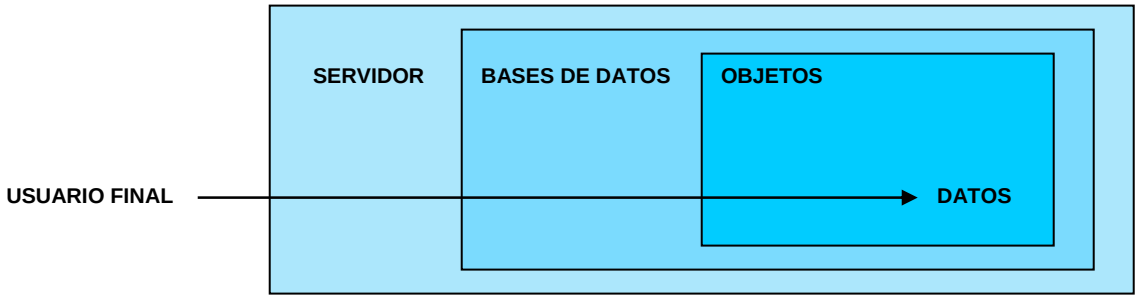
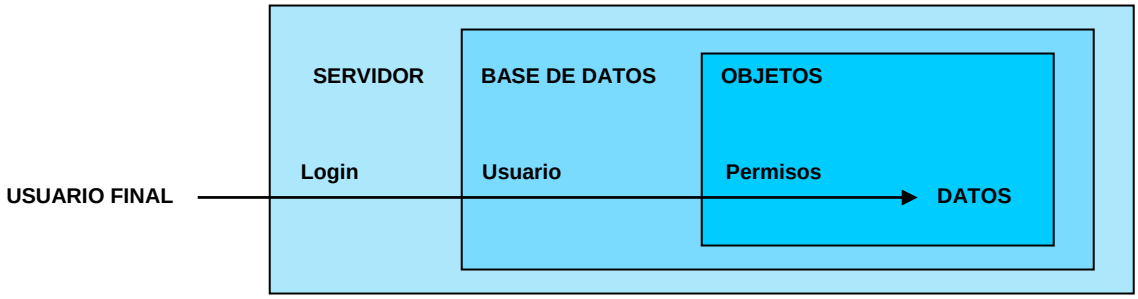


FIGURA 1. **ESQUEMA PARA ACCESAR A LAS DIFERENTES CAPAS DE UN SERVIDOR SYBASE**

- El usuario final debe tener permiso para entrar al Servidor.
- El usuario final debe tener permiso de acceso a una base de datos dada.
- Finalmente, debe tener permiso dentro de la capa de datos.



Para que las diferentes capas de acceso se cumplan, el usuario debe realizar lo siguiente:

- Tener un “login” válido en el servidor.
- Debe ser un “usuario” válido dentro de la base de datos
- Tener “permisos” para utilizar los objetos de la base de datos

3.2 Logines del Sistema

Una cuenta de login del SQL Server es una cuenta que permite a un usuario final conectarse hacia el servidor. Cuando se crea una cuenta de login, se debe especificar lo siguiente:

- Login
- Password

Cuando se instala el servidor por primera vez, se crea el login especial *"sa"*. Por default, el login *"sa"*:

- Se le ha asignado los tres roles especiales System Administrador (SA), System Security Officer (SSO) y el role Server Operator (OPER)
- Es el dueño de la base de datos master
- Ejecuta todos los comandos SQL
- Es tratado como dueño de la base de datos (dbo) en todas las bases de datos
- Tiene acceso a todas las bases de datos y todos los objetos de la base de datos.

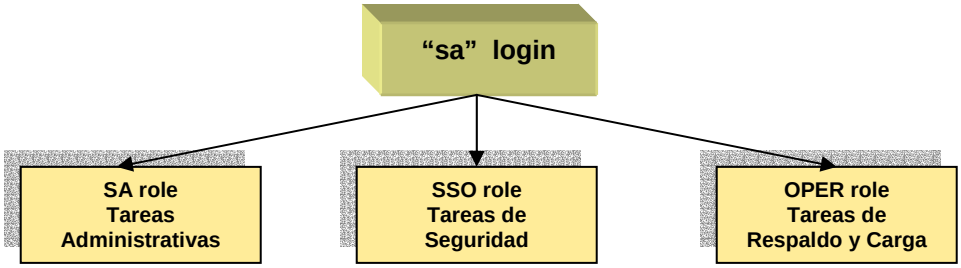
Inicialmente, el password de *"sa"* está definido como nulo, pero para garantizar la seguridad, el password debe ser cambiado antes de otorgar acceso a usuarios al servidor. Una vez cambiado, no puede ser nulo otra vez.

Este login no puede ser eliminado, pero puede ser bloqueado.

3.3 Roles de Sistema

Un role de sistema es un conjunto de privilegios asignados a un login dado, mediante los cuales se pueden realizar las tareas de administración y seguridad necesarias.

Roles Especiales del Login *"sa"*



En la instalación el login "sa" puede hacer todo,

- Puede mantener esos tres roles
- Otorgarlos a uno o más logines y luego bloquear el login *sa*
- Revocar los roles del login *sa*

Administradores del Sistema (Rol SA)

- ❖ Modifica parámetros de configuración
- ❖ Manejo de almacenamiento de disco
- ❖ Crea bases de datos de usuarios y otorga propiedad sobre ellas
- ❖ Borra usuarios de una base de datos
- ❖ Otorga y revoca roles SA hacia otros logines
- ❖ Otorga ciertos permisos a los usuarios del servidor
- ❖ Da de baja al servidor y sus procesos
- ❖ Modifica logines en cuanto a las siguientes opciones: "defdb", "deflanguage", "fullname"
- ❖ Monitorea la recuperación de la base de datos en el arranque del servidor
- ❖ Utiliza ciertas herramientas para el diagnóstico de problemas en el sistema (dbcc)

Oficiales de Seguridad del Sistema (Rol SSO)

- ❖ Crea logines en el servidor y asigna passwords iniciales

- Modifica logines en cuestiones de expiración del password, longitud mínima del password o máximo de intentos fallidos
- Cambia passwords
- Define un intervalo para la expiración del password
- Crea, otorga y revoca roles de usuario
- Otorga autorización para el uso de proxy
- Otorga y revoca roles SSO y OPER hacia otros logines
- Maneja el sistema de auditoría
- Bloquea y desbloquea logines
- Borra logines

Operadores del Servidor (Rol OPER)

- Los operadores pueden respaldar y cargar todas las bases de datos y logs de transacciones en el servidor.
  - Realizan dump db, dump transaction, load db y load transaction.
  - No tienen que ser propietarios de una base de datos para realizar tareas de mantenimiento.
- Los propietarios de base de datos realizan tareas de mantenimiento de sus propias bases de datos en lugar del operador.

3.4 Otorgando Acceso al Servidor

Creación de Logines de Usuario

Una cuenta de "login" es una cuenta que permite a un usuario final conectarse hacia el servidor. Para poder crear la cuenta se utiliza el siguiente procedimiento almacenado:

sp_addlogin

Descripción:

Añade una nueva cuenta de usuario al Adaptive Server.

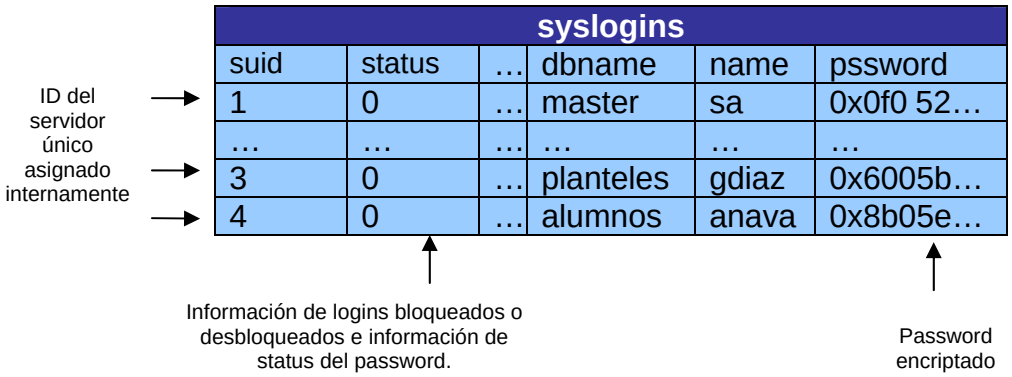
Sintaxis:

sp_addlogin loginame, passwd [, defdb]
[, deflanguage] [, fullname][, passwdexp]
[, minpwdlen] [, maxfailedlogins]

Parámetros:

Parámetro	Significado
loginame	Es el nombre de login de usuario.
passwd	Es el password del usuario. El password debe tener como mínimo una longitud de 6 caracteres.
defdb	Es el nombre de la base de datos asignada a un usuario por default al entrar al servidor.
deflanguage	Es el lenguaje por default asignado para esa cuenta.
fullname	Nombre completo del dueño de la cuenta de login.
passwdexp	Especifica el intervalo en días de la expiración del password
minpwdlen	Especifica la longitud mínima del password requerida para ese login.
maxfailedlogins	Especifica el número de intentos fallidos para una cuenta de login.

La información capturada se almacena en la tabla ***syslogins***, esta tabla se encuentra únicamente en la base de datos master. Esta tabla contiene un registro por cada cuenta de usuario válida.



Descripción de columnas:

Nombre	Descripción
suid	ID de usuario del servidor
status	Status de la cuenta
accdate	Fecha de ultima limpieza de totcpu y totio
totcpu	Tiempo de CPU acumulado por el login
totio	Tiempo de I/O acumulado por el login
spacelimit	Reservado
timelimit	Reservado
resultlimit	Reservado
dbname	Nombre de la base de datos por default en la cual se colocará el usuario al entrar al servidor.
name	Nombre de la cuenta de login
password	Password de la cuenta de login encriptado
language	Lenguaje asignado a esa cuenta
pwdate	Día de la última modificación del password
audflags	Definiciones de auditoría de usuarios
fullname	Nombre completo del usuario
srvname	Nombre del Servidor
logincount	Número de intentos fallidos para una cuenta de login
procid	

Si se requiere ver los detalles de una cuenta de login, es necesario utilizar el procedimiento almacenado ***sp_displaylogin***. Cualquier usuario puede hacer uso de este procedimiento para ver los detalles de su propia cuenta. Únicamente un SA o un SSO pueden ejecutar sp_displaylogin con el *loginame* y sus parámetros *expand*

Sintáxis:

sp_displaylogin [loginame [, expand_up | expand_down]]

Otros procedimientos almacenados que tienen que ver con la información de una cuenta de login son:

sp_modifylogin

Descripción:

Modifica la base de datos predeterminada, el idioma predeterminado o el nombre completo de una cuenta de login del Adaptive Server.

Sintaxis:

sp_modifylogin account, column, value

Parámetros:

Parámetro	Significado
account	Es la cuenta que se va a modificar.
column	Es la opción a la cual vamos a modificar: loginame, passwd, defdb, deflanguage, fullname, passwdexp, minpwdlen, maxfailedlogins
value	Es el valor para una opción especificada.

sp_password

Descripción:

Agrega o cambia una contraseña para una cuenta de login del Adaptive Server.

Sintaxis:

sp_password caller_passwd, new_passwd [, loginame]

Parámetros:

Parámetro	Significado
caller_passwd	Cuando se cambia el password de una cuenta propia, es aquí donde se coloca el password anterior. Cuando un SSO utiliza sp_password para cambiar el password de otra cuenta, utiliza este campo para introducir su propio password.
new_passwd	Es el nuevo password para una cuenta de login
loginame	Es la cuenta de login a la cual se le cambiará el password

Revocando el Acceso al Servidor

sp_droplogin

Descripción:

Omite un login de usuario del Adaptive Server eliminando la entrada del usuario de master.dbo.syslogins.

Sintaxis:

sp_droplogin loginame

Parámetros:

Parámetro	Significado
loginame	Es el nombre de la cuenta de usuario a eliminar.

Únicamente un SA o un SSO pueden ejecutar este procedimiento.

sp_locklogin

Descripción:

Bloquea una cuenta del Adaptive Server para que el usuario no pueda conectarse, o muestra una lista de todas las cuentas bloqueadas.

Sintaxis:

sp_locklogin [loginame, "{lock | unlock}"]

Parámetros:

Parámetro	Significado
loginame	Es la cuenta de login que va a ser bloqueada o desbloqueada
lock | unlock	Especifica si la cuenta va a ser bloqueada o desbloqueada

Otorgando y Revocando Roles de Sistema

Cuando es instalado el Adaptive Server por primera vez, el login *"sa"* tiene los roles SA, SSO y OPER. Para otorgar esos roles hacia otros logines, se utiliza el siguiente procedimiento:

sp_role

Sintaxis:

sp_role {"grant"|"revoke"}, rolename, loginame

donde el parámetro rolename puede ser:

- sa_role
- sso_role
- oper_role

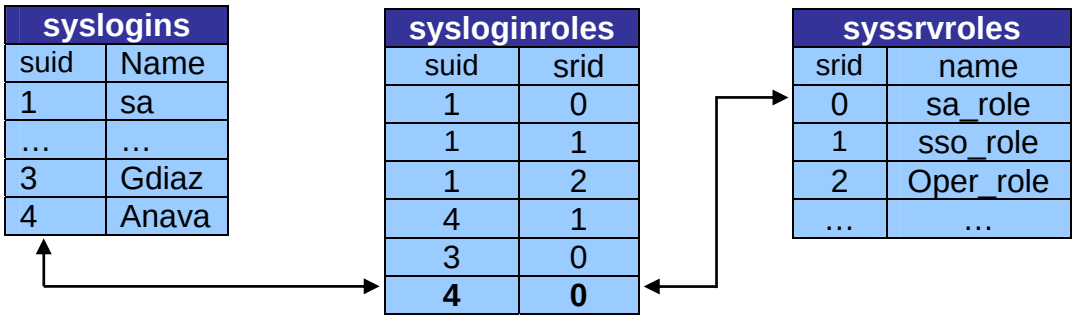
Ejemplo:

sp_role “grant”, sa_role, Alexander

Únicamente un SA puede otorgar y revocar roles SA.

Únicamente un SSO puede otorgar y revocar los roles SSO y OPER.

Al ejecutar este procedimiento se agrega o borra un registro en la tabla sysloginroles la cual se encuentra en la base de datos master.



Donde la tabla ***sysloginroles*** contiene un registro por cada role de sistema asignado a un login de usuario. Cada vez que se otorgue un role a un login, se agregará un nuevo registro en esta tabla.

Descripción de columnas:

Columna	Significado
suid	Es el ID del usuario
srid	Es el ID del role, ▪ 0 = sa_role ▪ 1 = sso_role ▪ 2 = oper_role ▪ 3 = navigator_role ▪ 4 = replication_role
status	Reservado

La tabla ***syssrvroles*** contiene un registro por cada role de sistema o role definido a usuario.

Descripción de columnas:

Columna	Significado
srid	Es el ID del *role*
name	Nombre del *role*
password	Password para el *role* (encriptado)
pwdate	Fecha de última modificación del password
status	Mapeo de bits del status del role (ver tabla A)
logincount	Número de intentos fallidos, envía un 0 por conexión acertada.

Tabla A. Bits de control de status en la tabla sysrvroles

Decimal	Hexadecimal	Status
2	0x2	*Role* bloqueado
4	0x4	*Role* expirado

Habilitando y Deshabilitando Roles de Sistema

Para habilitar o deshabilitar un role se utiliza lo siguiente:

Sintaxis:

Set role role_name {on | off}

Toma efecto inmediatamente y dura para toda la sesión, a menos que se reactive.

Quien puede Modificar los Parámetros de Configuración

Únicamente los Administradores del Sistema (SA), o bien los Oficiales de Seguridad del Sistema (SSO) pueden modificar los parámetros de configuración.

Administradores del Sistema (SA): Pueden modificar todos los parámetros excepto aquellos relacionados a la seguridad del sistema.

Oficiales de Seguridad del Sistema (SSO): Pueden modificar únicamente los parámetros relacionados a la seguridad del sistema.

Parámetros de Seguridad:

- allow procedure grouping
- allow remote access
- allow updates to system tables
- audit queue size
- auditing
- check password for digit
- current audit table
- max roles enabled per user
- maximum failed logins
- minimum password length
- msg confidentiality reqd
- msg integrity reqd
- secure default login
- select on syscomments.text column
- suspend audit when device full
- systemwide password expiration
- unified login required (Windows only)
- use security services (Windows only)

Configuración de Parámetros

sp_configure

Descripción:

Despliega o cambia la configuración de los parámetros del servidor.

Sintaxis:

sp_configure [configname [, configvalue] | group_name | non_unique_parameter_fragment] ["p|P|k|K|m|M|g|G"]]

Parámetros:

sp_configure;

Despliega la configuración de los parámetros del servidor ordenado en grupos, los valores por default, memoria utilizada, valor de la configuración, valor de los parámetros que han sido cambiados mas recientemente, el tipo de parámetro si es dinámico, estático o de sólo lectura. Muestra sólo los parámetros permitidos dependiendo de los *roles* del usuario.

sp_configure ["configname" |"group_name"];

Muestra los valores del parámetro o grupo indicado. En caso de no colocar ningún nombre o grupo, este procedimiento mostrará una lista de todos los parámetros existentes en el servidor organizados por grupos.

Nota: Cualquier usuario puede utilizar este procedimiento.

sp_configure [configname [, configvalue];

Cambia el valor de la configuración del parámetro indicado y despliega el valor por default, la cantidad de memoria utilizada y el tipo de parámetro.

sp_configure configname, 0, “default”;

Regresa a cero el valor de la configuración, asignándole posteriormente el valor por default.

sp_configure non_unique_parameter_fragment ;

Despliega el nombre de todos los parámetros que correspondan a non_unique_parameter_fragment, el valor por default, valor de configuración y la cantidad de memoria usada.

Sintaxis:

sp_configure "configuration file", 0, {"write" | "read" | "verify" | "restore"} " file_name"

Parámetros:

Parámetro	Significado
configuration file	(Incluya las comillas) especifica el parámetro de archivo de configuración.
0	Debe incluirse como segundo parámetro de sp_configure para mantener la compatibilidad hacia atrás.
write	Crea *file_name* a partir de la configuración actual. Si file_name ya existe, se escribe un mensaje en el log de

	errores; el nombre del archivo existente se cambia según la convención file_name.001, file_name.002, y así sucesivamente. Si ha cambiado un parámetro estático pero no ha reiniciado el servidor, *write* da el valor actual en ejecución de ese parámetro.
read	Realiza verificaciones de validación sobre los valores contenidos en *file_name* y lee en el servidor los valores que pasan la validación. Si faltan parámetros en *file_name*, se utilizan los valores actuales de los parámetros ausentes.
verify	Realiza verificaciones de validación sobre los valores contenidos en *file_name*. Esto es útil ya que impide que se intente configurar el servidor con valores inválidos.
restore	Crea un *file_name* con valores de sysconfigures. Esto es útil si se han perdido todas las copias del archivo de configuración y necesita generar una copia nueva.
file_name	Especifica el archivo de configuración que desea utilizar junto con cualquier subcomando.

Tablas sysconfigures y syscurconfigs

El informe mostrado por sp_configure se crea a partir de una tabla llamada spt_values. Esta es una tabla de búsqueda de la base de datos master y contiene registros que hacen referencia a parámetros de configuración, bloqueos, permisos y otra información del sistema.

Las tablas del sistema master..sysconfigures y master..syscurconfigs almacenan parámetros de configuración.

sysconfigures

Descripción de columnas:

Columna	Significado
config	Número del parámetro de configuración.
value	Es el valor del parámetro modificado por el usuario y es de tipo entero. Si el valor de la columna es 0 el parámetro es de tipo caracter.
comment	Nombre del parámetro de configuración.
status	Valor que representa el tipo de parámetro de configuración.
name	Nombre del parámetro de configuración.
parent	Número del parámetro de configuración padre; si existe más de un padre, los números adicionales del padre se almacenan en

	sysattributes.
value2	Es el valor del parámetro modificado por el usuario para el parámetro con tipo de dato caracter. Su valor es NULO para los parámetros con el datatype de número entero.
value3	Almacena el tamaño del wash del buffer pool.
value4	Almacena los porcentajes asincrónicos del prefetch de un buffer pool.

syscurconfigs

Contiene una entrada por cada parámetro de configuración, al igual que *sysconfigures,* pero con los valores actualmente utilizados por el Adaptive Server.

Descripción de columnas:

Columna	Significado
config	Número del parámetro de configuración
value	Es el valor actual para un parámetro de tipo entero. Si el valor de la columna es 0 el parámetro es de tipo caracter.
comment	Cantidad de memoria utilizada por cada parámetro de configuración representado en formato *string*.
status	El estado 1 significa “dinámico”, es decir, que los nuevos valores de estos parámetros tienen efecto inmediatamente cuando se ejecuta sp_configure. El estado 0 indica que el parámetro es estático y se aplica solo después de ejecutar sp_configure y reiniciar el servidor.
value2	El valor actual para un parámetro con tipo de dato caracter. El valor es Nulo para parámetros con tipo de dato entero.
defvalue	Valor por default de un parámetro de configuración.
minimum_value	Valor mínimo para un parámetro de configuración.
maximum_value	Valor máximo para un parámetro de configuración.
memory_used	Valor del número entero para la cantidad de memoria usada por cada parámetro de la configuración.
display_level	Muestra el nivel del parámetro de configuración (los valores son 1, 5, y 10).
data_type	Tipo de dato del parámetro de configuración.
message_num	Número del mensaje del sp_helpconfig para este parámetro de configuración.
apf_percent	

3.5 Configuración de Memoria

Introducción

Al llevar a cabo una instalación inicial de Adaptive Server, la configuración predeterminada del servidor es la mínima para que éste pueda arrancar. Como primer paso, el DBA debe establecer valores adecuados para los principales parámetros de configuración, incluyendo los tamaños del caché de datos ('default data cache') y del caché de procedimientos ('procedure cache').

Al aumentar o disminuir el parámetro 'max memory', Adaptive Server no cambia los tamaños de los cachés de datos o procedimientos (como ocurría al modificar el parámetro 'total memory' de versiones anteriores). A partir de la versión 12.5 Adaptive Server permite que otros parámetros, como 'number of user connections', 'number of devices', 'number of open database' y 'number of locks', que ahora son dinámicos, crezcan dinámicamente dentro del área de memoria asignada a Adaptive Server por el parámetro 'max memory'. La Figura 2 ilustra cómo se asigna la memoria a partir de ASE 12.5.

Los componentes de memoria de alto nivel del Servidor Adaptable incluyen:

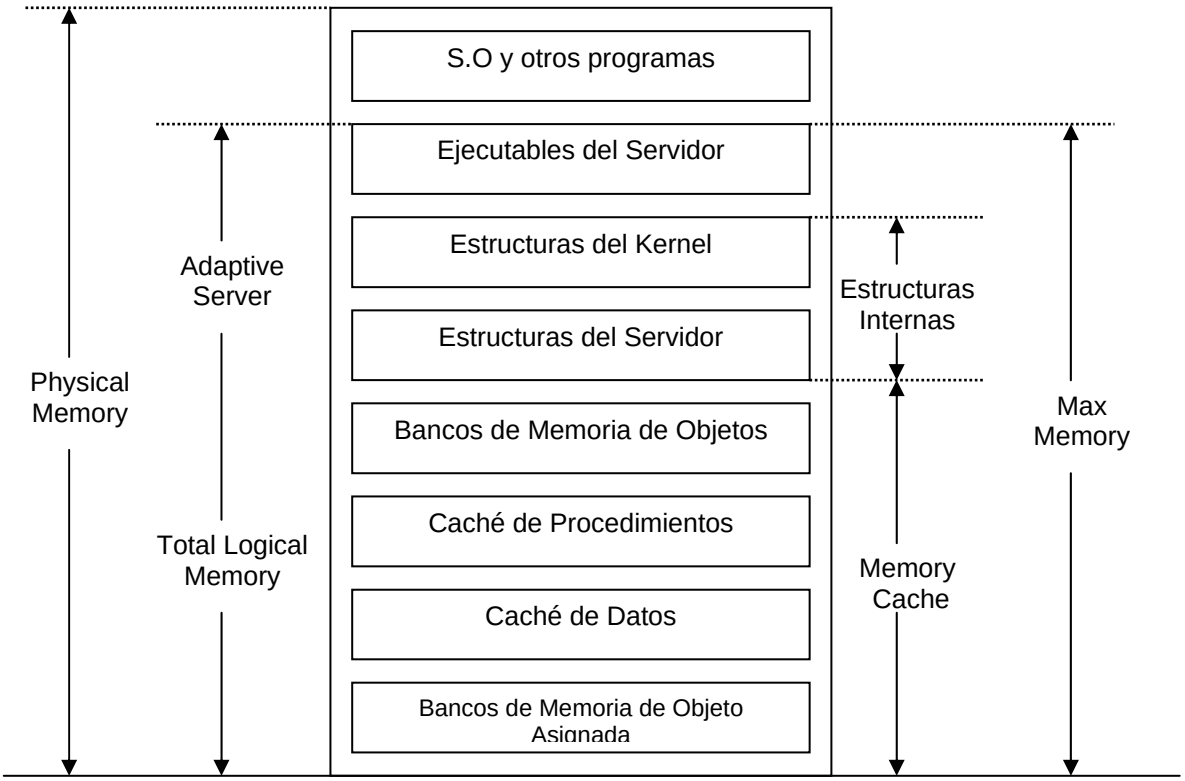


Figura 2. Asignación de Memoria en Adaptive Server 12.5

Nombre del Parámetro	Descripción
max memory	*max memory* es un parámetro de configuración que especifica el tamaño máximo de la memoria física en páginas de 2k que el Adaptive Server puede asignar. Es la memoria compartida que puede ser utilizada por el servidor. Es un parámetro dinámico. Entre más memoria este disponible, será mejor para el rendimiento del Adaptive Server.
total logical memory	Este parámetro muestra la memoria usada por la configuración actual de ASE, pero que puede o no estar siendo usada actualmente. Este parámetro es de sólo lectura.
total physical memory	Este parámetro muestra la cantidad de memoria que ASE está usando en un momento determinado. Este parámetro es de sólo lectura.
caché de datos	El caché de datos (data cache) es un área de memoria utilizada por todas las aplicaciones (datos, índices, páginas de log actualmente en uso por el servidor). Conjunto de buffers que contienen a las páginas. Cuando las páginas no están en uso por el servidor, éstas son almacenadas únicamente en disco.
caché de procedimientos	El caché de procedimientos (procedure cache) es un área de memoria donde los query plan actualmente en uso por el servidor son mantenidos.
estructuras del kernel	Utilizada por el servidor durante una corrida para almacenar información interna. No configurable
estructuras del servidor	Se utiliza para almacenar información acerca de las conexiones de usuario, bases de datos en uso, etc. Es configurable
memoria no asignada	Área dinámica

a) Para determinar el valor de *"max memory"* es necesario:

1. Determine el tamaño de la memoria física de su servidor.
2. Reste de ese valor la memoria requerida para el sistema operativo.
3. Reste la memoria requerida para aplicaciones diferentes a ASE.
4. La memoria restante es la memoria disponible para ASE.

Ejemplo:

Si el servidor tiene una memoria física de 512 MB
Memoria requerida para el sistema operativo y otras aplicaciones 128 MB

Memoria disponible para el Adaptive Server = 512 -128 = 384 MB

Para llevar a cabo la configuración, convertimos ese valor a páginas de 2k

Memoria disponible para el Adaptive Server = 384 * 1024/2= 196608

Utilizaremos el procedimiento sp_configure para definir un máximo de memoria para el servidor de 384 MB (196608 páginas de 2k)

```
sp_configure "max memory", 196608
go
```

b) Estimar los valores adecuados para los principales parámetros de configuración de ASE:

Los parámetros de configuración que más utilizan memoria en ASE son:

- number of user connections
- number of open databases
- number of open indexes
- number of open objects
- number of locks
- number of devices

Estime valores adecuados para éstos y para otros parámetros que consuman memoria y que sea necesario configurar para su sistema.

c) Configure el caché de procedimientos

Al llevar a cabo una instalación inicial de Adaptive Server, el tamaño del caché de procedimientos es mínimo (aproximadamente 7 Mb en ASE 12.5.1). El parámetro 'procedure cache size' determina el tamaño del caché de procedimientos en Adaptive Server 12.5.x.

El tamaño del caché de procedimientos debe ser suficiente para que al menos una copia de cada procedimiento principal resida allí.

Las siguientes fórmulas brindan un rango de tamaños posibles:

Tamaño máximo = (máx. # de usuarios concurrentes) * (tamaño del "query plan" más grande) *1.25

Tamaño mínimo = (# de procedimientos principales) * (tamaño promedio de los "query plan") * 1.25

El tamaño del caché de procedimientos puede estar entre los dos valores anteriores.

Para estimar el tamaño del "query plan" de un procedimiento almacenado particular, usted puede usar la siguiente secuencia de comandos:

```
use base_de_datos
go

select (count(*)/8) + 1
from sysprocedures
where id = object_id("nombre_procedimiento")
go
```

Para estimar el "query plan" más grande en una base de datos usted puede usar la siguiente secuencia de comandos:

```
use base_de_datos
go

select max(count(*)/8 + 1) as "size_in_2K_pages"
from sysprocedures
group by id
go
```

Estas consultas arrojan el resultado en páginas de 2 Kb.

Ejemplo:

Suponiendo que se tienen los siguientes valores:

Número de usuarios concurrentes: 50
“query plan” mas grande: 345 páginas de 2 kb
Número de procedimientos principales: 16
Tamaño promedio de los “query plan”: 220 páginas de 2 kb

Utilizando las fórmulas anteriores:

Tamaño máximo = 50 * 345 * 1.25 = 21562 páginas de 2 Kb

Tamaño mínimo = 16 * 220 * 1.25 = 4400 páginas de 2 Kb

Es decir, el administrador de la base de datos, puede definir el caché de procedimientos dentro de estos dos tamaños.

d) Configure el tamaño del caché de datos del Adaptive Server

Después de llevar a cabo una instalación inicial del Servidor, el tamaño del 'default data cache' es el mínimo para que el servidor pueda arrancar (normalmente unos 8 Mb en Adaptive Server 12.5.1). Ese tamaño usualmente no es suficiente para los requerimientos de procesamiento de un ambiente productivo, así que la memoria restante, después de llevar a cabo la configuración inicial de ASE (pasos 1 al 3), puede ser asignada en su totalidad al 'default data cache' de ASE.

Para determinar la memoria restante usted puede usar la siguiente fórmula:

```
memoria restante = ('max memory' - 'total logical memory') * 0.80
```

El parámetro 'total logical memory' refleja la cantidad de memoria que utiliza la configuración actual de ASE, incluyendo los cachés de datos y procedimientos; este es un parámetro de 'sólo lectura'.

Ejemplo:

Ejecute sp_configure para saber el valor de los parámetros “max memory” y “total logical memory”

```
sp_configure “max memory”
go

sp_configure “total logical memory”
go
```

y observamos que los valores que se obtienen son los siguientes:

“max memory” = 196608 páginas de 2 Kb

“total logical memory” = 36727 páginas de 2 Kb

memoria restante = (196608 - 36727) * 0.80 = 127904,8 páginas de 2 Kb

Para facilitar la configuración, convertimos el valor anterior a Mb:

memoria restante = 127904,8 / 1024 * 2 = 249,8 Mb

Esto quiere decir que hay aproximadamente 250 Mb que se le pueden agregar el 'default data cache' del Servidor.

Una vez calculada la memoria restante, determine el nuevo tamaño del caché de datos usando esta fórmula:

nuevo tamaño = tamaño actual del 'default data cache' + memoria restante

Lleve a cabo la configuración del 'default data cache' con el procedimiento almacenado del sistema sp_cacheconfig.

Ejemplo:

Ejecute el siguiente comando:

```
sp_cacheconfig "default data cache"
go
```

y determine que el tamaño actual del caché de datos es de 8 Mb, entonces:

nuevo tamaño = 8 + 250 = 258 Mb

Para definir el nuevo tamaño del caché, use nuevamente el procedimiento sp_cacheconfig:

```
sp_cacheconfig "default data cache", "258M"
go
```

En Adaptive Server 12.5.0.x esta configuración es estática.

Parámetros Adicionales

Así mismo, en la versión 12.5 se introdujeron dos nuevos parámetros relacionados a la administración de memoria:

Nombre del Parámetro	Descripción
'allocate max shared memory'	Este parámetro determina si ASE asigna toda la memoria especificada por 'max memory' durante el arranque, o sólo la cantidad de memoria que los

	parámetros de configuración requieren. Cuando está en 0 (predeterminado), ASE sólo usa la cantidad de memoria compartida requerida por los parámetros de configuración, que es un valor menor al de 'max memory'. Cuando está en 1, ASE asigna toda la memoria especificada por 'max memory' al arranque. Si el valor es 1, y usted aumenta 'max memory', ASE inmediatamente usa segmentos adicionales de memoria compartida. Esto significa que ASE siempre tiene la memoria requerida para cualquier cambio en la configuración de memoria y no hay degradación del rendimiento causado por el reajuste de la memoria adicional. Sin embargo, si usted no predice el crecimiento de memoria adecuadamente, y 'max memory' está definido con un valor muy alto, puede haber un gran desperdicio de memoria.
'dynamic allocation on demand'	Determina cuándo la memoria es asignada para parámetros dinámicos de configuración de memoria. Cuando está en 1 (predeterminado), la memoria es asignada a medida que es requerida. O sea, si usted cambia el parámetro de configuración 'number of user connections' de 100 a 200, la memoria para cada usuarios es agregada sólo cuando el usuario se conecta al servidor. Si está en 0, toda la memoria requerida para cualquier cambio dinámico de configuración es asignada inmediatamente. Es decir, cuando se cambia 'number of user connections' de 100 a 200, la memoria requerida para las 100 conexiones adicionales será asignada inmediatamente.

En términos generales, 'max memory' es mayor a 'total logical memory', que a su vez es mayor a 'total physical memory'.

3.6 Inicializando Dispositivos

El SQL Server puede tomar algunas decisiones predeterminadas razonables sobre muchos aspectos del manejo del almacenamiento, por ejemplo dónde situar las bases de datos, tablas e índices y cuánto espacio se asigna a cada uno. Sin embargo, el administrador del sistema tiene el control final sobre la asignación de recursos de disco a SQL Server y la ubicación física de las bases de datos, tablas e índices en esos recursos.

Al configurar un nuevo sistema, el administrador del sistema debe considerar aspectos que tienen impacto directo sobre el número y tamaño de los recursos de discos requeridos.

Un dispositivo de bases de datos debe prepararse y darse a conocer a SQL Server antes de que pueda emplearse para el almacenamiento de datos. Este proceso se denomina inicialización.

Una vez que es inicializado un dispositivo de base de datos, es posible:

- Asignarlo al conjunto predeterminado de dispositivos para los comandos create y alter database.
- Asignarlo al conjunto de espacio disponible de una base de datos de usuario.
- Asignarlo a una base de datos de usuario y emplearlo para almacenar uno o más objetos de base de datos.
- Asignarlo para almacenar el log de transacciones de una base de datos.

Uso del Comando Disk Init

El administrador del sistema inicializa los nuevos dispositivos de bases de datos con el comando disk init, el cual realiza lo siguiente:

- Correlaciona el dispositivo de disco físico o archivo del sistema operativo especificados, con un nombre de dispositivo de base de datos.
- Enumera el nuevo dispositivo en master…sysdevices.
- Prepara el dispositivo para el almacenamiento de bases de datos.
- Sólo un administrador del sistema puede ejecutar **disk init**.

Sintaxis:

```
disk init
name="logical_device_name",
physname="physical_name",
[vdevno= virtual_device_number,]
size=[number_of_pages|K|M|G]
[,vstart=virtual_address,
cntrltype= controller_number]
[,dsync={ true | false}]
```

Especificación de un nombre lógico de dispositivo con disk init

El *"logical_device_name"* debe ser un identificador válido, este nombre se utiliza en los comandos create database y alter database y en los procedimientos de sistema que administran segmentos. El nombre lógico de dispositivo es conocido sólo por SQL Server.

Especificación de un nombre físico de dispositivo con disk init

El *"physical_name"* del dispositivo de bases de datos da el nombre de una partición de disco en bruto (Unix) o el nombre de un archivo del sistema operativo.

Elección de un número de dispositivo para disk init

"vdevno" es un número identificador del dispositivo de base de datos, que debe ser exclusivo entre los dispositivos utilizados por el SQL Server.

Especificación del tamaño del dispositivo con disk init

El tamaño (size) es la cantidad de espacio a asignar a una base de datos en bloques de 2k. El tamaño puede ser especificado en las siguientes unidades 'k' o 'K' (kilobytes), 'm' o 'M' (megabytes), y 'g' o 'G' (gigabytes). Las comillas son opcionales se utilizan cuando se da una unidad especifica, si no se define una unidad no hay necesidad de ponerlas.

"vstart" es la dirección virtual inicial, en que el SQL Server empieza a usar el dispositivo de base de datos. *vstart* acepta las siguientes unidades k o K (kilobytes), m o M (megabytes), y g o G (gigabytes).

La palabra clave opcional *cntrltype* especifica el controlador del disco. Su valor predeterminado es 0. Cámbielo sólo si se le indica que lo haga.

Tabla sysdevices

La tabla sysdevices de la base de datos master contiene una fila por cada dispositivo de base de datos y puede contener una fila para cada dispositivo de volcado (cinta, disco, archivo del sistema operativo) disponible para SQL Server.

Descripción de columnas:

Columna	Significado
low	Representa el primer número de página virtual asignada a un dispositivo

high	Representa el último número de página virtual asignada a un dispositivo
status	Es un campo de mapa de bits que indica: el tipo de dispositivo; si se utiliza un dispositivo predeterminado de almacenamiento o de duplicación de disco. (Ver tabla A para bits de estado).
cntrltype	Tipo de controlador (dispositivo de base de datos = 0, dispositivo de volcado de disco = 2 , dispositivos de cinta = 3 – 8)
name	Nombre lógico del dispositivo.
phyname	Nombre físico del dispositivo, es el nombre real del dispositivo en el sistema operativo.
mirrorname	Nombre del dispositivo de duplicación.

Tabla A. Bits de estado

Decimal	Significado
1	Disco predeterminado
2	Disco físico
4	Disco Lógico (no utilizado)
8	Saltar encabezado (usado con dispositivos de volcado de cinta)
16	Dispositivo de volcado
32	Escrituras en serie
64	Dispositivos duplicado
128	Lecturas duplicadas
256	Sólo duplicación del lado secundario
512	Duplicación activada
2048	Uso interno; duplicación desactivada

Obtención de Información sobre los Dispositivos

Mediante el procedimiento del sistema ***sp_helpdevice*** se muestra información sobre los dispositivos incluidos en la tabla sysdevices.

Cuando se utiliza sin un nombre de dispositivo este procedimiento muestra información de todos los dispositivos que hay disponibles en SQL Server. Si se utiliza un nombre de dispositivo mostrará información sobre ese dispositivo en particular.

Sintaxis:

```
sp_helpdevice [logical_device_name]
```

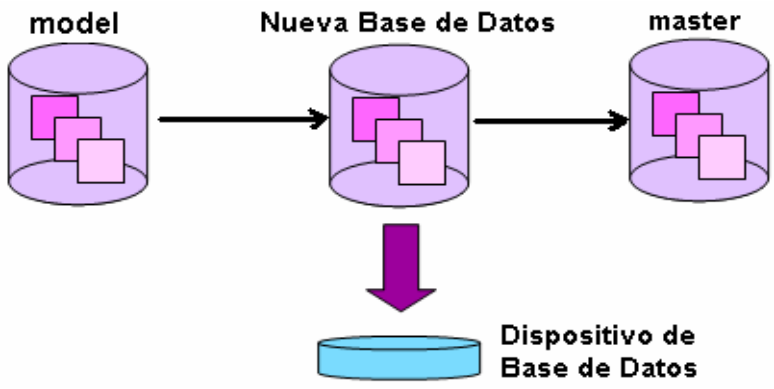
Borrando Dispositivos

Sintaxis:

sp_dropdevice logical_device_name

3.7 Creación y Uso de Bases de Datos

Que ocurre cuando se Crea una Base de Datos



1. El servidor reserva espacio en el dispositivo especificado para los datos y el log de transacciones.

2. El servidor copia todos los objetos dentro de la base model hacia la nueva base de datos.

3. Las opciones de base de datos asociadas a la base model se aplican a la nueva base de datos.

4. Se agregan registros a las siguientes tablas de sistema dentro de master:

- ***sysdatabases***, contiene una fila por cada base de datos en el servidor, en la cual se especifica el nombre de la base de datos, el dueño y además le asigna un ID (dbid) para esa base de datos.

  Descripción de columnas:

Columna	Significado
name	Nombre de la base de datos

dbid	ID de la base de datos
suid	ID de usuario del dueño de la base de datos
status	Bits de control; aquellas opciones que son definidas por un usuario con sp_dboption.
version	Sin usar
logptr	Puntero hacia el log de transacciones
crdate	Fecha de creación
dumptrdate	Fecha del último dump transaction
status2	Bits de control adicional
audflags	Definiciones de la auditoría para la base de datos
deftabaud	Máscara de bit que define las opciones de auditoría por default para tablas
defvwaud	Máscara de bit que define las opciones de auditoría por default para vistas
defpraud	Máscara de bit que define las opciones de auditoría para procedimientos almacenados
def_remote_type	Identifica el tipo de objetos por default que serán utilizados por tablas remotas, si no se provee de un lugar de almacenamiento vía el procedimiento almacenado sp_addobjectdef
status3	Control de bits adicionales
staus4	Control de bits adicionales

- ***sysusages,*** lleva cuenta de todo el espacio asignado a todas las bases de datos del SQL Server. Esta tabla contiene un registro para cada fragmento del dispositivo asignado a cada base de datos, indicando el tamaño y la dirección del comienzo lógico de disco para ese fragmento.

Descripción de columnas:

Columna	Significado
dbid	ID de la base de datos
segmap	Mapeo de bit de posibles asignaciones de segmento
lstart	Primer página (lógica) de la base de datos
size	Número de las páginas (lógicas) contiguas de la base de datos
vstart	Número de inicio de la pagina virtual
pad	Sin uso
unreservedpgs	Espacio libre, que no es parte de un extent asignado

Uso del Comando Create Database

Se utiliza el comando *create database* para crear bases de datos de usuarios. Es necesario tener permisos para ejecutar create database y de ser un usuario válido de master.

Sintaxis:

create database *database_name*
[on {default | *database_device*} [= *size*]
[, *database_device* [= *size*]]...]
[log on *database_device* [= *size*]
[, *database_device* [= *size*]]...]
[with {override | default_location = "pathname"}]
[for {load | proxy_update}]

Es posible controlar diferentes características de la nueva base de datos utilizando las cláusulas create database:

Parámetros:

Parámetro	Significado
database_name	Nombre de la base de datos
on	Especifica los nombres de uno o más dispositivos de base de datos y la asignación de espacio, en megabytes, para cada dispositivo.
log on	Sitúa el log de transacciones (la tabla syslogs) en un dispositivo aparte con el tamaño especificado o el predeterminado.
with override	Permite que el SQL Server en máquinas con espacio limitado mantenga sus logs en fragmentos de dispositivos separados de sus datos. Emplee esta opción sólo cuando vaya a poner el log y los datos en el mismo dispositivo lógico.
for load	Hace que SQL Server salte el paso de borrado de páginas durante la creación de la base de datos.

Tabla syslogs

Esta tabla contiene el log de transacciones.

Descripción de columnas:

Columna	Significado
xactid	ID de la transacción
op	Número de la operación de actualización

Tamaño de una Base de Datos

El tamaño de una base de datos es en megabytes, mínimo debe ser de 2MB. Al estimar el tamaño de una base de datos, considere principalmente:

- tablas
- índices
- log de transacciones

Utilice el procedimiento sp_estspace para estimar el tamaño de las tablas y sus índices.

Sintaxis:

```
sp_estspace table_name, no_of_rows [, fill_factor
    [, cols_to_max [, textbin_len [, iosec] ] ] ]
```

Parámetros:

Parámetro	Significado
table_name	Nombre de la tabla
no_of_rows	Número de filas estimado que contendrá la tabla
fill_factor	
cols_to_max	Es un listado de columnas de longitud variable separadas por comas que utilizan la longitud máxima
textbin_len	Es la longitud, por fila, de todas las columnas de texto e imagen.
iosec	Es el número del disco I/Os por segundo en esta máquina.

Tamaño del Log

El tamaño del log depende de la actividad (tipo y cantidad de transacciones) y la frecuencia de los respaldos.

Un buen punto de partida: 10% al 25% del tamaño global de la base de datos.

- Todos los inserts, deletes y updates son registrados

- Para create index, writetext, truncate table, select into y fast bulk copy, solo la asignación y la desasignación de espacios son registrados.

El log es fácil de extender e imposible de encoger.

Obteniendo Información de las bases de datos

Para hallar los nombres de los dispositivos en que reside una base de datos en particular, se utiliza el procedimiento **sp_helpdb**,

Sintaxis:

sp_helpdb [dbname]

Cuando el procedimiento es seguido de un nombre de base de datos, nos muestra información específica de esa base.

Espacio Utilizado

Para obtener un resumen de la cantidad de espacio de almacenamiento usado por una base de datos, se ejecuta el procedimiento ***sp_spaceused*** en la base de datos

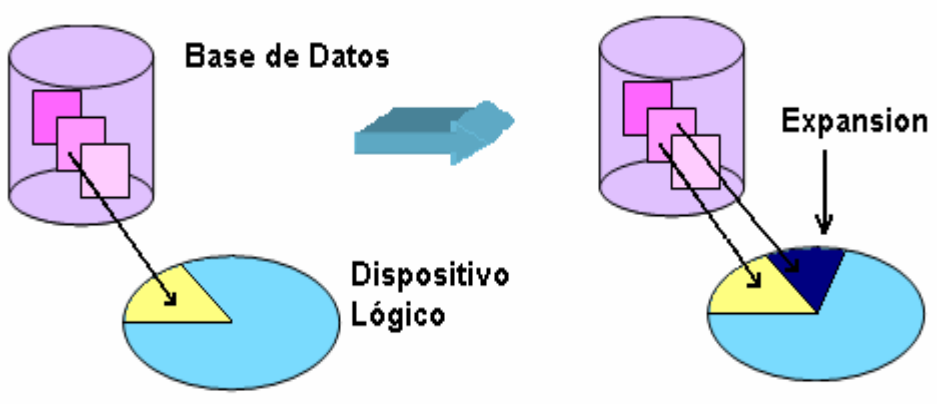
Sintaxis:

sp_spaceused [objname [,1]]

Ejecutando este procedimiento con regularidad se puede saber la cantidad de espacio disponible en la base de datos.

Que Hacer Cuando no se tiene Espacio

Si no se tiene espacio en el log se tiene que truncar. Si no se tiene espacio en los segmentos de datos intente liberar espacio eliminando objetos no usados. Otra alternativa es extender la base de datos (datos y/o log).



Extensión de una Base de Datos

Cuando una base de datos o log de transacciones crece hasta llenar todo el espacio asignado con **create database**, es posible utilizar el comando **alter database** para agregar almacenamiento. El espacio puede añadirse para los objetos de la base de datos, el log de transacciones o ambos.

Sintaxis:

alter database *database_name*
[on {default | *database_device* } [= size]
[, database_device [= *size*]]...]
[log on { default | *database_device* } [= *size*]
[, *database_device* [= *size*]]...]
[with override]
[for load]

Parámetros:

Parámetro	Significado
database_name	Nombre de la base de datos
on	Funciona como la cláusula equivalente del comando **create database**
log on	Funciona como la cláusula equivalente del comando **create database**
with override	Utilice ésta cláusula para crear un fragmento de dispositivo con espacio de log en un dispositivo que ya contiene datos.
for load	Utilice esta cláusula solo después de haber ejecutado **create database for load** para volver a crear la asignación de espacio de la base de datos que se está cargando desde un volcado.

Eliminar una Base de Datos

Utilice el comando drop database para quitar una base de datos del SQL Server, eliminando así la base de datos y todos lo objetos que contenga. Este comando libera espacio de almacenamiento asignado a la base de datos. Elimina las referencias de las tablas de sistema de la base de datos *master* a dicha base de datos.

Sintaxis:

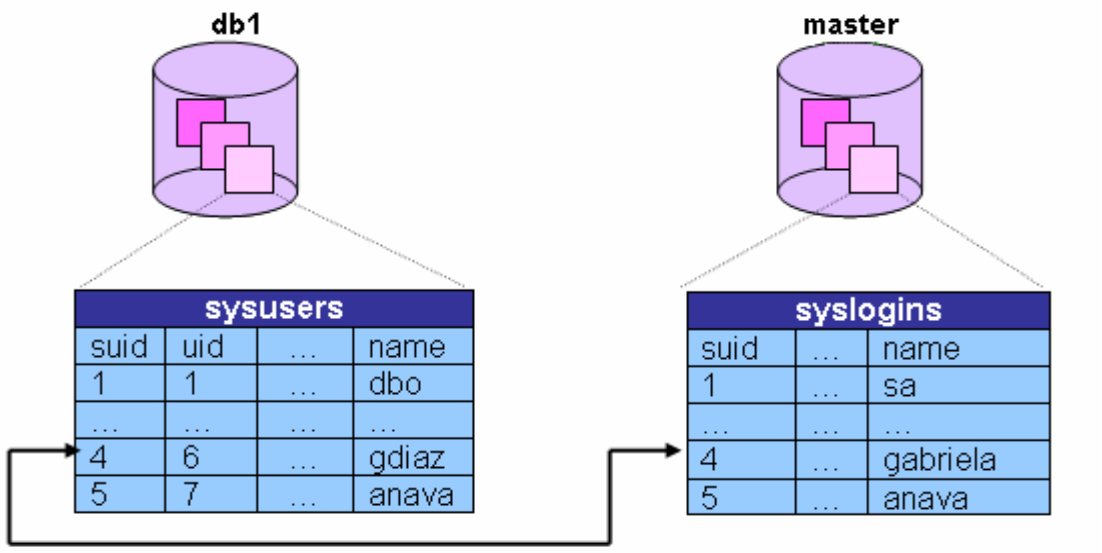
drop database *database_name* [, *database_name*]...

3.7.1 Otorgando el Acceso a la Base de Datos

Creación de usuarios para acceder a la Base de Datos

Para poder acceder a una base de datos, es necesario que el usuario esté dado de alta en esa base de datos en particular.

Los usuarios son listados en la tabla ***sysusers*** en cada base de datos.



Propietarios de Bases de Datos

Un propietario de una base de datos es:

- El creador de una base de datos

- Alguien a quien se le ha transferido la propiedad de una base de datos

Un propietario de base de datos:

- Se le ha otorgado la autorización de crear una base de datos
- Adiciona y retira usuarios de una base de datos con sp_adduser
- Otorga y revoca permisos a usuarios para crear objetos en la base de datos y ejecutar comandos con grant
- Ejecuta algunas tareas de operador de sistema en su propia base de datos
- Ejecuta Checkpoint y dbcc en la base de datos
- Tiene todos los privilegios sobre todos los objetos en la base de datos usando setuser

Agregando Usuarios a la Base de Datos

Para agregar usuarios a una base de datos se utiliza el siguiente procedimiento:

sp_adduser

Sintaxis:

sp_adduser loginame [, name_in_db [, grpname]]

Parámetros:

Parámetro	Significado
loginame	Es el nombre de usuario en master.dbo.syslogins
name_in_db	Es el nuevo nombre para el usuario dentro de la base de datos, si no se especifica uno toma el valor de loginame
grpname	Agrega al usuario a un grupo existente dentro de la base de datos

Sólo el propietario de la base de datos puede agregar usuarios a la base de datos. Los administradores del sistema son dbo automáticamente en cada base de datos que accedan y por lo tanto pueden agregar usuarios también.

Al agregar un usuario se inserta un registro en la tabla sysusers.

Descripción de columnas de la tabla sysusers:

Columna	Significado
suid	Es el ID de usuario del servidor
uid	ID de usuario, único dentro de una base de datos, se utiliza para otorgar o revocar permisos. El ID de usuario 1 es para "dbo"
gid	ID del grupo al que pertenece el usuario. Si uid = gid, esta entrada define a un grupo. El grupo "public" tiene el suid = -2; Todos los otros grupos tienen el suid = -gid.
name	Nombre del usuario o grupo, único en la base de datos.
environ	Reservado.

Para desplegar una lista con información de un usuario o usuarios de una base de datos, se utiliza el procedimiento:

sp_helpuser

Sintaxis:

sp_helpuser [name_in_db]

Parámetros:

Parámetro	Significado
name_in_db	Es el nuevo nombre para el usuario dentro de la base de datos, si no se especifica uno, toma el valor de *loginame*.

Si no se especifica un nombre, nos despliega todos los usuarios dados de alta en la base de datos.

Usuario "guest"

Agregar un usuario llamado *guest* en una base de datos permite a cualquier login acceder a la base como *guest*.

Cuando se instala el servidor, una cuenta de *guest* se crea automáticamente en master y tempdb.

- Esta cuenta no puede ser eliminada.
- Sin embargo para proteger master, los permisos para leer y/o modificar sus tablas pueden ser revocados.

No se puede agregar al servidor un login llamado *guest.*

Alias

Un alias nos sirve para tratar a más de un login como un mismo usuario de una base de datos, tomando todos sus privilegios.

- Hacer que otros logines sean dbo u otro nombre de usuario.
- Si las actividades del alias son auditadas, la identidad real lo será también.

Para crear un alias se utiliza el siguiente procedimiento:

sp_addalias

Sintaxis:

sp_addalias loginame, name_in_db

Parámetros:

Parámetro	Significado
loginame	Es el nombre de login de usuario.
name_in_db	Es el nuevo nombre para el usuario dentro de la base de datos, si no se especifica uno toma el valor de loginame

Este procedimiento agrega un registro a la tabla ***sysalternates*** en la base de datos.

Esta tabla se encuentra en todas las bases de datos, además contiene un renglón para cada *login* con un *alias* hacia un usuario de la base de datos. Cuando un usuario intenta acceder a una base de datos, el servidor busca una entrada válida uid en sysusers. Si ninguno es encontrado, mira en sysalternates.suid. Si suid del usuario es encontrado allí, el login es tratado como el usuario de la base de datos.

Descripción de columnas de la tabla sysalternates:

Columna	Significado
suid	Es el Server user ID del login, al que se le otorga el alias
altsuid	Server user ID, del usuario con el cual estamos entrando a la base

Para borrar el alias de un usuario utilice el procedimiento:

sp_dropalias

Sintaxis:

sp_dropalias loginame

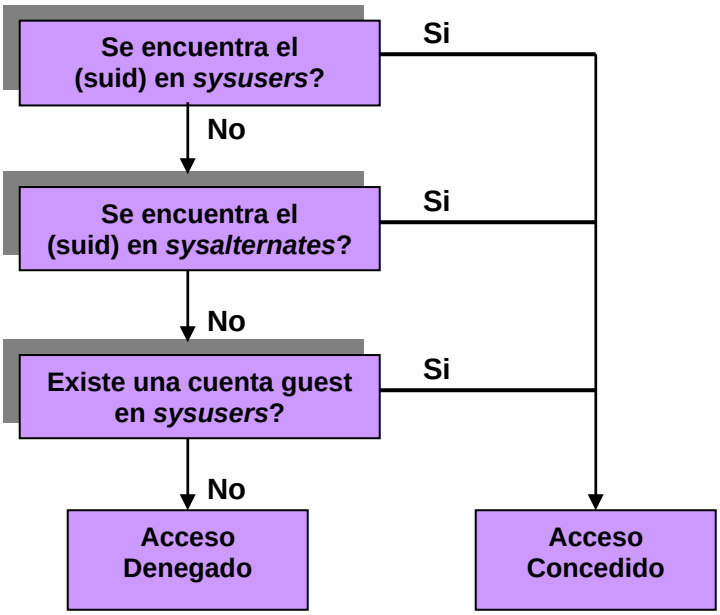
Parámetros:

Parámetro	Significado
loginame	Es el nombre de login de usuario.

Nota: Para borrar el alias se debe estar dentro de la base de datos.

Verificando el Acceso a la Base de Datos

Cuando se intenta acceder a una base de datos, el servidor realiza las siguientes tareas:



Agregando y Asignando Grupos

Un grupo es un nombre colectivo para múltiples usuarios de una base de datos. Cuando se crea un grupo, éste existe únicamente en la base de datos en la cual se creó.

- Los grupos son útiles por que permiten otorgar y revocar permisos a más de un usuario.

Agregar un Grupo

Para agregar un grupo a la base, se utiliza el siguiente procedimiento:

sp_addgroup

Sintaxis:

sp_addgroup group_name

Borrar un Grupo

Para borrar un grupo, se utiliza el siguiente procedimiento:

Sintaxis:

sp_dropgroup group_name

Para asignar o reasignar a un usuario a un grupo se utiliza, sp_changegroup:

sp_changegroup

Sintaxis:

sp_changegroup group_name, username

Desplegar Información de un Grupo

Utilice sp_helpgroup para desplegar información sobre los grupos existentes en la base de datos.

sp_helpgroup

Sintaxis:

sp_helpgroup [grpname]

Sin el parámetro grpname, muestra todos los grupos en la base de datos (incluyendo los *roles* de sistema y de usuario). Si se aplica el parámetro nos regresará información de los usuarios de ese grupo.

Dentro de la tabla ***sysusers*** se tiene un registro para cada grupo.

Borrando Usuarios de la base de datos.

Para borrar usuarios de una base de datos se utiliza el siguiente procedimiento:

sp_dropuser

Sintaxis:

sp_drop name_in_db

Parámetros:

Parámetro	Significado
name_in_db	Es el nuevo nombre para el usuario dentro de la base de datos, si no se especifica uno, toma el valor de loginame

No se puede borrar un usuario de una base de datos, si éste es dueño de objetos en la base de datos. Sin embargo se le puede denegar el acceso bloqueando su login de usuario.

Si "guest" es un usuario en la base de datos, el usuario eliminado puede seguir entrando a la base como guest.

Funciones Útiles

- **Funciones del Sistema:**

suser_id() y suser_name()

Para encontrar el ID de un usuario del servidor o el login.

Para encontrar	Función	Con el argumento
ID del usuario del servidor	suser_id()	(["servidor_usuario_nombre"])
Nombre del usuario en el servidor (login)	suser_name	([servidor_usuario_ID])

Los argumentos para estas funciones de sistema son opcionales. Si no se le indica uno el Servidor muestra la información del usuario.

Este ejemplo muestra como encontrar ID de un usuario en le servidor, en este caso es para el usuario “sandy”

Ejemplo 1:
Select suser_id (“sandy”)

3

Ejemplo 2:

Este ejemplo muestra como un login *sa* "mary” utiliza los comandos sin los argumentos:

Select suser_name(), suser_id()
--- ---
Mary 4

Para encontrar el número de ID de un usuario o el nombre dentro de una base de datos, use user_id y user_name.

Para encontrar	Función	Con el argumento
ID del usuario	user_id	([“db_usuario_nombre”])
Nombre del usuario	User_name	([db_usuario_ID])

El argumento de esta función es opcional. El servidor despliega la información del usuario.

Ejemplo1:

Select user_name(10)
Select user_name( )
Select user_id(“joe”)

db_id()

Devuelve el número de ID de la base de datos. Database_name debe ser una expresión alfanumérica; si es una expresión constante, debe escribirse entre comillas. Si no se proporciona database_name, db_id devuelve la ID de la base de datos actual.

Argumentos:

db_ib ([database_name])

db_name()

Devuelve el nombre de la base de datos. database_id debe ser una expresión numérica. Si no se suministra database_id, db_name devuelve el nombre de la base de datos actual.

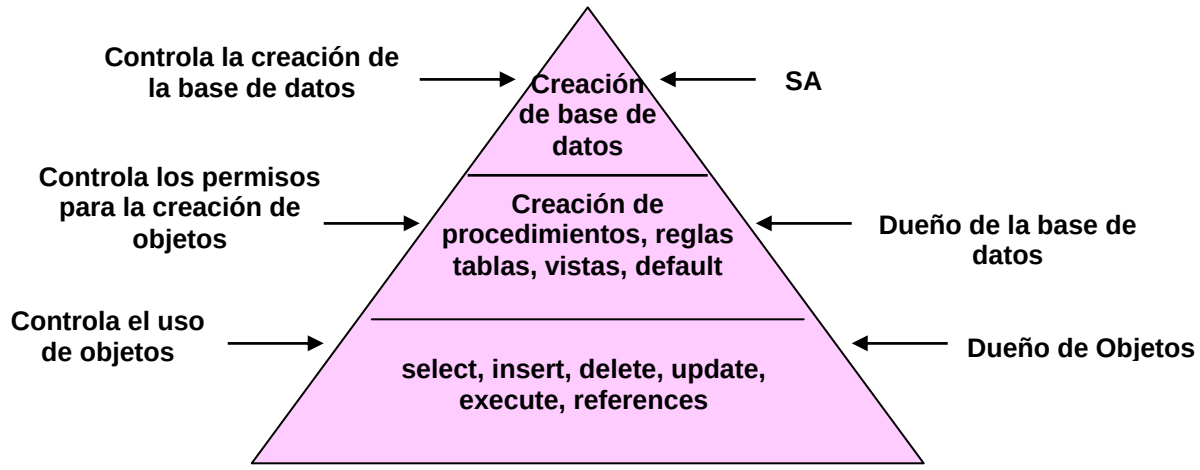
Argumentos:
([database_id])

proc_role()

Verifica si el usuario que ejecuta el procedimiento posee el rol adecuado para hacerlo. Si el usuario tiene el rol preciso, devuelve 1; en caso contrario, 0.

Argumentos:
(“ sa_role “ | “sso_role” | “oper_role”)

3.8 Otorgando y Revocando Privilegios a los Usuarios



Otorgando Privilegios en Comandos

Para dar permisos en comandos, utilice ***grant***:

Sintaxis:

grant {all [privileges] | *command_list*}

to {public | *name_list* | *role_name*}

Los permisos de comandos sobre bases de datos que no pueden ser otorgados son:

- dbcc, dump database, dump tran, load database, load transaction, setuser

Revocando Privilegios en Comandos

Para revocar los permisos en comandos se ejecuta el comando ***revoke***,

Sintaxis:

revoke {all [privileges] | *command_list* }
from {public | *name_list* | *role_name*}

Otorgando Privilegios en Objetos

Utilice el comando ***grant*** para otorgar privilegios sobre objetos específicos,

Sintaxis:

revoke [grant option for]
{all [privileges] | *permission_list*}
on { *table_name* [(*column_list*)]
| *view_name* [(*column_list*)]
| *stored_procedure_name*}
from {public | *name_list* | *role_name*}
[with grant option]

Permisos que no pueden ser otorgados:

- create index, create trigger, alter table, drop table, truncate table, update statistics.

Revocando Privilegios en Objetos

Utilice el comando ***revoke*** para revocar privilegios sobre objetos específicos,

Sintaxis:

revoke [grant option for]

{all [privileges] | *permission_list*}
on { *table_name* [(*column_list*)]
| *view_name* [(*column_list*)]
| *stored_procedure_name*}
from {public | *name_list* | *role_name*}
[cascade]

Mostrando Información Sobre Permisos

Cuando se otorgan privilegios se agregan un registro en la tabla ***sysprotects***,

Descripción de columnas de la tabla sysprotects:

Columna	Significado
id	Es el ID del objeto al cual se le aplica el permiso
uid	Es el ID del usuario, grupo o role, a los cuales se les otorgó permisos

Roles Definidos a Usuarios

Los *roles* definidos a usuario son una colección de permisos:

- Las asignaciones de *roles* en todo el servidor pueden incluir muchos logins.
- Son diferentes de los grupos ya que un grupo es específico de una base de datos y se le asocian ids de usuario.
- Son otorgados a un login específico o a otro *role*.
- Una vez definido los roles pueden ser apagados o encendidos dinámicamente por el usuario, teniendo así mayor flexibilidad con respecto a los grupos.

Cualquier login puede tener muchos roles.

Implementando Roles de Usuarios

Para implementar roles de usuarios, es necesario realizar los siguientes pasos:

1. Un login con privilegios SSO, debe crear el role de usuario, mediante el comando ***create role***,

Sintaxis:

create role *role_name* [with passwd "*password*"
[, {"passwd expiration" | "min passwd length" |
"max failed_logins" } *option_value*]]

Parámetros:

Parámetro	Significado
role_name	Es el nombre del nuevo role
with passwd	Opción para crear un password al role
password	Es el password para ese role
passwd expiration	Intervalo de expiración del password
min passwd length	Longitud mínima
max failed_logins	Máximo de intentos fallidos

2. Después de que el role es creado, cualquier login con el permiso with grant puede otorgar privilegios de acceso a un role de usuario.

3. El login con privilegios SSO puede otorgar a otros logines la membresía en un role de usuario.

4. A los usuarios a los que les fue otorgada la membresía a un role, deben activar explícitamente sus roles para obtener los privilegios asociados.

Agregando y Borrando Passwords

Para agregar un password a un role definido a usuario, se utiliza el siguiente comando:

Sintaxis:

alter role role_name { add passwd *password* / drop passwd }

Otorgando privilegios de Acceso a Roles de Usuario

Un SSO define los roles de usuario por motivos de seguridad. Los roles de usuario son a nivel servidor, y se almacenan en la base de datos master, y están asignados a los logins.

Se pueden otorgar o revocar permisos a los roles de usuario solo desde la base de datos donde residen actualmente los objetos.

Revocando Roles de Usuarios

Para revocar un role de usuario de un login u otro role de usuario.

Sintaxis:

revoke role {*role_name* [*, role_name* ...]} from
{*grantee* [*, grantee* ...]}

Borrando un Rol de Usuario

Sintaxis:

drop role *role_name* [with override]

Si se utiliza la opción override, la sentencia drop quita los permisos otorgados asociados al rol en cada base de datos, o todo el servidor.

Desplegando Información de un Role de Usuario

El procedimiento ***sp_displayroles*** nos ayuda para desplegar información sobre los roles otorgados a un login, roles contenidos por otro role o roles descendientes para un role particular.

sp_displayroles

Sintaxis:

sp_displayroles [grantee_name [, mode]]

Procedimientos Almacenados para Roles de Usuarios

Para ver que roles están activos, utilice:

sp_activeroles

Sintaxis:

sp_activeroles [expand_down]

Para desplegar información de permisos otorgados a un usuario, el cual incluye permisos para un grupo o miembro de un role, utilice ***sp_helprotect***,

sp_helprotect

Sintaxis:

sp_helprotect [name [, username [, "grant"
[, "none"|"granted"|"enabled"| role_name]]]]

3.9 Verificando la Consistencia de los Datos (dbcc)

Aunque la inconsistencia en los datos es rara, pueden llegar a presentarse en las bases de datos del SQL Server. Como administrador del sistema, es necesario encontrar y reparar la inconsistencia de los datos cuando éstas se presenten.

Por ejemplo, las inconsistencias pueden ocurrir en la asignación de la página de una tabla.

- El servidor registra una página según lo asignado a una tabla, pero la página no es una parte de la cadena de la página de la tabla, o viceversa.

Una página no señala a la página anterior o siguiente correcta en la cadena de la página

- Posibles causas,
  - Fallas en el Hardware
  - Fluctuaciones en la energía
  - Recuperación de las base de datos incorrecta

Utilerías para una Apropiada Verificación de Consistencia de los Datos

Las utilerías para la verificación de consistencia de los datos (dbcc) son un conjunto de comandos que verifican la consistencia lógica y física de una base de datos. Existen tres situaciones en las cuales es necesario verificar la consistencia de una base de datos:

1. Cuando se ha descubierto un error específico en una base de datos, utilice:

❖ Comandos que verifican el ligado de la página para los objetos de una base de datos:

- dbcc checktable
- dbcc checkdb
- dbcc checkcatalog

- ❖ los comandos que comprueban la asignación de la página para los objetos de la base de datos

    - dbcc tablealloc
    - dbcc indexalloc
    - dbcc checkalloc

2-3. Como una rutina de validación de una base de datos, o antes de volcar una base de datos, utilice:

- ❖ Comandos que verifiquen la consistencia completa de una base de datos:

    - dbcc checkstorage and dbcc checkverify

Cuando deben Ejecutarse los Comandos dbcc

Se debe verificar la consistencia de la base de datos,

- Antes de volcar una base de datos, para verificar que la base es consistente
- Si el log reporta un mensaje de error, tabla corrupta
- Como parte de un sistema de monitoreo

Compruebe la consistencia cuando se tenga el menor impacto sobre los usuarios.

Sólo un SA o role_sa puede ejecutar comandos dbcc, o bien otorgar o revocar los permisos de ejecución de algunos comandos dbcc hacia otros usuarios.

dbcc checktable

Comprueba la tabla especificada para ver que las páginas del índice y de los datos están ligadas correctamente, que los índices están en orden correctamente clasificados, que todos los indicadores son constantes, que la información de los datos en cada página es razonable, y que las compensaciones de la página son razonables. Si el segmento de log está en su propio dispositivo, ejecuta dbcc checktable sobre la tabla syslogs y hace un reporte del espacio libre y el espacio utilizado.

Sintaxis:

dbcc checktable ({table_name | table_id} [, skip_ncindex])

dbcc checkdb

Ejecuta la misma comprobación que checktable, pero sobre cada tabla, incluyendo a syslogs, sobre una base de datos especifica. Si no se especifica una base, la verificación la realiza sobre la base actual.

Sintaxis:

dbcc checkdb [(database_name [, skip_ncindex])]

dbcc checkcatalog

Ejecutar este comando sobre una base de datos. Si no se especifica una base, la verificación la realiza sobre la base donde se este parado en ese momento.
Verifica problemas de integridad referencial entre las tablas de sistema dentro de la base de datos.

Sintaxis:

dbcc checkcatalog [(database_name)]

dbcc tablealloc

Se ejecuta sobre una tabla específica. Comprueba que todas las páginas de la tabla y sus respectivos índices estén correctamente asignados.

Ésta es una versión más pequeña del checkalloc, proporcionando las mismas comprobaciones de la integridad en una tabla individual. Puede ser utilizada con (el nombre de la tabla o la identificación del objeto de la tabla la columna de la identificación de sysobjects).

Pueden generarse tres tipos diferentes de reportes con tablealloc: full, optimized y fast. El valor por default es optimized.

Sintaxis:

dbcc tablealloc ({*table_name* | *table_id*}
[, {full | optimized | fast | null}
[, fix | nofix]])|

Parámetros:

Parámetro	Significado
Full	Es equivalente al checkalloc a un nivel de tabla; reporta todos los tipos de errores de asignación.
Optimized	Produce un reporte basado en un listado de asignación de páginas en la tabla OAM.
Fase	Produce un reporte de excepción de páginas que están referenciadas pero no asignadas sobre el extent.
fix | nofix	Determina si tablealloc arreglo o no los errores de asignación encontrados en la tabla. El valor predeterminado es fix para todas las tablas a excepción de las tablas de sistema, para las cuales el valor predeterminado es nofix. Para utilizar esta opción con las tablas de sistema, primero debe ponerse la base en modo single user.

dbcc indexalloc

Se ejecuta sobre un índice específico. Para ese índice realiza lo siguiente:

Comprueba que todas las páginas del índice se asignen correctamente. Es una versión corta de checkalloc provee la misma comprobación de integridad en un índice individual. Incluye las mismas opciones de comandos que dbcc tablealloc

Sintaxis:

dbcc indexalloc ({*table_name* | *table_id*}, index_id
[, {full | optimized | fast | null}
[, fix | nofix]])

dbcc checkalloc

Se ejecuta sobre una base de datos. Este comando ejecuta un dbcc tablealloc sobre cada tabla dentro de la base de datos.

Sintaxis:

dbcc checkalloc [(database_*name* [, fix | nofix])]

dbcc checkstorage

El comando dbcc checkstorage identifica los problemas de:

- Bajo performance en el chequeo de consistencia.
- Informa de errores esporádicos

A diferencia de los otros comandos checkstorage, requiere de una base de datos especial, ejecuta en paralelo utilizando worker processes, minimiza el bloqueo de tablas durante el proceso, la integridad se reporta en una base de datos en lugar de un archivo de salida.

Sintaxis:

dbcc checkstorage [(*database_name*)]

dbcc checkverify

Verifica los resultados de la ejecución más reciente del comando dbcc checkstorage para la base de datos especificada.

Sintaxis:

dbcc checkverify [(*database_name*)]

3.10 Herramientas de Auditoría de la Base de Datos

La responsabilidad es un elemento primordial en la seguridad de un sistema. Una forma de asegurar esa responsabilidad es auditando los eventos sobre el sistema. Muchos de los eventos que ocurren en el Servidor pueden ser registrados.

La auditoría es una parte importante de la seguridad dentro de la administración del sistema de base de datos. Auditar puede ser útil para detectar la penetración del sistema y el uso erróneo de los recursos. Al examinar la auditoría, un oficial de seguridad del sistema puede vigilar el acceso a los objetos dentro de la base de datos, puede también monitorear la actividad de usuarios específicos.

El sistema de auditoría puede actuar como un impedimento hacia aquellos usuarios que pretenden dar un mal uso al sistema.

¿Que puede ser Auditado?

Las actividades que pueden ser auditadas son:

A nivel Servidor:

- logins
- logouts
- reboots
- remote procedure calls
- dbcc commands
- security
- disk
- adhoc

A nivel de Base de datos:

- grant
- revoke
- truncate
- drop
- load
- create
- alter
- bcp
- bind
- unbind
- dbaccess
- dump

A nivel Objetos:

- delete
- exec_procedure
- exec_trigger
- duna_obj_access
- insert
- referente
- select
- update

A nivel Usuario:

- cmdtext

3.10.1 El Sistema de Auditoría

El sistema de auditoría consiste de:

- De la base de datos ***sybsecurity***, la cual contiene las opciones globales de auditoría y el rastro de la auditoría.
- La configuración de parámetros para la administración de auditoría.
- Procedimientos de sistema para la administración de auditoría.

La Base de Datos *sybsecurity*

Esta base de datos es esencial para la auditoría del Adaptive Server. Se crea durante el proceso de instalación de la auditoría. En esta base se tienen las siguientes tablas de sistema,

Tabla de sistema	Contenido
sysauditoptions	Contiene una fila por cada opción de auditoría
sysaudits_01 – sysaudits_08	Contiene el registro de auditoría

sysauditoptions

Descripción: La tabla sysauditoptions contiene un registro por cada opción de auditoría server-wide e indica la opción definida actualmente. Otros tipos de opciones de auditoría y los settings son almacenados en otras tablas. Por ejemplo, el setting de una opción para una base de datos específica se almacena en sysdatabases y el setting de opción para un objeto específico se almacena en sysobjects. El valor predeterminado para cada opción es 0 u "off". Solo un oficial de seguridad de sistema puede entrar a esta tabla.

Descripción de columnas de la tabla sysauditoptions:

Columna	Significado
num	Número de la opción server-wide.
val	Valor actual; uno de los siguientes: 0 = off 1 = pass 2 = fail 3 = on
minval	Valor mínimo válido para esta opción.
maxval	Valor máximo válido para esta opción.
name	Nombre de la opción.
sval	Cadena equivalente del valor actual: por ejemplo, "on", "off", "nonfatal".
comment	Descripción de la opción.

sysaudits_01 – sysaudits_08

Descripción: Estas tablas de sistema contienen el registro de auditoría. Solamente una tabla a la vez puede estar activa. La tabla activa es determinada por el valor del parámetro de configuración actual de la tabla de auditoría.

Columna	Significado
event	Tipo de evento que está siendo auditado
eventmod	Información adicional acerca del evento auditado. Los valores posibles son: 0 = ninguna modificación para este evento

	1 = el evento pasó 2 = el evento falló
spid	ID, del proceso que ocasionó el evento.
eventtime	Fecha y hora del evento.
sequence	Secuencia de un registro para un evento con múltiples registros.
suid	ID, del login que causó el evento.
dbid	ID de la base de datos en donde ocurrió el evento o en dónde reside el evento auditado.
objid	ID del objeto auditado.
xactid	ID de la transacción que causó el evento.
loginname	Es el nombre del login correspondiente al suid.
dbname	Es el nombre de la base de datos correspondiente al dbid.
objname	Nombre del objeto correspondiente al objid.
objowner	Nombre del dueño del objid.
extrainfo	Información adicional del evento auditado. Este campo contiene una serie de puntos separados por puntos y comas. (Ver anexo2)

La columna extrainfo contiene una secuencia de puntos separados por puntos y comas como se muestra en la siguiente tabla.

Tabla 3.1: Puntos de la columna extrainfo

Item	Contenido
Roles	Listado de roles que se encuentran activos.
Subcommand	Es el nombre de una opción de comando o subcomando que fue utilizada por ese evento. Por ejemplo, para el comando "**alter table**", las opciones "add column" o "drop constraint" pueden ser usadas. Múltiples opciones o comandos son separados por comas.
Previous value	El valor antes de la actualización si el evento dio lugar a la actualización de un valor.
Current value	El nuevo valor si el acontecimiento dio lugar a la actualización de un valor.
Other information	Información adicional de seguridad relevante que se registra para el evento
Proxy information	El nombre original del login, si ocurrió el evento mientras que un set proxy estaba en efecto.
Principal information	El nombre principal del mecanismo subyacente de la seguridad, si la conexión segura del defecto del usuario, y el usuario registrado en el servidor adaptante vía unificado de este campo es NULL, si la conexión segura del defecto no se está utilizando.

Definiendo Opciones de Auditoría

Una vez instalada la auditoría, se pueden definir las opciones de auditoría utilizando el siguiente procedimiento:

sp_audit

Sintaxis:

sp_audit option, login_name, object_name [, setting]

Nota: Sólo un oficial de seguridad del sistema puede configurar las opciones de auditoría.

Parámetros:

Parámetro	Significado
option	Es el nombre de la opción de la auditoría a fijar. En la tabla 3.2 se enlistan las opciones de auditoría válidos.

Tabla 3.2 Opciones de Auditoría

Opción	Descripción
adhoc	Permite que los usuarios utilicen sp_addauditrecord para agregar sus propios registros definidos por el usuario de auditoría al rastro de la auditoría.
all	Audita todas las acciones realizadas por un usuario particular o por usuarios con un role particular.
alter	Audita la ejecución de los comandos “alter table” o “alter database”
bcp	Audita la ejecución de la utilería bcp.
bind	Audita la ejecución de los procedimientos del sistema sp_bindefault, sp_bindmsg y sp_bindrule.
cmdtext	Audita todos los comandos ejecutados por el usuario
create	Audita la creación de objetos en base de datos.
dbaccess	Audita el acceso a bases de datos actuales desde otra base de datos.
dbcc	Audita la ejecución de cualquier comando dbcc.
delete	Audita el borrado de filas desde una tabla o vista.
disk	Audita la ejecución de disk init, disk refit, disk reinit, disk mirror, disk unmirror y disk remirror.
drop	Audita el borrado de objetos de una base de datos.
dump	Audita la ejecución de los comandos dump database o dump transaction.

errors	Audita errores, sean fatales o no.
exec_procedure	Audita la ejecución de un procedimiento almacenado.
exec_trigger	Audita la ejecución de un trigger.
func_dbacces	Audita el acceso hacia una base de datos vía funciones de Transact-SQL.
func_obj_access	Audita el acceso hacia objetos de una base de datos vía funciones de Transact-SQL.
grant	Audita la ejecución del comando grant.
insert	Audita la inserción de filas dentro de tablas o vistas.
load	Audita la ejecución de los comandos load database o load transaction.
login	Audita todas las entradas al Adaptive Server.
logout	Audita todas las salidas al Adaptive Server.
reference	Audita referencias entre tablas.
revoke	Audita la ejecución del comando revoke.
rpc	Audita la ejecución de llamadas a procedimientos remotos.

security	Audita los siguientes eventos security-relevant:
	• Levantar o dar de baja el servidor
	• Activar o desactivar un role
	• Utilizar cualquiera de los siguientes comandos
	  • connect
	  • kill
	  • online database
	  • set proxy
	  • set session authorization
	  • sp_configure
	• utilizar cualquiera de las siguientes funciones:
	  • valid_user
	  • proc_role(desde cualquier procedimiento de sistema)
	• Regenerar los passwords de SSO

Opción	Descripción
select	Audita la ejecución del comando select.
setuser	Audita la ejecución del comando setuser.
table_access	Audita el acceso de un usuario específico a cualquier tabla.
truncate	Audita la ejecución del comando truncate table.
unbind	Audita la ejecución de los procedimientos de sistema sp_unbindrule, sp_unbindmsg y sp_unbindefault.
update	Audita las actualizaciones de filas en una tabla o vista.
view_access	Audita el acceso de un usuario específico a cualquier vista.

Parámetros:

Parámetro	Significado
login_name	Es el nombre de un login específico que será auditado. Para auditar todos los logins, especifique all en el parámetro ***option***. Si se especifica la opción all, se puede definir el parámetro ***login_name*** hacia un role de sistema específico para auditar todas las acciones de los usuarios que tengan activado ese role.
object_name	Es el nombre del objeto que será auditado. Los valores válidos, dependen del valor que se especifique en ***option***, son: • El nombre del objeto, incluyendo el nombre del dueño, si no es el dueño del objeto. • All para todos • Default table, default view, default procedure o default trigger para auditar el acceso a cualquier nueva tabla, vista, procedimiento o trigger.
setting	Es el nivel de auditoría. Si no se especifica un valor para *setting*. El Adaptive Server muestra la opción de setting actual. Los valores válidos para setting se describen a continuación en la siguiente tabla.

Valores para nivel (setting)

Setting value	Description
on	Activa la auditoría para la opción especificada. Adaptive Server genera los registros de auditoría para controlar los eventos definidos.
off	Desactiva la auditoría.
pass	para eventos que pasen la verificación de permisos
fail	para eventos que fallen la verificación de permisos

Si se especifica el paso para una opción y después falla la especificación para la misma opción, o viceversa, el resultado es equivalente a especificar on. El Adaptive Server genera los registros de la auditoría a pesar de que los eventos pasen o fallen en los permisos de controles. Los settings de on u off se aplican a todas las opciones que auditan. Los settings de paso y falla aplican a todas las opciones excepto errores y adhoc. Para estas opciones es off.

Parámetros que configuran el control de la auditoría

Parámetro de configuración	Efecto
auditing	Habilita o deshabilita la auditoría en el servidor.
audit_queue_size	Establece el tamaño de la auditoría.
current_audit_table	Define la tabla de auditoría actual.

suspend_auditing_when_full	Desactiva el sistema de auditoría si el espacio en *sybsecurity* se llegara a agotar.

3.11 Herramientas

3.11.1 Sybase

En 1984 se fundó Sybase, en un mercado donde Oracle, Ingress, Informix y DB2 eran productos RDBMS. Sybase es un Sistema Manejador de Bases de Datos Relacionales (RDBMS), Sybase fue el primero en incorporar la arquitectura cliente/servidor dentro de una base de datos relacional y el primero en promover el concepto de un nuevo camino a construir sistemas.

Plataformas Soportadas:

- Compaq Tru64
- HP/UX (32 y 64 bit)
- IBM AIX(32 y 64 bit)
- Microsoft Windows NT
- Microsoft Windows 98 (solo clientes)
- SGI IRIX (32 y 64 bit)
- Sun Solaris (32 y 64 bit)
- Linux

Características:

Especificaciones	**Número**
Bases de datos en Sybase	32,767
Tablas en un query	64
Logins por servidor	2,147,516,416
Usuarios por base de datos	2,146,484,223
Grupos por base de datos	1,032,193
	Tamaño
Servidor	8 Terabytes
Base de datos	4 Terabytes
Páginas	2k, 4k, 8k ,16k

- Operaciones dinámicas de respaldo continuo.
- Triggers o disparadores y procedimientos almacenados en al base de datos.
- Escalabilidad y flexibilidad
- Ejecución de reglas del negocio e integridad referencial.
- Soporte a tipo de datos definidos por el usuario.
- Soporte a terceras egresas con productos front-end.

- Conectividad con diversos lenguajes de programación a través de librerías.
- Consultas en línea.
- Procesador de transacciones.
- Consultas en línea.
- Procesador de transacciones.
- Arquitectura cliente/servidor.
- Lenguaje: structured query language.

Sybase Transact-SQL ha sido extendido para minimizar la complejidad de programación de alguna tarea deseada. Transact-SQL contiene muchas versiones comerciales de SQL, y añade características como lenguaje de control de flujo, triggers o disparadores, reglas y defaults.

Lenguaje de control de flujo puede ser utilizado como parte de una sentencia SQL o un conjunto de instrucciones. Algunas sentencias son:
Begin… end, break, continue, declare, goto, if-else, return y while.
Existen también técnicas especiales para el manejo de errores que están disponibles al programador de Transact-SQL.

3.11.2 ISQL

ISQL (Interactive SQL) es una interfaz interactiva a las bases de datos del gestor Sybase SQL. Permite ejecutar comandos del lenguaje de programación Transact-SQL, tales como consultas de bases de datos, comandos de mantenimiento y administración. ISQL es un programa que se encuentra en el cliente.

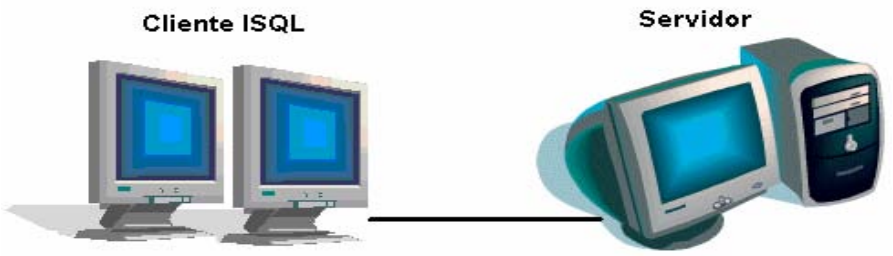


Figura 3.11.1 ISQL como cliente

3.11.3 SQSH

SQSH (SQL Shell), es una herramienta que mejora el funcionamiento del programa ISQL provisto por Sybase. Este programa surge después de años de trabajo y de susceptibilidades en el programa ISQL, intenta hacer un trabajo más rápido y eficiente.

SQSH es más que un bonito promt, intenta proporcionar la funcionalidad de un buen shell, por ejemplo el uso de variables, redireccionamiento de información, control de trabajos, historial de comandos.

SQSH fue diseñado con portabilidad y se ha compilado con éxito en las principales plataformas UNIX soportadas por Sybase, por ejemplo Linux, FreeBSD, HP-UX, AIX, IRIX, SunOS, Solaris, Dynix, OSF/1, DEC Unix, SCO, NeXT, DG/UX y CP/M. SQSH también se ha compilado para plataformas de Microsoft Windows NT.

3.114 Compilador GCC

Las siglas GCC significan GNU Compiler Collection (Colección de compiladores GNU). Antes eran siglas de GNU C Compiler (Compilador C GNU). Como su nombre indica es una colección de compiladores y admite diversos lenguajes: C, C++, Objetive C, Chill, Fortran y Java.

El compilador se distribuye bajo la licencia GPL (General Public License) lo que lo hace de libre distribución: se pueden hacer copias de él y regalarlas o venderlas siempre que se incluya el código fuente (o se indique como conseguirlo) y se mantenga la licencia.

Existen versiones para prácticamente todos los sistemas operativos. Viene incluido en la mayoría (si no en todas) las distribuciones de GNU/Linux. La versión DOS de este compilador es el DJGPP.

De las herramientas anteriormente descritas, empleamos a Sybase como manejador de base de datos, y el cliente predeterminado será SQSH por la versatilidad en las funciones que presenta, para compilar los programas hacemos uso del compilador GCC porque reúne características que permiten compilar programas desarrollados en lenguaje C y GTK+, además de que se distribuye de forma gratuita.

CAPITULO 4. DESARROLLO DE UNA HERRAMIENTA DE AUTOMATIZACIÓN DE AUDITORÍA

La creación de esta herramienta auditora tiene como fin facilitar la protección de los recursos y permitir diagnosticar los servidores de bases de datos, para incrementar la seguridad dentro de los mismos y su correcta configuración.

La auditoría a los servidores de bases de datos debe realizarse periódicamente por el administrador del sistema, para conocer el estado de los mismos y para comprobar que siguen cumpliendo con los requisitos del sistema. Si los resultados de la auditoría ponen en duda la validez de los datos, el administrador debe adoptar inmediatamente las acciones correctivas oportunas.

El DBA debe mantener un registro de las bases de datos auditadas, de los resultados de la auditoría y de las acciones correctivas que se hayan derivado de la misma. También debe documentar en procedimientos sus políticas, sistemas, programas, e instrucciones en la medida en que sea necesario.

Para el desarrollo de nuestra herramienta auditora se requirió de los conceptos y de las herramientas que se describieron en los capítulos anteriores.

El programa fue desarrollado con lenguaje C y las bibliotecas DB-Library, en Solaris 8^{+} y el compilador gcc 3.2.

4.1 Desarrollo de la Herramienta Auditora

A continuación se mostrarán las pantallas más representativas del programa.

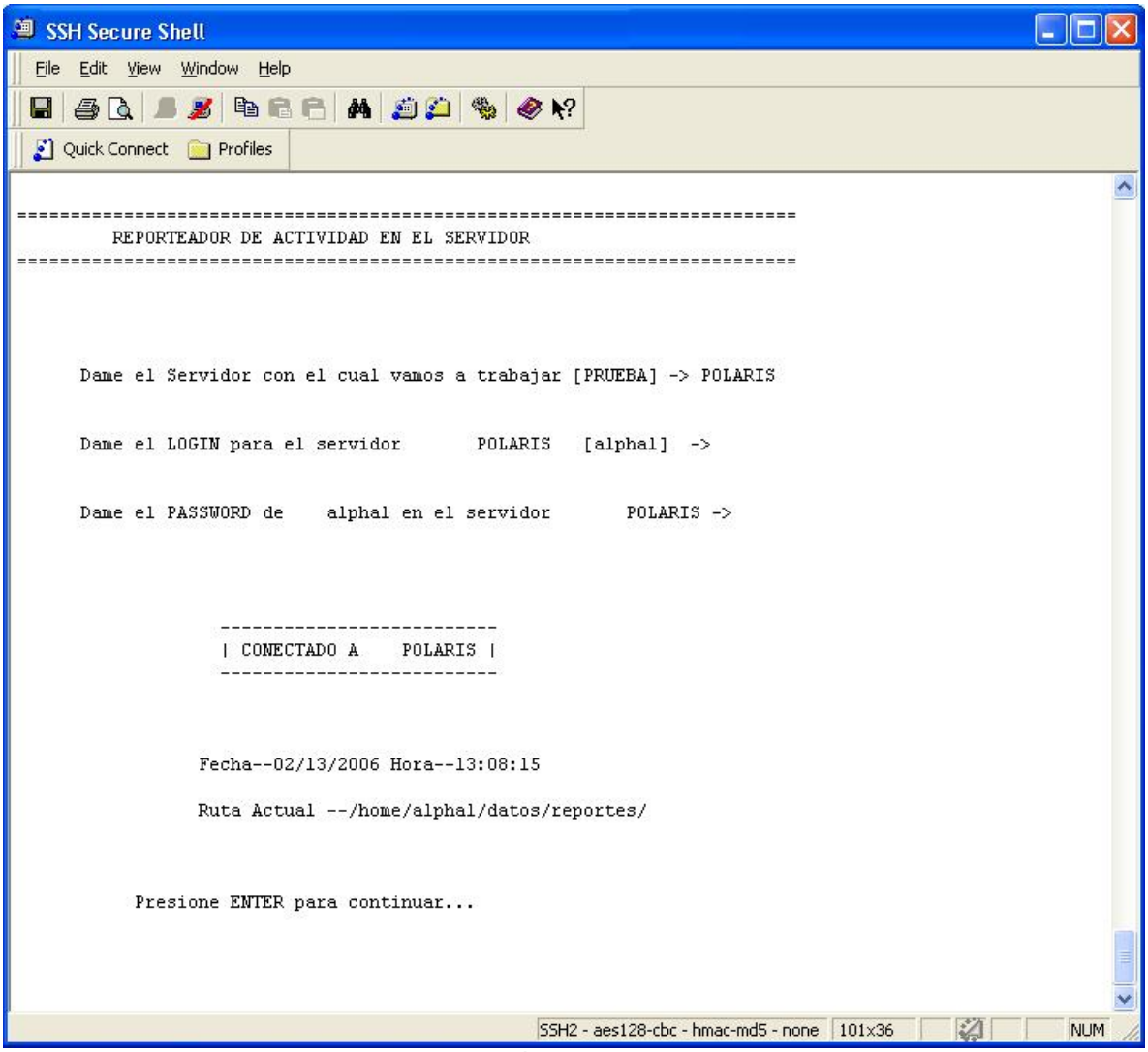


Figura 4.1 Ejecución de la herramienta auditora

La Figura 4.1 muestra la ventana principal de la herramienta auditora, como primer parámetro pide el nombre del servidor con el cual va a trabajar, en caso de no dar un nombre toma por default el servidor PRUEBA. Los siguientes parámetros de entrada que solicitará es el login y password del usuario que se conectará al servidor de base de datos, el usuario debe ser un "login" válido dentro del servidor y contar con los permisos necesarios para ejecutar este programa. Si al dar la información no existe ningún problema, el programa enviará un mensaje que el programa está conectado al servidor "X". Dará la fecha y hora en la que se ejecutó el programa, además de dar la ruta actual y espera que el usuario presione la tecla *enter* para continuar.

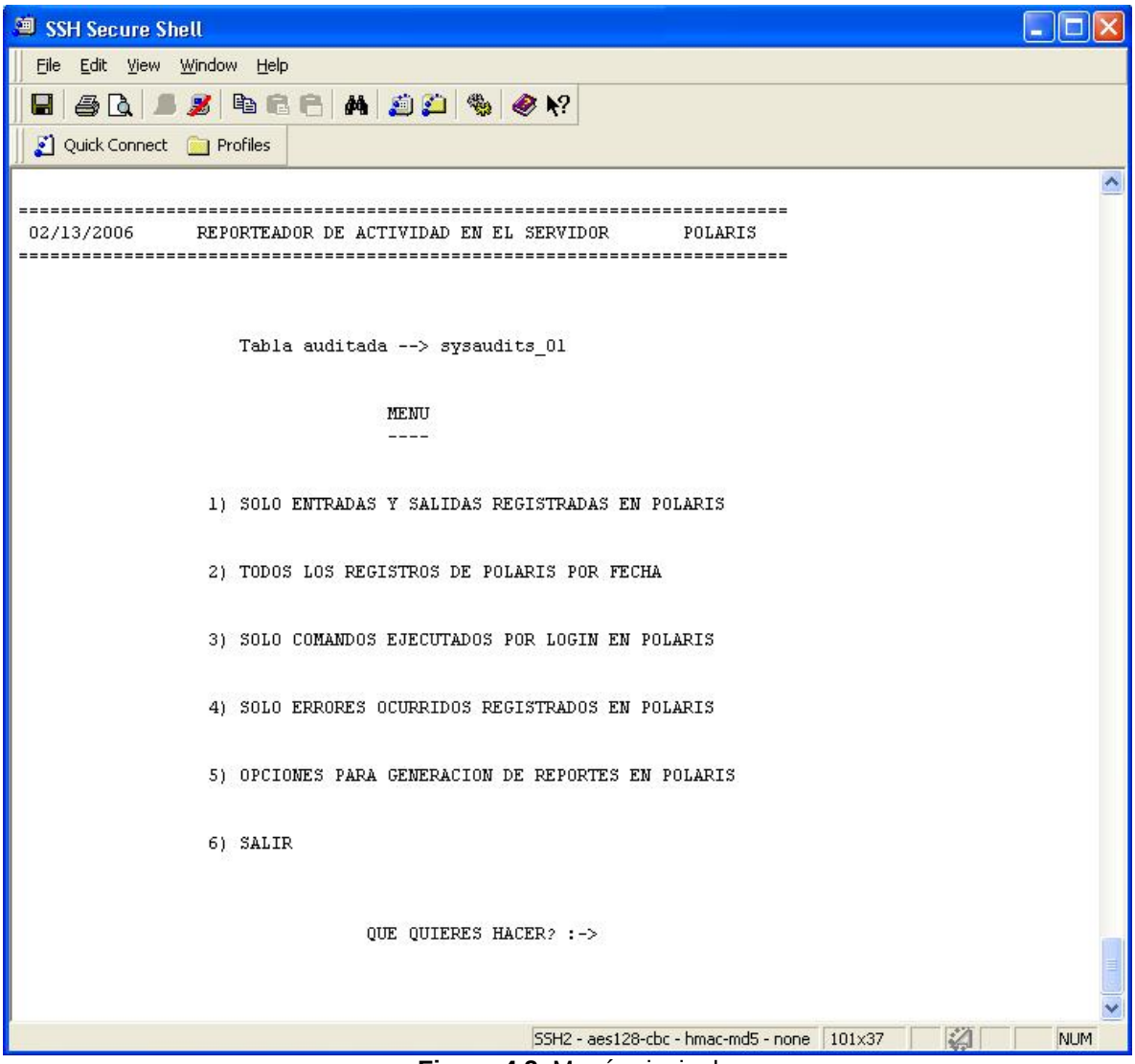


Figura 4.2 Menú principal

Cuando se tiene éxito en la conexión al servidor de base de datos aparecerá el menú de la figura 4.2, el cual permite llevar a cabo la auditoría del servidor y la generación de los reportes correspondientes. Podemos elegir de entre las diferentes opciones presentadas.

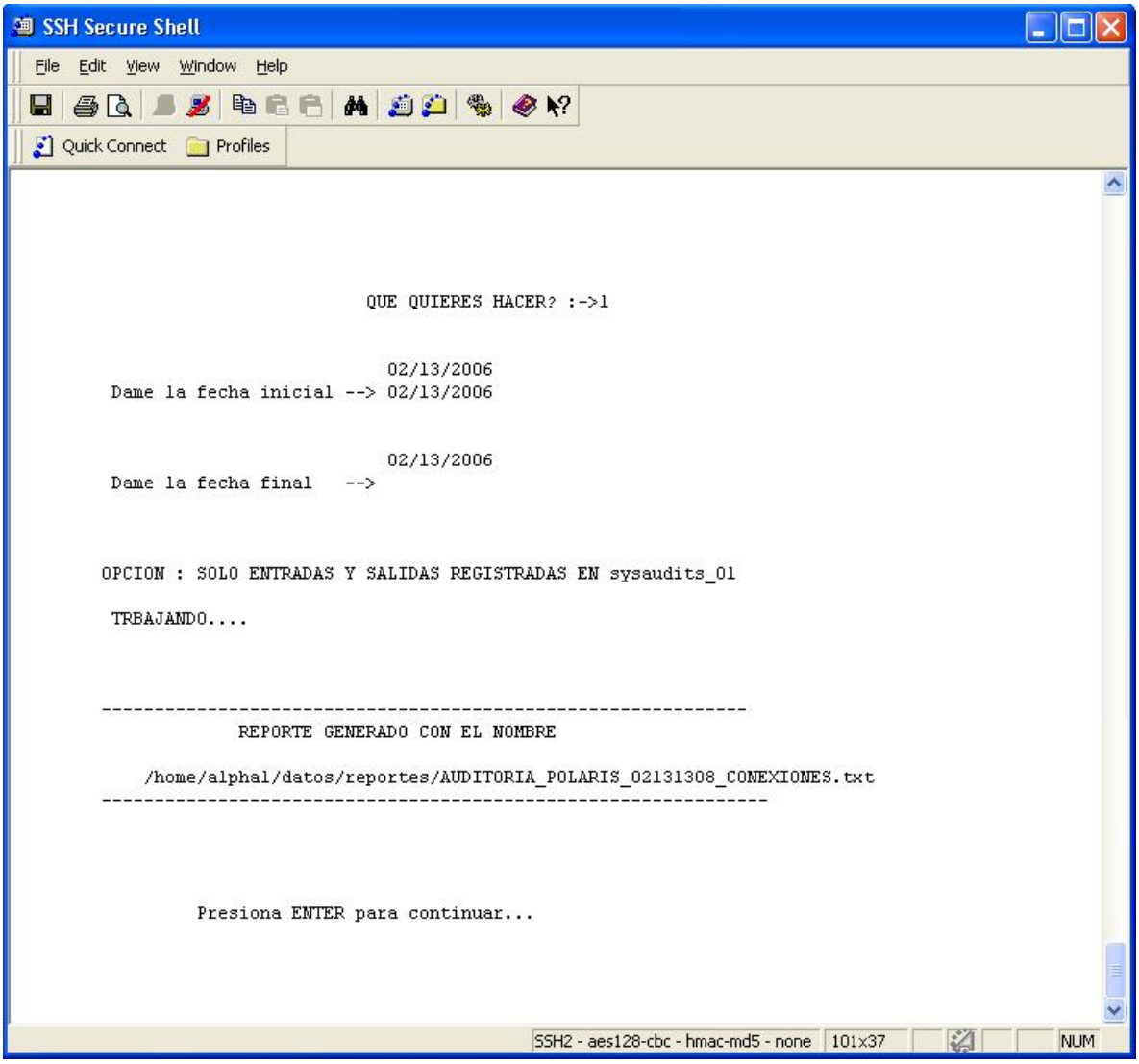


Figura 4.3 Solo entradas y salidas registradas en el Servidor

Si se elige la opción 1, mostrará la ventana de la figura 4.3, donde pedirá como parámetros de entrada la fecha inicial y la fecha final que se desea auditar, en caso de no dar ninguna opción tomará como default la fecha actual, desplegará un mensaje de la opción elegida y mostrará la ruta donde se depositará el reporte generado junto con el nombre del archivo que se creará. Finalmente el programa espera a que el usuario presione la tecla *enter* para terminar.

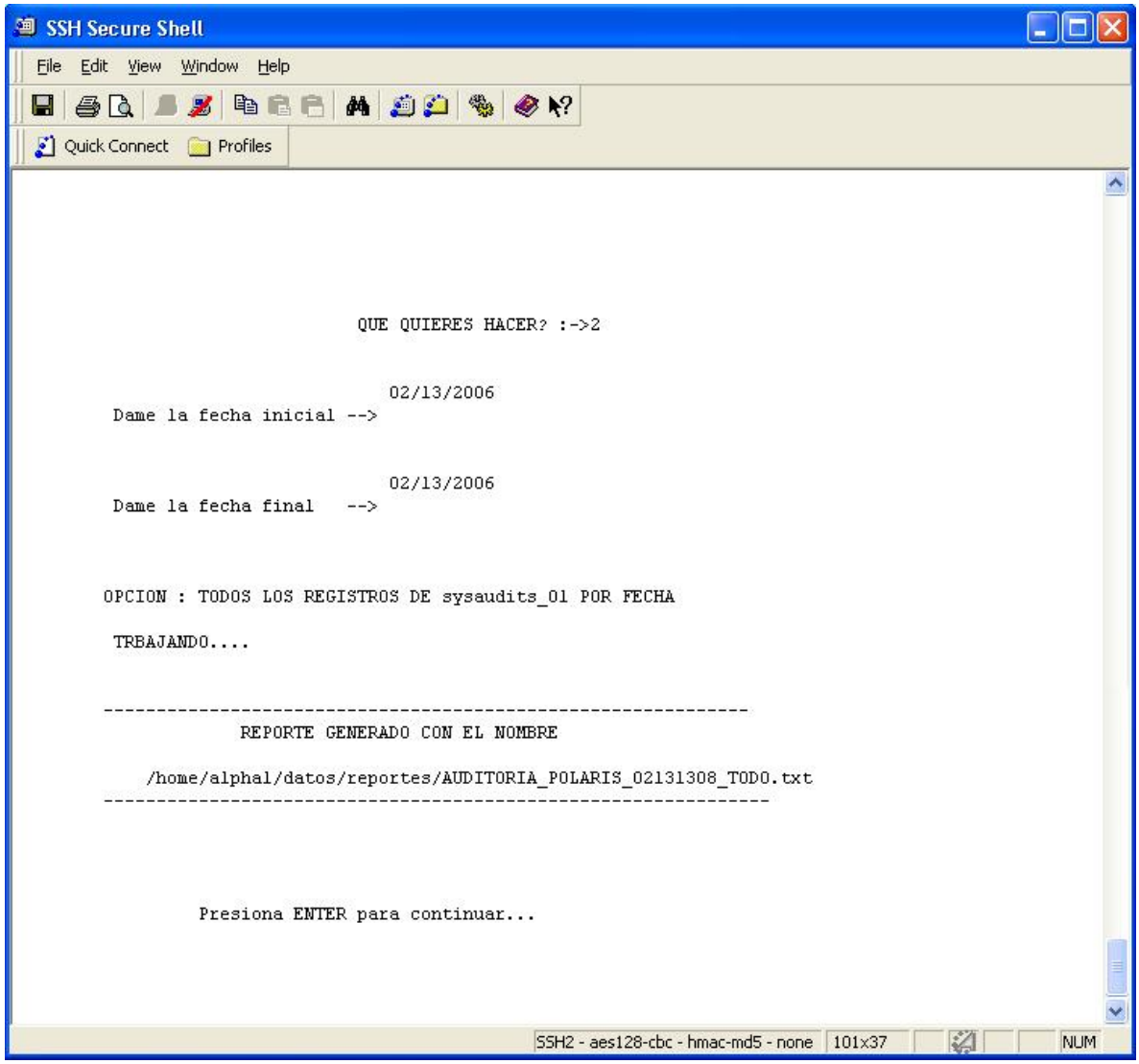


Figura 4.4 Todos los registros del servidor por fecha

Si la opción a elegir fue la 2, mostrará la siguiente ventana, de igual manera que en la opción anterior pedirá como parámetros la fecha inicial y final que se desea auditar. Finalmente el programa espera a que el usuario presione la tecla *enter* para terminar.

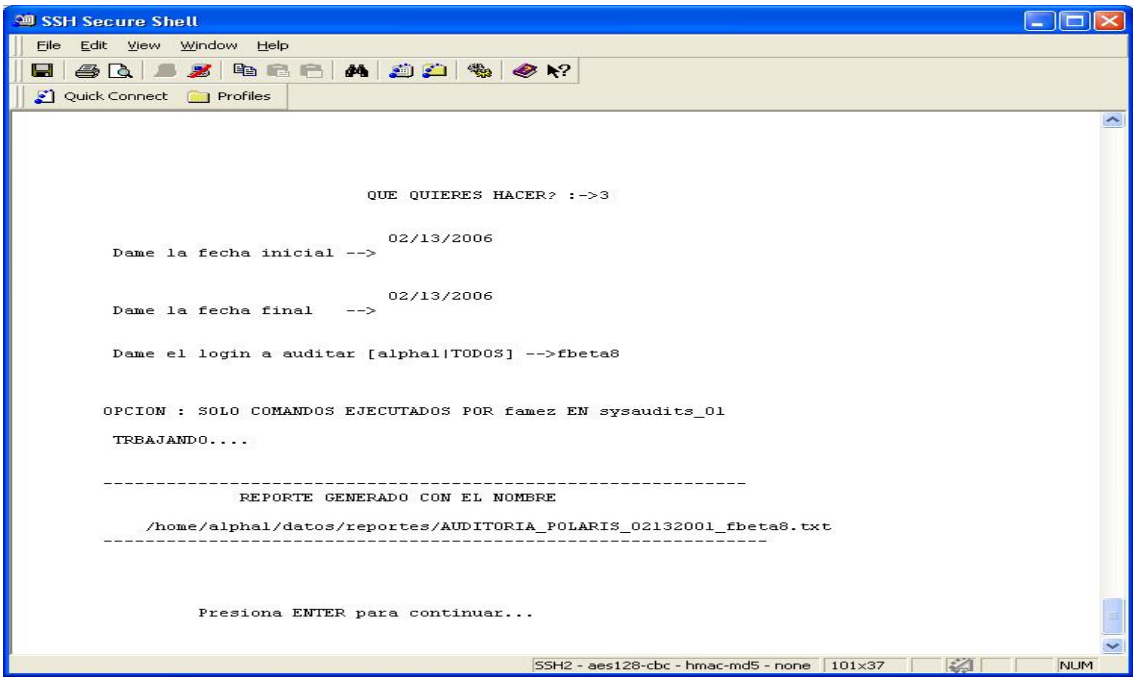


Figura 4.5 Solo comandos ejecutados por login en el servidor

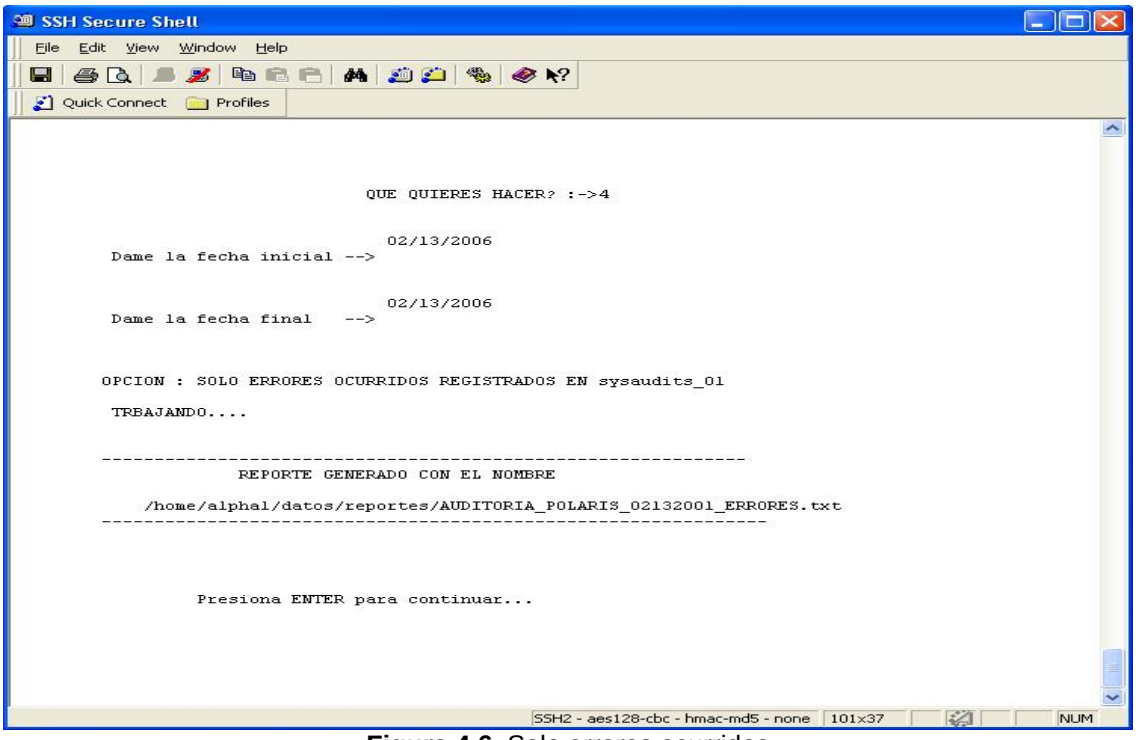


Figura 4.6 Solo errores ocurridos

Para las opciones 3 y 4 el programa mostrará las figuras 4.5 y 4.6 respectivamente, en las cuales mostrará el nombre del archivo que se generó y la ruta en la cual se guardará.

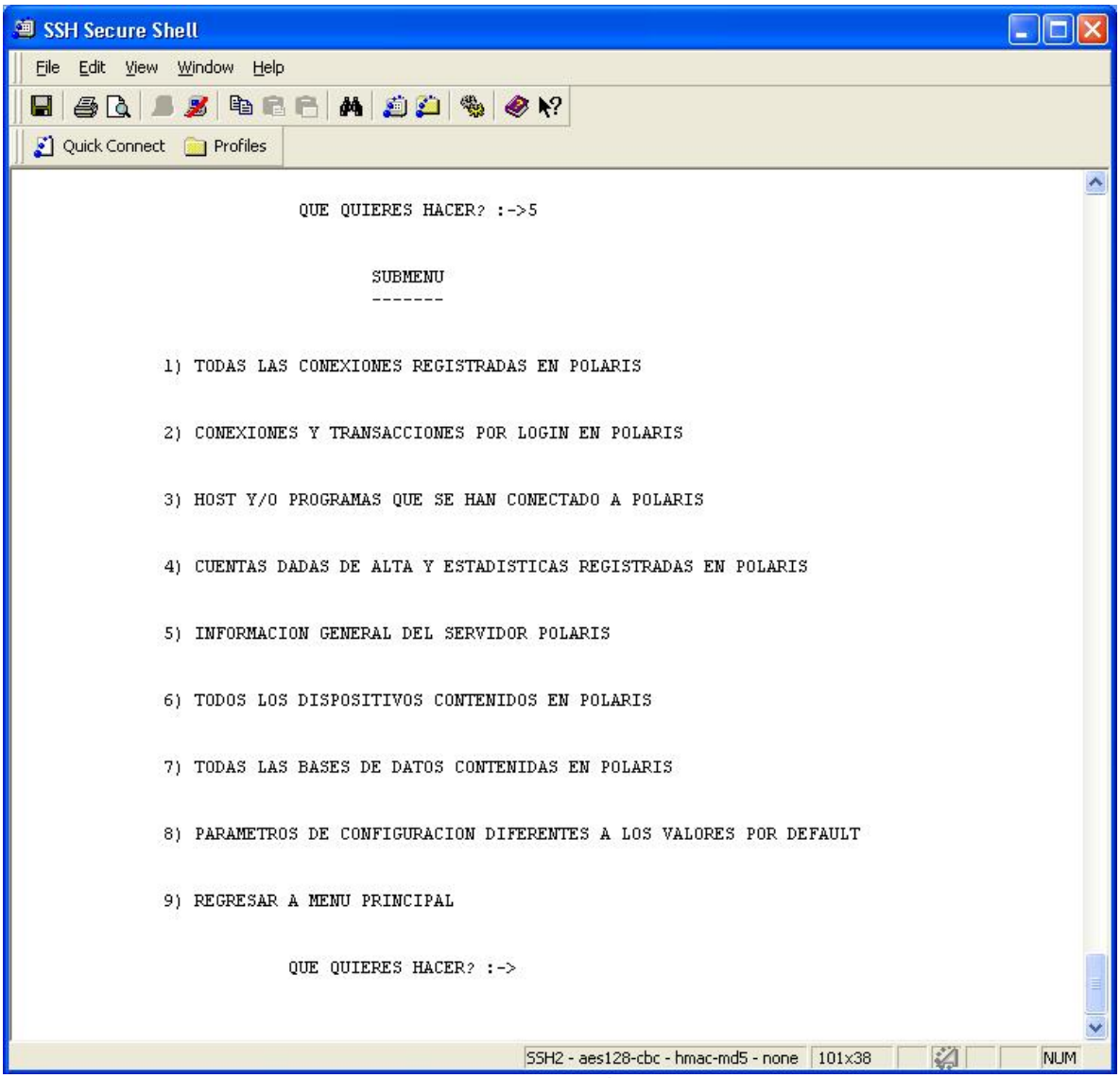


Figura 4.7 Opciones para generación de reportes en el servidor

Si la opción a elegir es la 5, el programa enviará un nuevo submenú como el presentado en la figura 4.7, en el cual se presentan más opciones para la generación de los reportes de auditoría.

Para las opciones del submenú, a continuación se presentan las figuras correspondientes a cada opción elegida, también puede observarse el nombre y la ruta del archivo que se generó:

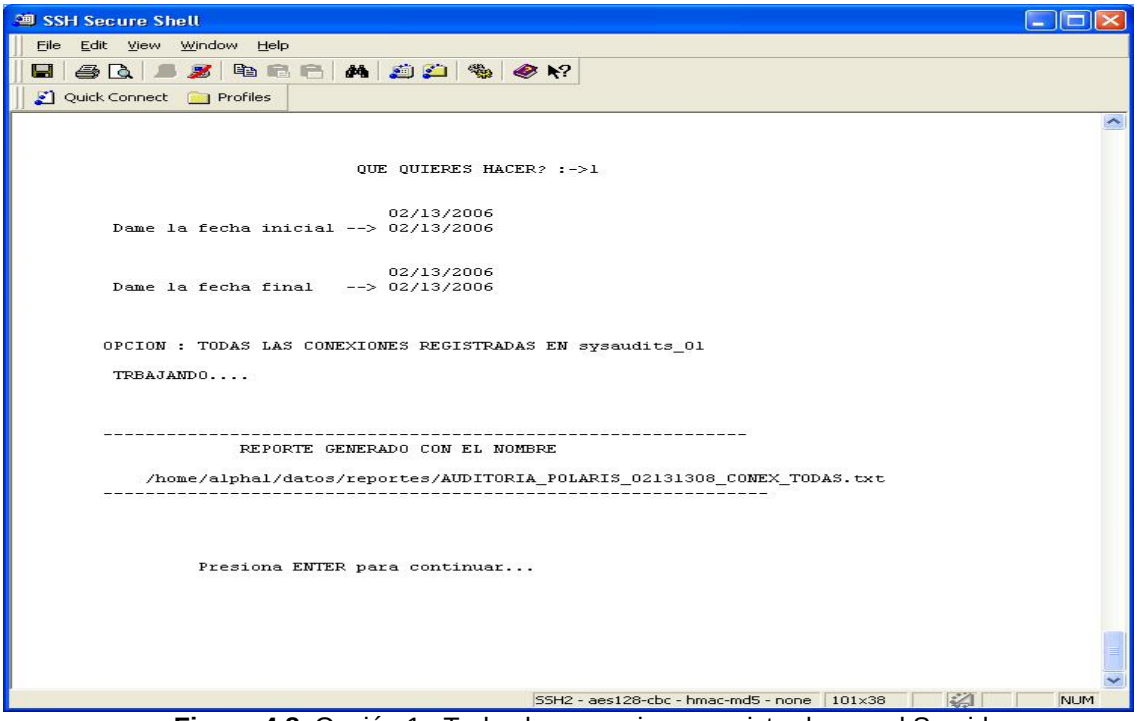


Figura 4.8 Opción 1. Todas las conexiones registradas en el Servidor

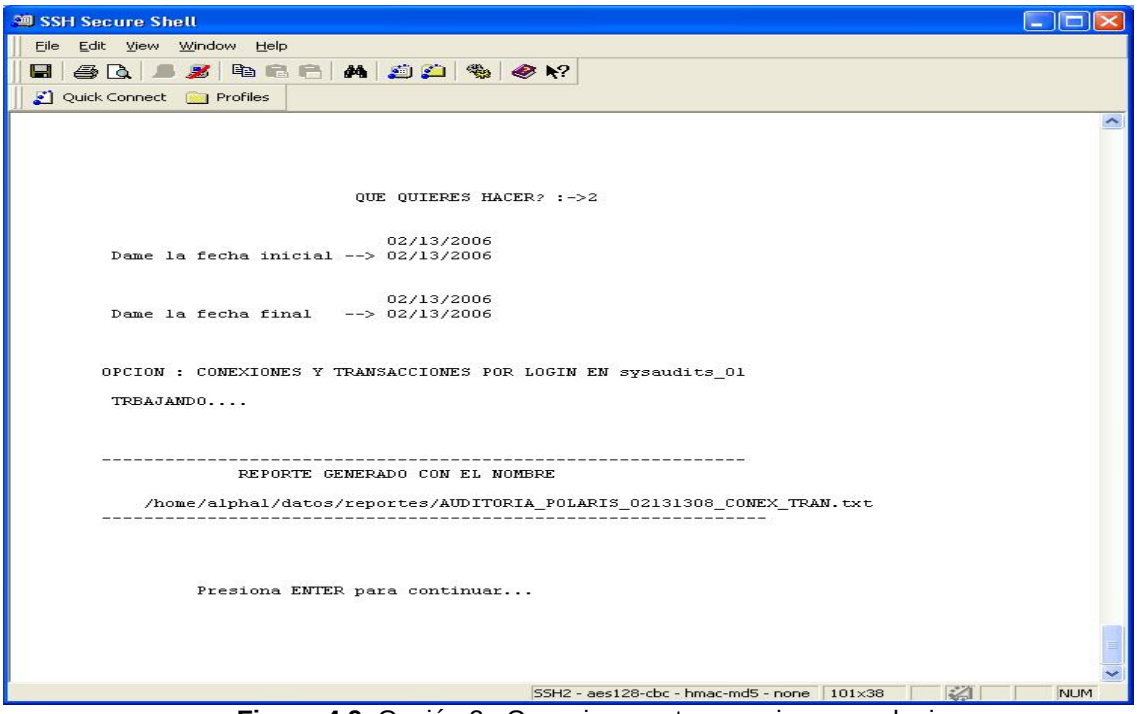


Figura 4.9 Opción 2. Conexiones y transacciones por login

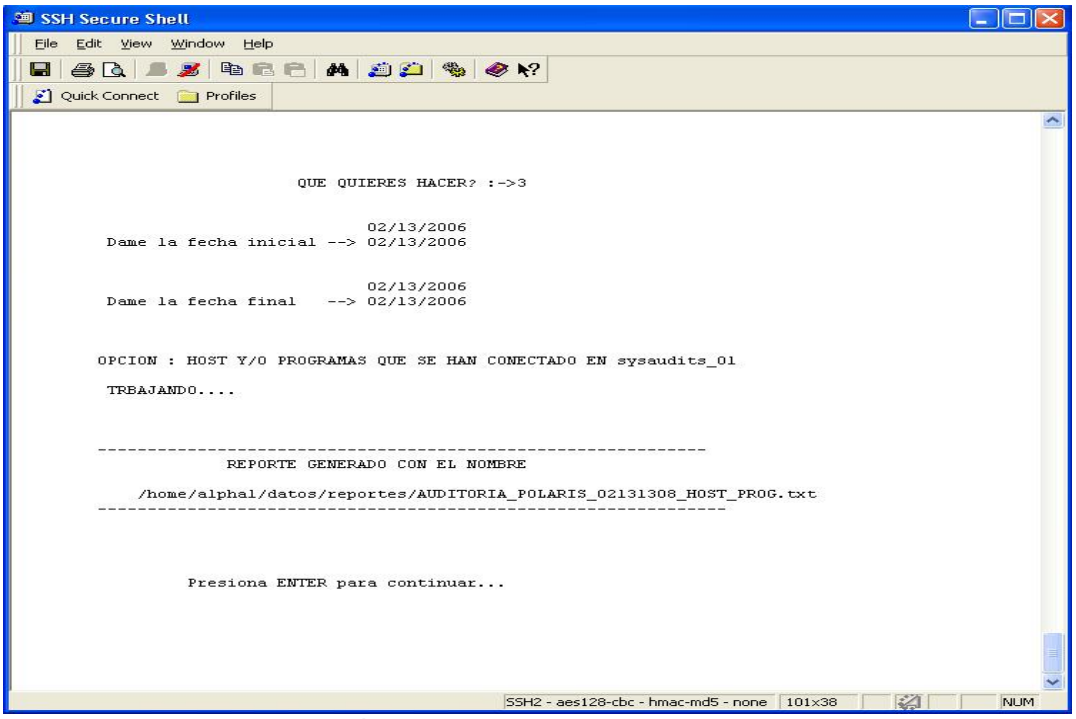


Figura 4.10 Opción 3. Host y/o programas que se han conectado

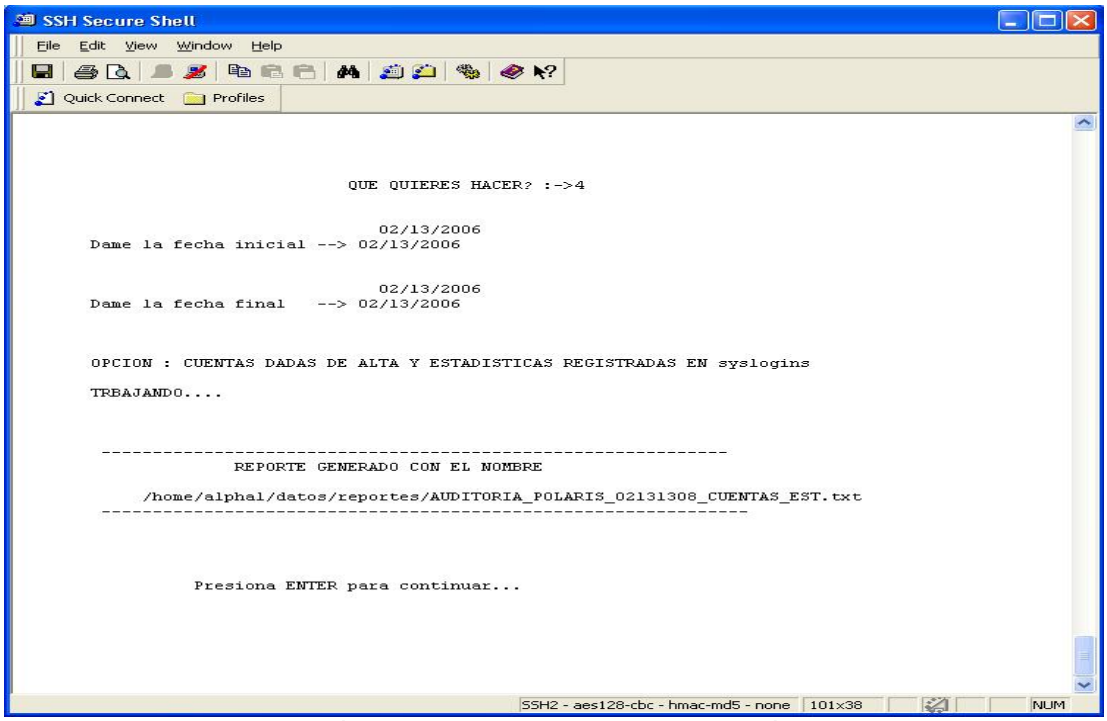


Figura 4.11 Opción 4. Cuentas dadas de alta u estadísticas registradas

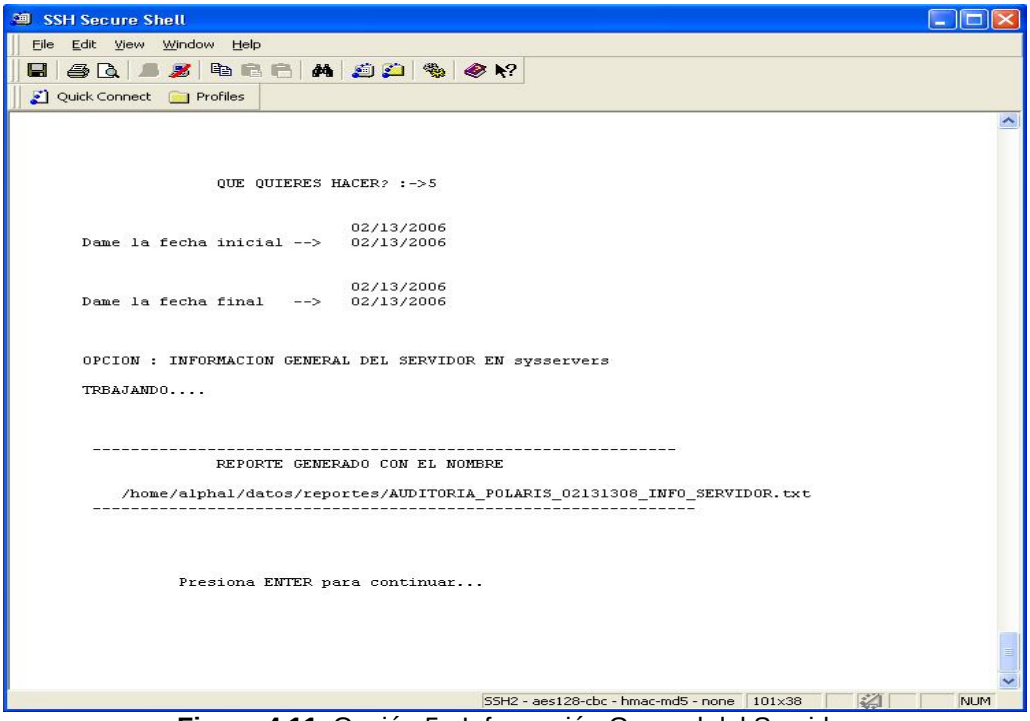


Figura 4.11 Opción 5. Información General del Servidor

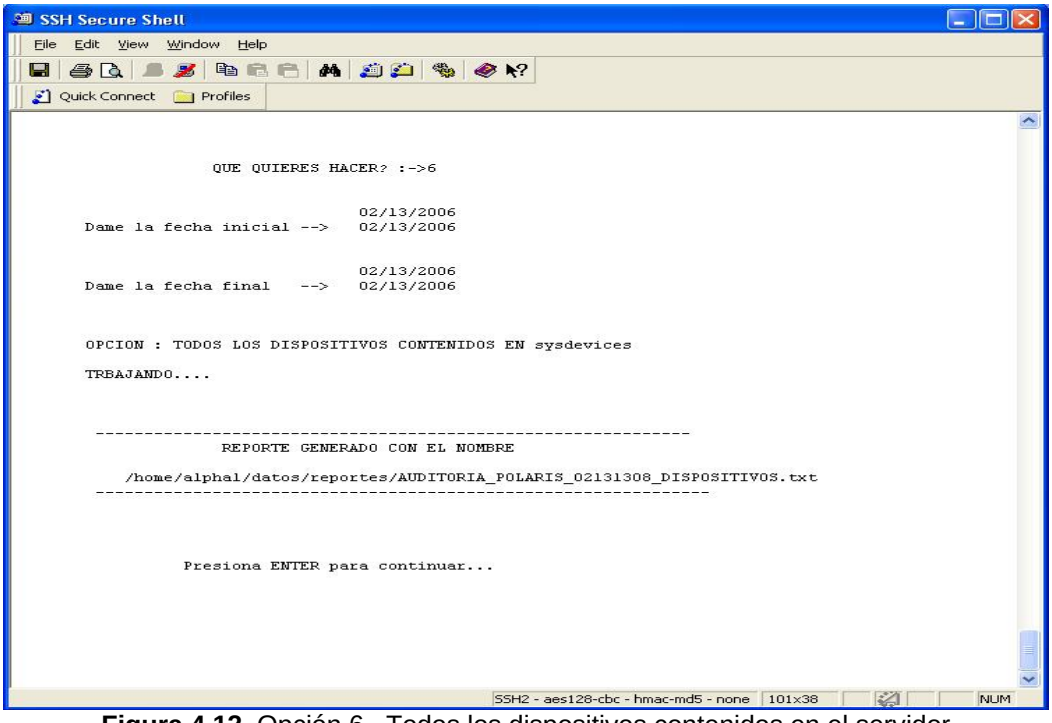


Figura 4.12 Opción 6. Todos los dispositivos contenidos en el servidor

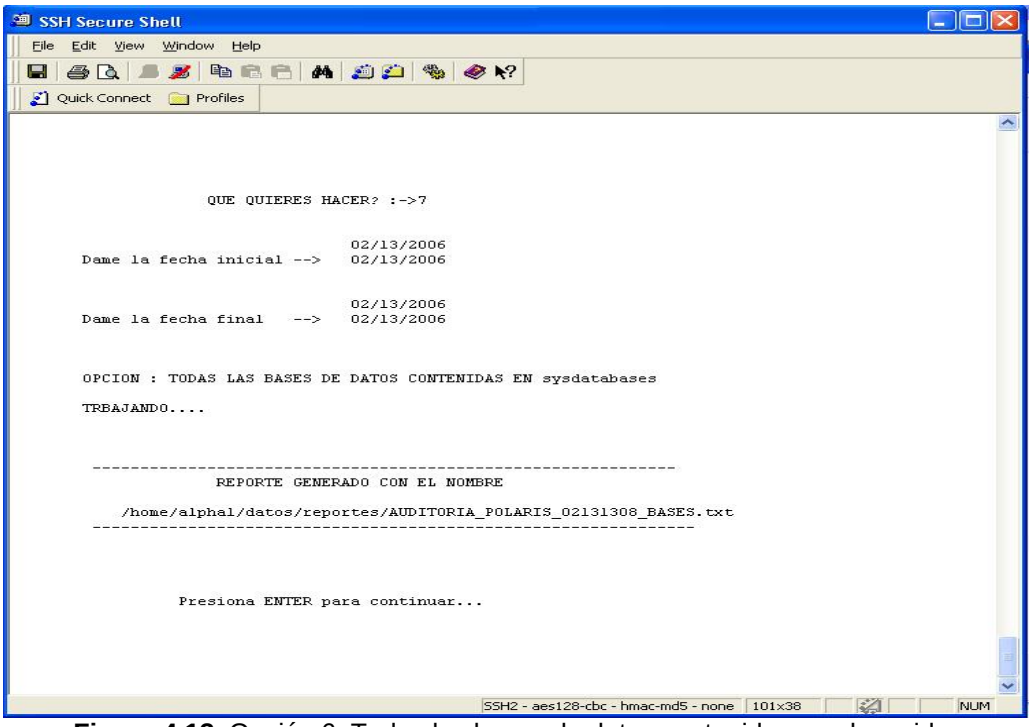


Figura 4.13 Opción 6. Todas las bases de datos contenidas en el servidor

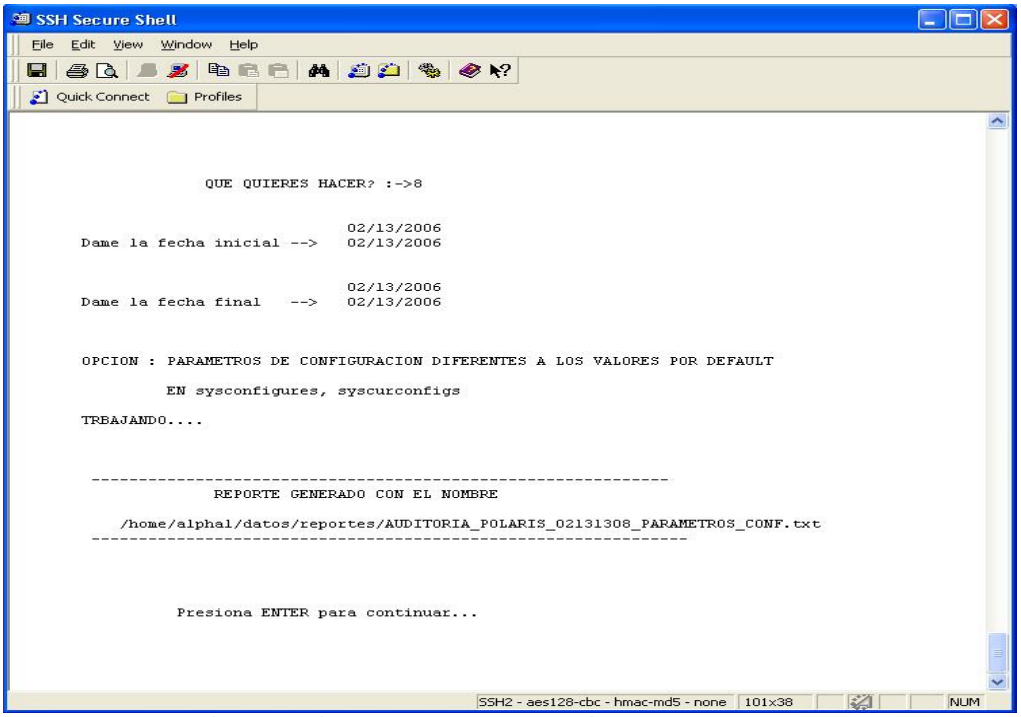
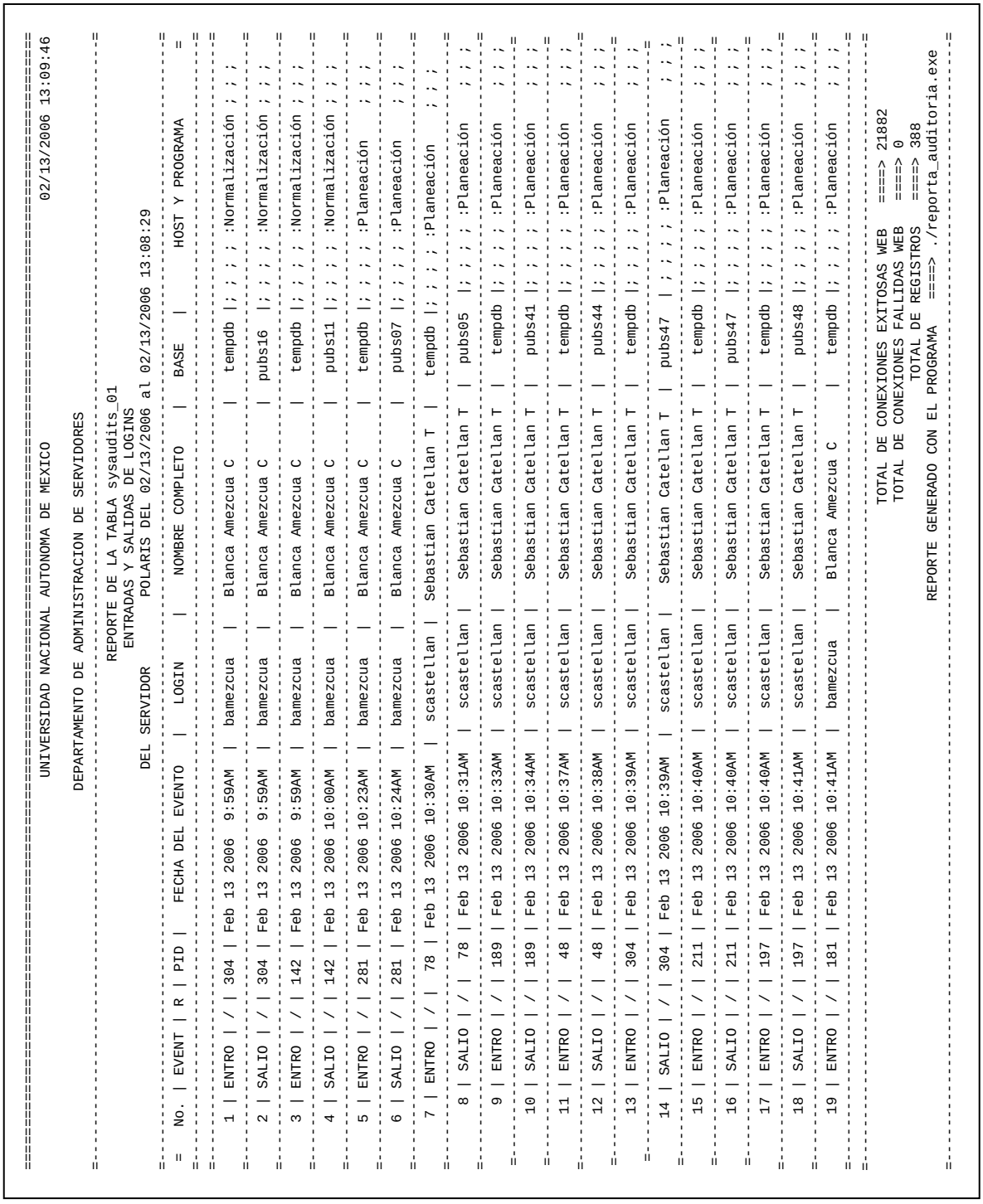


Figura 4.14 Opción 8. Parámetros de configuración diferentes a los valores por default

4.2 Generación de Reportes

Para la generación de los reportes el servidor que se utiliza es el que el usuario eligió al momento de ejecutar el programa. Las tablas que se toman en cuenta son las tablas de sistema sysaudits, syslogins, sysdatabases, sysdevices, syservers, sysusers.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO 02/13/2006 13:09:46

DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

REPORTE DE LA TABLA sysaudits_01
ENTRADAS Y SALIDAS DE LOGINS
DEL SERVIDOR POLARIS DEL 02/13/2006 al 02/13/2006 13:08:29

No.	EVENT	R	PID	FECHA DEL EVENTO	LOGIN	NOMBRE COMPLETO	BASE	HOST Y PROGRAMA
1	ENTRO	/	304	Feb 13 2006 9:59AM	bamezcua	Blanca Amezcua C	tempdb	; ; ; ; :Normalización ; ; ;
2	SALIO	/	304	Feb 13 2006 9:59AM	bamezcua	Blanca Amezcua C	pubs16	; ; ; ; :Normalización ; ; ;
3	ENTRO	/	142	Feb 13 2006 9:59AM	bamezcua	Blanca Amezcua C	tempdb	; ; ; ; :Normalización ; ; ;
4	SALIO	/	142	Feb 13 2006 10:00AM	bamezcua	Blanca Amezcua C	pubs11	; ; ; ; :Normalización ; ; ;
5	ENTRO	/	281	Feb 13 2006 10:23AM	bamezcua	Blanca Amezcua C	tempdb	; ; ; ; :Planeación ; ; ;
6	SALIO	/	281	Feb 13 2006 10:24AM	bamezcua	Blanca Amezcua C	pubs07	; ; ; ; :Planeación ; ; ;
7	ENTRO	/	78	Feb 13 2006 10:30AM	scastellan	Sebastian Catellan T	tempdb	; ; ; ; :Planeación ; ; ;
8	SALIO	/	78	Feb 13 2006 10:31AM	scastellan	Sebastian Catellan T	pubs05	; ; ; ; :Planeación ; ; ;
9	ENTRO	/	189	Feb 13 2006 10:33AM	scastellan	Sebastian Catellan T	tempdb	; ; ; ; :Planeación ; ; ;
10	SALIO	/	189	Feb 13 2006 10:34AM	scastellan	Sebastian Catellan T	pubs41	; ; ; ; :Planeación ; ; ;
11	ENTRO	/	48	Feb 13 2006 10:37AM	scastellan	Sebastian Catellan T	tempdb	; ; ; ; :Planeación ; ; ;
12	SALIO	/	48	Feb 13 2006 10:38AM	scastellan	Sebastian Catellan T	pubs44	; ; ; ; :Planeación ; ; ;
13	ENTRO	/	304	Feb 13 2006 10:39AM	scastellan	Sebastian Catellan T	tempdb	; ; ; ; :Planeación ; ; ;
14	SALIO	/	304	Feb 13 2006 10:39AM	scastellan	Sebastian Catellan T	pubs47	; ; ; ; :Planeación ; ; ;
15	ENTRO	/	211	Feb 13 2006 10:40AM	scastellan	Sebastian Catellan T	tempdb	; ; ; ; :Planeación ; ; ;
16	SALIO	/	211	Feb 13 2006 10:40AM	scastellan	Sebastian Catellan T	pubs47	; ; ; ; :Planeación ; ; ;
17	ENTRO	/	197	Feb 13 2006 10:40AM	scastellan	Sebastian Catellan T	tempdb	; ; ; ; :Planeación ; ; ;
18	SALIO	/	197	Feb 13 2006 10:41AM	scastellan	Sebastian Catellan T	pubs48	; ; ; ; :Planeación ; ; ;
19	ENTRO	/	181	Feb 13 2006 10:41AM	bamezcua	Blanca Amezcua C	tempdb	; ; ; ; :Planeación ; ; ;

TOTAL DE CONEXIONES EXITOSAS WEB ====> 21882
TOTAL DE CONEXIONES FALLIDAS WEB ====> 0
TOTAL DE REGISTROS ====> 388
REPORTE GENERADO CON EL PROGRAMA ====> ./reporta_auditoria.exe

El reporte anterior registra solo las entradas y salidas registradas en el servidor Polaris y la información que necesitamos para el reporte se tomó de las tablas de **sysaudits_01** y de syslogins. El reporte lleva por nombre *AUDITORIA_POLARIS_02131308_CONEXIONES.txt*

Procedemos a explicar las columnas contenidas en el reporte y de donde se tomó dicha información.

No.
Es un número consecutivo sólo para llevar un control de los registros y sea mas claro para el que está leyendo el reporte.

EVENT
Son valores de la columna Event que pertenece a la tabla **sysaudits_01** y para realizar nuestro reporte nos interesa que la columna contenga cualquiera de los siguientes valores:

45 corresponde a **LOGIN** que audita todas las entradas al Adaptive Server.
46 que corresponde a la opción de auditoria **LOGOUT** que se encarga de auditar todas las salidas al Adaptive Server.

R
Son valores simbólicos y dependen de la columna EVENTMOD que se encuentra en la tabla sysaudit_01 si esta columna contiene un valor igual a 1 que significa que el evento pasó la comprobación del evento y se representa con una (“ / “) en otro caso si el valor de esta columna es 0 que significa que no hubo ninguna modificación para el evento, o el valor de 2 que es cuando el evento falló la comprobación y cuando se toman estos valores se representará con una (“ X “).

PID
Son valores que se obtienen de la tabla sysaudit_01. De la columna que lleva el nombre de SPI en esta se almacena el ID del proceso que ocasionó el evento.

FECHA DEL EVENTO
Fecha y hora del evento estos datos se obtienen de la tabla de sysaudit_01 de su columna EVTIME.

LOGIN
Son valores de la columna loginname que se encuentra en la tabla sysaudits_01 y contiene el nombre del login correspondiente al suid, en caso de que éste sea nulo se le asignará la cadena "No hay login".

NOMBRE COMPLETO
Es el nombre completo del usuario, esta información la obtiene de la columna fullname que corresponde a la tabla syslogins, en esta columna se almacena el nombre completo del usuario y en caso de que este campo sea nulo se le asignará la cadena "login sin nombre"

BASE
Nombre de la base donde ocurrió el evento Los valores de esta columna se obtendrán de la tabla sysaudit_01 de la columna DBNAME.

HOST Y PROGRAMA
Información adicional del evento auditado, esta información corresponde a la columna extrainfo contenida en la tabla sysaudit_01.

```
=================================================================================================================================================
                                          UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO                                  02/13/2006 13:14:39
                                         DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES
=-----------------------------------------------------------------------------------------------------------------------------------------------=
                                               REPORTE DE LA TABLA sysaudits_01
=                                        ACTIVIDADES DEL 02/13/2006 00:00:01 AL 02/13/2006 13:09:46 EN EL SERVIDOR    POLARIS    =
=-----------------------------------------------------------------------------------------------------------------------------------------------=
=       | E | M |  FECHA DEL EVENTO    |           |              |      |                                                                          =
=  No.  | V | D |                      |   LOGIN   |BASE DE DATOS | OBJTO|                    COMANDO EJECUTADO                                     =
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     1 | 45|  1|  Feb 13 2006  9:59AM |      beta1 |       tempdb |      | ; ; ; ; :Normalización; ; ;
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     2 | 46|  1|  Feb 13 2006 10:31AM |      beta2 |        pubs5 |      | ; ; ; ; :Ciclo ; ; ;
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     3 | 46|  1|  Feb 13 2006 10:53AM |      beta3 |        pubs8 |      | ; ; ; ; epsilon; ; ;
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     4 | 45|  1|  Feb 13 2006 11:21AM |      beta4 |       tempdb |      | ; ; ; ; prueba_ins.c; ; ;
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     5 | 45|  2|  Feb 13 2006 11:27AM |      beta5 |       master |      | ; ; ; ; :Registros; ; ;
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     6 | 92|  1|  Feb 13 2006 11:59AM |     fbeta8 |       pubs13 |      | use pubs13
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     7 | 92|  1|  Feb 13 2006 11:59AM |     fbeta8 |       pubs13 |      | INSERT publishers (pub_id,pub_name,city,state)
       |   |   |                      |            |              |      | VALUES("1756","Jardn,Inc","Camden","NJ")
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     8 | 92|  1|  Feb 13 2006 11:59AM |     fbeta8 |       pubs13 |      | SELECT title_id, convert(int,(total_sales/price)),
       |   |   |                      |            |              |      | convert(varchar(20),pubdate,2) FROM titles
=-----------------------------------------------------------------------------------------------------------------------------------------------=
     9 | 92|  1|  Feb 13 2006 11:59AM |     fbeta8 |       pubs13 |      | SELECT    title_id,    total_sales,
       |   |   |                      |            |              |      | total_sales*2   from  titles
=-----------------------------------------------------------------------------------------------------------------------------------------------=
    10 | 92|  1|  Feb 13 2006 11:59AM |     fbeta8 |       pubs13 |      | SELECT     a.au_id,    a.au_lname,     a.au_fname,
       |   |   |                      |            |              |      | t.price*t.total_sales
       |   |   |                      |            |              |      | FROM     authors a,    titles t,sigl ,   titleauthor ti
=-----------------------------------------------------------------------------------------------------------------------------------------------=
    11 | 92|  1|  Feb 13 2006 11:59AM |     fbeta8 |       pubs13 |      | WHERE ti.au_id = a.au_id  and  t.title_id = ti.title_id
       |   |   |                      |            |              |      | order  by  a.au_id
       |   |   |                      |            |              |      | compute  sum(t.price*t.total_sales)  by  a.au_id
=-----------------------------------------------------------------------------------------------------------------------------------------------=
    12 | 46|  1|  Feb 13 2006 12:53PM |     alpha1 |       master |      | sa_role sso_role; ; ; ; :rep_ser_syb; ; ;
=-----------------------------------------------------------------------------------------------------------------------------------------------=
    13 | 46|  1|  Feb 13 2006  1:09PM |      beta9 |       pubs47 |      | ; ; ; ; epsilon:SQSH_1.0; ; ;
=-----------------------------------------------------------------------------------------------------------------------------------------------=
=-----------------------------------------------------------------------------------------------------------------------------------------------=
=                                                                TOTAL DE CONEXIONES EXITOSAS WEB   ====> 21977
=                                                                TOTAL DE CONEXIONES FALLIDAS WEB   ====> 0
=                                                                            TOTAL DE REGISTROS   ====> 3042
=                                                                REPORTE GENERADO CON EL PROGRAMA   ====> ./reporta_auditoria.exe
=-----------------------------------------------------------------------------------------------------------------------------------------------=
```

El reporte anterior tiene como nombre *AUDITORIA_POLARIS_02131308_TODO.txt,* y corresponde a la salida de la Opción 2. Todos los registros del servidor por fecha, del menú principal.

El reporte muestra la fecha y la hora en que se generó, el nombre del servidor que auditó. Las columnas que se obtienen son de la tabla sysaudits_01 y se explican a continuación:

No.
Es un número consecutivo que se le asigna a cada registro.

EV
Corresponde a la columna event, es el tipo de evento que está siendo auditado. El número 45 corresponde a las entradas al servidor, el 46 corresponde a las salidas del servidor, el número 92 se refiere a todos los comandos que realiza un usuario, etc. Para mayor referencia del significado de los eventos consultar el anexo 2, de la presente tesis.

MD
Corresponde a la columna eventmod de la tabla sysaudits_01, la cual guarda información adicional del evento auditado. Los valores posibles son:

0 = ninguna modificación para este evento

1 = el evento pasó

2 = el evento falló

FECHA DEL EVENTO
Corresponde a la columna eventtime de la tabla sysaudits_01, la cual registra la fecha y hora en que ocurrió el evento.

LOGIN
Es el nombre del login que causó el evento.

BASE DE DATOS
Es el nombre de la base de datos donde ocurrió el evento

OBJTO
Nombre del objeto auditado

COMANDO EJECUTADO
Información adicional del evento auditado. Este campo contiene una serie de puntos separados por puntos y comas. (Ver anexo2)

```
================================================================================================================================
                                          UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO                          02/13/2006 13:21:04
                                        DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES
=--------------------------------------------------------------------------------------------------------------------------------=
                                                REPORTE DE LA TABLA sysaudits_01
=               ERRORES OCURRIDOS DEL 02/13/2006 00:00:01 AL 02/13/2006 13:18:39 EN EL SERVIDOR       POLARIS
=--------------------------------------------------------------------------------------------------------------------------------=
=       | E | M |  FECHA DEL EVENTO    |           |              |      |                                                           =
=  No.  | V | D |                      |   LOGIN   |BASE DE DATOS | OBJTO|                   COMANDO EJECUTADO                       =
=--------------------------------------------------------------------------------------------------------------------------------=
     1 | 37|  2|  Feb 13 2006 11:27AM |      beta7 |       master |      | ; ; ; ; 4002.14.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     2 | 45|  2|  Feb 13 2006 11:27AM |      beta7 |       master |      | ; ; ; ; :Registros; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     3 | 37|  2|  Feb 13 2006 11:27AM |      beta7 |       pubs7  |      | ; ; ; ; 249.16.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     4 | 37|  2|  Feb 13 2006 11:27AM |      beta7 |       pubs7  |      | ; ; ; ; 249.16.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     5 | 37|  2|  Feb 13 2006 11:27AM |      beta7 |       pubs7  |      | ; ; ; ; 249.16.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     6 | 37|  2|  Feb 13 2006 11:31AM |      beta6 |       pubs24 |      | ; ; ; ; 515.16.3; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     7 | 37|  2|  Feb 13 2006 11:32AM |      beta7 |       master |      | ; ; ; ; 4002.14.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     8 | 45|  2|  Feb 13 2006 11:32AM |      beta7 |       master |      | ; ; ; ; :Registros; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
     9 | 37|  2|  Feb 13 2006 11:32AM |      beta7 |       master |      | ; ; ; ; 4002.14.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
    10 | 45|  2|  Feb 13 2006 11:32AM |      beta7 |       master |      | ; ; ; ; :Registros; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
    11 | 37|  2|  Feb 13 2006 11:35AM |      beta7 |       master |      | ; ; ; ; 4002.14.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
    12 | 45|  2|  Feb 13 2006 11:35AM |      beta7 |       master |      | ; ; ; ; :Registros; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
    13 | 37|  2|  Feb 13 2006 11:35AM |      beta7 |       pubs7  |      | ; ; ; ; 249.16.1; ; ;
=--------------------------------------------------------------------------------------------------------------------------------=
=--------------------------------------------------------------------------------------------------------------------------------=
=                                                    TOTAL DE CONEXIONES EXITOSAS WEB   ====> 22717
=                                                    TOTAL DE CONEXIONES FALLIDAS WEB   ====> 0
=                                                                TOTAL DE REGISTROS     ====> 48
=                                                    REPORTE GENERADO CON EL PROGRAMA   ====> ./reporta_auditoria.exe
=--------------------------------------------------------------------------------------------------------------------------------=
=--------------------------------------------------------------------------------------------------------------------------------=
```

El reporte anterior tiene como nombre *AUDITORIA_POLARIS_02132001_ERRORES.txt,* y corresponde a la salida de la opción 3, sólo errores ocurridos registrados en el servidor, del menú principal. El reporte muestra la fecha y la hora en que se generó, el nombre del servidor que auditó. Las columnas que se obtienen son de la tabla sysaudits_01 y se explican a continuación:

No.
Es un número consecutivo que se le asigna a cada registro

EV
Es el tipo de evento que está siendo auditado. El número 37 corresponde a los errores, 45 corresponde a las entradas al servidor. Para mayor referencia del significado de los eventos consultar el anexo 2, de la presente tesis.

MD
Corresponde a la columna eventmod de la tabla sysaudits_01, la cual guarda información adicional del evento auditado. Los valores posibles son:

0 = ninguna modificación para este evento

1 = el evento pasó

2 = el evento falló

FECHA DEL EVENTO
Corresponde a la columna eventtime de la tabla sysaudits_01, la cual registra la fecha y hora en que ocurrió el evento.

LOGIN
Es el nombre del login que causó el evento.

BASE DE DATOS
Es el nombre de la base de datos donde ocurrió el evento

OBJTO
Nombre del objeto auditado

COMANDO EJECUTADO
Información adicional del evento auditado. Este campo contiene una serie de puntos separados por puntos y comas. (Ver anexo2 y el Manual Troubleshooting and Error messages guide volume 2)

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO 02/13/2006 13:18:39

DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

REPORTE DE LA TABLA sysaudits_01
ACTIVIDADES DEL 02/13/2006 00:00:01 AL 02/13/2006 13:14:39
EN EL SERVIDOR POLARIS DEL LOGIN jsanchez
NOMBRE COMPLETO: JUAN

No.	EVENTO	RESULT	FECHA DEL EVENTO	DBNAME	COMANDO EJECUTADO
1	LOGIN	EXITO	Feb 13 2006 11:59AM	tempdb	; ; ; ; :Registros de Personal DEP; ; ;
2	COMANDO	EXITO	Feb 13 2006 11:59AM	tempdb	select @@langid
3	COMANDO	EXITO	Feb 13 2006 11:59AM	tempdb	select dateformat from master.dbo.syslanguages where langid = 0
4	COMANDO	EXITO	Feb 13 2006 11:59AM	tempdb	select @@ncharsize
5	COMANDO	EXITO	Feb 13 2006 11:59AM	tempdb	sp_server_info 1
6	COMANDO	EXITO	Feb 13 2006 11:59AM	tempdb	use pubs13
7	COMANDO	EXITO	Feb 13 2006 11:59AM	pubs13	select name from dbo.sysusers where suid = SUSER_ID('jsanchez')
8	COMANDO	EXITO	Feb 13 2006 11:59AM	pubs13	select xxx_xxx from xxx where xxx_xxx =13
9	COMANDO	EXITO	Feb 13 2006 11:59AM	pubs13	SELECT xxx.xxx_clav , xxx.xxx_xxx_xxx , x xxx.xxx_xxx_xxx , xxx.xxx_xxx FROM xxx WH xx (xxx.xxx_xxx_xxx = 'jsanchez')
10	COMANDO	EXITO	Feb 13 2006 11:59AM	pubs13	SELECT xxx.xxx_xxx_xxx , xxx.xxx_xxx , r xx.xxx_xxx , xxx.xxx_xxx_xxx , xxx.xxx_ xxxx FROM xxx , xxx WHERE (xxx.xxx_xxx = x or .xxx_xxx_enc) and ((xxx.xxx_xxx_xxx = 'jsanchez'
11	COMANDO	EXITO	Feb 13 2006 11:59AM	pubs13	(xxx.xxx_clav_usr = 'consulta'))
12	COMANDO	EXITO	Feb 13 2006 11:59AM	pubs13	SELECT xxx.xxx_xxx, xxx.xxx_xxx FROM xxx WHERE xxx.xxx_xxxx = 2062.000000000000000

En este reporte se registraron las actividades realizadas por un solo login dentro del servidor Polaris y la información que necesitamos para el reporte se tomó de la tabla **sysaudits_01.** Este reporte tiene como nombre AUDITORIA_POLARIS_02132001_ jsanchez

Procedemos a explicar las columnas contenidas en el reporte y de dónde se tomó dicha información.

No.
Es un número consecutivo solo para llevar un control de los registros y sea más claro para la persona que está leyendo el reporte.

EVENTO
Son valores de la columna event que pertenece a la tabla **sysaudits_01.** Estos valores corresponden al evento que está siendo auditado en este caso sólo nos interesan los siguientes:

45 corresponde a **LOGIN** que audita todas las entradas al Adaptive Server.
46 que corresponde a la opción de auditoria **LOGOUT** que se encarga de auditar todas las salidas al Adaptive Server.

RESULT
Estos valores dependen de la columna event correspondiente a la tabla sysaudit_01.
Si event toma el valor 45 ya anteriormente explicados RESULT se llenará con la cadena **LOGIN,** si event toma el valor 46 RESULT se llenará con la cadena **LOGOUT** y para cualquier otro valor RESULT se llenará con **COMANDO.**

FECHA DEL EVENTO
Fecha y hora del evento, estos datos se obtienen de la tabla sysaudit_01 de su columna eventtime.

DBNAME
Nombre de la base donde ocurrió el evento. Los valores de esta columna se obtendrán de la tabla sysaudit_01 de la columna dbname.

COMANDO EJECUTADO
Información adicional del evento auditado, esta información corresponde a la columna extrainfo contenida en la tabla sysaudit_01.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

FECHA DE HOY [13/02/2006] REPORTE DEL MES DE [FEBRERO] HORA [12:51:16]

REPORTE DE EL NUMERO DE CONEXIONES AL SERVIDOR [POLARIS]

	1										2										3										
DIAS / HORA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00:00	1B0	195	12E	E9	A5	E9	163	111	129	EF	99	88	D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00:30	129	14E	D6	AD	9B	E0	10B	D6	D6	C3	7D	85	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01:00	A4	BB	AB	7F	65	8A	D5	BE	76	7A	50	5C	65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01:30	77	7D	75	3E	51	76	64	67	56	44	45	42	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02:00	3C	55	55	31	3E	47	55	5D	38	38	29	2A	2A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02:30	34	37	32	2A	17	37	2F	36	18	28	24	10	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03:00	2E	22	20	19	19	24	17	11	15	16	16	14	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03:30	17	16	E	F	12	8	12	16	3	D	8	D	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04:00	1A	12	F	6	D	E	4	1B	6	F	D	E	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04:30	B	10	9	4	B	A	B	8	7	7	6	8	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05:00	13	E	3	6	1	6	5	1C	8	4	4	6	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05:30	B	D	5	4	4	7	15	14	4	F	A	4	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06:00	B	14	A	1	3	B	A	17	7	3	B	1	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06:30	2D	22	20	6	8	7	2D	23	D	13	6	2	A7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07:00	4B	4F	38	17	4	8	72	3C	33	8B	F	3	FB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07:30	97	96	5E	1D	5	13	B5	9D	55	97	B	9	C9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08:00	104	126	EE	21	1E	22	13F	113	8D	67	19	17	11D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08:30	1DB	1AF	176	2D	36	4A	1E1	162	117	155	28	1E	1D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTALES	A169	942B	7443	251D	23E0	4FE7	A39F	833A	69A3	511B	1C87	22DD	2827	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Este reporte da informe de todas las conexiones hechas en un mes, registradas por día, semana y por cada media hora. La información aquí presentada se encuentra en hexadecimal ya que si la información se dejara en forma decimal los números podrían ser demasiado grandes.

Para poder realizar este reporte se consulta la tabla sysaudit_01 la cual se encuentra en la base de datos sybsecurity. Las columnas que consultamos principalmente son:

event que es el tipo de evento que está siendo auditado; Para nuestro reporte nos interesa la columna event cuando toma el valor de 45 que es cuando se auditan todas las entradas al Adaptive Server.

Otra de las columnas importantes para nuestro reporte es la de eventmod la cual nos da información adicional acerca de los eventos auditados. Para nuestro repote nos interesa la columna eventmod cuando toma el valor de 1, revisar el capitulo 3.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

FECHA DE HOY [13/02/2006] REPORTE DEL MES DE [FEBRERO] HORA [12:51:16]

REPORTE DE EL NUMERO DE CONEXIONES DE CADA USUARIO Y CADA UNA DE LAS TRANSACCIONES REALIZADAS EN EL SERVIDOR [POLARIS]

USUARIO	NOMBRE COMPLETO	NO. CONEXIONES	EXITOSAS	FALLIDAS	NUMERO DE SENTENCIAS REALIZADAS POR EL USUARIO	
XXXXX	XXXXXXXX XXXXXXX XXXXX	21	21	0		
XXXXX	XXXXXX XXXXXX XXXXXX	116	116	0	SELECT	11201
					INSERT	72
					DELETE	73
					UPDATE	11
XXXXX	XXXXXX XXXXXX XXXXX	80	76	4		
XXXXX	XXXXXX XXXXXX	3	3	0		
XXXXX	XXXXXX	212	207	5	SELECT	201592
					UPDATE	1
XXXXX	XXXXX XXXXXXX XXX	178	177	1		
XXXXX	XXXXX XXXXXXX XXX	16	16	0		

LOGINS NO EXISTENTES EN EL SERVIDOR [POLARIS]

USUARIO	NO. CONEXIONES	EXITOSAS	FALLIDAS
XXXXXX	2	0	2

USUARIOS SIN CONEXION EN ESTE PERIODO

XXXXXXXX XXXX XXXXXXXX XXXXXXXXXX
XXXXXX XXX XXXXX XXXXXXX XXXXXX XXXX

TOTAL DE SELECTS : 220021	TOTAL DE INSERTS : 72	TOTAL DE DELETES : 73	TOTAL DE UPDATES : 23

El reporte presentado anteriormente lleva el nombre de *AUDITORIA_POLARIS_02132001_CONEX_TRAN.txt*, y se refiere a todas aquellas transacciones que han realizado los logins dentro del servidor.

Procedemos a explicar las columnas contenidas en el reporte y de donde se tomó dicha información.

USUARIO
Es el nombre de todas las cuentas de login.

NOMBRE COMPLETO
Es el nombre completo del usuario, que es dueño de esa cuenta de login, este dato se obtiene de la tabla syslogins.

NO. CONEXIONES
Es el número de veces en que se conectó un login al servidor este dato se obtiene de la tabla sysaudit tomando el loginname y la columna event.

EXITOSAS
Tomando en cuenta la información anterior y además la columna eventmod podemos hacer una comparación en la cual si el resultado de eventmod= 1, marcar como éxito la conexión.

FALLIDAS
En cambio si la información de la columna eventmod=2, podemos decir que fue una conexión fallida al servidor.

NUMERO DE SENTENCIAS REALIZADAS POR EL USUARIO
Tomando en cuenta el loginname de sysaudits y además apoyándonos en la columna extrainfo podemos saber si la transacción que realizó es un select, insert, delete o update y realizar así la suma total de cada una de las transacciones realizadas.

El reporte también nos permite saber si hay logins no existentes en el servidor, ya que nos otorga el nombre del login que pretendió entrar al servidor y si su conexión fue exitosa o no. Por otro lado nos presenta un reporte de aquellas cuentas de login que no han tenido actividad durante cierto periodo.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

FECHA DE HOY [13/02/2006] REPORTE DEL MES DE [FEBRERO] HORA [12:51:16]

REPORTE DE HOST Y/O PROGRAMAS QUE SE HAN CONECTADO AL SERVIDOR POLARIS

NOMBRE DE EL HOST Y/O PROGRAMA	NUMERO DE CONEXIONES	EXITOSAS	FALLIDAS
:Avance	90	69	21
:Ciclo	180	179	1
:Inscripcion	8	8	0
:Prueba	114	114	0
beta	294530	294530	0
epsilon:SQSH_1.0	305	300	5
epsilon	893	859	34
prueba.c	44	44	0
inscripcion.c	58	53	5
:reportes	2	2	0
:reporta_audit	1	1	0
gama	2	2	0

El reporte anterior da informe acerca del número de conexiones que los programas y/o host han tenido, tanto exitosas como fallidas en un mes.

La información que aquí se presenta es tomada de la tabla de sysaudits_01 que se encuentra en la base de datos sybsecurity.

NOMBRE DE EL HOST Y/O PROGRAMA
Esta información es tomada de la columna extrainfo que es información adicional del evento auditado y está contenida en la tabla sysaudit_01.

NÚMERO DE CONEXIONES
Es el registro que se tiene del número de conexiones realizadas al servidor para un solo evento, esta información es tomada de la columna sequense que se encuentra en la tabla sysaudits_01.

CONEXIONES EXITOSAS
Es el conteo que se hace del número de conexiones realizadas al servidor para un solo evento, este resultado depende de la columna eventmod ya que si ésta toma el valor de 1= el evento fue exitoso, se le suma un uno al contador y el total es lo que se presenta en el reporte.

CONEXIONES FALLIDAS
Es el conteo que se hace del número de conexiones realizadas al servidor para un solo evento, este resultado igual que el anterior depende de la columna eventmod si ésta toma los valores de 0 = ninguna modificación para este evento o el valor de 2 = el evento falló, se le suma un uno al contador y el total es lo que se presenta en el reporte.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

FECHA DE HOY [13/02/2006] REPORTE DEL MES DE [FEBRERO] HORA [12:51:16]

REPORTE DE CUENTAS DADAS DE ALTA Y ESTADISTICAS DEL SERVIDOR [POLARIS]

LOGIN	NOMBRE	ESTADO	BASE DATOS	DESDE	CPU UTIL	% CPU	E/S UTIL	% E/S
XXXXX	XXXXXX XXXXXXXX XXX	NORMAL	master	Mar 10 2004	0	0.0000%	0	0.0000%
XXXXX	XXXX XXXXXX XXX	BLOQUEADO	tempdb	Jul 8 2004	0	0.0000%	0	0.0000%
XXXXX	XXXXXX XXXXXX XXXX	NORMAL	tempdb	Jul 8 2004	3383	0.0478%	229858	0.2129%
XXXXX	XXXXX XXXXXXXX	NORMAL	tempdb	Jul 8 2004	187379	2.6475%	3053010	2.8283%
XXXXX	XXXXXXX XXXXX XXXXXXXX	NORMAL	tempdb	Jul 8 2004	8503	0.1201%	1463751	1.3560%
XXXXX	XXXXXXXX XXXXXX XXX	BLOQUEADO	tempdb	Jul 8 2004	0	0.0000%	0	0.0000%
XXXXX	XXXXXXXXX XXXXXXX XXXXXXX	NORMAL	tempdb	Jul 8 2004	0	0.0000%	0	0.0000%
XXXXX	XXXXXXXX XXXXXXXXX XXX	NORMAL	tempdb	Jul 8 2004	2137196	30.1966%	9944032	9.2122%
XXXXX	XXXXXXXXXX XXXXXXXX XXXX	NORMAL	tempdb	Jul 8 2004	3458	0.0489%	170363	0.1578%
XXXXX	XXXXXXXXXXX XXXXXXX XXXXX	NORMAL	tempdb	Jul 8 2004	0	0.0000%	0	0.0000%
XXXXX	XXXXXXXX XXXXXXX XXXX	NORMAL	tempdb	Jul 8 2004	412438	5.8274%	2112292	1.9569%
XXXXX	XXXXXXX XXXXXX XXXXXX	NORMAL	tempdb	Jul 8 2004	6	0.0001%	16913	0.0157%
XXXXXXX	XXXXXXX XXXXXXX	BLOQUEADO	tempdb	Jul 8 2004	0	0.0000%	0	0.0000%
XXXX	XXXXX XXXXX XXXXXXX XXX	NORMAL	master	May 23 2005	0	0.0000%	0	0.0000%
TOTAL DE CPU UTILIZADO : 7077599			TOTAL DE E/S UTILIZADO : 107943728					

El reporte anterior lleva por nombre *AUDITORIA_POLARIS_02132001_ CUENTAS_EST.txt*. Las columnas que se obtienen son de la tabla syslogins. Este reporte lleva un control de todas las cuentas que se encuentran dadas de alta en el servidor, además proporciona una estadística de la cantidad de CPU acumulado y de tiempo de I/O acumulado por cada login.

Procedemos a explicar las columnas contenidas en el reporte y de donde se tomó dicha información.

LOGIN
Nombre de la cuenta de login

NOMBRE
Proviene de la columna full name de la tabla syslogins. Es decir, es el nombre completo de la cuenta de login.

ESTADO
Estado de la cuenta. Revisar el anexo 4. Tablas del Sistema.

BASE DATOS
Nombre de la base de datos por default en la cual se colocará el usuario al entrar al servidor.

Para las siguientes columnas hace uso del procedimiento sp_reportstats, el cual consulta a la tabla syslogins en las siguientes columnas:

DESDE
Se refiere a la columna accdate de la tabla syslogins. Y se refiere a la fecha de última limpieza de las columnas totcpu y totio.

CPU UTIL
Tiempo de CPU acumulado por el login

%CPU
El procedimiento realiza las operaciones necesarias para sacar el porcentaje de CPU acumulado

E/S UTIL
Tiempo de I/O acumulado por el login

% E/S
El procedimiento realiza las operaciones necesarias para sacar el porcentaje de E/S acumulado

Finalmente realiza la suma total de tiempo de CPU acumulado y la suma total del tiempo de I/O acumulado.

```
 ------------------------------------------------------------------------------
 ------------------------------------------------------------------------------
|                    REPORTE DE SERVIDORES (INFORMACION GENERAL)                |
 ------------------------------------------------------------------------------
 ------------------------------------------------------------------------------
|  NOMBRE DE EL SERVIDOR |    POLARIS                                           |
 ------------------------------------------------------------------------------
|  VERSION --> Adaptive Server Enterprise/12.5/SWR 9609 GA/P/Sun_svr4/OS 5.8/main/1647/32-bit/FBO/Sat Jun  2 00:49:20 2001
 ------------------------------------------------------------------------------
|  SERVIDOR DE RESPALDO  |            POLARIS_back                              |
 ------------------------------------------------------------------------------
|  OTRO  SERVIDOR        |              POLARIS |
|  OTRO  SERVIDOR        |            EJBServer |
 ------------------------------------------------------------------------------------------------------------------------------
|                                                                                                                              |
|                                                                                                                              |
|                                                                                                                              |
 ------------------------------------------------------------------------------------------------------------------------------
|                                                                                                                              |
 ------------------------------------------------------------------------------------------------------------------------------
```

El reporte anterior contiene información general del servidor, y lleva por nombre *AUDITORIA_POLARIS_02132001_INFO_SERVIDOR.txt*. Procedemos a explicar las columnas contenidas en el reporte.

NOMBRE DEL SERVIDOR
En este caso se refiere al servidor local. También nos muestra la versión que está instalada del Adaptive Server, además proporciona información acerca del sistema operativo.

SERVIDOR DE RESPALDO
Como su nombre lo indica se refiere al servidor de respaldo (Backup Server).

OTRO SERVIDOR
En la cual se muestran todos los servidores que estén dados de alta.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

SISTEMA INTEGRAL DE ADMINISTRACION ESCOLAR

FECHA DE HOY [13/02/2006] REPORTE DEL MES DE [FEBRERO] HORA [12:51:16]

REPORTE DE DISPOSITIVOS CONTENIDOS EN EL SERVIDOR [POLARIS]

NOMBRE DE EL DISPOSITIVO	NOMBRE FISICO DE EL DISPOSITIVO	TAMANO	OCUPADO	LIBRE
master	/opt/sybase-12.5/serv_dev/POLARIS	60.0MB	27.0MB	33.0MB
S01	/dev/rdsk/c1t11d0s1	10000MB	4750.0MB	5250.0MB
S02	/dev/rdsk/c1t11d0s3	10000MB	2630.0MB	7370.0MB
S03	/dev/rdsk/c1t11d0s4	10000MB	3273.0MB	6727.0MB
S04	/dev/rdsk/c1t11d0s5	10000MB	4310.0MB	5690.0MB
S05	/dev/rdsk/c1t11d0s6	14000MB	4238.0MB	9762.0MB
S06	/dev/rdsk/c1t11d0s7	14000MB	1501.0MB	12499MB
sysprocsdev	/opt/sybase-12.5/serv_dev/POLARIS	120.0MB	120.0MB	0.0MB
TOTALES DE DISPOSITIVOS:		68180.00MB	20849.00MB	47331.00MB

El reporte anterior da informe de todos los dispositivos contenidos en el servidor Polaris esta información se obtiene del la tabla sysdevices ya que esta tabla contiene una fila por cada dispositivo. Este reporte lleva por nombre *AUDITORIA_POLARIS_02132001_DISPOSITIVOS.txt*.

NOMBRE DE EL DISPOSITIVO
Es el nombre lógico del dispositivo, esta información se obtiene de la columna name que se encuentra en la tabla **sysdevices.**

NOMBRE FISICO DE EL DISPOSITIVO
Es el nombre físico del dispositivo, es el nombre real del dispositivo en el sistema operativo; esta información se obtiene de la columna phyname que se encuentra en la tabla **sysdevices.**

TAMAÑO
Es el tamaño con que el dispositivo se inicializa. La información se obtiene de las tablas de sistema sysdatabases, sysdevices y sysusages.

OCUPADO
Tamaño del dispositivo que ya ha sido ocupado. La información se obtiene de las tablas de sistema sysdatabases, sysdevices y sysusages.

LIBRE
Tamaño del dispositivo que queda libre. La información se obtiene de las tablas de sistema sysdatabases, sysdevices y sysusages.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

FECHA DE HOY [13/02/2006] REPORTE DEL MES DE [FEBRERO] HORA [12:51:16]

REPORTE DE BASES DE DATOS CONTENIDAS EN EL SERVIDOR [POLARIS]

NOMBRE	DUEÑO	CREADA	MB DATOS	MB LIBRES	% LIBRE	MB D LOG	LOG LIBRE	% LIBRE	DISPOSITIVOS
pubs001	alpha1	03/22/2004	280	61.2	22	45	45.4	100	DAT -> S01 LOG -> S05
pubs002	alpha1	03/22/2004	210	20.5	10	45	39.6	88	DAT -> S01 LOG -> S05
pubs003	alpha1	03/22/2004	170	6.4	4	36	36.7	100	DAT -> S01 LOG -> S05 LOG -> S05
master	sa	02/12/2004	20	9.8	49	XXXXXX	XXXXXX	XXXXXX	D&L -> master
model	sa	02/12/2004	2	0.8	41	XXXXXX	XXXXXX	XXXXXX	D&L -> master
prueba	alpha1	07/10/2004	200	194.7	97	60	61.1	100	LOG -> S04 DAT -> S05
db1	alpha1	07/10/2004	250	250.8	100	80	81.6	100	LOG -> S04 DAT -> S05
sybsecurity	alpha1	07/10/2004	500	406.5	81	150	144.0	96	LOG -> S04 DAT -> S06
sybsystemdb	sa	02/12/2004	2	0.7	37	XXXXXX	XXXXXX	XXXXXX	D&L -> master
sybsystemprocs	sa	02/12/2004	120	72.1	60	XXXXXX	XXXXXX	XXXXXX	D&L -> sysprocsde
tempdb	sa	01/05/2006	1000	1005.5	101	XXXXXX	XXXXXX	XXXXXX	--------> master D&L -> S06

El reporte anterior lleva por nombre *AUDITORIA_POLARIS_02132001_BASES.txt* para obtener la información del reporte fue necesario hacer uso de las tablas sysdatabases, syslogins, sysusages y sysdevices, a continuación se explican brevemente las columnas, para mayor referencia consultar el capítulo 3 y el anexo 4:

NOMBRE
Contiene el nombre de la base de datos que fue auditada

DUEÑO
Es el nombre del dueño de la base de datos

CREADA
Proviene de la columna crdate de la tabla sysdatabases, y se refiere a la fecha en que se creó la base de datos

MB DATOS
Es el tamaño en megabytes, de la base de datos

MB LIBRES
Es la cantidad de espacio disponible de la base de datos

%LIBRE
Es el porcentaje de espacio disponible

MB D LOG
Es el tamaño en megabytes, del log de transacciones de la base de datos

LOG LIBRE
Es la cantidad de espacio disponible del log de transacciones

% LIBRE
Es el porcentaje de espacio disponible en el log de transacciones

DISPOSITIVOS
Se refiere a los dispositivos en los cuales está montada la base de datos.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
DEPARTAMENTO DE ADMINISTRACION DE SERVIDORES

FECHA DE HOY [13/02/2006] REPORTE DEL MES DE [FEBRERO] HORA [12:51:16]

REPORTE DE PARAMETROS DE CONFIGURACION DIFERENTES A LOS VALORES DE DEFAULT EN [POLARIS]

NOMBRE DEL PARAMETRO	DEFAULT	MEM USADA	VALOR CONF	VALOR EJEC
allow resource limits	0	#1517	1	1
auditing	0	0	1	1
configuration file	0	0	0	/opt/sybase
default sortorder id	50	0	57	57
executable codesize + overhead	0	487778	0	487778
max memory	33792	1000000	500000	500000
max online engines	1	1067484	2	2
max parallel degree	1	0	25	25
max scan parallel degree	1	0	25	25
number of aux scan descriptors	200	#612	600	600
number of devices	10	#14	20	20
number of locks	5000	5836	40000	40000
number of open databases	12	21234	200	200
number of open indexes	500	3508	3000	3000
number of open objects	500	3959	3000	3000
number of user connections	25	76641	600	600
number of worker processes	0	12885	100	100
o/s file descriptors	0	0	0	4096
procedure cache size	3271	86550	40960	40960
total data cache size	0	655698	0	655698
total logical memory	33792	941162	470581	470581
total physical memory	0	927816	0	463908

El reporte anterior da información acerca de los parámetros de configuración que han sido cambiados en sus valores de default, el título de este reporte es *AUDITORIA_POLARIS_02132001_PARAMETROS_CONF.txt*

La información que aquí se presenta se tomó de la tabla de syscurconfigs la cual sólo se encuentra en la base de datos master es construida dinámicamente. Esta contiene una entrada de cada parámetro de configuración y de sysconfigures la cual contiene un renglón por cada parámetro de configuración.

NOMBRE DEL PARÁMETRO
Es el nombre del parámetro de configuración. Este dato se obtiene de la columna comment que se encuentra en la tabla de sysconfigures.

DEFAULT
Es el valor por default del parámetro de configuración. Este dato se obtiene de la columna defvalue que se encuentra en la tabla syscurconfigs.

MEM USADA
Valor entero de la cantidad de memoria utilizada. Esta información se obtiene de la columna memory_used que contiene los valores para cada parámetro de configuración. Memory_used se encuentra en la tabla syscurconfigs.

VALOR CONF
Es el valor del parámetro modificado por el usuario y es de tipo entero. Si el valor de la columna es 0 el parámetro es de tipo carácter. Esta información se obtiene de la columna value que se encuentra en la tabla sysconfigures.

VALOR EJEC
Es el valor de tipo entero del parámetro en ejecución. Si este valor del parámetro es 0 el significa que dato es de tipo carácter. Esta información se obtiene de la columna value que se encuentra en la tabla syscurconfigs.

CONCLUSIONES

Con la realización de este trabajo, la conclusión a la que hemos podido llegar, es que toda empresa, pública o privada, que cuente con una Base de Datos medianamente compleja, se debe someter a un control estricto de evaluación de eficacia y eficiencia; ya que su éxito dependerá de esas características, pues aunque tenga un staff de gente de primera, si tiene un sistema propenso a errores, lento, vulnerable e inestable; y no hay un balance entre estas dos cosas, la empresa nunca saldrá adelante.

Debido a esta situación y además con la carencia de una herramienta propia que ayude a efectuar una auditoría que cumpla con todas las necesidades que la empresa requiere es que se tomó la decisión de crear una Herramienta Auditora cuyo principal objetivo sea el de supervisar y controlar entradas y salidas, comandos ejecutados, errores ocurridos, procedimientos específicos, procedimientos generales, dispositivos, objetos, bases de datos y de características propias del servidor como son sus valores de configuración y de esta manera poder garantizar integridad, autenticidad y confidencialidad de los datos.

Por lo anterior es que se puede decir que el objetivo de la tesis presentada si se cumplió ya que este sistema brinda un monitoreo que será a través de un proceso automatizado basado en los cuatro niveles de actividad a ser auditadas:

- A nivel Servidor
- A nivel Base de Datos
- A nivel Objetos
- A nivel Usuario

De esta manera la persona que se dedique a la profesión de Auditor del servidor tendrá una herramienta imprescindible para que con unos conocimientos mínimos pueda ser capaz de detectar las alteraciones en los principales procesos de nuestro sistema dando un diagnóstico mucho más preciso.

Con esto no queremos decir que se van a eliminar los riesgos y las vulnerabilidades, pero si se podrán detectar fallas a tiempo y tomar las medidas necesarias para que éstas no causen ningún daño severo dentro de nuestra empresa.

Entre las recomendaciones que podemos dar es que debido a la delicadeza de la información es necesario que la actividad de Auditoría sea realizada por un

usuario que tenga ciertos privilegios dentro del servidor, como por ejemplo el rol sso ya que este rol es el encargado de la seguridad del sistema, además de esto tiene que ser una persona de mucha confianza y sobre todo con ética.

Por último podemos decir que este trabajo no solo sirvió para concluir los estudios a nivel licenciatura, si no que también nos ayudó adquirir conocimiento para un buen desarrollo profesional en el área de administración de servidores de base de datos.

GLOSARIO DE TÉRMNOS.

Adaptive Server: El servidor en la arquitectura cliente/servidor de Sybase Adaptive Server maneja múltiples bases de datos y múltiples usuarios, guarda la ruta de la ubicación actual de los datos en el disco, conserva mapas de la descripción lógica de los datos para almacenamiento físico, y mantiene los datos y procedimientos en la memoria cache.

Administrador del Sistema: Usuario Autorizado para manejar la administración del sistema del SQL Server, incluida la creación de cuentas de usuarios, asignación de permisos y creación de nuevas bases de datos.

Activos de información: Aquellos componentes de la Institución (tangible e intangible) que son parte del patrimonio de la misma y necesitan ser resguardados. Se incluye todo aquel bien material, humano, equipamiento o de información para la ejecución de las actividades de la Institución.

Alias: Apodo o Pseudónimo - Nombre usualmente corto y fácil de recordar que se utiliza en lugar de otro nombre usualmente largo y difícil de memorizar.

Algoritmo: Descripción exacta de la secuencia en que se ha de realizar un conjunto de actividades tendientes a resolver un determinado tipo de problema o procedimiento**.**

Amenazas: Se definen como "los eventos que pueden desencadenar un incidente en la Institución, produciendo daños materiales o pérdidas en sus activos". Las amenazas pueden materializarse y transformarse en agresiones.

Análisis de Riesgos: Es la actividad relacionada con la identificación, análisis, y atenuación de los riesgos antes de que empiecen a amenazar el funcionamiento continuo y confiable de los sistemas de información.

Archivo: Colección de registros almacenados siguiendo una estructura homogénea.

ASCII: American Standard Code for Information Interchange. Es de facto el estándar del World Wide Web para el código utilizado por computadoras para representar todas las letras (mayúsculas, minúsculas, letras latinas, números, signos de puntuación, etc.). El código estándar ASCII es de 128 letras representadas por un digito binario de 7 posiciones (7 bits), de 0000000 a 1111111.

Atributo: Puede ser definido como una función que transforma un conjunto de entidades o relaciones.

Auditoría: Es una técnica especializada, orientada al examen de evaluación de los sistemas de información automatizadas, verificando la corrección y confiabilidad de la información procesada por medios electrónicos, de acuerdo a normas técnicas y reglamentarias.

Backup: Copia de seguridad o respaldo de una base de datos, servidor, computadora o archivos.

Base de datos: Es un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo.

Bit (Dígito Binario): Unidad mínima de almacenamiento de la información cuyo valor puede ser 0 ó 1; o bien verdadero o falso.

Bloqueo: Proceso de restringir el acceso a los recursos en un entorno multiusuario para mantener la seguridad e impedir problemas de acceso simultáneo. SQL Server aplica automáticamente bloqueos a las tablas o páginas.

Byte: Conjunto de 8 bits el cual suele representar un valor asignado a un carácter.

C/ C++: Lenguajes de programación (orientado a objetos en el caso de C++) utilizados en el WWW a través de un CGI, principalmente para realizar consultas a bases de datos tipo Oracle, SQL-Server, SyBase, etc; o a herramientas locales como WAIS. Generalmente el servidor donde se encuentra el programa funciona en ambiente UNIX.

Cache: Es un tipo de memoria especial, más rápida que la RAM normal (y mas cara), que se pone en el camino de los datos que van del procesador a la memoria RAM. Así, toda información que va de la RAM al procesador se deja almacenada temporalmente en la memoria cache.

Cliente: Aplicación que permite a un usuario obtener un servicio de un servidor localizado en la red. Sistema o proceso el cual le solicita a otro sistema o proceso la prestación de un servicio.

Código Fuente (Source Code): Conjunto de instrucciones que componen el programa informático mediante el cual se elabora un sitio web. Estos programas se escriben en determinados lenguajes como, por ejemplo, el HTML.

Contraseña (Password): Conjunto de caracteres alfanuméricos que le permite a un usuario el acceso a un determinado recurso o la utilización de un servicio dado. Se destaca que la contraseña no es visible en la pantalla al momento de ser tecleada con el propósito de que sólo pueda ser conocida por el usuario.

Copia de Respaldo o Seguridad (Backup): Acción de copiar archivos o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los

originales. Esta sencilla acción evita numerosos, y a veces irremediables, problemas si se realiza de forma habitual y periódica.

Dato: Unidad mínima que compone cualquier información.

Dump: La acción de hacer un backup de una entidad de base de datos, incluyendo lo datos y el log de transacciones, el cual va con el comando dump de la base de datos. También, los datos que resultan de esa acción.

Encriptación (Cifrado): Tratamiento de un conjunto de datos, contenidos o no en un paquete, a fin de impedir que nadie excepto el destinatario de los mismos pueda leerlos. Hay muchos tipos de cifrado de datos, que constituyen la base de la seguridad de la red.

Gigabyte (GB): Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un GB corresponde a 1.024 millones de bytes.

Hacker: Persona que accede a la información de una computadora de forma no autorizada.

Hardware (Maquinaria): Componentes físicos de una computadora o de una red, a diferencia de los programas o elementos lógicos que los hacen funcionar.

IDS (Intrusion Detection System): Un sistema de detección de intrusiones es un equipo de seguridad especializado en la protección de las redes. La misión de un sistema IDS es detectar posibles ataques a los componentes de la red interna (equipos de cómputo o elementos de red), que no han sido detectados por los firewalls debido a su naturaleza de tráfico permitido desde el punto de vista de acceso a la red.

Impacto: Se define como "daño producido a la Institución por un posible incidente" y es el resultado de la agresión sobre el activo. El impacto puede ser cuantitativo (si representa pérdidas cuantitativas monetarias directas o indirectas); cualitativo con pérdidas orgánicas (por ejemplo, daño de personas); y cualitativo con pérdidas funcionales.

Información de Acceso Público; Información de Acceso Autorizado: Toda aquella información que no presenta riesgos para la Institución, y cuyo acceso es libre a cualquier usuario que la necesite para el desarrollo de sus tareas habituales. Esta información es la que tiene menor exigencia de controles y seguridad.

Información Restringida: Toda aquella información que puede presentar riesgos para la Institución, y cuyo acceso debe ser expresamente autorizado por el dueño de los datos y restringido a un grupo reducido de usuarios que la necesite para el

desarrollo de sus tareas habituales. Esta información requiere un nivel medio de controles y seguridad.

Información Confidencial o Sensible: Toda aquella información de acceso autorizado que puede presentar riesgos importantes para la Institución, y que debe cumplir con medidas adicionales a las definidas para acceso autorizado. Esta información tiene una mayor exigencia de controles y seguridad.

Indices: Se usan en las aplicaciones de bases de datos para indicar la operación de ordenar los registros contenidos en ella de manera especial, en función de unos parámetros definidos previamente

Interfaz (Interface): Zona de contacto o conexión entre dos componentes de "hardware"; entre dos aplicaciones; o entre un usuario y una aplicación. Apariencia externa de una aplicación informática.

Internet: Es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP.

Intranet: Es una red local que utiliza herramientas de Internet. Puede considerarse como una Internet privada que funciona dentro de una Institución. Normalmente, dicha red local tiene como base el protocolo TCP/IP de Internet y utiliza un sistema de firewall que no permite acceder a la misma desde el exterior.

IP: Internet Protocol, Protocolo de Internet. Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet. Se han desarrollado diferentes familias de protocolos para comunicación por red de datos para los sistemas UNIX. El más ampliamente utilizado es el Internet Protocol Suite, comúnmente conocido como TCP / IP. Es un protocolo DARPA que proporciona transmisión fiable de paquetes de datos sobre redes. El nombre TCP / IP proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). Todos juntos llegan a ser más de 100 protocolos diferentes.
IP es la dirección numérica de una computadora en Internet de forma que cada dirección electrónica se asigna a una computadora conectada a Internet y por lo tanto es única. La dirección IP esta compuesta de cuatro octetos como por ejemplo, 132.248.53.10

Firewall: Sistema de protección de accesos no autorizados a las redes organizacionales. Su misión es controlar mediante mecanismos de filtrado de las comunicaciones el acceso a la información (quién entra en el sistema y a qué información accede).

Kilobyte: Unidad de medida de la capacidad de transmisión de una línea de telecomunicación equivalente a mil bytes aunque actualmente es usado como 1024 (dos elevado a la 10) bytes.

Lineamiento de seguridad: Es una declaración formal de las reglas por las cuales, el personal que tiene acceso a los activos de tecnología de información, se debe conducir.

Log: Archivo que registra movimientos y actividades de un determinado programa (log file).

Megabits por Segundo (Mbps): Unidad de medida de la capacidad de transmisión por una línea de telecomunicación donde cada Megabit está formado por 1.048.576 bits.

Mensaje de error: Mensaje que SQL Server presenta, normalmente en la terminal del usuario, cuando detecta una condición de error.

Mirror (Espejo): Término utilizado en Internet para hacer referencia a un servidor FTP, página web o cualquier otro recurso cuyo contenido es una copia exacta de otro. Estos mirrors se realizan automáticamente y en una frecuencia determinada de forma que pretenden tener una copia exacta del lugar del que hacen mirror.

Modificación de datos: Adición, eliminación o cambio de información en las bases de datos con los comandos insert, delete y update.

Motor: Proceso que ejecuta a un SQL Server comunicado con los procesos de otro servidor usando memoria compartida. Un motor puede describirse como el valor de la capacidad de procesamiento de una CPU. También es conocido como "motor servidor". Un SQL Server que se ejecute en una máquina de un solo procesador siempre tendrá un motor, el motor 0. Un SQL Server que se ejecute en una máquina de varios procesadores puede tener uno o más motores.

Nulo: Que no tiene un valor asignado explícitamente. NULL no equivale a cero ni a un espacio en blanco. Un valor de NULL no se considera mayor que, menor que, ni equivalente a ningún otro valor, incluso ningún otro valor de NULL.

Número de estado de error: Número vinculado a un mensaje de error de SQL Server que permite identificar inequívocamente la línea de código de SQL Server en la que se produjo el error.

Número de mensaje: Número que identifica inequívocamente un mensaje de error.

Número de nivel de gravedad: Gravedad de una condición de error: los errores con niveles de gravedad 19 y superiores son errores fatales.

Password: O contraseña. Se denomina así al método de seguridad que se utiliza para identificar a un usuario. Es frecuente su uso en redes, sistemas operativos y DBMS. Se utiliza para dar acceso a personas con determinados permisos.

Parámetro: Argumento de un procedimiento almacenado.

Partes de asignación de disco: Grupos de las unidades de asignación a partir de los cuales SQL Server construye un nuevo archivo de base de datos. El tamaño mínimo de una parte de asignación de disco es una unidad de asignación, o sea 256 páginas de 2k.

Permiso: Autoridad para realizar ciertas acciones sobre ciertos objetos de base de datos o para ejecutar ciertos comandos.

Procedimientos almacenados: Es un conjunto de sentencias SQL, que han sido almacenados en una base de datos que pueden ser ejecutados por su nombre.

Propietario de base de datos: Usuario que crea una base de datos. El propietario de una base de datos tiene control sobre todos los objetos de esa base de datos. El nombre de login del propietario de la base de datos es “dbo”.

Protocolo: Descripción formal de formatos de mensaje y de reglas que dos ordenadores deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina a máquina o intercambios de alto nivel entre programas de asignación de recursos.

Query: Consulta. Búsqueda en una base de datos.

RAM: Random Access Memory (memoria de acceso aleatorio). Por lo general el término RAM es comprendido generalmente como la memoria volátil (los datos e instrucciones se borran al apagarse la PC) que puede ser escrita y leída. La memoria del equipo permite almacenar datos de entrada, instrucciones de los programas que se están ejecutando en ese momento, los datos resultados del procesamiento y los datos que se preparan para la salida.

Recursos de sistemas de información: Los recursos incluyen, pero no se limitan a, todas las computadoras, servidores, bases de datos, programas y códigos fuente, así como toda la información en papel y cualquier información de uso interno.

Reglas de normalización: Reglas estándar del diseño de bases de datos en un sistema de administración de bases de datos relacionales.

Riesgo: Se define como la "posibilidad de que se produzca un impacto dado en la Institución".

Roles: Son conjuntos de privilegios que pueden concederse ‘de golpe’ a un usuario. Los Privilegios pueden ser de Objetos (para INSERT, SELECT, UPDATE, DELETE, EXECUTE…) o del Sistema (para crear tablas, vistas…).

Seguridad física: Significa tener el control total y a cualquier instante de la computadora. Estos métodos incluyen el acceso por clave que no permiten, incluso, el reinicio de la computadora como medida de seguridad.

Servidor: Un nodo de red que proporciona servicios a PCs clientes; por ejemplo, acceso a archivos, centro de impresión o ejecución remota.

Software: Conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadoras, es decir, la parte intangible o lógica de una computadora.

Software Malicioso: Cualquier programa, documento o mensaje, susceptible de causar perjuicios a los usuarios de sistemas informáticos.

Spyware: Los programas espía o aplicaciones que recopilan información sobre una persona u Institución sin su conocimiento.

Tabla de Sistema: Una de las tablas de diccionario de datos. Las tablas del sistema controlan información sobre el SQL Server en conjunto y sobre cada base de datos de usuario. La base de datos master contiene algunas tablas de sistema que no están en las bases de datos de usuario.

TCP/IP: Es un conjunto de protocolos de comunicación que definen cómo pueden comunicarse entre sí equipos de cómputo y otros dispositivos de distinto tipo.

Transacción: Mecanismo que garantiza que un grupo de acciones se trate como una sola unidad de trabajo.

Valor Predeterminado: Opción elegida por el sistema cuando no se especifica ninguna otra opción.

Violación de seguridad: Todo acto que ponga en riesgo a los activos de información de la Institución.

Virus: Los virus son programas que pueden introducirse en las computadoras y sistemas informáticos de formas muy diversas, produciendo efectos molestos, nocivos e incluso destructivos e irreparables.

Vulnerabilidades: Se define como la "ocurrencia real de materialización de una amenaza sobre un activo", la vulnerabilidad es una propiedad de la relación entre un activo y una amenaza.

Anexo 1. Procedimientos Almacenados del Sistema

1. **sp_activeroles**

   Descripción: Despliega todos los roles activos, otorgados a un login de usuario.

   ```
   sp_activeroles [expand_down]
   ```

2. **sp_addalias**

   Descripción: Permite que un usuario del Adaptive Server sea conocido dentro de una base de datos como otro usuario.

   ```
   sp_addalias loginame, name_in_db
   ```

3. **sp_addauditrecord**

   Descripción: Permite introducir registros de auditoria definidos por el usuario (comentarios) en la lista de auditoria.

   ```
   sp_addauditrecord [ text [, db_name [, obj_name
   [, owner_name [, dbid [, objid] ] ] ] ] ]
   ```

4. **sp_addaudittable**

   Descripción: Añade otra tabla de auditoria de sistema después de instalar la auditoria.

   ```
   sp_addaudittable devname
   ```

5. **sp_addengine**

   Descripción: Agrega un motor a un grupo motor existente, si no existe el grupo, lo crea y agrega el motor.

   ```
   sp_addengine engine_number, engine_group
   ```

6. **sp_addexeclass**

   Descripción: Crea o actualiza una clase de ejecución definida por el usuario que se pueda vincular hacia una aplicación cliente, a las conexiones y a los procedimientos almacenados.

   ```
   sp_addexeclass classname, priority, timeslice, engine_group
   ```

7. **sp_addextendedproc**

   Descripción: Crea un procedimiento almacenado extendido (ESP), en la base de datos master.

   ```
   sp_addextendedproc esp_name, dll_name
   ```

8. sp_addexternlogin

Descripción: Únicamente los Servicios de Integración de Componentes. Crea una cuenta de login y password alternos para utilizarse cuando se comunique con un servidor remoto mediante los Servicios de Integración de Componentes.

```
sp_addexternlogin server, loginname, externname [, externpassword]
```

9. sp_addgroup

Descripción: Añade un grupo a una base de datos. Los grupos se utilizan como nombres colectivos para conceder y revocar privilegios.

```
sp_addgroup grpname
```

10. sp_addlanguage

Descripción: Define los nombres de los meses, los días y su formato de fecha, para un idioma alterno.

```
sp_addlanguage language, alias, months, shortmons, days, datefmt,
datefirst
```

11. sp_addlogin

Descripción: Añade una nueva cuenta de usuario al Adaptive Server.

```
sp_addlogin loginame, passwd [, defdb]
 [, deflanguage ] [, fullname ][, passwdexp ]
 [, minpwdlen ] [, maxfailedlogins ]
```

12. sp_addmessage

Descripción: Añade mensajes definidos por el usuario a sysusermessages para ser utilizados por el procedimiento almacenado ***print***, llamadas ***raiserror***, y ***sp_bindmsg***.

```
sp_addmessage message_num, message_text [, language
  [, with_log [, replace] ] ]
```

13. sp_addobjectdef

Descripción: Especifica la correlación entre una tabla local y una restricción de almacenamiento externo. Únicamente los Servicios de Integración de Componentes.

```
sp_addobjectdef tablename, " objectdef" [," objecttype"]
```

14. sp_add_qpgroup

Descripción: Agrega un grupo de plan abstracto.

```
sp_add_qpgroup new_name
```

15. sp_addremotelogin

Descripción: Autoriza a un nuevo usuario del servidor remoto agregando una entrada a master.dbo.sysremotelogins.

```
sp_addremotelogin remoteserver [, loginame [, remotename] ]
```

16. sp_add_resource_limit

Descripción: Crea un límite en el número de los recursos del servidor que se pueden utilizar por una cuenta y/o una aplicación para ejecutar un query, query batch o una transacción.

```
sp_add_resource_limit name, appname, rangename, limittype, limitvalue
[, enforced [, action [, scope ] ] ]
```

17. sp_addsegment

Descripción: Define un segmento sobre un dispositivo de base de datos en la base de datos actual.

```
sp_addsegment segname, dbname, devname
```

18. sp_addserver

Descripción: Define un servidor remoto, o el nombre del servidor local.

```
sp_addserver lname [, class [, pname] ]
```

19. sp_addthreshold

Descripción: Crea un umbral para controlar el espacio en un segmento de base de datos. Cuando el espacio libre de un segmento se encuentra por debajo del nivel especificado, Adaptive Server ejecuta el procedimiento almacenado asociado.

```
sp_addthreshold dbname, segname, free_space, proc_name
```

20. sp_add_time_range

Descripción: Agrega un nombre de rango de tiempo a un Adaptive Server.

```
sp_add_time_range name, startday, endday, starttime, endtime
```

21. sp_addtype

Descripción: Crea un tipo de datos definido por el usuario.

```
sp_addtype typename, phystype [ ( length) | ( precision [, scale] ) ]
[, " identity" | nulltype ]
```

22. sp_addumpdevice

Descripción: Añade un dispositivo de volcado al Adaptive Server.

```
sp_addumpdevice { " tape" | " disk" }, logicalname, physicalname
  [, tapesize ]
```

23. sp_adduser

Descripción: Añade un usuario nuevo a la base de datos actual.

```
sp_adduser loginame [, name_in_db [, grpname ] ]
```

24. sp_altermessage

Descripción: Habilita e inhabilita el registro de un mensaje especifico definido por el sistema o definido por el usario en el error log del Adaptive Server.

```
sp_altermessage message_id, parameter, parameter_value
```

25. sp_audit

Descripción: Permite al Oficial de Seguridad del Sistema configurar las opciones de auditoria.

```
sp_audit option, login_name, object_name [, setting ]
```

26. sp_autoconnect

Descripción: Define una conexión passthrough a un servidor remoto para un usuario especifico, lo cual permite que el usuario elegido acceda en modo passthrough automáticamente en la cuenta. Únicamente los Servicios de Integración de Componentes.

```
sp_autoconnect server, { true | false } [, loginame]
```

27. sp_bindcache

Descripción: Vincula una base de datos, tabla, índice, objeto text u objeto image a un caché de datos.

```
sp_bindcache cachename, dbname [, [ ownername.] tablename
[, indexname | "text only"] ]
```

28. sp_bindefault

Descripción: Vincula un valor predeterminado definido por el usuario a una columna o tipo de datos definido por el usuario.

```
sp_bindefault defname, objname [, futureonly]
```

29. sp_bindexeclass

Descripción: Asocia una clase de ejecución con una aplicación cliente, una cuenta, o un procedimiento almacenado.

```
sp_bindexeclass " object_name", " object_type", " scope", " classname"
```

30. sp_bindmsg

Descripción: Vincula un mensaje de usuario a una restricción de integridad de referencia o a una restricción de verificación.

```
sp_bindmsg constrname, msgid
```

31. sp_bindrule

Descripción: Vincula una regla a una columna o a un tipo de datos definido por el usuario.

```
sp_bindrule rulename, objname [, futureonly]
```

32. sp_cacheconfig

Descripción: Crea, configura, reconfigura y omite cachés de datos, y proporciona información sobre ellos.

```
sp_cacheconfig [ cachename [, " cache_size[P|K|M|G]"]
  [, logonly|mixed][,strict | relaxed ] ]
  [, "cache_partition=[1|2|4|8|16|32|64]"]
```

33. sp_cachestrategy

Descripción: Habilita o inhabilita la estrategia de recobro previo (E/S grande) y de sustitución de cachés MRU para una tabla, índice, objeto text u objeto image.

```
sp_cachestrategy dbname, [ ownername.] tablename
  [, indexname | "text only" | "table only"
  [, { prefetch | mru }, { "on" | "off" } ] ]
```

34. sp_changedbowner

Descripción: Cambia el propietario de una base de datos.

```
sp_changedbowner loginame [, true ]
```

35. sp_changegroup

Descripción: Cambia el grupo de un usuario.

```
sp_changegroup grpname, username
```

36. sp_checknames

Descripción: Busca nombres con caracteres que no corresponden al juego ASCII de 7 bits en la base de datos actual.

```
sp_checknames
```

37. sp_checkreswords

Descripción: Detecta y muestra identificadores que son palabras reservadas de Transact-SQL. Verifica los nombres de servidores, nombres de dispositivos, nombres de bases de datos, segmentos, tipos de datos definidos por el usuario, nombres de objetos, columnas, usuarios, logins y logins remotos.

```
sp_checkreswords [ user_name_param]
```

38. sp_checksource

Descripción: Verifica la existencia del texto fuente del objeto compilado.

```
sp_checksource [ objname [, tabname [, username] ] ]
```

39. sp_chgattribute

Descripción: Cambia el valor de **max_rows_per_page**, **fillfactor**, **reservepagegap**, o **exp_row_size** para futuras asignaciones de espacio de una tabla o índice. Definir el **concurrency_opt_threshold** para una tabla.

```
sp_chgattribute objname, {"max_rows_per_page" | "fillfactor" |
  "reservepagegap" | "exp_row_size" concurrency_opt_threshold },
  optvalue
```

40. sp_clearpsexe

Descripción: Limpia los atributos de ejecución de una aplicación cliente, una cuenta o un procedimiento almacenado que fue determinado por sp_setpsexe.

```
sp_clearpsexe spid, exeattr
```

41. sp_clearstats

Descripción: Inicia un nuevo periodo de contabilidad para todos los usuarios del servidor o para un usuario especificado. Imprime estadísticas del periodo anterior mediante la ejecución de sp_reportstats.

```
sp_clearstats [ loginame]
```

42. sp_client_addr

Descripción: Muestra la dirección IP (Internet Protocol) de los clientes conectados, o proceso especifico conectado al Adaptive Server, incluyendo el spid y el hostname del cliente.

```
sp_client addr [ "spid"]
```

43. sp_cmp_all_qplans

Descripción: Compara todos los planes abstractos dentro de dos grupos de planes abstractos.

```
sp_cmp_all_qplans group1, group2 [, mode]
```

44. sp_cmp_qplans

Descripción: Compara dos planes abstractos.

```
sp_cmp_qplans id1, id2
```

45. sp_commonkey

Descripción: Define una clave común (columnas que se combinan con frecuencia) entre dos tablas o vistas.

```
sp_commonkey tabaname, tabbname, col1a, col1b
   [, col2a, col2b, ..., col8a, col8b]
```

46. sp_companion

Descripción: Realiza operaciones cluster tales como configuración del Adaptive Server, como un servidor acompañante secundario en un sistema de alta disponibilidad y mueve un servidor acompañante a partir de un modo failover a otro. sp_companion se ejecuta desde el servidor acompañante secundario.

```
sp_companion
[ server_name
{ ,configure
[, { with_proxydb | NULL } ]
[, srvlogin ]
[, server_password ]
[, cluster_login ]
[, cluspassword ] ]
| drop
| suspend
| resume
| prepare_failback
| do_advisory }
{, all
| help
| group attribute_name
| base attribute_name}
```

47. sp_configure

Descripción: Muestra los parámetros de configuración por grupo, sus valores actuales, sus valores por default, el valor que se le ha fijado recientemente y la cantidad de memoria utilizada por este ajuste.

```
sp_configure [ configname [, configvalue] | group_name |
  non_unique_parameter_fragment] ["p|P|k|K|m|M|g|G"]]
```

o

```
sp_configure "configuration file", 0, {"write" | "read" | "verify"
   | "restore"} "file_name"
```

48. sp_copy_all_qplans

Descripción: Copia todos los planes desde un grupo de plan abstracto hacia otro grupo.

```
sp_copy_all_qplans src_group, dest_group
```

49. sp_copy_qplan

Descripción: Copia un plan abstracto hacia otro grupo de plan abstracto.

```
sp_copy_qplan src_id, dest_group
```

50. sp_countmetadata

Descripción: Muestra el número de índices, objetos, o bases de datos en el Adaptive Server.

```
sp_countmetadata " configname" [, dbname]
```

51. sp_cursorinfo

Descripción: Proporciona información acerca de un cursor específico o de todos los cursores activos en la sesión actual.

```
sp_cursorinfo [{ cursor_level|NULL}] [, cursor_name]
```

52. sp_dboption

Descripción: Muestra o cambia las opciones de base de datos.

```
sp_dboption [ dbname, optname, { true|false } ]
```

53. sp_dbrecovery_order

Descripción: Especifica el orden en la cual se recuperaran las bases de datos de un usuario y enlista el orden de recuperación, definido por el usuario, de una base de datos o todas las bases de datos.

```
sp_dbrecovery_order [ database_name [, rec_order [, force] ] ]
```

54. sp_dbremap

Descripción: Obliga al Adaptive Server a reconocer los cambios realizados por ***alter database***. Ejecute este procedimiento únicamente si así lo indica un mensaje del Adaptive Server.

```
sp_dbremap dbname
```

55. sp_defaultloc

Descripción: Define una restricción para almacenar un valor predeterminado para los objetos de una base de datos local. Únicamente los Servicios de Integración de Componentes.

```
sp_defaultloc dbname, {“ defaultloc" | NULL }[, " defaulttype"]
```

56. sp_depends

Descripción: Muestra información sobre las dependencias de objetos de base de datos: vistas, disparadores y procedimientos que dependan de una tabla o vista especificadas, y las tablas o vistas de las que dependen la vista, disparador o procedimiento especificados.

```
sp_depends objname
```

57. sp_deviceattr

Descripción: Plataformas Unix únicamente. Habilita o deshabilita la opción dsync de un dispositivo de base de datos tipo archivo.

```
sp_deviceattr logicalname, optname, optvalue
```

58. sp_diskdefault

Descripción: Especifica si puede utilizarse o no un dispositivo de base de datos para almacenar la base de datos si el usuario no especifica ningún dispositivo de base de datos o especifica ***default*** con los comandos ***create database*** o ***alter database***.

```
sp_diskdefault logicalname, {defaulton|defaultoff}
```

59. sp_displayaudit

Descripción: Muestra el estado de las opciones de auditoria.

```
sp_displayaudit ["procedure"|"object"|"login"| "database" | "global" |
  "default_object" | "default_procedure" [, name] ]
```

60. sp_displaylevel

Descripción: Define o muestra los parámetros de configuración del Adaptive Server que aparecen en la salida de sp_configure.

```
sp_displaylevel [ loginame [, level] ]
```

61. sp_displaylogin

Descripción: Muestra información sobre una cuenta de login.

```
sp_displaylogin [ loginame [, expand_up | expand_down ] ]
```

62. sp_displayroles

Descripción: Despliega todos los roles concedidos a otro role, o muestra completamente la jerarquía de árbol de los roles en formato de tabla.

```
sp_displayroles [ grantee_name [, mode ] ]
```

63. sp_dropalias

Descripción: Elimina el nombre del alias establecido con sp_addalias.

```
sp_dropalias loginame
```

64. sp_drop_all_qplans

Descripción: Elimina todos los planes abstractos en un grupo de plan abstracto.

```
sp_drop_all_qplans name
```

65. sp_dropdevice

Descripción: Omite un dispositivo de base de datos o un dispositivo de volcado del Adaptive Server.

```
sp_dropdevice logicalname
```

66. sp_dropengine

Descripción: Omite un motor de un grupo motor especificado o si el motor es el último en el grupo, omite el grupo motor.

```
sp_dropengine engine_number, engine_group
```

67. sp_dropexeclass

Descripción: Elimina una clase de ejecución definida por el usuario.

```
sp_dropexeclass classname
```

68. sp_dropextendedproc

Descripción: Remueve un procedimiento almacenado extendido (ESP) de la base de datos master.

```
sp_dropextendedproc esp_name
```

69. sp_dropexternlogin

Descripción: Elimina la definición de un login remoto previamente definido por sp_addexternlogin. Únicamente los Servicios de Integración de Componentes.

```
sp_dropexternlogin remote_server [, login_name]
```

70. sp_dropglockpromote

Descripción: Elimina valores de promoción de bloqueos de una tabla o base de datos.

```
sp_dropglockpromote { "database" | "table" }, objname
```

71. sp_dropgroup

Descripción: Elimina un grupo de una base de datos.

```
sp_dropgroup grpname
```

72. sp_dropkey

Descripción: Quita una clave definida con sp_primarykey, sp_foreignkey o sp_commonkey de la tabla syskeys.

```
sp_dropkey keytype, tabname [, deptabname ]
```

73. sp_droplanguage

Descripción: Omite un idioma alternativo del servidor y quita su fila de master.dbo.syslanguages.

```
sp_droplanguage language [, dropmessages ]
```

74. sp_droplogin

Descripción: Omite un login de usuario del Adaptive Server eliminando la entrada del usuario de master.dbo.syslogins.

```
sp_droplogin loginame
```

75. sp_dropmessage

Descripción: Omite mensajes definidos por el usuario de sysusermessages.

```
sp_dropmessage message_num [, language ]
```

76. sp_dropobjectdef

Descripción: Suprime la correlación de almacenamiento externo proporcionado para un objeto local. Únicamente los Servicios de Integración de Componentes.

```
sp_dropobjectdef " object_name"
```

77. sp_drop_qpgroup

Descripción: Omite un grupo del plan abstracto.

```
sp_drop_qpgroup group
```

78. sp_drop_qplan

Descripción: Omite un plan abstracto.

```
sp_drop_qplan id
```

79. sp_dropremotelogin

Descripción: Omite un login de usuario remoto.

```
sp_dropremotelogin remoteserver [, loginame [, remotename] ]
```

80. sp_drop_resource_limit

Descripción: Quita uno o más limites de recurso del Adaptive Server.

```
sp_drop_resource_limit { name, appname } [, rangename, limittype,
  enforced, action, scope ]
```

81. sp_droprowlockpromote

Descripción: Quita valores de promoción de umbrales de una fila en una tabla o base de datos.

```
sp_droprowlockpromote {"database" | "table"}, objname
```

82. sp_dropsegment

Descripción: Omite un segmento de una base de datos o elimina la correlación de un segmento desde un dispositivos de base de datos en particular.

```
sp_dropsegment segname, dbname [, device ]
```

83. sp_dropserver

Descripción: Elimina un servidor de la lista de servidores conocidos o elimina logins remotos y logins externos en la misma operación.

```
sp_dropserver server [, droplogins ]
```

84. sp_dropthreshold

Descripción: Quita un umbral de espacio libre de un segmento.

```
sp_dropthreshold dbname, segname, free_space
```

85. sp_drop_time_range

Descripción: Remueve un rango de tiempo definido por el usuario del Adaptive Server.

```
sp_drop_time_range name
```

86. sp_droptype

Descripción: Elimina un tipo de datos definido por el usuario.

```
sp_droptype typename
```

87. sp_dropuser

Descripción: Elimina un usuario de la base de datos actual.

```
sp_dropuser name_in_db
```

88. sp_dumpoptimize

Descripción: Especifica la cantidad de datos descargados por Backup Server durante la operación de volcado de la base de datos.

```
sp_dumpoptimize [ "archive_space = { maximum | minimum | default }" ]
sp_dumpoptimize [ "reserved_threshold = { nnn | default }" ]
sp_dumpoptimize [ "allocation_threshold = { nnn | default }" ]
```

89. sp_engine

Descripción: Le permite traer un engine en línea o fuera de línea.

sp_engine {“online” | [offline | can_offline] [, *engine_id*] | [“shutdown”, *engine_id*]}

90. sp_estspace

Descripción: Calcula la cantidad de espacio necesario para una tabla y sus índices, y el tiempo necesario para crear el índice.

```
sp_estspace table_name, no_of_rows [, fill_factor [, cols_to_max
  [, textbin_len [, iosec] ] ] ]
```

91. sp_export_qpgroup

Descripción: Exporta todos los planes para un usuario específico y un grupo del plan abstracto hacia una tabla de usuario.

```
sp_export_qpgroup usr, group, tab
```

92. sp_extendsegment

Descripción: Extiende el alcance de un segmento a otro dispositivo de base de datos.

```
sp_extendsegment segname, dbname, devname
```

93. sp_extengine

Descripción: Inicia o detiene un servidor EJB. Muestra el estado de la información sobre el servidor EJB.

```
sp_extengine ' ejb_server', '{start | stop | status}'
```

94. sp_familylock

Descripción: Reporta información sobre todos los bloqueos mantenidos por una familia (coordinando procesos y sus procesos de trabajo) que ejecutaba una declaración en paralelo.

```
sp_familylock [ fpid1 [, fpid2] ]
```

95. sp_find_qplan

Descripción: Encuentra un plan abstracto, dado un patrón del texto de un query o del texto del plan.

```
sp_find_qplan pattern [, group ]
```

96. sp_fixindex

Descripción: Repara el índice de una de las tablas del sistema cuando éste se ha corrompido.

```
sp_fixindex dbname, table_name, index_id
```

97. sp_flushstats

Descripción: Limpia las estadísticas desde su almacenamiento en memoria hasta la tabla de sistema systabstats.

```
sp_flushstats objname
```

98. sp_forceonline_db

Descripción: Proporciona el acceso a todas las páginas en una base de datos que se encontraba marcada como sospechosa por la recuperación.

```
sp_forceonline_db dbname, { "sa_on" | "sa_off" | "all_users" }
```

99. sp_forceonline_object

Descripción: Proporciona el acceso a un índice marcado previamente como sospechoso por la recuperación.

```
sp_forceonline_object dbname, objname, indid,
  { sa_on | sa_off | all_users } [, no_print]
```

100. sp_forceonline_page

Descripción: Proporciona el acceso a las páginas marcadas previamente como sospechosas por la recuperación.

```
sp_forceonline_page dbname, pgid, {"sa_on" | "sa_off" | "all_users"}
```

101. sp_foreignkey

Descripción: Define una clave externa en una tabla o vista en la base de datos actual.

```
sp_foreignkey tabname, pktabname, col1 [, col2] ... [, col8]
```

102. sp_freedll

Descripción: Descarga una biblioteca de acoplamiento dinámica (DLL) que fue cargada previamente en memoria del Servidor XP para apoyar la ejecución de un procedimiento almacenado extendido (ESP).

```
sp_freedll dll_name
```

103. sp_getmessage

Descripción: Recupera cadenas de mensajes almacenados de sysmessages y sysusermessages para las instrucciones ***print*** y ***raiserror***.

```
sp_getmessage message_num, result output [, language]
```

104. sp_grantlogin

Descripción: Únicamente Windows NT. Si está activo el modo de seguridad integrada o al modo mixto (con canales con nombre), asigna roles del Adaptive Server o permisos predeterminados a usuarios y grupos de Windows NT.

```
sp_grantlogin { login_name | group_name } [ " role_list" | default ]
```

105. sp_ha_admin

Descripción: Realiza tareas administrativas en el Adaptive Server configurados con Sybase Failover en un sistema de alta disponibilidad. sp_ha_admin es instalado con el script installhavss (insthasv en Windows NT).

```
sp_ha_admin [ cleansessions | help ]
```

106. sp_help

Descripción: Proporciona información sobre un objeto de base de datos (cualquier objeto enumerado en sysobjects), y sobre tipos de datos definidos por el usuario o suministrados por el Adaptive Server.

```
sp_help [ objname]
```

107. sp_helpartition

Descripción: Enumera la primera página y la página de control de cada partición en una tabla con particiones.

```
sp_helpartition [ table_name]
```

108. sp_helpcache

Descripción: Muestra información sobre los objetos vinculados a un caché de datos o la cantidad de sobrecarga necesaria para un tamaño de caché especificado.

```
sp_helpcache { cache_name | " cache_size[P|K|M|G]" }
```

109. sp_helpconfig

Descripción: Proporciona información de la ayuda sobre los parámetros de configuración.

```
sp_helpconfig " configname", [" size"]
sp_helpconfig "number of ccbs"
sp_helpconfig "caps per ccb"
sp_helpconfig "average cap size"
```

110. sp_helpconstraint

Descripción: Proporciona información sobre cualquier restricción de integridad especificada para una tabla. Esta información incluye el nombre de la restricción y la definición de la restricción predeterminada vinculada, la restricción de clave exclusiva o primaria, la restricción de referencia o de verificación.

```
sp_helpconstraint [ objname] [, detail]
```

111. sp_helpdb

Descripción: Proporciona información sobre una base de datos en particular o sobre todas las bases de datos.

```
sp_helpdb [ dbname]
```

112. sp_helpdevice

Descripción: Proporciona información sobre un dispositivo en particular o sobre todos los dispositivos de bases de datos y dispositivos de volcado del Adaptive Server.

```
sp_helpdevice [ devname]
```

113. sp_helpextendedproc

Descripción: Despliega procedimientos almacenados extendidos (ESPs), en la base de datos actual, junto con sus archivos asociados DLL..

```
sp_helpextendedproc [ esp_name]
```

114. sp_helpexternlogin

Descripción: Proporciona información sobre nombres de login externos. Únicamente los Servicios de Integración de Componentes.

```
sp_helpexternlogin [ remote_server [, login_name] ]
```

115. sp_helpgroup

Descripción: Proporciona información sobre un grupo en particular o sobre todos los grupos de la base de datos actual.

```
sp_helpgroup [ grpname]
```

116. sp_helpindex

Descripción: Proporciona información sobre los índices creados en una tabla.

```
sp_helpindex objname
```

117. sp_helpjava

Descripción: Despliega información acerca de las clases Java y los JARs asociados que están instalados en la base de datos.

```
sp_helpjava ["class" [, java_class_name [, "detail" | "depends" ] ]
  | "jar" [, jar_name [, "depends" ] ] ]
```

118. sp_helpjoins

Descripción: Enumera las columnas en dos tablas o vistas que pueden ser candidatas de combinación.

```
sp_helpjoins lefttab, righttab
```

119. sp_helpkey

Descripción: Proporciona información sobre una clave primaria, externa o común de una tabla o vista en particular, o sobre todas las claves de la base de datos actual.

```
sp_helpkey [ tabname]
```

120. sp_helplanguage

Descripción: Proporciona información sobre un idioma alternativo en particular o sobre todos los idiomas.

```
sp_helplanguage [ language]
```

121. sp_helplog

Descripción: Proporciona el nombre del dispositivo que contiene la primera página del diario de transacciones.

```
sp_helplog
```

122. sp_helpobjectdef

Descripción: Reporta dueños, objetos, y el tipo información para las definiciones de objetos remotos. Únicamente los Servicios de Integración de Componentes.

```
sp_helpobjectdef [ object_name]
```

123. sp_help_qpgroup

Descripción: Reporta información sobre un grupo de plan abstracto.

```
sp_help_qpgroup [ group [, mode ] ]
```

124. sp_help_qplan

Descripción: Reporta información acerca de un plan abstracto.

```
sp_help_qplan id [, mode ]
```

125. sp_helpremotelogin

Descripción: Proporciona información sobre logins de un servidor remoto en particular o sobre los logins de todos los servidores remotos.

```
sp_helpremotelogin [ remoteserver [, remotename] ]
```

126. sp_help_resource_limit

Descripción: Proporciona información sobre todos los límites de un recurso, los límites dados a un login o una aplicación, los límites efectuados en un tiempo o día de la semana, o límites con un alcance o una acción dado.

```
sp_help_resource_limit [ name [, appname [, limittime [, limitday
  [, scope [, action] ] ] ] ] ]
```

127. sp_helpprotect

Descripción: Informa sobre permisos para objetos, usuarios, grupos o roles de base de datos.

```
sp_helprotect [ name [, username [, "grant"
[, "none"|"granted"|"enabled"| role_name]]]]
```

128. sp_helpsegment

Descripción: Proporciona información sobre un segmento en particular o sobre todos los segmentos de la base de datos actual.

```
sp_helpsegment [ segname]
```

129. sp_helpserver

Descripción: Proporciona información sobre un servidor remoto en particular o sobre todos los servidores remotos.

```
sp_helpserver [ server]
```

130. sp_helpsort

Descripción: Muestra el criterio de ordenación y el juego de caracteres predeterminado del Adaptive Server.

```
sp_helpsort
```

131. sp_helptext

Descripción: Imprime el texto de un procedimiento del sistema, disparador, vista, valor predeterminado o restricción de verificación de integridad.

```
sp_helptext objname
```

132. sp_helpthreshold

Descripción: Informa sobre el segmento, valor de espacio libre, estado y procedimiento almacenado asociado a todos los umbrales de la base de datos actual o a todos los umbrales de un segmento en particular.

```
sp_helpthreshold [ segname]
```

133. sp_helpuser

Descripción: Proporciona información sobre un usuario en particular, grupo o alias, o sobre todos los usuarios de la base de datos actual.

```
sp_helpuser [ name_in_db]
```

134. sp_hidetext

Descripción: Oculta el texto original para el objeto compilado especificado.

```
sp_hidetext [ objname [, tabname [, username] ] ]
```

135. sp_import_qpgroup

Descripción: Importa planes abstractos desde una tabla de usuario en un grupo de plan abstracto.

```
sp_import_qpgroup tab, usr, group
```

136. sp_indsuspect

Descripción: Verifica los índices marcados como sospechosos en tablas de usuario durante la recuperación que sigue a un cambio de criterio de ordenación.

```
sp_indsuspect [ tab_name]
```

137. sp_listsuspect_db

Descripción: Enumera todas las bases de datos que tengan actualmente páginas fuera de línea debido a la corrupción detectada en la recuperación.

```
sp_listsuspect_db
```

138. sp_listsuspect_object

Descripción: Enumera todos los índices en una base de datos que estén actualmente fuera de línea debido a la corrupción detectada en la recuperación.

```
sp_listsuspect_object [ dbname]
```

139. sp_listsuspect_page

Descripción: Enumera todas las páginas en una base de datos que estén actualmente fuera de línea debido a la corrupción detectada en la recuperación.

```
sp_listsuspect_page [ dbname]
```

140. sp_lock

Descripción: Proporciona información sobre los procesos que actualmente mantienen bloqueos.

```
sp_lock [ spid1 [, spid2]]
```

141. sp_locklogin

Descripción: Bloquea una cuenta del Adaptive Server para que el usuario no pueda conectarse, o muestra una lista de todas las cuentas bloqueadas.

```
sp_locklogin [ loginame, "{ lock|unlock }"]
```

142. sp_logdevice

Descripción: Desplaza el diario de transacciones de una base de datos con log y datos en el mismo dispositivo, a un dispositivo de base de datos distinto.

```
sp_logdevice dbname, devname
```

143. sp_loginconfig

Descripción: Muestra el valor de uno o todos los parámetros de seguridad integrada. Únicamente Windows NT.

```
sp_loginconfig [" parameter_name"]
```

144. sp_logininfo

Descripción: Muestra todos los roles concedidos a usuarios y grupos de Windows NT con sp_grantlogin. Únicamente Windows NT

```
sp_logininfo [ " login_name" | " group_name" ]
```

145. sp_logiosize

Descripción: Cambia el tamaño de E/S del diario utilizado por el Adaptive Server a un banco de memoria distinto, al realizar las E/S para el diario de transacciones de la base de datos actual.

```
sp_logiosize [ "default" | " size" | "all" ]
```

146. sp_modifylogin

Descripción: Modifica la base de datos predeterminada, el idioma predeterminado o el nombre completo de una cuenta de login del Adaptive Server.

```
sp_modifylogin account, column, value
```

147. sp_modify_resource_limit

Descripción: Modifica un límite del recurso por un nuevo valor de límite especificado, o la acción a tomar cuando se excede el límite, o a ambas.

```
sp_modify_resource_limit { name, appname } , rangename , limittype ,
  limitvalue , enforced , action , scope
```

148. sp_modify_time_range

Descripción: Cambia el día y la hora de inicio, el día y/o la hora de termino asociado a un rango de tiempo.

```
sp_modify_time_range name, startday, endday, starttime, endtime
```

149. sp_modifystats

Descripción: Permite al Administrador del Sistema modificar los valores de densidad de una columna –o columnas – en sysstatistics.

```
sp_modifystats [ database].[ owner][ table_name], column_name,
  REMOVE_SKEW_FROM_DENSITY
```

or

```
sp_modifystats [ db].[ owner][ table_name],
  { column_name} [, column_name] [,...] ] },
  MODIFY_DENSITY,
  { range | total}, { absolute | total}, " value"
```

150. sp_modifythreshold

Descripción: Modifica un umbral al asociarlo a un procedimiento de umbral, nivel de espacio libre o nombre de segmento distinto. No se puede utilizar sp_modifythreshold para cambiar la cantidad de espacio libre o el nombre de segmento del umbral de última oportunidad.

```
sp_modifythreshold dbname, segname, free_space
[, new_proc_name] [, new_free_space] [, new_segname]
```

151. sp_monitor

Descripción: Muestra estadísticas sobre el Adaptive Server.

```
sp_monitor
```

152. sp_monitorconfig

Descripción: Muestra el caché de estadísticas usadas por los descriptores de metadatos para índices, objetos y bases de datos. sp_monitorconfig también reporta estadísticas sobre la exploración de los descriptores auxiliares utilizados para consultas con integridad. Brinda estadísticas de uso de memoria para los parámetros que asignan memoria. No todos los parámetros son soportados en el momento.

```
sp_monitorconfig " configname"
```

153. sp_object_stats

Descripción: Imprime las estadísticas de bloqueo para tablas e índices.

```
sp_object_stats interval [, top_n [, dbname, objname [, rpt_option ] ] ]
```

154. sp_passthru

Descripción: Este procedimiento permite la ejecución de un comando y, opcionalmente, capturar en variables locales los resultados arrojados por el comando. Únicamente los Servicios de Integración de Componentes.

```
sp_passthru server, command, errcode, errmsg, rowcount
   [, arg1, arg2, ... argn]
```

155. sp_password

Descripción: Agrega o cambia una contraseña para una cuenta de login del Adaptive Server.

```
sp_password caller_passwd, new_passwd [, loginame]
```

156. sp_placeobject

Descripción: Coloca futuras asignaciones de espacio de una tabla o índice en un segmento en particular.

```
sp_placeobject segname, objname
```

157. sp_plan_dbccdb

Descripción: Recomienda los tamaños convenientes para las nuevas bases de datos dbccdb y dbccalt, lista los dispositivos convenientes para dbccdb y dbccalt, y sugiere un tamaño para el caché y un número conveniente de procesos de trabajo para la base de datos final.

```
sp_plan_dbccdb [ dbname]
```

158. sp_poolconfig

Descripción: Crea, omite, redimensiona y proporciona información sobre bancos de memoria incluidos en un caché de datos.

Crea un banco de memoria en un cache existente, o cambia el tamaño del banco:

```
sp_poolconfig cache_name
[, " mem_size[P|K|M|G]", " config_poolK" [, " affected_poolK"] ]
```

Changing a pool's wash size:

```
sp_poolconfig cache_name, " io_size", " wash= size[P|K|M|G]"
```

Changing a pool's asynchronous prefetch percentage:

```
sp_poolconfig cache_name, " io_size", "local async prefetch limit=
percent"
```

159. sp_primarykey

Descripción: Define una clave primaria en una tabla o vista.

```
sp_primarykey tabname, col1 [, col2, col3, ..., col8]
```

160. sp_processmail

Descripción: Únicamente Windows NT. Lee, procesa, envía y elimina mensajes de la bandeja de entrada del Adaptive Server, utilizando los procedimientos almacenados extendidos de sistema (ESPs) xp_findnextmsg, xp_readmail, xp_sendmail, and xp_deletemail.

```
sp_processmail [ subject] [, originator
  [, dbuser [, dbname [, filetype  [, separator] ] ] ] ]
```

161. sp_procqmode

Descripción: Muestra el modo de procesamiento de consultas de un procedimiento almacenado, vista o disparador.

```
sp_procqmode [ object_name [, detail] ]
```

162. sp_procxmode

Descripción: Muestra o cambia los modos de transacción asociados con procedimientos almacenados.

```
sp_procxmode [ procname [, tranmode] ]
```

163. sp_recompile

Descripción: Hace que cada procedimiento almacenado y disparador que utiliza la tabla elegida vuelva a compilarse la próxima vez que se ejecute.

```
sp_recompile objname
```

164. sp_remap

Descripción: Vuelve a correlacionar un procedimiento almacenado, disparador, regla, valor predeterminado o vista de versiones posteriores a la 4.8 y anteriores a la 10.0 para que sean compatibles con la versión 10.0 y posteriores. Use sp_remap en los objetos de versiones anteriores a la 11.0 que el procedimiento de la versión mejorada no haya podido correlacionar.

```
sp_remap objname
```

165. sp_remoteoption

Descripción: Muestra o cambia las opciones de un login remoto.

```
sp_remoteoption [ remoteserver [, loginame
[, remotename [, optname [, optvalue]]]]]
```

166. sp_remotesql

Descripción: Establece una conexión hacia un servidor remoto, transmite un query almacenado desde el servidor remoto cliente, y retransmite los resultados de nuevo al cliente. Únicamente los Servicios de Integración de Componentes.

```
sp_remotesql server, query [, query2, ... , query254]
```

167. sp_rename

Descripción: Cambia el nombre de un objeto creado por el usuario o un tipo de dato definido por el usuario en la base de datos actual.

```
sp_rename objname, newname
```

168. sp_renamedb

Descripción: Cambia el nombre de una base de datos de usuario. No se puede cambiar el nombre de las bases de datos de sistema o de bases de datos con restricciones de integridad de referencia externas.

```
sp_renamedb dbname, newname
```

169. sp_rename_qpgroup

Descripción: Renombra un grupo de plan abstracto.

```
sp_rename_qpgroup old_name, new_name
```

170. sp_reportstats

Descripción: Reporta estadísticas sobre el uso del sistema.

```
sp_reportstats [ loginame]
```

171. sp_revokelogin

Descripción: Solamente Windows NT. Cuando el modo de seguridad integrada o el modo mixto (con canales con nombre) esta activo, revoca roles del Adaptive Server y permisos predeterminados de usuarios y grupos de Windows NT.

```
sp_revokelogin { login_name | group_name }
```

172. sp_role

Descripción: Concede o revoca roles de sistema para una cuenta de login del Adaptive Server.

```
sp_role {"grant"|"revoke"}, rolename, loginame
```

173. sp_sendmsg

Descripción: Envía un mensaje hacia un puerto User Datagram Protocol (UDP).

```
sp_sendmsg ip_address, port_number, message
```

174. sp_serveroption

Descripción: Muestra o cambia las opciones de un servidor remoto.

```
sp_serveroption [ server, optname, optvalue]
```

175. sp_setlangalias

Descripción: Asigna o cambia el alias para un idioma alternativo.

```
sp_setlangalias language, alias
```

176. sp_setpglockpromote

Descripción: Define o cambia los umbrales de promoción de bloqueos para una base de datos, una tabla o para el Adaptive Server.

```
sp_setpglockpromote {"database"|"table"}, objname, new_lwm,
  new_hwm, new_pct

sp_setpglockpromote server, NULL, new_lwm, new_hwm, new_pct
```

177. sp_setpsexe

Descripción: Fija los atributos de ejecución "sobre la marcha" para una sesión mientras ésta este activa.

```
sp_setpsexe spid, exeattr, value
```

178. sp_set_qplan

Descripción: Cambia el texto del plan abstracto de un plan existente sin cambiar la consulta asociada.

```
sp_set_qplan id, plan
```

179. sp_setrowlockpromote

Definición: Define o cambia los umbrales de promoción de row-lock, para una tabla con datarows-locked en una base de datos o para todas las tablas con datarows-locked en un servidor.

```
sp_setrowlockpromote "server", NULL, new_lwm, new_hwm, new_pct

sp_setrowlockpromote {"database" | "table"}, objname, new_lwm,
  new_hwm, new_pct
```

180. sp_setsuspect_granularity

Descripción: Despliega y fija el modo de aislamiento de fallas de la recuperación para una base de datos de usuario, el cual definirá como se comportara la recuperación cuando detecte corrupción en los datos.

```
sp_setsuspect_granularity [ dbname [, "database" | "page" [,
  "read_only"] ] ]
```

181. sp_setsuspect_threshold

Descripción: Muestra o define el número máximo de páginas sospechosas que el Adaptive Server permitirá en una base de datos, antes de marcar completamente la base de datos como sospechosa.

```
sp_setsuspect_threshold [ dbname [, threshold]]
```

182. sp_showcontrolinfo

Descripción: Muestra información sobre asignaciones del grupo motor, limita aplicaciones cliente, cuentas, y procedimientos almacenados.

```
sp_showcontrolinfo [ object_type, object_name, spid]
```

183. sp_showexeclass

Descripción: Despliega los atributos de la clase de ejecución y los motores en cualquier grupo motor asociado a la clase de ejecución especificada.

```
sp_showexeclass [ execlassname]
```

184. sp_showplan

Descripción: Muestra la salida del showplan para cualquier conexión de usuario para la declaración actual del SQL o una declaración anterior en el mismo batch. El query plan se despliega en formato showplan.

```
sp_showplan spid, batch_id output, context_id output, stmt_num output
```

Muestra la salida de showplan para una declaración actual del SQL sin especificar el batch_id, context_id, o stmt_num:

```
sp_showplan spid, null, null, null
```

185. sp_showpsexe

Descripción: Despliega la clase de ejecución, la prioridad actual, y la afinidad para todos los sesiones clientes que corren en el Adaptive Server.

```
sp_showpsexe [ spid]
```

186. sp_spaceused

Descripción: Muestra el número de filas, el número de páginas de datos, el tamaño de los indices y el espacio utilizado por una tabla o por todas las tablas de la base de datos actual.

```
sp_spaceused [ objname [,1] ]
```

187. sp_ssladmin

Descripción: Agrega, borra, o despliega una lista de los certificados del servidor para el Adaptive Server.

```
sp_ssladmin { [addcert, certificate_path, password] [dropcert,
certificate_path] [lscert] [help] }
```

188. sp_syntax

Descripción: Despliega la sintaxis de las declaraciones de Transact-SQL, de los procedimientos de sistema, de las utilidades, y de otras rutinas, dependiendo sobre que productos y que scripts de sp_syntax existen en el Adaptive Server.

sp_syntax word [, mod][, language]

189. sp_sysmon

Descripción: Muestra información del funcionamiento para afinamiento y rendimiento.

```
sp_sysmon begin_sample

sp_sysmon { end_sample | interval } [, section [, applmon] ]

sp_sysmon {end_sample| interval} [, applmon]
```

190. sp_thresholdaction

Descripción: Se ejecuta automáticamente cuando el número de páginas libres del segmento de diario se encuentra por debajo del umbral de última oportunidad (a menos que el umbral se haya asociado a un procedimiento diferente). Sybase no proporciona este procedimiento.

```
sp_thresholdaction @ dbname,
@ segment_name,
@ space_left,
@ status
```

191. sp_transactions

Descripción: Reporta información acerca de las transacciones activas.

sp_tranactions ["xid", *xid_value*] | ["state", {"heuristic_commit" | "heuristic_abort" | "prepared" | "indoubt"} [, "xactname"]] | ["gtrid", *gtrid_value*]

192. sp_unbindcache

Descripción: Desvincula una base de datos, tabla, índice, objeto text u objeto image de un caché de datos.

```
sp_unbindcache dbname [, [ owner.] tablename [, indexname|"text only"] ]
```

193. sp_unbindcache_all

Descripción: Desvincula todos los objetos limitados a un caché.

```
sp_unbindcache_all cache_name
```

194. sp_unbindefault

Descripción: Desvincula un valor predeterminado creado, de una columna o de un tipo de datos definido por el usuario.

```
sp_unbindefault objname [, futureonly]
```

195. sp_unbindexeclass

Descripción: Quita los atributos de la clase de ejecución asociada previamente a una aplicación cliente, a una cuenta, o a un procedimiento almacenado para el alcance especificado.

```
sp_unbindexeclass object_name, object_type, scope
```

196. sp_unbindmsg

Descripción: Desvincula un mensaje definido por el usuario desde una restricción.

```
sp_unbindmsg constrname
```

197. sp_unbindrule

Descripción: Desvincula una regla desde una columna o un tipo de datos definido por el usuario.

```
sp_unbindrule objname [, futureonly]
```

198. sp_volchanged

Descripción: Notifica a Backup Server que el operador realizó el manejo del volumen solicitado durante un volcado o carga.

```
sp_volchanged session_id, devname, action [, fname [, vname] ]
```

199. sp_who

Descripción: Proporciona información sobre todos los usuarios y procesos actuales del Adaptive Server, o acerca de un usuario o proceso en particular.

```
sp_who [ loginame | " spid"]
```

Procedimientos Almacenados del Catalogo

1. **sp_column_privileges**

   Descripción: Devuelve información sobre permisos para una o más columnas de una tabla o vista.

```
sp_column_privileges table_name [, table_owner
  [, table_qualifier [, column_name] ] ]
```

2. **sp_columns**

   Descripción: Devuelve información sobre el tipo de datos que puede almacenarse en una o más columnas.

```
sp_columns table_name [, table_owner ] [, table_qualifier]
  [, column_name]
```

3. **sp_databases**

   Descripción: Devuelve una lista de las bases de datos de un servidor.

```
sp_databases
```

4. **sp_datatype_info**

   Descripción: Devuelve información sobre un tipo de datos ODBC en particular o sobre todos los tipos de datos ODBC admitidos.

```
sp_datatype_info [ data_type]
```

5. **sp_fkeys**

   Descripción: Devuelve información sobre restricciones de clave externa creadas en la base de datos actual con el comando ***create table*** o ***alter table***.

```
sp_fkeys pktable_name [, pktable_owner] [, pktable_qualifier]
  [, fktable_name] [, fktable_owner] [, fktable_qualifier]
```

6. **sp_pkeys**

   Descripción: Devuelve información sobre restricciones de clave primaria creadas para una sola tabla con el comando ***create table*** o ***alter table***.

```
sp_pkeys table_name [, table_owner] [, table_qualifier]
```

7. **sp_server_info**

   Descripción: Devuelve una lista de nombres de atributos y de valores coincidentes para un servidor.

```
sp_server_info [ attribute_id]
```

8. sp_special_columns

Descripción: Devuelve el conjunto optimo de columnas que identifican de forma exclusiva una fila de una tabla o vista; también puede devolver una lista de las columnas actualizadas automáticamente cuando una transacción actualiza algún valor de la fila.

```
sp_special_columns table_name [, table_owner] [, table_qualifier]
  [, col_type]
```

9. sp_sproc_columns

Descripción: Devuelve información sobre los parámetros de entrada y retorno de un procedimiento almacenado.

```
sp_sproc_columns procedure_name [, procedure_owner]
  [, procedure_qualifier][, column_name]
```

10. sp_statistics

Descripción: Devuelve una lista de índices de una sola tabla.

```
sp_statistics table_name [, table_owner] [, table_qualifier]
  [, index_name][, is_unique]
```

11. sp_stored_procedures

Descripción: Devuelve información sobre uno o más procedimientos almacenados.

```
sp_stored_procedures [ sp_name [, sp_owner [, sp_qualifier] ] ]
```

12. sp_table_privileges

Descripción: Devuelve información sobre privilegios para todas las columnas de una tabla o vista.

```
sp_table_privileges table_name [, table_owner [, table_qualifier] ]
```

13. sp_tables

Descripción: Devuelve una lista de objetos que pueden aparecer en una cláusula from.

```
sp_tables [ table_name] [, table_owner] [, table_qualifier]
  [, table_type]
```

System Extended Stored Procedures

1. xp_cmdshell

Descripción: Ejecuta un comando del sistema operativo desde el Adaptive Server. Soporta todas las DLL.

xp_cmdshell command [, no_output]

2. xp_deletemail

Descripción: Borra un mensaje del la bandeja de entrada del Adaptive Server. Solamente Windows NT

xp_deletemail [msg_id]

3. xp_enumgroups

Descripción: Despliega grupos para un dominio especifico de Windows NT. Solamente Windows NT

xp_enumgroups [domain_name]

4. xp_findnextmsg

Descripción: Recupera el identificador del siguiente mensaje en la bandeja de entrada de Adaptive Server. Solamente Windows NT.

xp_findnextmsg @ msg_id = @ msg_id output [, type] [, unread_only = { true|false }]

5. xp_logevent

Descripción: Prevé registrar un evento definido por el usuario en el Log de Windows NT. Solamente Windows NT.

xp_logevent error_number, message [, type]

6. xp_readmail

Descripción: Lee un mensaje de la bandeja de entrada del Adaptive Server. Solamente Windows NT

xp_readmail [msg_id][, recipients output]
[, sender output]
[, date_received output]
[, subject output]
[, cc output]
[, message output]
[, attachments output]
[, suppress_attach = { true | false }]
[, peek = { true | false }]
[, unread = { true | false }]
[, msg_length output]
[, bytes_to_skip [output]]
[, type [output]]

7. xp_sendmail

Descripción: Envía un mensaje a los recipientes especificados usando la interfaz de MAPI. Solamente Windows NT.

xp_sendmail recipient [; recipient]...
[, subject]
[, cc_recipient]...
[, bcc_recipient]...
[, { query | message }]
[, attachname]
[, attach_result = { true | false }]
[, echo_error = { true | false }]
[, include_file [, include_file]...]
[, no_column_header = { true | false }]
[, no_output = { true | false }]
[, width]
[, separator]
[, dbuser]
[, dbname]
[, type]
[, include_query = { true | false }]

8. xp_startmail

Descripción: Inicia una sesión de correo en el Adaptive Server. Solamente Windows NT.

xp_startmail [mail_user] [, mail_password]

9. xp_stopmail

Descripción: Finaliza una sesión de correo del Adaptive Server. Solamente Windows NT.

xp_stopmail

dbcc Stored Procedures

1. sp_dbcc_alterws

Descripción: Cambia el tamaño de un espacio de trabajo específico a un valor concreto, e inicializa el espacio de trabajo.

sp_dbcc_alterws dbname, wsname, " wssize[K|M]"

2. sp_dbcc_configreport

Descripción: Genera un reporte que describe la información de configuración de una base de datos específica usada por la operación dbcc checkstorage.

sp_dbcc_configreport [dbname]

3. sp_dbcc_createws

Descripción: Crea un espacio de trabajo de un tipo y tamaño específico, en una base de datos y segmento específicos.

sp_dbcc_createws dbname, segname, [wsname], wstype, "wssize[K | M]"

4. sp_dbcc_deletedb

Descripción: Borra toda la información de la base de datos especificada de dbccdb.

sp_dbcc_deletedb [dbname]

5. sp_dbcc_deletehistory

Descripción: Borra los resultados de las operaciones de dbcc checkstorage sobre la base de datos especificada de dbccdb.

sp_dbcc_deletehistory [cutoffdate [, dbname]]

6. sp_dbcc_differentialreport

Descripción: Genera un informe que destaca los cambios en las estadísticas y los errores de I/O que ocurrieron entre dos operaciones del dbcc. Compara las operaciones completadas para la base de datos en fechas específicas.

sp_dbcc_differentialreport [dbname [, objectname]], [db_op] [, " date1" [, " date2"]]

7. sp_dbcc_evaluatedb

Descripción: Recomienda valores para los parámetros de configuración usando los resultados de operaciones *dbcc checkstorage* previas.

sp_dbcc_evaluatedb [dbname]

8. sp_dbcc_faultreport

Descripción: Genera un informe sobre los errores de estadísticas para las operaciones del dbcc checkstorage realizadas sobre un objeto especifico en una base de datos en la fecha especificada.

sp_dbcc_faultreport [report_type [, dbname [, objectname [, date]]]]

9. sp_dbcc_fullreport

Descripción: Ejecuta un reporte extendido de dbcc checkstorage, ejecutando los siguientes reportes en el siguiente orden: sp_dbcc_summaryreport, sp_dbcc_configreport, sp_dbcc_statisticsreport, y sp_dbcc_faultreport.

sp_dbcc_fullreport [dbname [, objectname [, date]]]

10. sp_dbcc_runcheck

Descripción: Ejecuta el dbcc checkstorage de una base de datos específica, y después ejecuta sp_dbcc_summaryreport o un reporte que uno especifique.

sp_dbcc_runcheck dbname [, user_proc]

11. sp_dbcc_statisticsreport

Descripción: Genera un reporte de asignación de estadísticas sobre un objeto especifico en una base de datos.

sp_dbcc_statisticsreport [dbname [, objectname [, date]]]

12. sp_dbcc_summaryreport

Descripción: Genera un reporte sobre las verificaciones realizadas por dbcc checkstorage en una base de datos especifica.

sp_dbcc_summaryreport [dbname [, op_name]]

13. sp_dbcc_updateconfig

Descripción: Actualiza los parámetros de configuración para la base de datos especificada.

sp_dbcc_updateconfig dbname, type, " str1" [, " str2"]

Anexo 2. Valores en evento y columna extrainfo

event	Audit option	Command or access audited	extrainfo
1	adhoc	User-defined audit record	extrainfo is filled by the *text* parameter of sp_addauditrecord
2	alter	alter database	*Roles:* Current active roles *Subcommand:* "ALTER SIZE" *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
3	alter	alter table	*Roles:* Current active roles *Subcommand:* "ADD COLUMN", "REPLACE COLUMN", "ADD CONSTRAINT", or "DROP CONSTRAINT" *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
4	bcp	bcp in	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
6	bind	sp_bindefault	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* Name of default *Proxy information:* Original login name, if a set proxy is in effect
7	bind	sp_bindmsg	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* Message ID *Proxy information:* Original login name, if a set proxy is in effect
8	bind	sp_bindrule	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* Name of the rule *Proxy information:* Original login name, if a set proxy is in effect
9	create	create database	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
10	create	create table	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
11	create	create procedure	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect

event	Audit option	Command or access audited	extrainfo
12	create	create trigger	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
13	create	create rule	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
14	create	create default	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
15	create	sp_addmessage	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* Message Number *Proxy information:* Original login name, if a set proxy is in effect
16	create	create view	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
17	dbaccess	Any access to the database by any user	*Roles:* Current active roles *Subcommand:* “USE CMD” or “OUTSIDE REFERENCE” *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
18	delete	delete from a table	*Roles:* Current active roles *Subcommand:* “DELETE” *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
19	delete	delete from a view	*Roles:* Current active roles *Subcommand:* “DELETE” *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
20	disk	disk init	*Roles:* Current active roles *Subcommand:* “disk init” *Previous value:* NULL *Current value:* NULL *Other Information:* Name of the disk *Proxy information:* Original login name, if a set proxy is in effect
21	disk	disk refit	*Roles:* Current active roles *Subcommand:* “disk refit” *Previous value:* NULL *Current value:* NULL *Other Information:* Name of the disk *Proxy information:* Original login name, if a set proxy is in effect
22	disk	disk reinit	*Roles:* Current active roles

event	Audit option	Command or access audited	extrainfo
			Subcommand: “disk reinit” *Previous value:* NULL *Current value:* NULL *Other Information:* Name of the disk *Proxy information:* Original login name, if a set proxy is in effect
23	disk	disk mirror	*Roles:* Current active roles *Subcommand:* “disk mirror” *Previous value:* NULL *Current value:* NULL *Other Information:* Name of the disk *Proxy information:* Original login name, if a set proxy is in effect
24	disk	disk unmirror	*Roles:* Current active roles *Subcommand:* “disk unmirror” *Previous value:* NULL *Current value:* NULL *Other Information:* Name of the disk *Proxy information:* Original login name, if a set proxy is in effect
25	disk	disk remirror	*Roles:* Current active roles *Subcommand:* “disk remirror” *Previous value:* NULL *Current value:* NULL *Other Information:* Name of the disk *Proxy information:* Original login name, if a set proxy is in effect
26	drop	drop database	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
27	drop	drop table	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
28	drop	drop procedure	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
29	drop	drop trigger	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
30	drop	drop rule	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
31	drop	drop default	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
32	drop	sp_dropmessage	*Roles:* Current active roles *Subcommand:* NULL

event	Audit option	Command or access audited	extrainfo
			Previous value: NULL *Current value:* NULL *Other Information:* Message number *Proxy information:* Original login name, if a set proxy is in effect
33	drop	drop view	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL
34	dump	dump database	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
35	dump	dump transaction	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
36	errors	Fatal error	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* Error number.Severity.State *Proxy information:* Original login name, if a set proxy is in effect
37	errors	Non-fatal error	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* Error number.Severity.State *Proxy information:* Original login name, if a set proxy is in effect
38	exec_procedure	Execution of a procedure	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other Information:* All input parameters *Proxy information:* Original login name, if a set proxy is in effect
39	exec_trigger	Execution of a trigger	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
40	grant	grant	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
41	insert	insert into a table	*Roles:* Current active roles *Subcommand:* If insert: “INSERT” If select into: “INSERT INTO” followed by the fully qualified object name *Previous value:* NULL *Current value:* NULL *Other information:* NULL

event	Audit option	Command or access audited	extrainfo
			Proxy information: Original login name, if a set proxy is in effect
42	insert	insert into a view	*Roles:* Current active roles *Subcommand:* "INSERT" *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
43	load	load database	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
44	load	load transaction	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
45	login	Any login to Adaptive Server	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other Information:* Host name of the machine from which login was done *Proxy information:* Original login name, if a set proxy is in effect
46	logout	Any logouts from Adaptive Server	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other Information:* Host name of the machine from which login was done *Proxy information:* Original login name, if a set proxy is in effect
47	revoke	revoke	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
48	rpc	Remote procedure call from another server	*Roles:* Current active roles *Subcommand:* Name of client program *Previous value:* NULL *Current value:* NULL *Other information:* Server name, host name of the machine from which the RPC was done. *Proxy information:* Original login name, if a set proxy is in effect
49	rpc	Remote procedure call to another server	*Roles:* Current active roles *Subcommand:* Procedure name *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
50	security	Server start	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other Information:* -d*masterdevicename* -i*interfaces file path* -S*servername* -e*errorfilename*

event	Audit option	Command or access audited	extrainfo
			Proxy information: Original login name, if a set proxy is in effect
51	security	Server shutdown	*Roles:* Current active roles *Subcommand:* "shutdown" *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
55	security	Role toggling	*Roles:* Current active roles *Subcommand:* NULL *Previous Value:* "on" or "off" *Current Value:* "on" or "off" *Other Information:* Name of the role being set *Proxy information:* Original login name, if a set proxy is in effect
61	table_access	Table access	*Roles:* Current active roles *Subcommand:* SELECT, SELECT INTO, INSERT, UPDATE, DELETE, REFERENCE, READTEXT, or WRITETEXT *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
62	select	select from a table	*Roles:* Current active roles *Subcommand:* "SELECT INTO", "SELECT", or "READTEXT" *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
63	select	select from a view	*Roles:* Current active roles *Subcommand:* "SELECT", "SELECT INTO", or "READTEXT" *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
64	truncate	truncate table	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
67	unbind	sp_unbindefault	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
68	unbind	sp_unbindrule	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
69	unbind	sp_unbindmsg	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
70	update	update to a table	*Roles:* Current active roles *Subcommand:* "UPDATE" or "WRITETEXT" *Previous value:* NULL

event	Audit option	Command or access audited	extrainfo
			Current value: NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
71	update	update to a view	*Roles:* Current active roles *Subcommand:* "UPDATE" or "WRITETEXT" *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
73	**Note** This event is udited automatically. It is not controlled by an audit option.	Turning the auditing parameter on with sp_configure	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
74	**Note** This event is udited automatically. It is not controlled by an audit option.	Turning the auditing parameter off with sp_configure	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
76	security	Regeneration of a password by a System Security Officer (SSO)	*Roles:* Current active roles *Subcommand:* Setting SSO password *Previous value:* NULL *Current value:* NULL *Other information:* Login name *Proxy information:* Original login name, if a set proxy is in effect
80	security	proc_role within a system procedure	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other Information:* Required roles *Proxy information:* Original login name, if a set proxy is in effect
81	dbcc	dbcc	*Roles:* Current active roles *Subcommand:* The dbcc subcommand name *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
82	security	sp_configure	*Roles:* Current active roles *Subcommand:* Name of the configuration parameter *Previous Value:* The old parameter value if the command is setting a new value *Current Value:* The new parameter value if the command is setting a new value *Other Information:* Number of configuration parameter, if a parameter is being set; Name of the configuration file, if a configuration file is being used to set parameters *Proxy information:* Original login name, if a set proxy is in effect
83	security	online database	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
84	setuser	setuser	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL

event	Audit option	Command or access audited	extrainfo
			Other Information: Name of the user being set *Proxy information:* Original login name, if a set proxy is in effect
85	func_obj_access, func_dbaccess	Accesses to objects and databases via Transact-SQL functions	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
88	security	set proxy or set session authorization	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* Previous suid *Current value:* New suid *Other information:* NULL *Proxy information*: Original login name, if set proxy or set session authorization had no parameters; otherwise, NULL.
89	?????	kill	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
90	?????	connect	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
91	?????	????? reference	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect
92	cmdtxt	All actions of a particular user, or by users with a particular role	*Roles:* Current active roles *Subcommand:* NULL *Previous value:* NULL *Current value:* NULL *Other information:* NULL *Proxy information:* Original login name, if a set proxy is in effect

ANEXO 3. List of Routines DB-Library

Rutina	Descripción
db12hour	Determina si el idioma especificado usa el formato de 12 horas o 24 horas. **Sintaxis:** DBBOOL db12hour(dbproc, language) DBPROCESS *dbproc; char *language;
dbadata	Devuelve un apuntador a los datos para una columna compute. **Sintaxis:** BYTE *dbadata(dbproc, computeid, column) DBPROCESS *dbproc; int computeid; int column;
dbadlen	Devuelve la longitud real de los datos para una columna compute. **Sintaxis:** DBINT dbadlen(dbproc, computeid, column) DBPROCESS *dbproc; int computeid; int column;
dbaltbind	Asigna una columna compute a una variable del programa. **Sintaxis:** RETCODE dbaltbind(dbproc, computeid, column, vartype, varlen, varaddr) DBPROCESS *dbproc; int computeid; int column; int vartype; DBINT varlen; BYTE *varaddr;
dbaltbind_ps	Asigna una columna compute a una variable del programa, con la precisión y escala que soporta para tipos de datos numeric y decimal. **Sintaxis:** RETCODE dbaltbind_ps(dbproc, computeid, column, vartype, varlen, varaddr, typeinfo) DBPROCESS *dbproc; int computeid;

Rutina	Descripción
	int column; int vartype; DBINT varlen; BYTE *varaddr; DBTYPEINFO *typeinfo;
dbaltcolid	Devuelve el id de la columna para una columna compute. **Sintaxis:** int dbaltcolid(dbproc, computeid, column) DBPROCESS *dbproc; int computeid; int column;
dbaltlen	Devuelve la longitud máxima de los datos para una particular columna compute. **Sintaxis:** DBINT dbaltlen(dbproc, computeid, column) DBPROCESS *dbproc; int computeid; int column;
dbaltop	Devuelve el tipo de operador agregado para una particular columna compute. **Sintaxis:** int dbaltop(dbproc, computeid, column) DBPROCESS *dbproc; int computeid; int column;
dbalttype	Devuelve el tipo de datos para una columna compute. **Sintaxis:** int dbalttype(dbproc, computeid, column) DBPROCESS *dbproc; int computeid; int column;
dbaltutype	Devuelve el tipo de datos usuario definido para una columna compute. **Sintaxis:** DBINT dbaltutype(dbproc, computeid, column) DBPROCESS *dbproc; int computeid; int column;

Rutina	Descripción
dbanullbind	Asocia un variable con una columna-renglón de un compute. **Sintaxis:** RETCODE dbanullbind(dbproc, computeid, column,indicator) DBPROCESS *dbproc; int computeid; int column; DBINT *indicator;
dbbind	Asigna una columna de resultados a una variable del programa. **Sintaxis:** RETCODE dbbind(dbproc, column, vartype, varlen, varaddr) DBPROCESS *dbproc; int column; int vartype; DBINT varlen; BYTE *varaddr;
dbbind_ps	Asigna una columna de resultados a una variable del programa, con la precisión y escala soportada por el tipo de datos numeric y decimal. **Sintaxis:** RETCODE dbbind_ps(dbproc, column, vartype, varlen, varaddr, typeinfo) DBPROCESS *dbproc; int column; int vartype; DBINT varlen; BYTE *varaddr; DBTYPEINFO *typeinfo;
dbbufsize	Devuelve el tamaño de un renglón del buffer del DBPROCESS. **Sintaxis:** int dbbufsize(dbproc) DBPROCESS *dbproc;
dbbylist	Devuelve el bylist para un renglón compute. **Sintaxis:** BYTE *dbbylist(dbproc, computeid, size) DBPROCESS *dbproc; int computeid; int *size;
dbcancel	Cancela el lote del comando actual.

Rutina	Descripción
	Sintaxis: RETCODE dbcancel(dbproc) DBPROCESS *dbproc;
dbcancel_a	(VMS sólo) Cancela el lote de comandos actual asincrónicamente. **Sintaxis:** RETCODE dbcancel_a(dbproc, final_result, ast_proc, ast_param) DBPROCESS *dbproc; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbcanquery	Cancele cualquier fila pendiente del query recientemente ejecutado. **Sintaxis:** RETCODE dbcanquery(dbproc) DBPROCESS *dbproc;
dbcanquery_a	(VMS sólo) Cancela cualquier fila pendiente del query recientemente ejecutado asincrónicamente. **Sintaxis:** RETCODE dbcanquery_a(dbproc, final_result, ast_proc, ast_param) DBPROCESS *dbproc; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbchange	Determine si un lote de comandos se ha cambiado de la base de datos actual. **Sintaxis:** char *dbchange(dbproc) DBPROCESS *dbproc;
dbclose	Cierre y libere una estructura de DBPROCESS. **Sintaxis:** void dbclose(dbproc) DBPROCESS *dbproc;
dbclrbuf	Limpia de renglones un buffer. **Sintaxis:** void dbclrbuf(dbproc, n) DBPROCESS* dbproc; DBINT n;

Rutina	Descripción
dbclropt	Limpia las opciones asignadas con dbsetopt. **Sintaxis:** RETCODE dbclropt(dbproc, option, param) DBPROCESS *dbproc; int option; char* param;
dbcmd	Agrega texto al buffer de comandos de un DBPROCESS. **Sintaxis:** RETCODE dbcmd(dbproc, cmdstring) DBPROCESS *dbproc; char *cmdstring;
DBCMDROW	Determina si el comando actual puede devolver renglones. **Sintaxis:** RETCODE DBCMDROW(dbproc) DBPROCESS *dbproc;
dbcolbrowse	Determina si la fuente de una columna de resultados es una columna actualizable vía los medios de las DB-library en modo browse. **Sintaxis:** DBBOOL dbcolbrowse(dbproc, colnum) DBPROCESS *dbproc; int colnum;
dbcollen	Devuelve la longitud máxima de los datos en una columna de resultados. **Sintaxis:** DBINT dbcollen(dbproc, column) DBPROCESS *dbproc; int column;
dbcolname	Devuelve el nombre de una columna de resultados. **Sintaxis:** char *dbcolname(dbproc, column) DBPROCESS *dbproc; int column;
dbcolsource	Devuelve un apuntador al nombre de la columna de la base de datos de la columna de resultados que fue derivada.

Rutina	Descripción
	Sintaxis: char *dbcolsource(dbproc, colnum) DBPROCESS *dbproc; int colnum;
dbcoltype	Devuelve el tipo de datos para una columna de resultados. **Sintaxis:** int dbcoltype(dbproc, column) DBPROCESS *dbproc; int column;
dbcoltypeinfo	Devuelve información de la precisión y escala para una columna de resultados para tipo numeric o decimal. **Sintaxis:** DBTYPEINFO * dbcoltypeinfo(dbproc, column) DBPROCESS *dbproc; int column;
dbcolutype	Devuelve el tipo de datos definido por un usuario para una columna de resultados. **Sintaxis:** DBINT dbcolutype(dbproc, column) DBPROCESS *dbproc; int column;
dbconvert	Convierte los datos de un tipo de datos a otro. **Sintaxis:** DBINT dbconvert(dbproc, srctype, src, srclen,desttype, dest, destlen) DBPROCESS *dbproc; int srctype; BYTE *src; DBINT srclen; int desttype; BYTE *dest; DBINT destlen;
dbconvert_ps	Convierte los datos de un tipo de datos a otro, con la precisión y escala soportada para el tipo de datos numeric y decimal. **Sintaxis:** DBINT dbconvert_ps(dbproc, srctype, src, srclen,desttype, dest, destlen, typeinfo) DBPROCESS *dbproc; int srctype; BYTE *src; DBINT srclen;

Rutina	Descripción
	int desttype; BYTE *dest; DBINT destlen; DBTYPEINFO *typeinfo;
DBCOUNT	Devuelve el número de filas afectado por un comando Transact-SQL. **Sintaxis:** DBINT DBCOUNT(dbproc) DBPROCESS *dbproc;
DBCURCMD	Devuelve el número de comandos actual. **Sintaxis:** int DBCURCMD(dbproc) DBPROCESS *dbproc;
DBCURROW	Devuelve el número de la fila que actualmente se esta leyendo. **Sintaxis:** DBINT DBCURROW(dbproc) DBPROCESS *dbproc;
dbcursor	Inserta, actualiza, borra, bloquea o refresca una fila en particular en el buffer. **Sintaxis:** RETCODE dbcursor(hc, optype, bufno, table, values) DBCURSOR *hc; DBINT optype; DBINT bufno; BYTE *table; BYTE *values
dbcursorbind	Registra la información ligada sobre las columnas del cursor. **Sintaxis:** RETCODE dbcursorbind(hc, col, vartype, varlen,poutlen, pvaraddr, typeinfo) DBCURSOR *hc; int col; int vartype; DBINT varlen; DBINT *poutlen; BYTE *pvaraddr; DBTYPEINFO *typeinfo;
dbcursorclose	Cierra el cursor asociado con a la conexión y suelte todos los datos

Rutina	Descripción
	que agarro el cursor. **Sintaxis:** void dbcursorclose(hc) DBCURSOR *hc;
dbcursorcolinfo	Devuelve información de una columna en particular especificando el número de la columna del cursor abierto. **Sintaxis:** RETCODE dbcursorcolinfo(hcursor, column, colname, coltype, collen, usertype) DBCURSOR *hcursor DBINT column; DBCHAR *colname; DBINT *coltype; DBINT *collen; DBINT *usertype;
dbcursorfetch	Toma un bloque de renglones y las inserta en variables del programa declaradas por el usuario con el dbcursorbind. **Sintaxis:** RETCODE dbcursorfetch(hc, fetchtype, rownum) DBCURSOR *hc; DBINT fetchtype; DBINT rownum;
dbcursorinfo	Devuelve el número de columnas y el número de filas en el keyset si los keyset y si estos llegaron al final del conjunto de resultados. **Sintaxis:** RETCODE dbcursorinfo(hcursor, ncols, nrows); DBCURSOR *hcursor; DBINT *ncols DBINT *nrows;
dbcursoropen	Abre un cursor y especifica la opción de barrido, la opción del concurrencia y el tamaño del buffer (el número de filas que recupera en un solo fetch). **Sintaxis:** DBCURSOR *dbcursoropen(dbproc, stmt, scrollopt,concuropt, nrows, pstatus) DBPROCESS *dbproc; BYTE *stmt; SHORT scrollopt; SHORT concuropt; USHORT nrows; DBINT *pstatus

Rutina	Descripción
dbdata	Devuelve un apuntador a los datos en una columna de resultados. **Sintaxis:** BYTE *dbdata(dbproc, column) DBPROCESS *dbproc; int column;
dbdate4cmp	Compara dos valores de DBDATETIME4. **Sintaxis:** int dbdate4cmp(dbproc, d1, d2) DBPROCESS *dbproc; DBDATETIME4 *d1; DBDATETIME4 *d2;
dbdate4zero	Inicializa una variable DBDATETIME4 a Ene 1, 1900 12:00AM. **Sintaxis:** RETCODE dbdate4zero(dbproc, dateptr) DBPROCESS *dbproc; DBDATETIME4 *dateptr;
dbdatechar	Convierte un valor entero de un DBDATETIME en formato carácter. **Sintaxis:** RETCODE dbdatechar(dbproc, charbuf, datepart, value) DBPROCESS *dbproc; char *charbuf; int datepart; int value;
dbdatecmp	Compara dos valores de DBDATETIME. **Sintaxis:** int dbdatecmp(dbproc, d1, d2) DBPROCESS *dbproc; DBDATETIME *d1; DBDATETIME *d2;
dbdatecrack	Convierte un DBDATETIME recuperado de la máquina en formato accesible por el usuario. **Sintaxis:** RETCODE dbdatecrack(dbproc, dateinfo, datetime) DBPROCESS *dbproc; DBDATEREC *dateinfo; DBDATETIME *datetime;
dbdatename	Convierte el componente especificado de una estructura de

Rutina	Descripción
	DBDATETIME en su valor correspondiente en carácter. **Sintaxis:** int dbdatename(dbproc, charbuf, datepart, datetime) DBPROCESS *dbproc; char *charbuf; int datepart; DBDATETIME *datetime;
dbdateorder	Devuelve el orden de componente de fecha para un idioma dado. **Sintaxis:** char *dbdateorder(dbproc, language) DBPROCESS *dbproc; char *language;
dbdatepart	Devuelve la parte especificada de un valor de DBDATETIME como un valor numérico. **Sintaxis:** DBINT dbdatepart(dbproc, datepart, datetime) DBPROCESS *dbproc; int datepart; DBDATETIME *datetime;
dbdatezero	Inicializa un valor de DBDATETIME a Ene 1, 1900 12:00:00:000AM. **Sintaxis:** RETCODE dbdatezero(dbproc, dateptr) DBPROCESS *dbproc; DBDATETIME *dateptr;
dbdatlen	Devuelve la longitud de los datos en una columna de resultados. **Sintaxis:** DBINT dbdatlen(dbproc, column) DBPROCESS *dbproc; int column;
dbdayname	Determine el nombre de un día de la semana especificado en un idioma especificado. **Sintaxis:** char *dbdayname(dbproc, language, daynum) DBPROCESS *dbproc; char *language; int daynum;

Rutina	Descripción
DBDEAD	Determina si un DBPROCESS está muerto. **Sintaxis:** DBBOOL DBDEAD(dbproc) DBPROCESS *dbproc;
dberrhandle	Instala la función del usuario para ocuparse de errores de las DB-library. **Sintaxis:** int (*dberrhandle(handler))() int (*handler)();
dbexit	Cierra y libera las estructuras de todos los DBPROCESS, y limpia cualquier estructura inicializada por el dbinit. **Sintaxis:** void dbexit()
dbfcmd	Agrega el texto al buffer de comandos de un DBPROCESS utilizando parámetros de su programa en C. **Sintaxis:** RETCODE dbfcmd(dbproc, cmdstring, args...) DBPROCESS *dbproc; char *cmdstring; ??? args...;
DBFIRSTROW	Devuelve el número de la primera fila buffer de renglones. **Sintaxis:** DBINT DBFIRSTROW(dbproc) DBPROCESS *dbproc;
dbfreebuf	Limpia el buffer de comandos. **Sintaxis:** void dbfreebuf(dbproc) DBPROCESS *dbproc;
dbfreequal	Libera la memoria asignada por el dbqual. **Sintaxis:** void dbfreequal(qualptr) char *qualptr;
dbfreesort	Libera una estructura de orden asignada por el dbloadsort.

Rutina	Descripción
	Sintaxis: RETCODE dbfreesort(dbproc, sortorder) DBPROCESS *dbproc; DBSORTORDER *sortorder;
dbgetchar	Devuelve un apuntador a un carácter en el buffer de comandos. **Sintaxis:** char *dbgetchar(dbproc, n) DBPROCESS *dbproc; int n;
dbgetcharset	Trae el nombre del juego de caracteres del cliente desde la estructura del DBPROCESS. **Sintaxis:** char *dbgetcharset(dbproc) DBPROCESS *dbproc;
dbgetloginfo	Transfiere la información de un login en su estructura de DBPROCESS TDS (Tabular Data Stream) a una estructura de DBLOGINFO recientemente asignada. **Sintaxis:** RETCODE dbgetloginfo(dbproc, loginfo) DBPROCESS *dbproc; DBLOGINFO **loginfo;
dbgetlusername	Devuelve el nombre del usuario de una estructura de LOGINREC. **Sintaxis:** int dbgetlusername(login, name_buffer, buffer_len) LOGINREC *login; BYTE *name_buffer; int buffer_len;
dbgetmaxprocs	Determine el número máximo actual de DBPROCESSes simultáneamente abierto. **Sintaxis:** int dbgetmaxprocs()
dbgetnatlang	Trae el idioma de la estructura de DBPROCESS. **Sintaxis:** char* dbgetnatlang(dbproc) DBPROCESS *dbproc;
dbgetoff	Checa la existencia de una estructura de Transact-SQL en el buffer

Rutina	Descripción
	de comandos. **Sintaxis:** int dbgetoff(dbproc, offtype, startfrom) DBPROCESS *dbproc; DBUSMALLINT offtype; int startfrom;
dbgetpacket	Devuelve el tamaño del paquete TDS actualmente en uso. **Sintaxis:** int dbgetpacket(dbproc) DBPROCESS *dbproc;
dbgetrow	Lee la fila especificada en el buffer de renglones. **Sintaxis:** STATUS dbgetrow(dbproc, row) DBPROCESS *dbproc; DBINT row;
DBGETTIME	Devuelve el número de segundos que las DB-library esperará por una contestación del servidor a un comando de SQL. **Sintaxis:** int DBGETTIME()
dbgetuserdata	Devuelve un apuntador a los datos de usuario-asignados desde una estructura de DBPROCESS. **Sintaxis:** BYTE *dbgetuserdata(dbproc) DBPROCESS *dbproc;
dbhasretstat	Determina si el comando actual o la llamada del procedimiento remoto generaron un número de estado de retorno. **Sintaxis:** DBBOOL dbhasretstat(dbproc) DBPROCESS *dbproc;
dbinit	Inicializa las DB-library. **Sintaxis:** RETCODE dbinit()
DBIORDESC	(sólo UNIX y AOS/VS) Proporciona acceso del programa al descriptor de archivo de UNIX o AOS/VS usado por un DBPROCESS para leer datos que vienen del servidor.

Rutina	Descripción
	Sintaxis: int DBIORDESC(dbproc) DBPROCESS *dbproc;
DBIOWDESC	(sólo UNIX y AOS/VS) Proporcione acceso del programa al descriptor de archivo de UNIX o AOS/VS usado por un DBPROCESS para escribir los datos al servidor. **Sintaxis:** int DBIOWDESC(dbproc) DBPROCESS *dbproc;
DBISAVAIL	Determina si un DBPROCESS está disponible para el uso general. **Sintaxis:** DBBOOL DBISAVAIL(dbproc) DBPROCESS *dbproc;
dbisopt	Verifica el estatus de un servidor u opciones de las DB-library. **Sintaxis:** DBBOOL dbisopt(dbproc, option, param) DBPROCESS *dbproc; int option; char *param;
DBLASTROW	Devuelve el número de la última fila en el buffer de renglones. **Sintaxis:** DBINT DBLASTROW(dbproc) DBPROCESS *dbproc;
dbloadsort	Carga un set order a un servidor. **Sintaxis:** DBSORTORDER *dbloadsort(dbproc) DBPROCESS *dbproc;
dblogin	Asigna un registro del login para el uso en el dbopen. **Sintaxis:** LOGINREC *dblogin()
dbloginfree	Libera un registro de login. **Sintaxis:** void dbloginfree(loginptr) LOGINREC *loginptr;

Rutina	Descripción
dbmny4add	Agrega dos valores de DBMONEY4. **Sintaxis:** RETCODE dbmny4add(dbproc, m1, m2, sum) DBPROCESS *dbproc; DBMONEY4 *m1; DBMONEY4 *m2; DBMONEY4 *sum;
dbmny4cmp	Compara dos valores de DBMONEY4. **Sintaxis:** int dbmny4cmp(dbproc, m1, m2) DBPROCESS *dbproc; DBMONEY4 *m1; DBMONEY4 *m2;
dbmny4copy	Copia un valor de DBMONEY4. **Sintaxis:** RETCODE dbmny4copy(dbproc, src, dest) DBPROCESS *dbproc; DBMONEY4 *src; DBMONEY4 *dest;
dbmny4divide	Divide un valor DBMONEY4 entre otro. **Sintaxis:** RETCODE dbmny4divide(dbproc, m1, m2, quotient) DBPROCESS *dbproc; DBMONEY4 *m1; DBMONEY4 *m2; DBMONEY4 *quotient;
dbmny4minus	Niega un valor de DBMONEY4. **Sintaxis:** RETCODE dbmny4minus(dbproc, src, dest) DBPROCESS *dbproc; DBMONEY4 *src; DBMONEY4 *dest;
dbmny4mul	Multiplica dos valores de DBMONEY4. **Sintaxis:** RETCODE dbmny4mul(dbproc, m1, m2, product) DBPROCESS *dbproc; DBMONEY4 *m1; DBMONEY4 *m2; DBMONEY4 *product;

Rutina	Descripción
dbmny4sub	Substrae un valor DBMONEY4 de otro. **Sintaxis:** RETCODE dbmny4sub(dbproc, m1, m2, difference) DBPROCESS *dbproc; DBMONEY4 *m1; DBMONEY4 *m2; DBMONEY4 *difference;
dbmny4zero	Inicializa una variable DBMONEY4 a $0.0000. **Sintaxis:** RETCODE dbmny4zero(dbproc, mny4ptr) DBPROCESS *dbproc; DBMONEY4 *mny4ptr;
dbmnyadd	Suma dos valores DBMONEY. **Sintaxis:** RETCODE dbmnyadd(dbproc, m1, m2, sum) DBPROCESS *dbproc; DBMONEY *m1; DBMONEY *m2; DBMONEY *sum;
dbmnycmp	Compara dos valores DBMONEY. **Sintaxis:** int dbmnycmp(dbproc, m1, m2) DBPROCESS *dbproc; DBMONEY *m1; DBMONEY *m2;
dbmnycopy	Copia un valor de DBMONEY. **Sintaxis:** RETCODE dbmnycopy(dbproc, src, dest) DBPROCESS *dbproc; DBMONEY *src; DBMONEY *dest;
dbmnydec	Decrementa un valor de DBMONEY en un diez-milésimo de dólar. **Sintaxis:** RETCODE dbmnydec(dbproc, mnyptr) DBPROCESS *dbproc; DBMONEY *mnyptr;

Rutina	Descripción
dbmnydivide	Divide un valor DBMONEY entre otro. **Sintaxis:** RETCODE dbmnydivide(dbproc, m1, m2, quotient) DBPROCESS *dbproc; DBMONEY *m1; DBMONEY *m2; DBMONEY *quotient;
dbmnydown	Divide un valor DBMONEY por un entero positivo. **Sintaxis:** RETCODE dbmnydown(dbproc, mnyptr, divisor, remainder) DBPROCESS *dbproc; DBMONEY *mnyptr; int divisor; int *remainder;
dbmnyinc	Incrementa un valor DBMONEY en un diez-milésimo de dólar. **Sintaxis:** RETCODE dbmnyinc(dbproc, mnyptr) DBPROCESS *dbproc; DBMONEY *mnyptr;
dbmnyinit	Prepara un valor DBMONEY para las llamadas con dbmnyndigit. **Sintaxis:** RETCODE dbmnyinit(dbproc, mnyptr, trim, negative) DBPROCESS *dbproc; DBMONEY *mnyptr; int trim; DBBOOL *negative;
dbmnymaxneg	Devuelve el valor DBMONEY negativo máximo soportado. **Sintaxis:** RETCODE dbmnymaxneg(dbproc,dest) DBPROCESS *dbproc; DBMONEY *dest;
dbmnymaxpos	Devuelve el valor DBMONEY positivo máximo soportado. **Sintaxis:** RETCODE dbmnymaxpos(dbproc, dest) DBPROCESS *dbproc; DBMONEY *dest;

Rutina	Descripción
dbmnyminus	Niega un valor DBMONEY. **Sintaxis:** RETCODE dbmnyminus(dbproc, src, dest) DBPROCESS *dbproc; DBMONEY *src; DBMONEY *dest;
dbmnymul	Multiplica dos valores DBMONEY. **Sintaxis:** RETCODE dbmnymul(dbproc, m1, m2, product) DBPROCESS *dbproc; DBMONEY *m1; DBMONEY *m2; DBMONEY *product;
dbmnyndigit	Devuelve el digito menos significativo de un valor DBMONEY como un DBCHAR. **Sintaxis:** RETCODE dbmnyndigit(dbproc, mnyptr, value, zero) DBPROCESS *dbproc; DBMONEY *mnyptr; DBCHAR *value; DBBOOL *zero;
dbmnyscale	Multiplica un valor DBMONEY por un entero positivo y agrega una cantidad especificada. **Sintaxis:** RETCODE dbmnyscale(dbproc, mnyptr, multiplier, addend) DBPROCESS *dbproc; DBMONEY *mnyptr; int multiplier; int addend;
dbmnysub	Resta un valor DBMONEY de otro. **Sintaxis:** RETCODE dbmnysub(dbproc, m1, m2, difference) DBPROCESS *dbproc; DBMONEY *m1; DBMONEY *m2; DBMONEY *difference;
dbmnyzero	Inicializa un valor DBMONEY a $0.0000. **Sintaxis:** RETCODE dbmnyzero(dbproc, mnyptr)

Rutina	Descripción
	DBPROCESS *dbproc; DBMONEY *mnyptr;
dbmonthname	Determine el nombre de un mes especificado en un idioma especificado. **Sintaxis:** char *dbmonthname(dbproc, language, monthnum,shortform) DBPROCESS *dbproc; char *language; int monthnum; DBBOOL shortform;
DBMORECMDS	Indique si hay más órdenes a ser procesadas. **Sintaxis:** RETCODE DBMORECMDS(dbproc) DBPROCESS *dbproc;
dbmoretext	Envía parte de un texto o de imagen al servidor. **Sintaxis:** RETCODE dbmoretext(dbproc, size, text) DBPROCESS *dbproc; DBINT size; BYTE *text;
dbmsghandle	Instala una función para el usuario, para manejar los mensajes del servidor. **Sintaxis:** int (*dbmsghandle(handler))() int (*handler)();
dbname	Devuelve el nombre de la base de datos actual. **Sintaxis:** char *dbname(dbproc) DBPROCESS *dbproc;
dbnextrow	Lee la próxima fila del buffer de resultados. **Sintaxis:** STATUS dbnextrow(dbproc) DBPROCESS *dbproc;
dbnextrow_a	(Sólo VMS) Asincrónicamente lea la próxima fila del buffer de resultados.

Rutina	Descripción
	Sintaxis: STATUS dbnextrow_a(dbproc, final_result, ast_proc, ast_param) DBPROCESS *dbproc; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbnpcreate	Cree un procedimiento de notificación. **Sintaxis:** RETCODE dbnpcreate(dbproc) DBPROCESS *dbproc;
dbnpdefine	Define un procedimiento de notificación. **Sintaxis:** RETCODE dbnpdefine(dbproc, procedure_name, namelen) DBPROCESS *dbproc; DBCHAR *procedure_name; DBSMALLINT namelen;
dbnullbind	Asocie una variable con una columna de fila de resultados. **Sintaxis:** RETCODE dbnullbind(dbproc, column, indicator) DBPROCESS *dbproc; int column; DBINT *indicator;
dbnumalts	Devuelve el número de columnas compute la una fila. **Sintaxis:** int dbnumalts(dbproc, computeid) DBPROCESS *dbproc; int computeid;
dbnumcols	Determina el número de columnas regulares para el conjunto actual de resultados. **Sintaxis:** int dbnumcols(dbproc) DBPROCESS *dbproc;
dbnumcompute	Devuelve el número de cláusulas compute en el conjunto actual de resultados. **Sintaxis:** int dbnumcompute(dbproc)

Rutina	Descripción
	DBPROCESS *dbproc;
DBNUMORDERS	Devuelve el número de columnas que están involucradas en la sentencia order by de una orden de select de transact-SQL. **Sintaxis:** int DBNUMORDERS(dbproc) DBPROCESS *dbproc;
dbnumrets	Determina el número de parámetros de retorno generado por un procedimiento almacenado. **Sintaxis:** int dbnumrets(dbproc) DBPROCESS *dbproc;
dbopen	Crea e inicializa una estructura de DBPROCESS. **Sintaxis:** DBPROCESS *dbopen(login, server) LOGINREC *login; char *server;
dbopen_a	(sólo VMS) Crea e inicializa una estructura de DBPROCESS asincrónicamente. **Sintaxis:** RETCODE dbopen_a(login, server, dbproc, final_result, ast_proc, ast_param) LOGINREC *login; char *server; DBPROCESS **dbprocptr; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbordercol	Devuelve el numero de columna que ocupa en la clausula del query y que esta listada en la cláusula order by. **Sintaxis:** int dbordercol(dbproc, order) DBPROCESS *dbproc; int order;
dbpoll	Checa si una contestación del servidor ha llegado por un DBPROCESS. **Sintaxis:** RETCODE dbpoll(dbproc, milliseconds, ready_dbproc,

Rutina	Descripción
	return_reason) DBPROCESS *dbproc; long milliseconds; DBPROCESS **ready_dbproc; int *return_reason;
dbpoll_a	(sólo VMS) Asincrónicamente Checa si una contestación del servidor ha llegado por un DBPROCESS. **Sintaxis:** RETCODE dbpoll(dbproc, milliseconds, ready_dbproc, return_reason, final_result, ast_proc, ast_param) DBPROCESS *dbproc; long milliseconds; DBPROCESS **ready_dbproc; int *return_reason; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbprhead	Imprime los títulos de la columna para filas devueltas por el servidor. **Sintaxis:** void dbprhead(dbproc) DBPROCESS *dbproc;
dbprrow	Imprime todas las filas devueltas por el servidor. **Sintaxis:** RETCODE dbprrow(dbproc) DBPROCESS *dbproc;
dbprtype	Convierte una cadena de valor token a una cadena entendible. **Sintaxis:** char *dbprtype(token) int token;
dbqual	Devuelve un apuntador a una cláusula where para ser usada en una actualización en el actual renglón de la tabla consultada. **Sintaxis:** char *dbqual(dbproc, tabnum, tabname) DBPROCESS *dbproc; int tabnum; char *tabname;
DBRBUF	(sólo UNIX y AOS/VS) Determina si las DB-library contienen en el buffer de red algún byte no leído.

Rutina	Descripción
	Sintaxis: DBBOOL DBRBUF(dbproc) DBPROCESS *dbproc;
dbreadpage	Lee una página de datos binarios del servidor. **Sintaxis:** DBINT dbreadpage(dbproc, dbname, pageno, buf) DBPROCESS *dbproc; char *dbname; DBINT pageno; BYTE buf[];
dbreadtext	Lee parte de un texto o de imagen del servidor. **Sintaxis:** STATUS dbreadtext(dbproc, buf, bufsize) DBPROCESS *dbproc; void *buf; DBINT bufsize;
dbrecftos	Graba todos los comandos de SQL enviados de la aplicación al servidor. **Sintaxis:** void dbrecftos(filename) char *filename;
dbrecvpassthru	Recibe un paquete de TDS de un servidor. **Sintaxis:** RETCODE dbrecvpassthru(dbproc, recv_bufp) DBPROCESS *dbproc; DBVOIDPTR *recv_bufp;
dbregdrop	Borra un procedimiento registrado. **Sintaxis:** RETCODE dbregdrop(dbproc, procedure_name, namelen) DBPROCESS *dbproc; DBCHAR *procedure_name; DBSMALLINT namelen;
dbregexec	Ejecute un procedimiento registrado. **Sintaxis:** RETCODE dbregexec(dbproc, options) DBPROCESS *dbproc;

Rutina	Descripción
	DBUSMALLINT options;
dbreghandle	Instala una rutina para una notificación del procedimiento registrado. **Sintaxis:** RETCODE dbreghandle(dbproc, procedure_name, namelen, handler) DBPROCESS *dbproc; DBCHAR *procedure_name; DBSMALLINT namelen; INTFUNCPTR handler;
dbreginit	Inicializa la ejecución de un procedimiento registrado. **Sintaxis:** RETCODE dbreginit(dbproc, procedure_name, namelen) DBPROCESS *dbproc; DBCHAR *procedure_name; DBSMALLINT namelen;
dbreglist	Devuelve una lista de procedimientos registrados definida en el Servidor Abierto actualmente. **Sintaxis:** RETCODE dbreglist(dbproc) DBPROCESS *dbproc;
dbregnowatch	Cancela una demanda a ser notificada cuando un procedimiento registrado se ejecuta. **Sintaxis:** RETCODE dbregnowatch(dbproc, procedure_name, namelen) DBPROCESS *dbproc; DBCHAR *procedure_name; DBSMALLINT namelen;
dbregparam	Define o describa un parámetro del procedimiento registrado. **Sintaxis:** RETCODE dbregparam(dbproc,param_name, type, datalen, data) DBPROCESS *dbproc; char *param_name; int type; DBINT datalen; BYTE *data;
dbregwatch	Pide ser notificado cuando un procedimiento registrado se ejecuta.

Rutina	Descripción
	Sintaxis: RETCODE dbregwatch(dbproc, procedure_name,namelen,options) DBPROCESS *dbproc; DBCHAR *procedure_name; DBSMALLINT namelen; DBUSMALLINT options;
dbregwatchlist	Devuelve una lista de procedimientos registrados que un DBPROCESS está vigilando. **Sintaxis:** RETCODE dbregwatchlist(dbproc) DBPROCESS *dbproc;
dbresults	Prepare los resultados del siguiente query. **Sintaxis:** RETCODE dbresults(dbproc) DBPROCESS *dbproc;
dbresults_a	(sólo VMS) Asincrónicamente prepare los resultados siguiente query. **Sintaxis:** RETCODE dbresults_a(dbproc, final_result, ast_proc, ast_param) DBPROCESS *dbproc; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbretdata	Devuelve un apuntador a un valor de parámetro de retorno generado por un procedimiento almacenado. **Sintaxis:** BYTE *dbretdata(dbproc, retnum) DBPROCESS *dbproc; int retnum;
dbretlen	Determine la longitud de un valor de parámetro de retorno generada por un procedimiento almacenado. **Sintaxis:** DBINT dbretlen(dbproc, retnum) DBPROCESS *dbproc; int retnum;

Rutina	Descripción
dbretname	Determine el nombre del parámetro del procedimiento almacenado asociado con un valor de parámetro de retorno en particular. **Sintaxis:** char *dbretname(dbproc, retnum) DBPROCESS *dbproc; int retnum;
dbretstatus	Determine el número de estado del procedimiento almacenado devuelto por el comando actual o la llamada del procedimiento remoto. **Sintaxis:** DBINT dbretstatus(dbproc) DBPROCESS *dbproc;
dbrettype	Determine el tipo de datos de un valor de parámetro de retorno generado por un procedimiento almacenado. **Sintaxis:** int dbrettype(dbproc, retnum) DBPROCESS *dbproc; int retnum;
DBROWS	Indica si el comando actual devolvió realmente renglones. **Sintaxis:** RETCODE DBROWS(dbproc) DBPROCESS *dbproc;
DBROWTYPE	Devuelve el tipo de la fila actual. **Sintaxis:** STATUS DBROWTYPE(dbproc) DBPROCESS *dbproc;
dbrpcinit	Inicializa una llamada de procedimiento remoto. **Sintaxis:** RETCODE dbrpcinit(dbproc, rpcname, options) DBPROCESS *dbproc; char *rpcname; DBSMALLINT options;
dbrpcparam	Agrega un parámetro a una llamada del procedimiento remoto. **Sintaxis:** RETCODE dbrpcparam(dbproc, paramname, status, type,maxlen, datalen, value)

Rutina	Descripción
	DBPROCESS *dbproc; char *paramname; BYTE status; int type; DBINT maxlen; DBINT datalen; BYTE *value;
dbrpcsend	Indica el fin de una llamada de procedimiento remoto. **Sintaxis:** RETCODE dbrpcsend(dbproc) DBPROCESS *dbproc;
dbrpwclr	Limpia todas las contraseñas remotas de la estructura de LOGINREC. **Sintaxis:** void dbrpwclr(loginrec) LOGINREC *loginrec;
dbrpwset	Agrega una contraseña remota a la estructura de LOGINREC. **Sintaxis:** RETCODE dbrpwset(loginrec, srvname, password, pwlen) LOGINREC *loginrec; char *srvname; char *password; int pwlen;
dbsafestr	Doble las cuotas en una cadena de caracteres. **Sintaxis:** RETCODE dbsafestr(dbproc, src, srclen, dest, destlen, quotetype) DBPROCESS *dbproc; char *src; DBINT srclen; char *dest; DBINT destlen; int quotetype;
dbsechandle	Instala una función de el usuario para ocuparse de login seguro. **Sintaxis:** RETCODE *dbsechandle(type, handler) DBINT type; INTFUNCPTR (*handler)();

Rutina	Descripción
dbsendpassthru	Envía un paquete de TDS a un servidor. **Sintaxis:** RETCODE dbsendpassthru(dbproc, send_bufp) DBPROCESS *dbproc; DBVOIDPTR send_bufp;
dbservcharset	Consiga el nombre del juego de carácter del servidor. **Sintaxis:** char *dbservcharset(dbproc) DBPROCESS *dbproc;
dbsetavail	Marque un DBPROCESS como estar disponible para el uso general. **Sintaxis:** void dbsetavail(dbproc) DBPROCESS *dbproc;
dbsetbusy	Llame una función del usuario cuando las DB-library están leyendo del servidor. **Sintaxis:** void dbsetbusy(dbproc, busyfunc) DBPROCESS *dbproc; int (*(*busyfunc)())();
dbsetdefcharset	Pone un juego de caracteres predefinido para una aplicación. **Sintaxis:** RETCODE dbsetdefcharset(charset) char *charset;
dbsetdeflang	Pone un idioma predefinido para una aplicación. **Sintaxis:** RETCODE dbsetdeflang(language) char *language;
dbsetidle	Llama una función del usuario cuando las DB-library terminaron la lectura desde el servidor. **Sintaxis:** void dbsetidle(dbproc, idlefunc) DBPROCESS *dbproc; void (*idlefunc)();
dbsetifile	Especifica el nombre y localización del archivo de interfaces de Sybase.

Rutina	Descripción
	Sintaxis: void dbsetifile(filename) char *filename;
dbsetinterrupt	Llama una función del usuario para ocuparse de interrupciones mientras espera una lectura desde el servidor. **Sintaxis:** void dbsetinterrupt(dbproc, chkintr, hndlintr) DBPROCESS *dbproc; int (*chkintr)(); int (*hndlintr)();
DBSETLAPP	Pone el nombre de la aplicación en la estructura de LOGINREC. **Sintaxis:** RETCODE DBSETLAPP(loginrec, application) LOGINREC *loginrec; char *application;
DBSETLCHARSET	Pone el juego de caracteres en la estructura de LOGINREC. **Sintaxis:** RETCODE DBSETLCHARSET(loginrec, char_set) LOGINREC *loginrec; DBCHAR *char_set;
DBSETLENCRYPT	Especifica si o no la encriptación de contraseña será usado para acceder al Servidor de SQL (10.0+) **Sintaxis:** RETCODE DBSETLENCRYPT(loginrec, enable) LOGINREC *loginrec; DBBOOL enable;
DBSETLHOST	Pone el nombre del servidor en la estructura de LOGINREC. **Sintaxis:** RETCODE DBSETLHOST(loginrec, hostname) LOGINREC *loginrec; char *hostname;
DBSETLLABELED	Pone un bit para informar SQL Server™ Seguro que se enviarán las etiquetas de seguridad del login al momento de login. **Sintaxis:** RETCODE DBSETLLABELED(loginrec, enable) LOGINREC *loginrec;

Rutina	Descripción
	DBBOOL enable;
DBSETLNATLANG	Pone el idioma en la estructura de LOGINREC. **Sintaxis:** RETCODE DBSETLNATLANG(loginrec, language) LOGINREC *loginrec; char *language;
dbsetloginfo	Transfiera la información de una estructura de DBLOGINFO a una estructura de LOGINREC. **Sintaxis:** RETCODE dbsetloginfo(loginrec, loginfo) LOGINREC *login; DBLOGINFO *loginfo;
dbsetlogintime	Pone el número de segundos que las DB-library esperaran por una contestación del servidor a una demanda para una conexión de DBPROCESS. **Sintaxis:** RETCODE dbsetlogintime(seconds) int seconds;
DBSETLPACKET	Pone el tamaño del paquete TDS en la estructura de LOGINREC de una aplicación. **Sintaxis:** RETCODE DBSETLPACKET(login, packet_size) LOGINREC *login; short packet_size;
DBSETLPWD	Pone la contraseña de usuario del servidor en la estructura de LOGINREC. **Sintaxis:** RETCODE DBSETLPWD(loginrec, password) LOGINREC *loginrec; char *password;
DBSETLUSER	Pone el username en la estructura de LOGINREC. **Sintaxis:** RETCODE DBSETLUSER(loginrec, username) LOGINREC *loginrec; char *username;

Rutina	Descripción
dbsetmaxprocs	Pone el número máximo de estructuras de DBPROCESS simultáneamente abiertas. **Sintaxis:** RETCODE dbsetmaxprocs(maxprocs) int maxprocs;
dbsetnotifs	(sólo VMS) Habilite o desactive las notificaciones del procedimiento registrados. **Sintaxis:** RETCODE dbsetnotifs(notif_state) DBBOOL notif_state;
dbsetnull	Define el valor de substitución para ser usada al ligar los valores nulos. **Sintaxis:** RETCODE dbsetnull(dbproc, bindtype, bindlen, bindval) DBPROCESS *dbproc; int bindtype; int bindlen; BYTE *bindval;
dbsetopt	Pone opciones para un servidor o las DB-library. **Sintaxis:** RETCODE dbsetopt(dbproc, option, char_param,int_param) DBPROCESS *dbproc; int option; char *char_param; int int_param;
dbsetrow	Pone un renglón del buffered como el actual. **Sintaxis:** STATUS dbsetrow(dbproc, row) DBPROCESS *dbproc; DBINT row;
dbsetsecurity	Fija los valores de seguridad del login para el uso del servidor al logear en un Servidor de SQL Seguro. **Sintaxis:** RETCODE dbsetsecurity(loginrec, labelname, labelvalue) LOGINREC *loginrec; CHAR *labelname; CHAR *labelvalue;

Rutina	Descripción
dbsettime	Pone el número de segundos que las DB-library esperarán por una contestación del servidor a un comando SQL. **Sintaxis:** RETCODE dbsettime(seconds) int seconds;
dbsetuserdata	Usa una estructura de DBPROCESS para guardar un apuntador a los datos de usuario asignados. **Sintaxis:** void dbsetuserdata(dbproc, ptr) DBPROCESS *dbproc; BYTE *ptr;
dbsetversion	Especifique un nivel de versión de la DB-library. **Sintaxis:** RETCODE dbsetversion(version) DBINT version;
dbspid	Recupera el id del proceso de servidor para el DBPROCESS especificado. **Sintaxis:** int dbspid(dbproc) DBPROCESS *dbproc;
dbspr1row	Pone un renglón de resultados del servidor en un buffer. **Sintaxis:** RETCODE dbspr1row(dbproc, buffer, buf_len) DBPROCESS *dbproc; char *buffer; DBINT buf_len;
dbspr1rowlen	Determina el tamaño del buffer para asignar los resultados devueltos por las funciones dbsprhead, dbsprline, y dbspr1row. **Sintaxis:** DBINT dbspr1rowlen(dbproc) DBPROCESS *dbproc;
dbsprhead	Pone los titulos de un query dentro del buffer. **Sintaxis:** RETCODE dbsprhead(dbproc, buffer, buf_len) DBPROCESS *dbproc; char *buffer;

Rutina	Descripción
	DBINT buf_len;
dbsprline	Crea una estructura que contiene el subrayado para los nombres de las columnas producido por el dbsprhead. **Sintaxis:** RETCODE dbsprline(dbproc, buffer, buf_len, linechar) DBPROCESS *dbproc; char *buffer; DBINT buf_len; DBCHAR linechar;
dbsqlexec	Envía un lote de comandos al servidor. **Sintaxis:** RETCODE dbsqlexec(dbproc) DBPROCESS *dbproc;
dbsqlexec_a	(sólo VMS) Envía un lote de comandos al servidor y verifique su exactitud asincrónicamente. **Sintaxis:** RETCODE dbsqlexec_a(dbproc,final_result,ast_proc,ast_param) DBPROCESS *dbproc; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbsqlok	Espere por los resultados del servidor y verifique la exactitud de las instrucciones que el servidor está respondiendo. **Sintaxis:** RETCODE dbsqlok(dbproc) DBPROCESS *dbproc;
dbsqlok_a	(sólo VMS) Espera por los resultados del servidor y verifique la exactitud de las instrucciones asincrónicamente. **Sintaxis:** RETCODE dbsqlok_a(dbproc,final_result,ast_proc,ast_param) DBPROCESS *dbproc; RETCODE *final_result; void (*ast_proc)(); BYTE *ast_param;
dbsqlsend	Envíe un lote de comandos al servidor y no espere por una contestación.

Rutina	Descripción
	Sintaxis: RETCODE dbsqlsend(dbproc) DBPROCESS *dbproc;
dbstrbuild	Construya una cadena imprimible de texto que contiene el placeholders por variable. **Sintaxis:** int dbstrbuild(dbproc, charbuf, bufsize,text [, formats [, arg] ...]) DBPROCESS *dbproc; char *charbuf; int bufsize; char *text; char *formats; ??? args???;
dbstrcmp	Compara dos cadenas de caracteres especificando el tipo de ordenamiento. **Sintaxis:** int dbstrcmp(dbproc, str1, len1, str2, len2,sortorder) DBPROCESS *dbproc; char *str1; int len1; char *str2; int len2; DBSORTORDER *sortorder;
dbstrcpy	Copia una porción del buffer de comandos a una variable. **Sintaxis:** RETCODE dbstrcpy(dbproc, start, numbytes, dest) DBPROCESS *dbproc; int start; int numbytes; char *dest;
dbstrlen	Devuelve la longitud, en caracteres, del buffer de comandos. **Sintaxis:** int dbstrlen(dbproc) DBPROCESS *dbproc;
dbstrsort	Determina cual de dos cadenas de caracteres deben aparecer primero en una lista ordenada. **Sintaxis:** int dbstrsort(dbproc, str1, len1, str2, len2, sortorder) DBPROCESS *dbproc;

Rutina	Descripción
	char *str1; int len1; char *str2; int len2; DBSORTORDER *sortorder;
dbtabbrowse	Determine si la tabla especificada es una tabla actualizable vía los medios del modo browse de la DB-library. **Sintaxis:** DBBOOL dbtabbrowse(dbproc, tabnum) DBPROCESS *dbproc; int tabnum;
dbtabcount	Devuelve el número de tablas involucradas en un select. **Sintaxis:** int dbtabcount(dbproc) DBPROCESS *dbproc;
dbtabname	Devuelve el nombre de una tabla basado en su número. **Sintaxis:** char *dbtabname(dbproc, tabnum) DBPROCESS *dbproc; int tabnum;
dbtabsource	Devuelve el nombre y número de la tabla desde una particular columna de resultado cuando fue derivada. **Sintaxis:** char *dbtabsource(dbproc, colnum, tabnum) DBPROCESS *dbproc; int colnum; int *tabnum;
DBTDS	Determina qué versión de TDS (Tabular Data String protocolo) está usándose. **Sintaxis:** int DBTDS(dbproc) DBPROCESS *dbproc;
dbtextsize	Regresa el número de bytes de texto o datos de la imagen que requiere ser leídos para el renglón actual. **Sintaxis:** DBINT dbtextsize(dbproc) DBPROCESS *dbproc;

Rutina	Descripción
dbtsnewlen	Devuelve el tamaño del nuevo valor de la columna del timestamp después de una actualización en modo browse. **Sintaxis:** int dbtsnewlen(dbproc) DBPROCESS *dbproc;
dbtsnewval	Devuelve el nuevo valor de la columna del timestamp después de una actualización en modo browse. **Sintaxis:** DBBINARY *dbtsnewval(dbproc) DBPROCESS *dbproc;
dbtsput	Pone el nuevo valor de la columna del timestamp en la fila actual de la tabla dada en el DBPROCESS. **Sintaxis:** RETCODE dbtsput(dbproc, newts, newtslen, tabnum,tabname) DBPROCESS *dbproc; DBBINARY *newts; int newtslen; int tabnum; char *tabname;
dbtxptr	Devuelve el valor del apuntador del texto para una columna en la fila actual. **Sintaxis:** DBBINARY *dbtxptr(dbproc, column) DBPROCESS *dbproc; int column;
dbtxtimestamp	Devuelve el valor del texto para una columna timestamp en la fila actual. **Sintaxis:** DBBINARY *dbtxtimestamp(dbproc, column) DBPROCESS *dbproc; int column;
dbtxtsnewval	Devuelve el nuevo valor de un timestamp de texto después de una llamada a dbwritetext. **Sintaxis:** DBBINARY *dbtxtsnewval(dbproc) DBPROCESS *dbproc;
dbtxtsput	Pone el nuevo valor de un timestamp de texto en la columna

Rutina	Descripción
	especificada de la fila actual en el DBPROCESS. **Sintaxis:** RETCODE dbtxtsput(dbproc, newtxts, colnum) DBPROCESS *dbproc; DBBINARY *newtxts; int colnum;
dbuse	Use una base de datos en particular. **Sintaxis:** RETCODE dbuse(dbproc, dbname) DBPROCESS *dbproc; char *dbname;
dbvarylen	Determina si los datos de la columna de resultados especificada pueden variar en su longitud. **Sintaxis:** DBBOOL dbvarylen(dbproc, column) DBPROCESS *dbproc; int column;
dbversion	Determine qué versión de DB-library está en uso. **Sintaxis:** char *dbversion()
dbwillconvert	Determina si una conversión de tipo de datos específica está disponible dentro de las DB-Library. **Sintaxis:** DBBOOL dbwillconvert(srctype, desttype) int srctype; int desttype;
dbwritepage	Escribe una página de datos binarios hacia el servidor. **Sintaxis:** RETCODE dbwritepage(dbproc, dbname, pageno, size, buf) DBPROCESS *dbproc; char *dbname; DBINT pageno; DBINT size; BYTE buf[];
dbwritetext	Envíe un texto o imagen hacia el servidor.

Rutina	Descripción
	Sintaxis: RETCODE dbwritetext(dbproc, objname, textptr,textptrlen, timestamp, log,size, text) DBPROCESS *dbproc; char *objname; DBBINARY *textptr; DBTINYINT textptrlen; DBBINARY *timestamp; DBBOOL log; DBINT size; BYTE *text;
dbxlate	Traduzca una cadena desde un conjunto caracteres hacia otro. **Sintaxis:** int dbxlate(dbproc, src, srclen, dest, destlen, xlt, srcbytes_used, srcend, status) DBPROCESS dbproc; char *src; int srclen; char *dest; int destlen; DBXLATE *xlt; int *srcbytes_used; DBBOOL srcend; int *status;

Anexo 4. Sybase Adaptive Server Enterprise 12.5. Tablas de Sistema

Tablas del Sistema en Master:

syscharsets

La tabla syscharsets contiene una fila por cada juego de caracteres y criterios de ordenación definido para el uso del Adaptive Server. Uno de los criterios de ordenación está marcado en master..sysconfigures como el criterio de ordenación predeterminado, que es el único que está actualmente en uso.

Descripción de columnas:

Columna	Significado
type	El tipo de entidad que esta fila representa. Los Números a partir de 1001 a 1999 representan juegos de caracteres. Los Números a partir de 2000 a 2999 representan criterios de ordenamiento
id	Es el Id para el juego de caracteres o criterio de ordenamiento. El criterio de ordenamiento se define de la combinación del Id de criterio de ordenamiento y el Id del juego de caracteres (csid). El juego de caracteres se define por id, el cual debe ser único. Sybase reserva Números Id del 0-200
csid	Si la fila representa un juego de caracteres, este campo no se utiliza. Si la fila representa un criterio de ordenamiento, éste es el id del juego de caracteres que el criterio de ordenamiento construyó. Una fila del juego de caracteres debe existir en esta tabla
status	Bits de estado de información interna del sistema
name	Nombre único para cada juego de caracteres o criterio de ordenamiento. Deben contener solamente las letras A-z o a-z de 7-bit ASCII, los dígitos 0-9, y guiones bajos (_), y comienzan con una letra.
description	Una descripción opcional de las características del juego de caracteres o criterio de ordenamiento
definition	Definición interna del juego de caracteres o criterio de ordenamiento. La estructura de los datos en este campo depende de type
sortfile	Nombre del archivo del criterio de ordenamiento

sysconfigures

La tabla sysconfigures contiene un registro para cada parámetro de configuración que pueda ser establecido por el usuario.

Descripción de columnas:

Columna	Significado
config	Número del parámetro de configuración.
value	Es el valor del parámetro modificado por el usuario y es de tipo entero. Si el valor de la columna es 0 el parámetro es de tipo caracter.
comment	Nombre del parámetro de configuración.
status	Valor que representa el tipo de parámetro de configuración.
name	Nombre del parámetro de configuración.
parent	Número del parámetro de configuración padre; si existe más de un padre, los Números adicionales del padre se almacenan en sysattributes.
value2	Es el valor del parámetro modificado por el usuario para el parámetro con tipo de dato caracter. Su valor es NULO para los parámetros con el datatype de Número entero.
value3	Almacena el tamaño del wash del buffer pool.
value4	Almacena los porcentajes asincrónicos del prefetch de un buffer pool.

syscurconfigs

Contiene una entrada por cada parámetro de configuración, al igual que *sysconfigures,* pero con los valores actualmente utilizados por el Adaptive Server.

Descripción de columnas:

Columna	Significado
config	Número del parámetro de configuración
value	Es el valor actual para un parámetro de tipo entero. Si el valor de la columna es 0 el parámetro es de tipo caracter.
comment	Cantidad de memoria utilizada por cada parámetro de configuración representado en formato *string*.
status	El estado 1 significa “dinámico”, es decir, que los nuevos valores de estos parámetros tienen efecto inmediatamente cuando se ejecuta sp_configure. El estado 0 indica que el parámetro es estático y se aplica solo después de ejecutar sp_configure y reiniciar el servidor.
value2	El valor actual para un parámetro con tipo de dato caracter. El valor es Nulo para parámetros con tipo de dato entero.
defvalue	Valor por default de un parámetro de configuración.

minimum_value	Valor mínimo para un parámetro de configuración.
maximum_value	Valor máximo para un parámetro de configuración.
memory_used	Valor del Número entero para la cantidad de memoria usada por cada parámetro de la configuración.
display_level	Muestra el nivel del parámetro de configuración (los valores son 1, 5, y 10).
data_type	Tipo de dato del parámetro de configuración.
message_num	Número del mensaje del sp_helpconfig para este parámetro de configuración.
apf_percent	

sysdatabases

Contiene una fila por cada base de datos en el servidor, en la cual se especifica el nombre de la base de datos, el dueño y además le asigna un ID (dbid) para esa base de datos.

Descripción de columnas:

Columna	Significado
name	Nombre de la base de datos
dbid	ID de la base de datos
suid	ID de usuario del dueño de la base de datos
status	Bits de control; aquellas opciones que son definidas por un usuario con sp_dboption.
version	Sin usar
logptr	Puntero hacia el log de transacciones
crdate	Fecha de creación
dumptrdate	Fecha del último dump transaction
status2	Bits de control adicional
audflags	Definiciones de la auditoría para la base de datos
deftabaud	Máscara de bit que define las opciones de auditoría por default para tablas
defvwaud	Máscara de bit que define las opciones de auditoría por default para vistas
defpraud	Máscara de bit que define las opciones de auditoría para procedimientos almacenados
def_remote_type	Identifica el tipo de objetos por default que serán utilizados por tablas remotas, si no se provee de un lugar de almacenamiento vía el procedimiento almacenado sp_addobjectdef
status3	Control de bits adicionales
staus4	Control de bits adicionales

sysdevices

La tabla sysdevices de la base de datos master contiene una fila por cada dispositivo de base de datos y puede contener una fila para cada dispositivo de volcado (cinta, disco, archivo del sistema operativo) disponible para SQL Server.

Descripción de columnas:

Columna	Significado
low	Representa el primer Número de página virtual asignada a un dispositivo
high	Representa el último Número de página virtual asignada a un dispositivo
status	Es un campo de mapa de bits que indica: el tipo de dispositivo; si se utiliza un dispositivo predeterminado de almacenamiento o de duplicación de disco. (Ver tabla A para bits de estado).
cntrltype	Tipo de controlador (dispositivo de base de datos = 0, dispositivo de volcado de disco = 2 , dispositivos de cinta = 3 – 8)
name	Nombre lógico del dispositivo.
phyname	Nombre físico del dispositivo, es el nombre real del dispositivo en el sistema operativo.
mirrorname	Nombre del dispositivo de duplicación.

Tabla 1. Bits de estado

Decimal	Significado
1	Disco predeterminado
2	Disco físico
4	Disco Lógico (no utilizado)
8	Saltar encabezado (usado con dispositivos de volcado de cinta)
16	Dispositivo de volcado
32	Escrituras en serie
64	Dispositivos duplicado
128	Lecturas duplicadas
256	Solo duplicación del lado secundario
512	Duplicación activada
2048	Uso interno; duplicación desactivada

sysengines

sysengines contiene una fila para cada motor del Adaptive Server actualmente en línea.

Descripción de columnas:

Columna	Significado
engine	Número de motor
osprocid	Id de proceso del sistema operativo (puede ser NULL)
osprocname	Nombre de proceso del sistema operativo (puede ser NULL)
status	Puede ser: online, in offline, in create, in destroy, debug, bad status
affinitied	Número de los procesos del Adaptive Server con afinidad a este motor
cur_kpid	Id de proceso del kernel, actualmente en ejecución en este motor, si lo hay
last_kpid	Id de proceso del kernel, de un proceso previamente ejecutado en este motor
Idle_1	Reservado
Idle_2	Reservado
Idle_3	Reservado
starttime	Fecha y hora en que el motor estuvo en línea.

syslanguages

Contiene un registro por cada idioma conocido por el Adaptive Server. us_english no se encuentra en syslanguages, pero siempre está disponible para el servidor.

Descripción de columnas:

Columna	Significado
langid	Id único del idioma
dateformat	Orden de fecha: por ejemplo, “dmy”
datefirst	Primer día de la semana, 1 para lunes, 2 martes, etcétera, hasta el 7 para domingo
upgrade	Versión del Adaptive Server de la última actualización del idioma
name	Nombre oficial del idioma, por ejemplo “french”
alias	Nombre alterno del idioma, por ejemplo “français”
months	Lista de los nombres completos del mes separada por comas, en orden de Enero a Diciembre, cada nombre en la mayoría tiene una longitud de 20 caracteres
shortmonths	Lista de nombres cortos del mes separada por comas, en orden de Enero a Diciembre, cada nombre en la mayoría tiene una longitud de 9 caracteres

days	Lista de nombres de los días separada por comas, en orden de lunes a domingo, cada nombre en la mayoría tiene una longitud de 30 caracteres

syslisteners

syslisteners contiene un registro por cada protocolo de red disponible para la conexión con el Adaptive Server actual. Adaptive Server construye syslisteners cuando la aplicación del usuario o cliente consulta la tabla.

Descripción de columnas:

Columna	Significado
net_type	Protocolo de red
address_info	Información que identifica únicamente este servidor en la red, generalmente el nombre del servidor actual y un número identificador, como el número de puerto del servidor para el protocolo

syslocks

syslocks contiene información de los bloqueos activos. Se construye dinámicamente cuando es consultado por un usuario. Las actualizaciones a syslocks no están permitidas.

Descripción de columnas:

Columna	Significado
id	Id de la tabla
dbid	Id de la base de datos
page	Número de página
type	Tipo de bloqueo (los valores de bit para la columna type se listan en la tabla 2)
spid	Id del proceso que mantiene el bloqueo
class	Nombre del cursor con el cual se asocia el bloqueo
fid	Id de la familia (coordinando el proceso y sus procesos de trabajo a la cual pertenece el bloqueo. Los valores para fid se enlista en la tabla 3)
context	Tipo de contexto de la petición de bloqueo. Los valores de contexto se enlistan en la tabla 4
row	Número de fila
loid	Id único del dueño del bloqueo

Tabla 2. Lista de representaciones de bit para la columna type.

Tabla 2. Bits de control para la columna type

Decimal	status
1	Bloqueo exclusivo de tabla
2	Bloqueo compartido de tabla
3	Intento de bloqueo exclusivo
4	Intento de bloqueo compartido
5	Bloqueo exclusivo de página
6	Bloqueo compartido de página
7	Bloqueo para actualización de página
8	Bloqueo exclusivo de fila
9	Bloqueo compartido de fila
10	Bloqueo de actualización de fila
11	Bloqueo compartido de llave siguiente
256	Bloqueo que bloquea otro proceso
512	Bloqueo de demanda

Tabla 3. Lista de valores para la columna fid.

Tabla 3. Lista de valores para la columna fid de la tabla syslocks

Valor	Interpretación
0	La tarea representada por el spid es una sola tarea que ejecuta una declaración en la serie.
Nonzero	La tarea (spid) que sostiene el bloqueo es un miembro de una familia que ejecuta una declaración en paralelo. Si el valor es igual al spid, indica que la tarea es el proceso que coordina a una familia que ejecuta una consulta en paralelo.

Tabla 4. Lista de valores para la columna context.

Tabla 4. Lista de valores para la columna context de la tabla syslocks

Valor	Interpretación
null	La tarea que mantiene este bloqueo es cualquiera que ejecuta un query en serie, o es un query que está siendo ejecutado en paralelo dentro de un transaction isolation level 1.
0x1	La tarea que mantiene el bloqueo seguirá bloqueada hasta que el query se complete. El contexto del bloqueo puede ser “Fam dur” cuando:

	• El bloqueo es un bloqueo de tabla mantenido como parte de un query en paralelo • El bloqueo se mantiene por un worker process en un transaction isolation level 3. • El bloqueo se mantiene por un worker process dentro de un query en paralelo y debe mantenerse durante la duración de la transacción.
0x2	Bloqueo sostenido por un "work process" en un query en paralelo y debe mantenerse durante la duración de una transacción.
0x4	Bloqueo de afinidad de llave
0x8	Bloqueo conseguido en las páginas de índice de una tabla con "allpages-locked"
0x10	Bloqueo sobre una página o fila conseguido para borrar una fila
0x20	Dirección de bloqueo conseguida sobre una página de índice durante un *shrink* o *split operation*
0x40	Bloqueo intencionado sostenido por una transacción para realizar lecturas repetidas. Válido únicamente para la intención de lograr un bloqueo compartido o un bloqueo exclusivo sobre datos únicamente de tablas bloqueadas.

sysloginroles

Esta tabla ***sysloginroles*** contiene un registro por cada role de sistema asignado a un login de usuario. Cada vez que se otorgue un role a un login, se agregará un nuevo registro en esta tabla.

Descripción de columnas:

Columna	Significado
suid	Es el ID del usuario
srid	Es el ID del role, ▪ 0 = sa_role ▪ 1 = sso_role ▪ 2 = oper_role ▪ 3 = navigator_role ▪ 4 = replication_role
status	Reservado

syslogins

Esta tabla contiene un registro por cada cuenta de usuario válida.

Descripción de columnas:

Nombre	Descripción
suid	ID de usuario del servidor
status	Status de la cuenta (ver tabla 5)
accdate	Fecha de ultima limpieza de totcpu y totio
totcpu	Tiempo de CPU acumulado por el login
totio	Tiempo de I/O acumulado por el login
spacelimit	Reservado
timelimit	Reservado
resultlimit	Reservado
dbname	Nombre de la base de datos por default en la cual se colocara el usuario al entrar al servidor.
name	Nombre de la cuenta de login
password	Password de la cuenta de login encriptado
language	Lenguaje asignado a esa cuenta
pwdate	Día de la ultima modificación del password
audflags	Definiciones de auditoría de usuarios
fullname	Nombre completo del usuario
srvname	Nombre del Servidor
logincount	Número de intentos fallidos para una cuenta de login
procid	Almacena el trigger de conexión registrado con el script en la opción login de sp_modifylogin.

Tabla 5. Lista de Representaciones de bit para la columna status.

Tabla 5. Bits de control para la columna status de la tabla syslogins

Decimal	Interpretación
1	El password contiene menos de 6 caracteres o es NULL
2	La cuenta esta bloqueada
4	El password ha expirado

syslogshold

syslogshold contiene información acerca de cada transacción activa más antigua de una base de datos o el punto de truncado de Replication Server para el diario de la base de datos, pero no es una tabla normal. Más bien es construido dinámicamente cuando es consultado por un usuario. Las actualizaciones hacia sysholds no están permitidas.

Descripción de columnas:

Nombre	Descripción
dbid	Id de la base de datos
reserved	Sin uso
spid	Id de proceso del usuario que es dueño de la transacción anterior (0 para Replication Server)
page	Número de inicio de página de una porción activa en syslogs definida por una transacción antigua (o la página de truncamiento en syslogs para Replication Server)
xactive	Id de la transacción activa más antigua (siempre 0x000000 para Replication Server)
masterxactid	Id de una transacción *master*, para las transacciones multi-base de datos; de otra manera 0x000000 (siempre 0x000000 para Replication Server).
starttime	Fecha y hora del inicio de una transacción (o cuando el punto de truncamiento fue definido por Replication Server)
name	Nombre de la transacción activa más antigua. Es el nombre definido con begin transaction, "$user_transaction" si no se especifica ningún valor con begin transaction o "$chained_transaction" para el inicio de transacciones implícitas por el modo ANSI chained. El inicio de transacciones internas por el servidor tienen nombres que comienzan con el signo de dólar ($) y son designadas por la operación, o son llamadas "$replication_truncation_point" para Replication Server.

sysmessages

sysmessages contiene un registro por cada error o advertencia del sistema que puede ser devuelta por el Adaptive Server. El Servidor muestra la descripción del error sobre la pantalla del usuario.

Descripción de columnas:

Columna	Significado
error	Número de error único
severity	Nivel de severidad del error
dlevel	Reservado
description	Explicación del error con los placeholders para los parámetros
langid	Lenguaje ; null para us_english
sqlstate	Valor SQLSTATE para el error

sysmonitors

sysmonitors contiene un registro por cada contador de monitoreo.

Descripción de columnas:

Columna	Significado
field_name	Nombre del contador
group_name	Grupo al cual pertenece el contador
field_id	Identificador único para la fila
value	Valor actual del contador
description	Descripción del contador; no utilizado

sysprocesses

Contiene información acerca de los procesos en el Adaptive Server. Se construye dinámicamente cuando es consultada por un usuario. Las actualizaciones hacia sysprocesses no están permitidas. Utilice la declaración kill para matar un proceso.

Descripción de columnas:

Columna	Significado
spid	Id del proceso
kpid	Id de proceso del kernel
enginenum	Número del motor sobre el cual se está ejecutando el proceso
status	Estado del Id del proceso (ver tabla 6)
suid	Es el ID del usuario que utilizó el comando
hostname	Nombre del host
program_name	Nombre de la aplicación específica
hostprocess	Número Id del hostprocess
cmd	Comando o proceso que está ejecutándose actualmente. La evaluación de una declaración condicional, como un if o un while loop, devuelve cond.
cpu	Tiempo de cpu acumulado por el proceso en ticks.
physical_io	Número de lectura y escritura en disco para el comando actual
memusage	Cantidad de memoria asignada al proceso
blocked	Id del proceso bloqueado, si es que lo hay
dbid	Id de la base de datos
uid	Id del usuario que ejecutó el proceso
gid	Id de grupo del usuario que ejecutó el comando
tran_name	Nombre de la transacción activa
time_blocked	Tiempo del bloqueo en segundos
network_pktsz	Tamaño del paquete de la conexión de red actual

fid	Id de los procesos de trabajo padre
execlass	Ejecución de clase a la cual el proceso está destinado
priority	Prioridad base asociada al proceso
affinity	Nombre del motor con el cual, el proceso tiene afinidad
Id	ID del procedimiento que corre actualmente (ó 0 si ningún procedimiento esta corriendo)
stmtnum	Número de la declaración actual dentro de la ejecución de un procedimiento (o el Número de la declaración de un SQL batch si ningún procedimiento se está ejecutando)
linenum	El Número de la línea de la declaración actual dentro de la ejecución de un procedimiento almacenado (o el Número de la línea de la declaración actual de un SQL batch si ningún procedimiento se está ejecutando)
origsuid	Id del usuario del servidor original. Si este valor no es NULO, un usuario con un suid del origsuid que ejecute set proxy o set session authorization para imitar al usuario que ejecutó el comando.
block_xloid	Identificador único de un bloqueo que está bloqueando una transacción
clientname	Nombre con el cual es conocido un usuario dentro de una sesión actual. Este parámetro es opcional
clienthostname	Nombre con el cual es conocido el host de una sesión actual. Este parámetro es opcional
clientapplname	Nombre con el cual es conocida la aplicación de una sesión actual. Este parámetro es opcional
sys_id	Identidad única del nodo acompañante
ses_id	Identidad única de cada sesión del cliente
loggedindatetime	Muestra la fecha y hora en que el cliente se conectó con el Adaptive Server.
ipaddr	Dirección IP desde la cual se conecta el cliente

Tabla 6. Lista de valores para la columna status.

Tabla 6. Valores para la columna status de la tabla sysprocesses

status	Significado
alarm sleep	Espera una alarma, como waitfor delay
background	Un proceso, como un procedimiento de umbral, ejecutado por SQL Server, no por un proceso de usuario
infected	El servidor ha detectado un error grave; situación muy rara
latch sleep	Espera lograr un match
lock sleep	Espera lograr un bloqueo
PLC sleep	Espera tener acceso a un log caché de usuario
recv sleep	Espera una lectura de la red
runnable	Está en la cola de procesos a ejecutar

running	Se ejecuta activamente en uno de los motores del servidor
send sleep	Espera un envío de la red
sleeping	Espera E/S del disco o algún otro recurso (indica probablemente un proceso que está en ejecución, pero haciendo un uso intenso de la E/S de disco)
stopped	Procesos detenidos
sync sleep	Espera un mensaje de sincronización de otro proceso en la familia

sysremotelogins

sysremotelogins contiene una fila para cada usuario remoto al cual permite ejecutar una llamada a un procedimiento remoto sobre el Adaptive Server.

Descripción de columnas:

Columna	Significado
remoteserverid	Identifica al servidor remoto
remoteusername	Nombre del login del usuario del servidor remoto
suid	Id de usuario del servidor local
status	Bitmap de opciones

sysresourcelimits

La tabla sysresourcelimits contiene un registro por cada límite de recurso definido por Adaptive Server. Los límites de recursos son un conjunto de restricciones que un Administrador de Sistema puede crear para prevenir que un login (y/o una aplicación) monopolicen los recursos del servidor.

Descripción de columnas:

Columna	Significado
name	Nombre del login al que se le aplica el límite
appname	Nombre de la aplicación a la que el límite aplica
rangeid	Columna Id en la tabla systimeranges. Rango de tiempo durante el cual el límite trabaja
limitid	Columna Id en la tabla spt_limit_types
enforced	Subconjunto de la columna enforced en la tabla spt_limit_types • 1 = prior to execution • 2 = during execution • 3 = both
actiontotake	Acción a asumir cuando ocurra una violación al límite • 1 = límites que emiten una advertencia • 2 = límites que abortan el query batch • 3 = límites que abortan la transaction • 4 = límites que matan la sesión

limitvalue	Valor del límite
scope	Alcance del límite (éste es uno de los siguientes:) • 1 = límites que aplican a “query” • 2 = límites que aplican a “query batch” • 4 = límites que aplican a “transaction”
spare	Reservado

syssecmechs

La tabla syssecmechs contiene la información sobre los servicios de seguridad respaldados por cada mecanismo de seguridad que esté disponible para el Adaptive Server.

Descripción de columnas:

Columna	Significado
sec_mech_name	Nombre del mecanismo de seguridad, por ejemplo, “NT LANMANAGER”
available_service	Nombre del servicio de seguridad respaldado por un mecanismo de seguridad; por ejemplo, " unified login”

sysservers

La tabla sysservers contiene un registro por cada Adaptive Server remoto, Backup Server™ u Open Server™, sobre el cual este Adaptive Server puede ejecutar llamadas de procedimiento remotos.

Descripción de columnas:

Columna	Significado
srvid	Número ID (únicamente para uso local) del servidor remoto
srvstatus	Btimap de opciones (Ver tabla 7)
srvname	Nombre del servidor
srvnetname	Nombre en el archivo de interfaces para el servidor
srvclass	Categoría del servidor, definida por el parámetro class en el procedimiento sp_addserver (ver tabla 8)
srvsecmech	Mecanismo de seguridad
srvcost	Proporciona el costo de red en milisegundos para tener acceso a un servidor sobre una red. Utilizado únicamente por el query optimizer del Adaptive Server para evaluar el costo de un query al tener acceso a una tabla proxy, el default esta definido a 1.000 ms.

Tabla 7. Lista de Representaciones de bit para la columna srvstatus.

Tabla 7. Bits de control de status de la tabla sysservers

Decimal	Status
0	Las interrupciones están permitidas
1	Las interrupciones están permitidas
2	Se permite la encriptación del password de red
4	El servidor remoto es de solo lectura
8	Utilice el modelo de seguridad A del rpc
16	Utilice el modelo de seguridad B del rpc
64	Utilice la confidencialidad en el mensaje
128	Utilice integridad en el mensaje
256	Autenticación mutua

Tabla 8. Lista de categorías del servidor para la columna srvclass.

Tabla 8. Categorías del servidor en la tabla sysservers

srvclass	Server Category
0	Servidor Local (este servidor)
1	Otro Adaptive Server o Servidor de Servicios de Integración Componentes
3	Servidor cifrado a la especificación de DirectCONNECT
4	Servidor accesible por Net-Gateway o MDI Database Gateway
5	Servidor cifrado a la especificación de Generic Access Module

syssessions

sysessions se utiliza únicamente cuando el Adaptive Server se configura para el Failover de Sybase en un sistema de alta disponibilidad. syssessions contiene un registro por cada cliente que se conecta al Adaptive Server con la propiedad failover (por ejemplo, isql-Q). Los clientes que tienen una entrada en syssessions durante un failover son movidos al acompañante secundario. Los clientes que no tengan una entrada en syssessions son eliminados durante un failover. Los clientes que cuentan con una entrada en syssessions durante un failback son movidos al acompañante primario. Los clientes que no cuentan con una entrada en syssessions durante un failback son eliminados.

Descripción de columnas:

Columna	Significado
sys_id	Identificador único de un nodo acompañante
ses_id	Identificados único para cada sesión de un cliente
state	Describe si la sesión está activa o no
spare	Reservado para funcionalidad futura
status	Reservado para funcionalidad futura
dbid	Reservado para funcionalidad futura
name	De igual manera el nombre de login de un cliente es especificado en syslogins

syssrvroles

La tabla ***syssrvroles*** contiene un registro por cada role de sistema o role definido a usuario.

Descripción de columnas:

Columna	Significado
srid	Es el ID del role
name	Nombre del role
password	Password para el role (encriptado)
pwdate	Fecha de última modificación del password
status	Mapeo de bits del estado del role (ver tabla 9)
logincount	Número de intentos fallidos, envía un 0 por conexión acertada.

Tabla 9. Bits de control de status en la tabla sysrvroles

Decimal	Hexadecimal	Status
2	0x2	Role bloqueado
4	0x4	Role expirado

systimeranges

Esta tabla almacena “nombres de rangos de tiempo”, que son utilizados por el servidor para controlar cuando un límite de recurso este activo.

Descripción de columnas:

Columna	Significado
name	Nombre único para el rango de tiempo
id	Identificador Id único para el rango de tiempo. 1 representa el límite “at all times”

startday	Día de la semana (1-7) para el inicio del rango. Lunes = 1, Domingo = 7
endday	Día de la semana (1-7) para el fin del rango. Lunes = 1, Domingo = 7
starttime	Hora del día para el inicio del rango
endtime	Hora del día para el fin del rango

systransactions

La tabla systransactions contiene información de las transacciones del Adaptive Server. Partes de la tabla son construidas dinámicamente cuando son consultadas por un usuario, mientras que otras partes son almacenadas en la base de datos master. Actualizar las columnas construidas dinámicamente no está permitido.

Descripción de columnas:

Columna	Significado
xactkey	*Transaction key* única del Adaptive Server
starttime	Fecha de inicio de la transacción
failover	Valor que indica el estado de la transacción *failover* (Ver tabla 10)
type	Valor que indica el tipo de transacción (Ver tabla 11)
coordinator	Valor que indica el método de coordinación o protocolo (Ver tabla 12)
state	Valor que indica el estado actual de la transacción (Ver tabla 13)
connection	Valor que indica el estado de la conexión (Ver tabla 14)
status	Bandera de estado de transacción interna
status2	Banderas de estado de transacción adicionales internas.
spid	Identificación de proceso del servidor, ó 0 si se separa el proceso
masterdbid	Comenzar la transacción en la base de datos
loid	Bloquear el Id del dueño
namelen	Longitud del “xactname"
xactname	Nombre de la transacción o XID
srvname	Nombre del servidor remoto (Null para el servidor local)

Tabla 10. Lista de valores para la columna failover.

Tabla 10. Valores para la columna failover de la tabla systransactions

Valor failover	Failover State
0	Resident Tx
1	Failed-over Tx
2	Tx by Failover-Conn

Tabla 11. Lista de valores para la columna type.

Tabla 11. Valores para la columna type de la tabla systransactions

Valor type	Transaction type
1	Local
3	External
98	Remote
99	Dtx_State

Tabla 12. Lista de valores para la columna coordinator.

Tabla 12. Valores para la columna coordinator de la tabla systransactions

Valor coordinator	método de coordinación o protocolo
0	None
1	Syb2PC
2	ASTC
3	XA
4	DTC

Tabla 13. Lista de valores para la columna state.

Tabla 13. Valores para la columna state de la tabla systransactions

Valor state	State transaction
1	Begun
2	Done Command
3	Done
4	Prepared
5	In Command
6	In Abort Cmd
7	Committed
8	In Post Commit
9	In Abort Tran
10	In Abort Savept
65537	Begun-Detached
65538	Done Cmd-Detached
65539	Done-Detached
65540	Prepared-Detached
65548	Heur Committed
65549	Heur Rolledback

Tabla 14. Lista de valores para la columna connection.

Tabla 14. Valores para la columna connection de la tabla systransactions

Valor connection	Connection State
1	Attached
2	Detached

sysusages

Lleva cuenta de todo el espacio asignado a todas las bases de datos del SQL Server. Esta tabla contiene un registro para cada fragmento del dispositivo asignado a cada base de datos, indicando el tamaño y la dirección del comienzo lógico de disco para ese fragmento.

Descripción de columnas:

Columna	Significado
dbid	ID de la base de datos
segmap	Mapeo de bit de posibles asignaciones de segmento
lstart	Primer página (lógica) de la base de datos
size	Número de las páginas (lógicas) contiguas de la base de datos
vstart	Número de inicio de la página virtual
pad	Sin uso
unreservedpgs	Espacio libre, que no es parte de un extent asignado

Tablas del Sistema en todas las Bases de Datos:

sysalternates

Esta tabla se encuentra en todas las bases de datos, además contiene un renglón para cada *login* con un *alias* hacia un usuario de la base de datos. Cuando un usuario intenta acceder a una base de datos, el servidor busca una entrada válida uid en sysusers. Si ninguno es encontrado, mira en sysalternates suid. Si suid del usuario es encontrado allí, el login es tratado como el usuario de la base de datos.

Descripción de columnas:

Columna	Significado
suid	Es el Server user ID del login, al que se le otorga el alias
altsuid	Server user ID, del usuario con el cual estamos entrando a la base

syscolumns

Contiene un registro por cada columna en cada tabla y vista, y un registro por cada parámetro en un procedimiento.

Descripción de columnas:

Columna	Significado
id	Identificador de la tabla a la cual esta columna pertenece o del procedimiento con el cual este parámetro es asociado
number	Número del subprocedimiento cuando el procedimiento está agrupado.
colid	Id de la columna
status	Bits 0-2 (valores 1, 2 y 4) indican la posición del bit, si la columna utiliza el tipo de dato bit. Si la columna utiliza el tipo de dato text/image, los bits 0 y 1 indicarán el estado de replica como sigue: • 01 = always replicate • 10 = replicate only if changed • 00 = never replicate Bit 3 (valor 8) indica si los valores NULOS son legales en esta columna. Bit 4 (valor 16) indica si más de un check constraint existe para la columna. Bit 5 y 6 son utilizados internamente. Bit 7 (valor 128) indica una columna identity. Bit 8 sin uso.
type	Tipo de almacenaje físico; copiado de systypes
length	Longitud física de datos; copiado de systypes o suministrado por el usuario
offset	Compensación en la fila donde ésta columna aparece; de ser negativo, sería una columna de longitud variable
usertype	Tipo de usuario ID; copiado de systypes
cdefault	ID del procedimiento que genera el valor de default para esta columna
domain	Constraint ID de la primera regla o check constraint para esta columna
name	Nombre de la Columna
printfmt	Reservado
prec	Número de dígitos significativos
scale	Número de dígitos a la derecha del punto decimal
remote_type	Correlaciona nombres locales hacia nombres remotos. Requerido por los métodos de acceso de servicios componentes de la integración para permitir que el software pase la información nativa del datatype de la columna en parámetros a los servidores del access_server de la clase.

remote_name	Correlaciona nombres locales hacia nombres remotos. Requerido por los métodos de acceso de Servicios de Integración Componentes para construir un query utilizando los nombres apropiados de la columna para una tabla remota.
xtype	ID de la clase.
xdbid	ID de la base de datos de la clase. Para clases de sistema, el valor es-1. De otra manera, el valor es la base de datos actual ID.
accessrule	Id del objeto de la regla de acceso en sysprocedures

sysindexes

La tabla sysindexes contiene una fila para cada índice clustered, una fila para cada índice no clustered, una fila para cada tabla que no tiene ningún índice clustered, y una fila para cada tabla que contiene columnas de imagen o texto.

Descripción de columnas:

Columna	Significado
name	Nombre de la tabla o Indice.
id	ID de la tabla o ID de la tabla a la cual pertenece el indice.
indid	0 si es una tabla. 1 si es un indice clustered en una tabla allpages-locked; >1 si es un indice nonclustered o un indice clustered en una tabla con data-only-locked; 255 si el texto o imagen o estructura Java off-row (LOB structure).
doampg	Número de página para el mapeo de asignación de un objeto de una tabla.
ioampg	El Número de página para el mapeo de asignación de un índice o (LOB estructure).
oampgtrips	Es el Número de veces de un ciclo de páginas OAM dentro del caché sin ser reutilizado, antes de ser flushed.
status2	Información del estado del sistema interno (Ver tabla 16)
ipgtrips	Es el Número de veces de un ciclo de páginas index en el caché, sin ser reutilizado, antes de ser flushed.
first	Si el indid es 0 ó 1, es el Número de la primer página de datos, si indid esta entre 2 y 250, es el Número de la primer página del indice leaf-level.
root	Si el indid es 0 y la tabla es una tabla unpartitioned allpages-locked, es el Número de la última página de la cadena de páginas, sin uso para otro tipo de páginas. Si el indid esta entre 1 y 250, es el Número de página del

	root del árbol del índice.
distribution	Sin uso.
usagecnt	Reservado.
segment	Número del segmento en el cual reside el objeto.
status	Información del estado del sistema interno (Ver tabla 15).
maxrowsperpage	Número máximo de filas por página.
minlen	Tamaño mínimo de la fila.
maxlen	Tamaño máximo de la fila.
maxirow	Tamaño máximo de una fila non-leaf del índice.
keycnt	Número de llaves para un índice clustered en una tabla allpages-locked; 1 para el resto de los índices.
keys1	La descripción de columnas llaves si la entrada es un índice.
keys2	La descripción de columnas llaves si la entrada es un índice.
soid	ID de ordenamiento con el cual fue creado el índice, 0 si no hay datos caracter en la llave.
csid	ID del conjunto de caracteres con el cual fue creado el índice, 0 si no hay datos caracter en la llave.
base_partition	Número de la partición, incrementada por el comando alter table...unpartition
fill_factor	Valor para el fillfactor de una tabla, definido con sp_chgattribute
res_page_gap	Valor para el reservepagegap en una tabla
exp_rowsize	Tamaño previsto de las filas de datos
keys3	La descripción de columnas llaves si la entrada es un índice.
identitygap	Hueco de identidad para una tabla.

Tabla 15. Lista de Representaciones de bit para la columna status.

Tabla 15. Bits de estado en la columna status de la tabla sysindexes

Decimal	Status
1	Aborta el comando o trigger actual, si hay tentativa de insertar llaves duplicadas.
2	Indice único.
4	Aborta el comando o trigger actual, si hay tentativa de insertar filas duplicadas; siempre es 0 para tablas con data-only-locked.
16	Indice clustered.
64	El indice permite filas duplicadas, si una tabla tiene allpages-locked; es 0 para tablas con data-only-locked.
128	Ordenamiento de objeto; no definido para las tablas sin índices clustered o para los objetos text.

512	La opción de ordenamiento de datos utilizada en la declaración create index.
2048	Índices en la llave primaria.
32768	Índices suspect; indice que fue creado durante otro ordenamiento.

Tabla 16. Lista de Representaciones de bit para la columna status2.

Tabla 16. Bits de estado en la columna status2 de la tabla sysindexes

Decimal	Status
1	El índice soporta foreign key constraint .
2	El índice soporta primary key/unique declarative constraint.
4	El índice incluye una columna IDENTITY.
8	Nombre del constraint no especificado.
16	I/Os grande (prefetch) no permitido para una tabla, índice, o cadena de texto.
32	Estrategia MRU caché no permitido para una tabla, índice, o cadena de texto.
64	inserts ascendentes conectados a una tabla
256	El índice es preordenado y no tiene que ser copiado hacia nuevos extents.
512	La tabla es una tabla data-only-locked con un índice clustered.
8192	El índice en una tabla data-only-locked es suspect.

syslogs

Esta tabla contiene el log de transacciones. No es útil a los usuarios.

Descripción de columnas:

Columna	Significado
xactid	ID de la transacción
Op	Número de la operación de actualización

sysobjects

Contiene un registro por cada tabla, vista, procedimiento almacenado, procedimiento almacenado extendido, log, regla, default, trigger, check constraint, referential constraint y (únicamente en tempdb) objetos temporales.

Descripción de columnas:

Columna	Significado
name	Nombre del objeto

id	ID de objeto
uid	Id de usuario dueño del objeto
type	Uno de lo siguientes tipos de objeto: • D = default • F = función SQLJ • L = log • P = Transact-SQL o procedimiento SQLJ • PR = objetos preparados (creados por Dinamyc SQL) • R = reglas • RI = referencial constraint • S = tablas de sistema • TR = trigger • U = tablas de usuario • V = vista • XP = Procedimiento almacenado extendido
userstat	Información de tipo aplicación-dependiente. (32768 decimal [0x8000 hex] Indica a Data Workbench® que un procedimiento es un reporte.
sysstat	Información de estado interna (256 decimal [0x100 hex] indica que la tabla es de solo lectura).
indexdel	Contador de un índice delete (incrementado si un índice es suprimido).
schemacnt	Cuenta los cambios en el esquema de un objeto (incrementado si se agrega una regla o un default).
sysstat2	Información de estado adicional interna (Ver tabla 17).
crdate	Fecha de creación del objeto.
expdate	Reservado.
deltrig	Id del procedimiento almacenado de un *trigger delete* si la entrada es una tabla.
instrig	Id del procedimiento almacenado de un *trigger insert* si la entrada es una tabla.
updtrig	Id del procedimiento almacenado de un *trigger update* si la entrada es una tabla.
seltrig	Reservado
ckfirst	Id del primer check constraint en una tabla
cache	Reservado.
audflags	Definiciones de auditoría del objeto
objspare	Spare
versionts	
loginame	Nombre del login que creo el objeto.

Tabla 17. Lista de Representaciones de bit para la columna sysstat2.

Tabla 17. Bits de estado en la columna sysstat2 de la tabla sysobjects

Decimal	Status
1	La tabla tiene un referential constraint.
2	La tabla tiene un foreign key constraint.
4	La tabla tiene más de un check constraint.
8	La tabla tiene un primary key constraint.
16	El procedimiento almacenado puede ser ejecutado solamente en modo chained transaction.
32	El procedimiento almacenado puede ejecutarse en cualquier modo de transacción.
64	La tabla tiene un campo IDENTITY.
512	La tabla no contiene columnas de longitud variable.
1024	Es una tabla remota.
2048	La tabla es una tabla proxy creada con la palabra clave existente.
8192	La tabla utiliza allpages locking scheme.
16384	La tabla utiliza datapages locking scheme.
32768	La tabla utiliza datarows locking scheme.
65536	La tabla fue creada en versión 11.9 o anterior a la del servidor.
131072	La tabla tiene un índice clustered.
242144	El objeto representa un procedimiento Embedded SQL.
33554432	El objeto representa un procedimiento almacenado SQLJ.
16777216	El objeto representa una regla de acceso.
67108864	El objeto representa una regla de acceso OR.

sysprotects

Cuando se otorgan privilegios se agregan un registro en la tabla ***sysprotects***.

Descripción de columnas:

Columna	Significado
id	Es el ID del objeto al cual se le aplica el permiso
uid	Es el ID del usuario, grupo o role, a los cuales se les otorgó permisos

systhresholds

La tabla systhresholds contiene un registro por cada umbral definido para una base de datos.

Descripción de columnas:

Columna	Significado
segment	El Número de segmento de espacio libre el cual está siendo monitoreado.
free_space	Tamaño del umbral, en páginas de 2k(status de 4k)
status	Bit 1 igual a 1 para el umbral de última oportunidad del logsegment.
proc_name	Nombre del procedimiento que se ejecutó cuando el Número de páginas no utilizadas sobre el segmento están por debajo del espacio libre.
suid	Id del usuario que agregó el umbral o lo modificó recientemente.
currauth	Máscara de bit que indica qué roles estaban activos para el suid al momento de agregar el umbral o durante la última modificación. Cuando se cruza el umbral, proc_name se ejecuta con este conjunto de roles, menos aquellos que han sido desactivados cuando se agregó el umbral o se modificó.

sysusers

Al agregar un usuario dentro de una base de datos se inserta un registro en la tabla sysusers.

Descripción de columnas de la tabla sysusers:

Columna	Significado
suid	Es el ID de usuario del servidor
uid	ID de usuario, único dentro de una base de datos, se utiliza para otorgar o revocar permisos. El ID de usuario 1 es para “dbo”
gid	ID de grupo al que pertenece el usuario. Si uid = gid, esta entrada define a un grupo. El grupo “public” tiene el suid = -2; Todos los otros grupos tienen el suid = -gid.
name	Nombre de usuario o grupo, único en la base de datos.
environ	Reservado.

Tablas del Sistema en sybsecurity:

sysauditoptions

Descripción: La tabla sysauditoptions contiene un registro por cada opción de auditoría server-wide e indica la opción definida actualmente. Otros tipos de opciones de auditoría y los settings son almacenados en otras tablas. Por ejemplo, el setting de una opción para una base de datos específica se almacena en sysdatabases y el setting de opción para un objeto específico se almacena en sysobjects. El valor predeterminado para cada opción es 0 u "off". Solo un oficial de seguridad de sistema puede entrar a esta tabla.

Descripción de columnas de la tabla sysauditoptions:

Columna	Significado
num	Número de la opción server-wide.
val	Valor actual; uno de los siguientes: 0 = off 1 = pass 2 = fail 3 = on
minval	Valor mínimo válido para esta opción.
maxval	Valor máximo válido para esta opción.
name	Nombre de la opción.
sval	Cadena equivalente del valor actual: por ejemplo, "on", "off", "nonfatal".
comment	Descripción de la opción.

sysaudits_01 – sysaudits_08

Descripción: Estas tablas de sistema contienen el registro de auditoría. Solamente una tabla a la vez puede estar activa. La tabla activa es determinada por el valor del parámetro de configuración actual de la tabla de auditoría.

Descripción de columnas de la tabla sysaudits_01- sysaudits_08:

Columna	Significado
event	Tipo de evento que está siendo auditado
eventmod	Información adicional acerca del evento auditado. Los valores posibles son: 0 = ninguna modificación para este evento 1 = el evento pasó la comprobación 2 = el evento falló la comprobación

spid	ID, del proceso que ocasionó el evento.
eventtime	Fecha y hora del evento.
sequence	Secuencia de un registro para un evento con múltiples registros.
suid	ID, del login que causó el evento.
dbid	ID de la base de datos en donde ocurrió el evento o en donde reside el evento auditado.
objid	ID del objeto auditado.
xactid	ID de la transacción que causó el evento.
loginname	Es el nombre del login correspondiente al suid.
dbname	Es el nombre de la base de datos correspondiente al dbid.
objname	Nombre del objeto correspondiente al objid.
objowner	Nombre del dueño del objid.
extrainfo	Información adicional del evento auditado. Este campo contiene una serie de puntos separados por puntos y comas. (Ver anexo2)

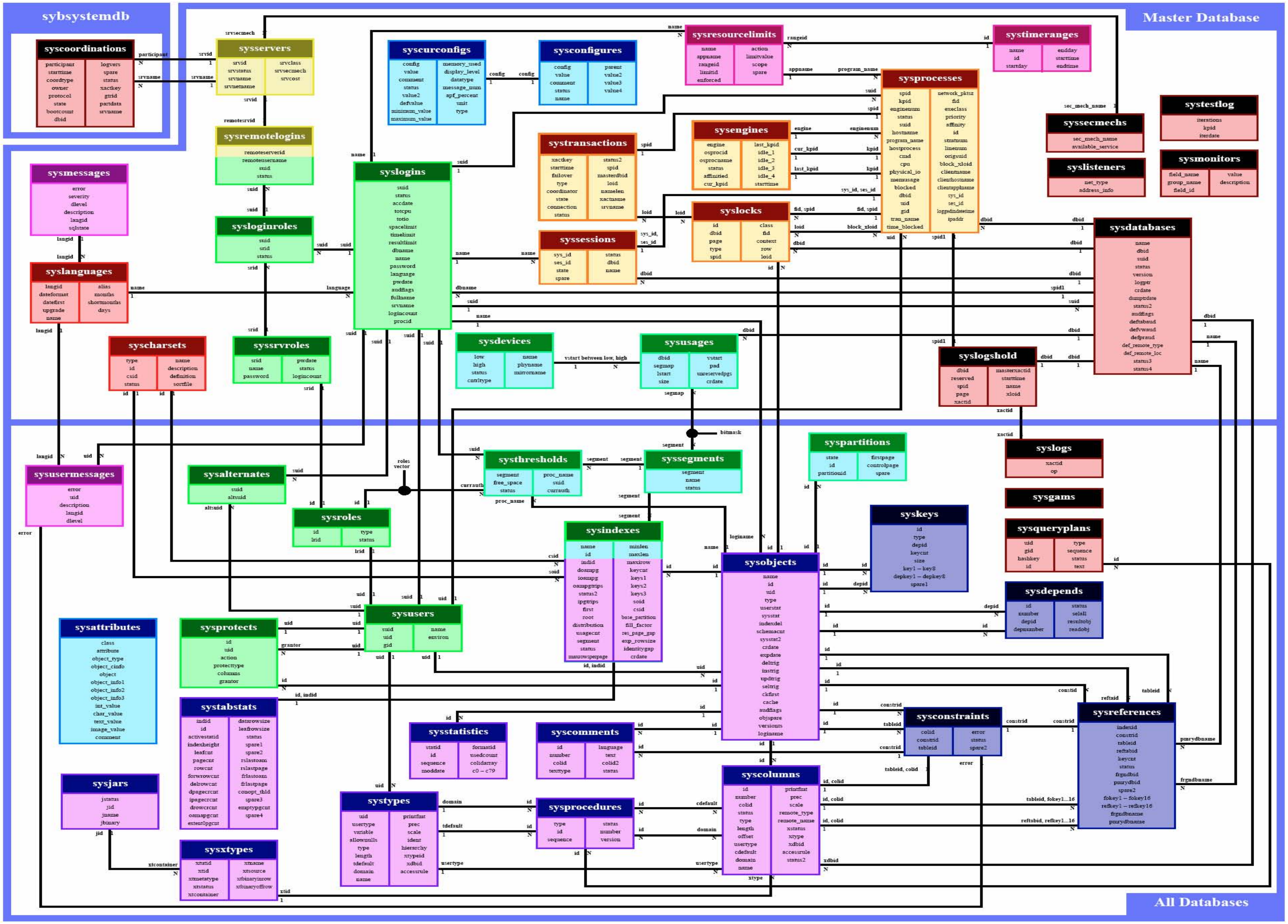
La columna extrainfo contiene una secuencia de puntos separados por puntos y comas como se muestra en la siguiente tabla.

Tabla 18: Puntos de la columna extrainfo

Item	Contenido
Roles	Listado de roles que se encuentran activos.
Subcommand	Es el nombre de una opción de comando o subcomando que fue utilizada por ese evento. Por ejemplo, para el comando "**alter table**", las opciones "add column" o "drop constraint" pueden ser usadas. Múltiples opciones o comandos son separados por comas.
Previous value	El valor antes de la actualización si el evento dio lugar a la actualización de un valor.
Current value	El nuevo valor si el acontecimiento dio lugar a la actualización de un valor.
Other information	Información adicional de seguridad relevante que se registra para el evento
Proxy information	El nombre original del login, si ocurrió el evento mientras que un set proxy estaba en efecto.
Principal information	El nombre principal del mecanismo subyacente de la seguridad, si la conexión segura del defecto del usuario, y el usuario registrado en el servidor adaptante vía unificado de este campo es NULL, si la conexión segura del defecto no se está utilizando.

A continuación se presenta el diagrama de las tablas de sistema del Adaptive Server Enterprise 12.5 de Sybase. En este anexo sólo se contemplaron aquellas tablas que eran necesarias para el desarrollo de nuestra herramienta auditora.

Sybase Adaptive Server Enterprise 12.5. Tablas de Sistema



Bibliografía

Libros

- Piattini, M. y del Peso, E. Auditoría informática. Un enfoque práctico. Ra-ma, 1998.
- By Marlene Theriault & William Heney. Oracle Security. O'Reilly 1st Edition October 1998.
- By Simson Garfinkel & Gene Spafford. Practical Unix Security. O'Reilly & Associates, 1998.
- Maiwald, Eric, Fundamentos de Seguridad de Redes, Segunda Edición, McGraw-Hill, México 2005.
- Pegler Swift, Stacia Sambar (Traducción al español Armando Vega Alvarado). Open Client DB-Library/C, Sybase, 1994.
- De Miguel, Adoración y Piattini, Mario Gerardo, Concepción y Diseño de Base de Datos, Segunda Edición, Addison-Wesley Iberoamericana, México 1999.
- Tsai, Alice, Sistema de Base de Datos: Administración y Uso, Prentice-Hall Hispanoamericana, S.A. México 1994.
- Ullman, Jeffrey D. y Widon, Jennifer, Introducción a los Sistemas de Base de Datos, Prentice-Hall Hispanoamericana, S.A., México 1999.
- Date, C. J. Introducción a los Sistemas de Bases de Datos, Séptima Edición, Pearson Educación de México, S.A. de C.V., México 2001.
- Hansen, Gary y Hansen, James, Diseño y Administración de Base de Datos, Tercera Edición, Prentice-Hall Hispanoamericana, México 2000.

Direcciones Electrónicas:

❖ Información acerca de Auditoría en Base de datos:

www.virtual.unal.edu.co/cursos/sedes/manizales/4060035/lecciones/Capitulo4.html

❖ Información acerca de Seguridad en Sistemas de Información:

http://ulises.umh.es/cc/personal/marco/ssi/default.html

❖ Información acerca de Base de datos:

http://www.itlp.edu.mx/publica/tutoriales/basedat1/
http://www.itlp.edu.mx/publica/tutoriales/basedat2/
http://macine.epublish.cl/tesis/index.html
http://academicos.cualtos.udg.mx/Informatica/Ceneval2003/Bases%20de%20Datos1.htm
http://informatica.uv.es/iiguia/2000/BD2/BD2Tema6.pdf

Información acerca de Apuntes de Ficheros y Bases de datos:
http://www3.uji.es/~mmarques/f47/apun/apun.html

Información acerca de Auditoría:
http://www.monografias.com/trabajos6/audi/audi.shtml

Información acerca de Auditoría y Seguridad en Base de datos.
http://bbdd.escet.urjc.es/.../Auditoria%20y%20Seguridad%20en%20BBDD.Enfoque%20practico%20con%20Oracle%208i.pdf

http://alarcos.inf-cr.uclm.es/doc/adbd/

http://alarcos.inf-cr.uclm.es/doc/calidadSI/Tema4.pdf

Información acerca de productos y documentación de Sybase
http://www.sybase.com

Información acerca de productos y documentación de Sybase en español
http://www.mtbase.com/inicio

Página Oficial de SQShell, documentación, fuentes y binarios
http://www.sqsh.org

Página Oficial del compilador GCC, documentación, fuentes y binarios
http://www.gnu.org/software/gcc/gcc.html