UNICIENCIA 22
pp. 135-140
2008

AUDITANDO EN LAS BASES DE DATOS

Johnny Villalobos Murillo
Facultad de Ciencias Exactas y Naturales, Escuela de Informática,
Universidad Nacional de Costa Rica
Heredia. Costa Rica
E-Mail: jvillalo@una.ac.cr

RESUMEN

La importancia de establecer controles que permitan minimizar el riesgo inherente que tienen los datos contenidos en una base de datos hace necesario implementar procedimientos de auditoría. Existen esencialmente dos tipos de auditorías aplicables a las bases de datos: la auditoría de objetos y la auditoría de transacciones. Algunos sistemas gestores de bases de datos proporcionan mecanismos para el primer tipo, mientras que para realizar las auditorías de transacciones, es necesario crear nuestros propios procedimientos o acudir a soluciones de terceros.

Palabras claves: Base de datos, auditoría, objetos, transacciones.

ABSTRACT

The importance of establishing controls that allow you to diminish the inherent risk that has the data contained in a database make necessary implement audit procedures. Two types of audits applicable to the database exist essentially, the audit of object and the audit of transactions, some system database manager provide mechanisms for the first type, whereas to make the audits of transaction, is necessary to create our own procedures or to go to solution of third.

Keywords: Database, audit, objects, transactions.

1. AUDITORÍA DE BASES DE DATOS

La auditoría de bases de datos consiste en un proceso de monitoreo continuo y riguroso de los controles que la administración ha establecido dentro de los sistemas de bases de datos y todos sus componentes para obtener una seguridad razonable de la utilización adecuada de los datos que son almacenados por los usuarios mediante los sistemas de información. El monitoreo y pruebas a los controles determinan la pertinencia y suficiencia de éstos, permitiendo entonces ajustar, eliminar o implementar nuevos controles para asegurar su adecuada utilización.

El propósito de los controles de las bases de datos es minimizar el riesgo inherente que tiene este valioso recurso. Los datos contenidos en las bases de datos pueden considerarse uno de los activos más importantes que tiene la organización, ellos finalmente producirán la información que necesita la empresa para su funcionamiento día a día o para su planificación estratégica.

Por esta razón, la gerencia debe establecer políticas de seguridad, procedimientos de utilización y controles pertinentes, las políticas deberán ser divulgadas en la organización.

2. DÓNDE UBICAR LOS CONTROLES

Cuando se plantea la forma más adecuada de hacer la auditoría, existe además la interrogante de ¿cuál es el lugar más adecuado para implantar los controles? Existen tres posibilidades, la primera es poner los controles en el nivel de las aplicaciones, es decir, directamente en los programas o interfaces

Recibido 5/7/2006 / Aprobado 30/11/2006

del usuario. La segunda posibilidad es ponerlos en medio de las aplicaciones y la base de datos, lo que llamamos controles en la red y la tercera posibilidad es ubicar el control en la base de datos, lo que se llama control en la fuente.

2.1 Los controles en las aplicaciones

La primera alternativa implica que a cada sistema, módulo o programa, se le debe adicionar el control; esto implica modificaciones de código, compilaciones de programas y otros procesos necesarios para asegurar que todas las aplicaciones contengan el control. Esta alternativa resulta ser muy vulnerable, ya que es posible acceder a la base de datos por otros medios. Tiene la desventaja de la actualización del control, con esto se quiere decir que si el control sufre alguna modificación, entonces será necesario actualizar todas las aplicaciones, y en este caso es posible que alguna quede sin actualizar.

2.2 Los controles en la red

La segunda alternativa, que consiste en poner el control entre el usuario y la base de datos, significa realmente tenerlo en la red. Esta alternativa tiene la ventaja de un único control, ubicado en una posición estratégica fácil de modificar y de administrar. El control es como un programa espía que captura todos los mensajes entre los usuarios y la base de datos, se puede capturar todos los estatutos de tipo SQL, entre ellos y las actividades de conexión, las horas y días que ellos interactúan. Existe un único problema en esta alternativa de ubicación, que consiste en conocer cómo eran los valores de los datos antes y después de que el usuario los actualizara. Imagínase que una tabla de la base de datos tiene el monto de un recibo pendiente de pagar, si un usuario no autorizado puede de alguna forma cambiar este valor, sería difícil conocer el monto anterior del recibo, ya que el programa espía lo que detecta es que el usuario envió una sentencia de actualización como la siguiente:

```
Update recibo set saldo = 10,000
where recibo = 34,567
```

En este ejemplo perdemos la información del monto anterior. El programa espía sólo graba la sentencia pero no el valor, esto puede traer consecuencias no deseadas en el departamento de cuentas por cobrar.

2.3 Control en la fuente

La tercera alternativa y la que recomiendo es la de poner el control directamente en la base de datos. Al igual que la segunda, hablamos de un único control, fácil de implementar y de administrar. Con esta alternativa y en este caso sí es posible conocer los valores de los datos antes y después de la actualización, ya que podemos utilizar mecanismos de la base de datos que permiten llegar a ese nivel.

3. TIPOS DE AUDITORÍAS DE BASES DE DATOS

Consideramos dos grandes tipos de auditorías de la base de datos, esta división está relacionada directamente con las actividades que los usuarios realizan, suponga que un usuario desea cambiar la dirección de envío de pedidos a un cliente. Para lograr esto se requiere que el usuario realice una serie de pasos, por ejemplo:

P 1. Conectarse a la base de datos.
P 2. Ejecutar el cambio de dirección.
P 3. Desconectarse.

Los pasos 1 y 3 se lograrían si cuenta con los privilegios o derechos necesarios para conectarse o desconectarse; para el paso 2 el usuario debe tener privilegios de acceso a la tabla de clientes, y la posibilidad de hacer una modificación. Basándonos en esta secuencia de pasos establecemos el primer tipo de auditoría, y la llamamos auditoría de actividades de los usuarios.

El segundo tipo de auditoría está inmersa en el paso 2. ¿Qué ocurrió realmente cuando cambió el dato, qué valores quedaron?, ¿qué valores había?, en otras palabras, ¿qué cambios produjo la transacción? Los controles que establezcamos para conocer lo que realmente ocurrió los llamaremos auditoría de transacciones.

3.1 Auditoría de actividades

El primer tipo de auditoría lo llamamos au-

ditoría de actividades, que consiste en controlar las actividades que realizan los usuarios en los objetos de la base de datos y entenderemos como objetos todas las tablas, vistas, restricciones de integridad que los usuarios crean en la base de datos.

Este proceso de monitoreo de las actividades de los usuarios permite encontrar posibles accesos a objetos no autorizados, conexiones en horas o días fuera de horarios normales. Toda esta actividad se va almacenando en una tabla o en un archivo que llamaremos el registro de auditoría.

El registro de auditoría (RA) tiene un crecimiento muy alto, por lo que es necesario administrarlo adecuadamente, muchas veces este registro llega a ser igual o mayor en tamaño que la base de datos. Imagínese el RA de una organización de más de 2000 empleados, que además permite el acceso de clientes por Internet, o el registro de auditoría de un banco, que puede recibir más de 20 transacciones por minuto.

3.2 Auditoría de transacciones

La auditoría de transacciones consiste en implementar una serie de controles que permiten llevar una bitácora de todas las transacciones que los usuarios realizan, pero a un nivel tal que podamos establecer una historia de cómo se produjeron los cambios. Al igual que en el tipo anterior, es necesario crear un registro de auditoría al que llamaremos registro de auditoría de transacciones (RAT). El crecimiento del RAT es superior al del RA, ya que es posible que un usuario que se conecte una sola vez, pueda hacer 30 transacciones. En este caso el RA almacena las actividades de conexión y desconexión, básicamente mientras que el RAT almacena 30 registros correspondientes a la información antes y después de cada transacción.

4. LA ADMINISTRACIÓN DEL REGISTRO DE AUDITORÍA

El gran volumen de información que se almacena en el RA y el RAT hace pensar en algunas organizaciones donde este control puede ser innecesario, que el costo de almacenamiento es muy alto, y el rendimiento de la base de datos se puede ver afectado. Es posible que tengan razón, y eviten con estas justificaciones establecer los controles, dejando la base de datos completamente vulnerable.

Es necesario pensar entonces en una serie de consideraciones respecto a la administración del RA, para evitar que afecte y se opte por eliminarlo, dos de las principales recomendaciones son:

- Audite sólo lo necesario o sólo las excepciones.
- Haga respaldo y limpieza del RA y el RAT.

Como buena práctica, es necesario analizar qué tipo de información deseamos auditar, qué es relevante y qué no, qué nos sirve en una investigación y qué información no tiene sentido auditar, de este modo no produciremos tanta información y se disminuye la posibilidad de afectar el rendimiento del servidor.

4.1 Actividades sospechosas

Una guía adecuada cuando se requiere auditar actividades sospechosas es, primero auditar en forma general y posteriormente auditar lo específico. Esto se refiere que para empezar con el proceso de la auditoría, comenzamos a auditar todo tipo de acciones que realicen los usuarios, luego auditamos algunas acciones, aquellas que realmente sean pertinentes y por último auditamos al usuario o grupo de usuarios que nos preocupan.

5. HABILITANDO LA AUDITORÍA

Para habilitar la auditoría es necesario ejecutar algunas instrucciones que permitan crear una serie de procesos “espías” y el RA.

El sistema gestor de la base de datos relacionales Oracle provee un conjunto de instrucciones SQL, a lo que llamamos script, que permite habilitar la auditoría de actividades. El script debe ejecutarse por un usuario que tenga los privilegios necesarios.

Un detalle interesante es que se puede deshabilitar el proceso de auditoría cuando el usuario, ya sea el DBA o el encargado de la auditoría, lo desee.

En el caso de Oracle, la habilitación requiere la ejecución de una secuencia de comandos SQL:

Habilitar: CATAUDIT.SQL
Deshabilitar: CATNOAUDIT.SQL

5.1 Composición del RA de Oracle

El RA llamado Audit.Trail, de Oracle, puede contener diversos tipos de información. La información básica que encontramos es:

- Usuario.
- Sesión.
- Terminal.
- Nombre del objeto accesado.
- Operación realizada o intentada.
- Fecha y hora.

El RA permitirá saber:

- ¿Quién se conectó?
- ¿Cuándo se conectó?
- ¿Qué hizo en la base de datos?
- ¿Tuvo éxito o no?
- ¿Qué privilegios tenía?
- ¿Quién se los otorgó?

Con este tipo de auditoría, no podemos saber qué datos estaban involucrados, es decir, si la acción del usuario genera un cambio en el saldo de la cuenta de un cliente, no podemos saber qué saldo tenía antes y que saldo queda después de la transacción, ya que esta información no se almacena en el RA.

5.1.1 Los tres niveles de auditoría de actividades

Estatutos SQL, privilegios y objetos, estos son los tres niveles que podemos auditar en la base de datos. Los estatutos son todas aquellas instrucciones de tipo SQL que podamos realizar, las más frecuentes son las instrucciones que recibe el servidor de la base de datos para que permita una conexión (inicio de sesión) o una desconexión.

Los privilegios, por su parte, son los permisos que se le otorgan a los usuarios para que realicen transacciones o consulten información. Existen más de 80 privilegios diferentes en el servidor de la base de datos Oracle.

Finalmente, las manipulaciones directas sobre los objetos, como borrados, creaciones, modificaciones de objetos de usuario o de sistema, como se mencionó anteriormente.

Para auditar en Oracle es necesario contar con el privilegio AUDIT SYSTEM, para que se pueda habilitar o deshabilitar la auditoría. Este privilegio inicialmente lo tiene el administrador de la base de datos, y es él quien lo otorga al usuario designado. Una vez realizada la asignación, puede entonces iniciar.

Supongamos que deseamos auditar las veces en que Ana se conecta o se desconecta. Para esto entonces desde una interfaz de usuario como el SQL PLUS, el auditor escribe:

```
AUDIT SESSION BY ANA
```

Si se desea saber todas las consultas que hace Ana en las tablas a las que tiene derecho, escribimos:

```
AUDIT SELECT ANY TABLE
```

Si deseamos saber las transacciones que realiza Ana, en cada tabla, y en que momento, escribimos:

```
AUDIT INSERT TABLE, DELETE TABLE,
EXECUTE PROCEDURE BY ACCESS
WHENEVER NOT SUCCESSFUL
```

En este caso, se audita cualquier transacción en las tablas sin importar quien sea el que lo haga.

La cláusula WHENEVER NOT SUCCESSFUL se refiere a las transacciones que fallan, es decir, aquellos intentos de actualizar una tabla por parte del usuario y este no lo logra, posiblemente por los privilegios que se le han otorgado. En este caso estaríamos monitoreando usuarios que tratan de hacer transacciones en una tabla a las que no tienen derecho.

Posteriormente y como ya se mencionó, podemos ser más específicos en lo que queremos auditar. Supongamos que sólo queremos monitorear la tabla cuentas, entonces escribimos:

AUDIT SELECT, INSERT, DELETE UPDATE ON CUENTAS BY ACCESS WHENEVER SUCCESSFUL

Recordemos que es mejor auditar las excepciones, ya que de lo contrario, el RA se haría enorme.

5.1.2 Consultando el registro de auditoría

El RA puede ser consultado por el usuario que tenga el privilegio adecuado. Para realizar las consultas se cuenta con varios tipos de vistas en el diccionario de datos, como por ejemplo:

- SYS.DBA_PRIV_STMT_OPTS
- SYS.DBA_PRIV_AUDIT_OPTS

El nombre de la vista empieza con SYS, ya que éste es el dueño, el prefijo DBA se utiliza para acceder a las tablas de administradores de las bases de datos, posteriormente el nombre de la base de datos denota el tipo de información, por ejemplo, PRIV_AUDIT_OPTS se refiere a los estatutos SQL que utilizaron los usuarios auditados.

Para realizar la consulta, se implementa una instrucción SQL como la siguiente:

SELECT * FROM
SYS.DBA_PRIV_AUDIT_OPTS

Si desea conocer la composición de alguna de las tablas o vistas, es decir, qué información almacena, puede utilizar el comando DESCRIBE, de esta forma se detallan todas las columnas que tiene la tabla.

DESCRIBE SYS.DBA_PRIV_AUDIT_OPTS

6. AUDITANDO LAS TRANSACCIONES

Los sistemas gestores de bases de datos relacionales tienen un objeto especial llamado TRIGGER (desencadenador).

El TRIGGER consiste en un conjunto de sentencias SQL que se le adhieren a una tabla, con el propósito de que en caso que la tabla reciba una transacción, el TRIGGER se ejecutará y aplicará las sentencias que tiene almacenadas.

La idea de implementar el TRIGGER como auditoría de transacciones es excelente, siempre y cuando se tome en cuenta que él interfiere en el rendimiento de la transacción, aun así, si es necesario implementarlo tenga presente hacerlo en forma adecuada y moderada.

Supongamos que deseamos registrar todos los cambios que se realizan en la tabla cuentas, quién los realiza y cuándo los realiza. Para hacer estos primeros cambios necesitamos crear nuestro propio registro de auditoría.

La sentencia SQL para crear la tabla sería la siguiente:

```
Create table RAT
    ( usuario varchar (10),
     Fecha date,
     Cliente varchar (6),
     Saldo_anterior number (10,2)
     Saldo_actual number (10,2) )
```

En esta tabla, la columna usuario sirve para registrar el usuario que realiza la transacción; fecha almacena el día y la hora en que se efectúa la transacción; cliente se refiere al cliente al cual se le actualizó el saldo y finalmente saldo anterior y saldo actual registran los valores que tenía el saldo del cliente antes y después de llevar a cabo la transacción.

El TRIGGER, que se creará, sólo reaccionará en el caso de que se haga una actualización, pero es posible que reaccione a inserciones y borrados, el código necesario es el siguiente:

```
Create trigger auditor
after update on cuentas
for each row
begin
insert into RAT
values ( user, date, :new.cliente,
 :old.saldo, :new.saldo ) end
```

De esta forma se almacenan en nuestro

registro de auditoría los valores de usuario (user), fecha (date), cliente (:new.cliente), saldo_anterior (:old.saldo), saldo_actual (:new.saldo).

Las variables, :new.saldo y :old.saldo son proporcionadas por Oracle.

Dependiendo de la cantidad de transacciones que se reciban, nuestro registro de auditoría crecerá cada vez más.

Si esto mismo se les aplica a todas las tablas que conforman un sistema de información en una corporación, el tamaño del archivo de auditoría sería muy grande, el rendimiento de las transacciones se vería afectado. Considerando lo anterior debemos aplicar los siguientes consejos:

1. Utilice TRIGGER en tablas críticas.
2. Audite sólo transiciones relevantes.
3. Construya varios registros de auditoría, no use solo uno.
4. Ubique los registros de auditoría en un disco aparte de los discos de datos de los sistemas.
5. Revise el tamaño del registro de auditoría.
6. Elimine información que ya no sea necesaria.
7. Respalde en forma periódica.