2011

Auditoria Oracle 11g

Francisco Bellido Rodríguez
*

06/03/2011

1. Objetivos

Este proyecto consiste en realizar un estudio y análisis de una auditoria en Oracle 11g y sus tipos, con el objetivo de poder mostrar a los demás compañeros las posibilidades que presenta para el administrador el buen entendimiento y aplicación de medidas para mejorar el rendimiento de nuestra base de datos. Se pretende estudiar las características de las herramientas disponibles y presentar las distintas opciones que nos ofrece. El estudio se realizará de forma teórico-práctica así iremos experimentando y sacaré conclusiones sobre el comportamiento y rendimiento de las diferentes modalidades de auditoría que Oracle 11g. Se busca obtener resultados y comparativas que se utilizarán para la realización de éste manual de buenas prácticas acerca de la auditoría.

2. Introducción

En la actualidad, las empresas almacenan cada vez más datos sensibles en sus servidores. Información de clientes y personas, información estratégica, proyectos, investigaciones, etc que son de vital importancia para la organización y el buen funcionamiento de ésta. Supone un reto para las empresas la salvaguarda de éstos datos sensibles ya que la custodia de ésta información es una necesidad y una obligación, incluso están protegidas y castigadas por ley (**Ley Orgánica 15/1999, de 13 de diciembre**). Tan importante es el acceso restringido a ésta información sensible que se hace imprescindible ofrecer garantías de confidencialidad para alguien no autorizado, es fundamental tener la capacidad de detectar si se ha realizado un acceso autorizado o no y si éste ha sido o no adecuado, quién lo ha hecho, cuando y qué información se ha podido comprometer. El objetivo final del administrador de la base de datos es obtener un nivel de seguridad que nos permita asegurar que los datos están disponibles para aquellos usuarios que deben estarlo y no para otros y a su vez que el acceso a éstos de hace de forma que se optimicen al máximo los recursos de la base de datos.

3. Empecemos por el principio

3.1 ¿Qué es una auditoria?

Inspección, interna o externa, de los distintos procesos académicos o de gestión. Es un término que se incorpora del mundo empresarial y judicial. Se refiere al proceso de **evaluación** de una institución o programa. También denominada auditoría de calidad. Un examen o estudio que evalúa e informa sobre la medida en que una condición, proceso o desempeño se ajusta a estándares o criterios predeterminados.

3.2 ¿Qué es un auditor?

Persona especializada en el análisis de la situación y evolución de las Aseguradoras, mediante cuyos conocimientos estadísticos, económicos, jurídicos y financieros, evalúa el estado de las entidades en un momento determinado, cuya función primordial es el asesoramiento y certificación de los documentos (Balances, Declaraciones Juradas, etc.), que las aseguradoras tienen obligadamente que elaborar de acuerdo a la normativa vigente.

3.3 ¿Qué entendemos por Auditar?

Auditar consiste principalmente en estudiar los mecanismos de control que están implantados en una empresa u organización, determinando si los mismos son adecuados y cumplen unos determinados objetivos o estrategias, estableciendo los cambios que se deberían realizar para la consecución de los mismos. Los mecanismos de control pueden ser directivos, preventivos, de detección, correctivos o de recuperación ante una contingencia.

4. Auditoria Oracle

En el caso de las BBDD Oracle, la auditoría es un conjunto de características que permite al DBAs y a los usuarios hacer un seguimiento del uso de la BBDD. El DBAs puede definir una actividad de auditoría predeterminada. La información de las auditorías se almacena en el diccionario de datos, en la tabla SYS.AUD$ o en la pista de auditoría del sistema operativo (si lo permite). Existen varias vistas que se basan en esta tabla (SYS.AUD$) para mostrar distintos resultados, según la información que se quiera obtener.
Lo anterior viene definido en el parámetro AUDIT_TRAIL, esto quiere decir que para que los datos de auditoría se vayan almacenando en la BBDD este parámetro Estudio e Implantación de Audit Vault debe estar activado como ya se explicará en el siguiente apartado. Cabe destacar que la BBDD Oracle tiene varias capas de seguridad y proporciona la capacidad de auditar cada nivel.
En una auditoría es importante auditar tres tipos de acciones: intentos de inicio de sesión, accesos a objetos y acciones de la base de datos. Cuando se realizan auditorías, la funcionalidad de la BBDD es dejar constancia de los comandos correctos e incorrectos que se realizan sobre ésta. Esto puede modificarse cuando se configura cada tipo de auditoría.

Los pasos principales para llevar a cabo una auditoría tradicional son los siguientes:

Comprobar activación de una instancia de Oracle para auditoría

La activación de la auditoría en Oracle Database viene definida por el valor del parámetro: AUDIT_TRAIL. Para comprobar si la auditoría de la base de datos está activada ejecutaremos el siguiente comando SQL:

```
select name, value
from v$parameter
where name like 'audit_trail';
```

```
SQL> select name, value
  2  from v$parameter
  3  where name like 'audit_trail';

NAME
--------------------------------------------------------------------------------

VALUE
--------------------------------------------------------------------------------

audit_trail
DB
```

Como vemos en mi caso la respuesta que nos da como resultado BD, podemos ir al apartado donde se especifican los valores de respuesta y ver que significa.

También podemos hacerlo tocando directamente el archivo init.ora cambiando el parámetro *Audit_trail*

```
db_name='ORCL'
memory_target=1G
processes = 150
audit_file_dest='<ORACLE_BASE>/admin/orcl/adump'
audit_trail ='db'
db_block_size=8192
db_domain=''
db_recovery_file_dest='<ORACLE_BASE>/flash_recovery_area'
db_recovery_file_dest_size=2G
diagnostic_dest='<ORACLE_BASE>'
dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB)'
open_cursors=300
remote_login_passwordfile='EXCLUSIVE'
undo_tablespace='UNDOTBS1'
# You may want to ensure that control files are created on separate physical
# devices
control_files = (ora_control1, ora_control2)
compatible ='11.2.0'
```

Captura fichero C:\app\usuario\product\11.2.0\dbhome_1\dbs

```
SQL> show parameter audit

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
audit_file_dest                      string      C:\APP\USUARIO\ADMIN\ORCL\ADUM
                                                 P
audit_sys_operations                 boolean     FALSE
audit_trail                          string      DB
SQL>
```

Captura comando show parameter audit

Y después reiniciamos la base de datos.
Posibles valores del parámetro AUDIT_TRAIL:

NONE	**Desactiva la auditoría de la base de datos. Es igual a FALSE.**
OS	activa la auditoría de la base de datos. Los sucesos auditados se escribirán en la pista de auditoría del sistema operativo, no se auditará en Oracle sino en el sistema operativo anfitrión. (No funciona en todos los sistemas operativos).
DB	activa la auditoría y los datos se almacenarán en la taba SYS.AUD$ de Oracle. Es equivalente a TRUE.
DB_EXTENDED	activa la auditoría y los datos se almacenarán en la taba SYS.AUD$ de Oracle. Además se escribirán los valores correspondientes en las columnas SQLBIND y SQLTEXT de la tabla SYS.AUD$.
XML	activa la auditoría de la base de datos, los sucesos será escritos en ficheros del sistema operativo.

XML, EXTENDED	activa la auditoría de la base de datos, los sucesos será escritos en el formato del sistema operativo, además se incluirán los valores deSqlText y SqlBind. [Bob Bryla, kevin Loney (2008)]

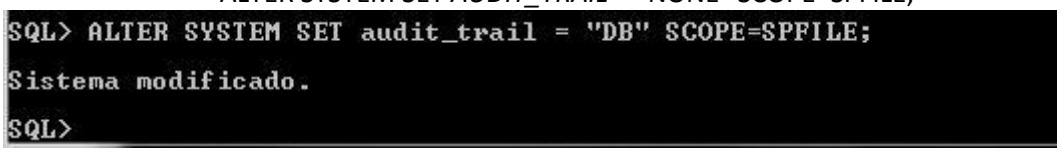
Activar la Auditoría de la B.D.

Para generar registros en las tablas de auditoría no basta con utilizar el comando AUDIT, sino que también es necesario activar la escritura en las tablas de auditoría activando el parámetro de inicialización AUDIT_TRAIL.
Por un lado se puede activar o desactivar modificando dicho parámetro desde el INIT.ora y por otro lado por ejemplo se puede hacer a través de sentencias SQL.

ALTER SYSTEM SET audit_trail = "DB" SCOPE=SPFILE;

Para **desactivar** la auditoría ejecutaremos el siguiente comando:

ALTER SYSTEM SET *AUDIT_TRAIL* = "NONE" SCOPE=SPFILE;



Nota:
En Oracle 9i la auditoría viene desactivada por defecto, el valor del parámetro " AUDIT_TRAIL " está a "NONE". Al igual que ocurre con Oracle 10g.
En Oracle 11g la auditoría viene activada por defecto, el valor del parámetro" AUDIT_TRAIL "está a "DB".

Recolección de datos

La recolección de datos se basa en un conjunto de vistas que actúan sobre la tabla donde se guardan los registros de auditoría. Ésta tabla es AUD$ para la auditoría normales y FGA_LOG$ para la auditoría de grano fino.

Cada vista ofrece distintos tipos de información de forma más clara para el auditor o administrador de la base de datos.

Análisis de la recolección de auditoría genérica

Debido a que la auditoría genérica guarda los registros de auditoría en una tabla llamada AUD$, el motor de base de datos *Oracle* ofrece al auditor un conjunto de vistas que permiten el análisis de los datos de dicha tabla desde diferentes enfoques.

A continuación se explicarán dichas vistas:

DBA_OBJ_AUDIT_OPTS	describe las políticas de auditoría genérica activadas sobre objetos, incluyendo las opciones de auditoría de todos ellos (operaciones que se auditan sobre los objetos) e información relevante: propietario del objeto, nombre del objeto y tipo de objeto.
USER_OBJ_AUDIT_OPTS	esta vista no es única ni exclusiva del administrador. Cada usuario tienen en su esquema esta vista, llamada de la misma manera, y que muestra las políticas de auditoría genérica creadas sobre objetos de su esquema. Es similar a DBA_OBJ_AUDIT_OPTS sólo que no lleva la columna del propietario del objeto, pues es implícito que el propietario sea el mismo que la vista.
DBA_PRIV_AUDIT_OPTS	describe las políticas de auditoría sobre privilegios del sistema que están activas en un momento dado.
DBA_STMT_AUDIT_OPTS	describe las políticas de auditoría sobre privilegios del sistema que están activas en un momento dado. Se diferencia de DBA_PRIV_AUDIT_OPTS en que, en lugar de mostrar sólo el privilegio que se audita, muestra sus opciones.
DBA_AUDIT_EXISTS	contiene los registros de los eventos sobre la existencia o no existencia de los objetos, incluyendo en dichos eventos todas las políticas producidas por *audit exists* y *audit not exists*.
DBA_AUDIT_OBJECT	muestra los registros de auditoría producidos por políticas de auditoría genérica que están relacionadas con objetos (tablas, vistas, índices, secuencias, enlaces, disparadores, espacios de tablas, etc.).
USER_AUDIT_OBJECT	muestra los registros de auditoría producidos por políticas de auditoría genérica que están relacionadas con objetos (tablas, vistas, índices, secuencias, enlaces, disparadores, espacios de tablas, etc.) y que has sido producidas por el usuario propietario de la vista. Esta vista no es única, y la tienen todos los usuarios en su esquema.
DBA_AUDIT_SESSION	muestra los registros de auditoría producidos por políticas de auditoría genérica que están relacionadas con inicio y fin de

	sesión (CONNECT y DISCONNECT).
USER_AUDIT_SESSION	muestra los registros de auditoría producidos por políticas de auditoría genérica que están relacionadas con inicio y fin de sesión (CONNECT y DISCONNECT) y que has sido producidas por el usuario propietario de la vista. Esta vista no es única, y la tienen todos los usuarios en su esquema.
DBA_AUDIT_STATEMENT	muestra los registros de auditoría producidos por políticas de auditoría genérica que están relacionadas con las siguientes operaciones: GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM.
USER_AUDIT_STATEMENT	muestra los registros de auditoría producidos por políticas de auditoría genérica que están relacionadas con las siguientes operaciones: GRANT, REVOKE, AUDIT, NOAUDIT, y ALTER SYSTEM y que has sido producidas por el usuario propietario de la vista. Esta vista no es única, y la tienen todos los usuarios en su esquema.
DBA_AUDIT_TRAIL	muestra todos los registros de auditoría genérica.
USER_AUDIT_TRAIL	muestra todos los registros de auditoría genérica y que has sido producidas por el usuario propietario de la vista. Esta vista no es única, y la tienen todos los usuarios en su esquema.

Análisis de la recolección de auditoría de grano fino

Debido a que la auditoría de grano fino guarda los registros de auditoría en una tabla llamada FGA_LOG$, el motor de base de datos *Oracle* ofrece al auditor un conjunto de vistas que permiten el análisis de los datos de dicha tabla desde diferentes enfoques. A continuación se explicarán dichas vistas:

ALL_AUDIT_POLICIES	**describe todas las políticas de auditoría de grano fino declaradas en el sistema, ya sean habilitadas o**

	no.
ALL_AUDIT_POLICY_COLUMNS	describe todas las políticas de auditoría de grano fino que estén realizadas para realizar una operación sobre una o varias columna específica de ciertas tablas.
ALL_DEF_AUDIT_OPTS	contiene las opciones por defecto de auditoría aplicados cuando las políticas sean creadas. Si el auditor no especifica ciertas opciones, dichas opciones se configurarán por defecto según esta tabla.
AUDIT_ACTIONS	contiene códigos de las acciones que pueden ser auditadas. Es una especie de catálogo para la optimización de la auditoría de grano fino por parte de *Oracle*.
DBA_AUDIT_POLICIES	muestra exactamente los mismos datos que la vista ALL_AUDIT_POLICIES
DBA_COMMON_AUDIT_TRAIL	contiene todos los registros de auditoría, tanto genérica como de grano fino.
DBA_FGA_AUDIT_TRAIL	muestra todos los registros de auditoría realizados con políticas de auditoría de grano fino.
STMT_AUDIT_OPTION_MAP	es un mapa de opciones de auditoría que contiene códigos de las opciones que pueden tener las políticas de auditoría. Es una especie de catálogo para la optimización de la auditoría de grano fino por parte de *Oracle*.
V$XML_AUDIT_TRAIL	contiene todos los registros de auditoría, tanto genérica como de grano fino, auditoría de SYS y registros en XML. Cuando los registros de auditoría se traducen a un formato XML OS, se pueden leer con un editor de texto o a través de esta vista, que contiene información similar a la vista DBA_AUDIT_TRAIL.

Auditoria Tradicional

La sintaxis del comando Audit es la siguiente:

AUDIT

opc_sentencia.
{ BY usuario}
[BY { SESSION | ACCESS }]
[WHENEVER [NOT] SUCCESSFUL] ;

opc_sentencia	Especifica la sentencia/s SQL que se desea auditar.
BY usuario	Indica que se quieren auditar las sentencias SQL requeridas para el usuario/s indicados. Si se omite, la auditoría se realiza para todos los usuarios de la B.D.
BY SESSION	Provoca que Oracle inserte un único registro resumen en la tabla de auditoría aunque la sentencia se ejecute varias veces en la misma sesión.
BY ACCESS	Provoca la escritura de un registro en las tablas de auditoría cada vez que la sentencia se ejecuta. Cuando se especifican auditorías de sentencias DDL o de privilegios del sistema, la auditoría por defecto es por accesos. Cuando se auditan sobre objetos o sentencias DML, la auditoría por defecto es por sesión
WHENEVER SUCCESSFUL	Se realiza la auditoría cuando la sentencia auditada haya concluido satisfactoriamente
WHENEVER NOT SUCCESSFUL	Se realiza la auditoría cuando la sentencia auditada NO concluya satisfactoriamente.

Auditoria de Sesión

Auditorías de inicio de sesión: cada intento de conexión con la base de datos por parte de un usuario puede ser auditado. El comando para iniciar la auditoría de los intentos de inicio de sesión es, Audit Session, este comando auditará tanto los intentos fallidos como los aciertos.

Para auditar sólo los intentos fallidos se utiliza el comando:
audit session whenever not successful;
Para auditar sólo las conexiones **correctas** se utiliza el comando:
audit session whenever successful;

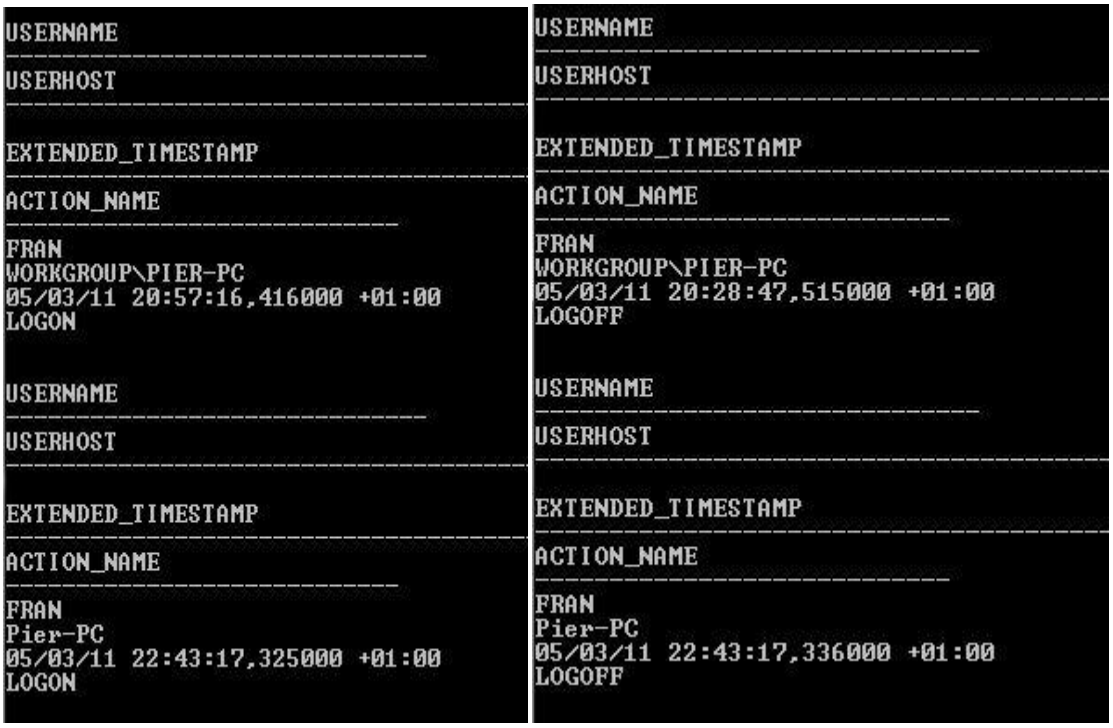
Vamos a auditor las sesiones de un usuario (Fran) cuando se desconecta y conecta así como el usuario sys.

```
SQL> audit session;

Auditoría terminada correctamente.
```

```
SQL> audit session by fran, pier;

Auditoría terminada correctamente.
```

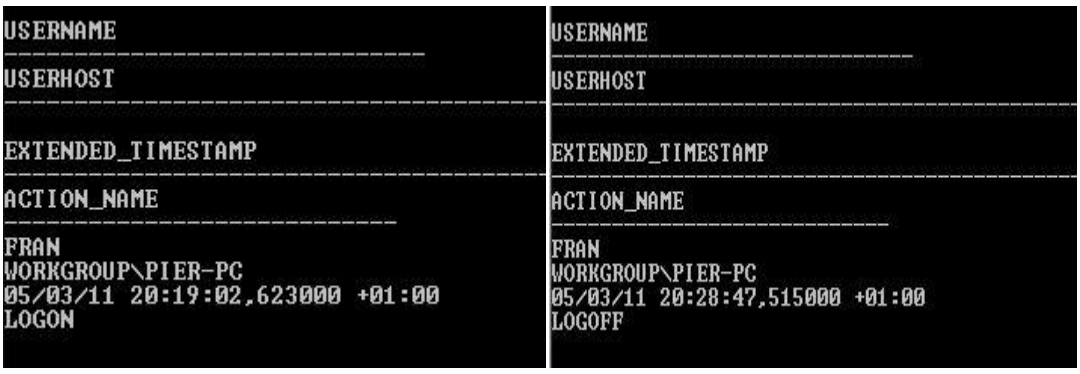
El resultado de la auditoria de la auditoria a la hora de consultar la vista es el siguiente:

Select Username, userhost, extended_timestamp, action_name from dba_audit_session where username='FRAN';



Podemos ver en la vista de usuario cuando el usuario Fran se ha conectado

Select username, timestamp, action_name, comment_text, priv_used from dba_audit_trail where username='FRAN';



Auditoria de Acción

Auditorías de acción: cualquier acción que afecte a un objeto de la base de datos (tabla, enlace de base de datos, espacio de tablas, sinónimo, segmento de anulación, usuario, índice, etc.) puede auditarse. Las posibles acciones que pueden auditarse (create, alter, drop) sobre estos objetos pueden agruparse para simplificar la cantidad de esfuerzo administrativo necesario para determinar y mantener las opciones de configuración de la auditoría.
audit role; Este comando activará la auditoría de las acciones:

create role, alter role, drop role y set role.

También se puede ser más selectivo, por ejemplo, si se quiere auditar a un usuario concreto cuando realiza la acción "update" .Ejemplo:

audit update table by nombre_usuario;

De esta forma se activará la auditoría para el usuario "nombre_usuario" sólo cuando ejecute el comando "update" para cualquier tabla.

Se puede auditar cualquier acción que afecte a cualquier objeto de la BD. Para facilitar la gestión, las acciones a auditar se encuentran agrupadas según los grupos que se muestran en la siguiente tabla:

Grupo	***Comandos Auditados***
CLUSTER	Todas las sentencias que afecten a *clusters*.
DATABASE LINK	Todas las sentencias que afecten a enlaces de BD.
EXISTS	Todas las sentencias que fallen porque ya existe un objeto en la BD.
INDEX	Todas las sentencias que afecten a índices.
NOT EXISTS	Todas las sentencias que fallen porque un determinado objeto no existe.
PROCEDURE	Todas las sentencias que afecten a procedimientos.
PROFILE	Todas las sentencias que afecten a perfiles.
PUBLIC DATABASE LINK	Todas las sentencias que afecten a enlaces públicos de BD.
PUBLIC SINONYM	Todas las sentencias que afecten a sinónimos públicos.
ROLE	Todas las sentencias que afecten a roles.
ROLLBACK SEGMENT	Todas las sentencias que afecten a segmentos de *rollback*.
SEQUENCE	Todas las sentencias que afecten a secuencias.
SESSION	Todas las sentencias de acceso a la BD.
SYNONYM	Todas las sentencias que afecten a sinónimos.
SYSTEM AUDIT	Todas las sentencias `AUDIT` y `NOAUDIT`.
SYSTEM GRANT	Todas las sentencias afecten a privilegios.
TABLE	Todas las sentencias que afecten a tablas.
TABLESPACE	Todas las sentencias que afecten a espacios de tablas.
TRIGGER	Todas las sentencias que afecten a disparadores.
USER	Todas las sentencias que afecten a las cuentas de usuarios.
VIEW	Todas las sentencias que afecten a vistas.

```
SQL> audit create table by fran;
Auditoría terminada correctamente.
SQL>
```

Nos vamos al usuario y creamos una tabla, una vez hecho esto podemos irnos al usuario sys y con el siguiente comando obtendremos estos resultados:

```
SQL> Select * from dba_priv_audit_opts where user_name='FRAN';

USER_NAME                      PROXY_NAME
------------------------------ ------------------------------
PRIVILEGE                                SUCCESS    FAILURE
---------------------------------------- ---------- ----------
FRAN
GRANT ANY OBJECT PRIVILEGE               BY ACCESS  NOT SET

FRAN
GRANT ANY PRIVILEGE                      BY ACCESS  NOT SET

FRAN
SELECT ANY TABLE                         BY ACCESS  NOT SET


USER_NAME                      PROXY_NAME
------------------------------ ------------------------------
PRIVILEGE                                SUCCESS    FAILURE
---------------------------------------- ---------- ----------
FRAN
DROP ANY TABLE                           BY ACCESS  NOT SET

FRAN
CREATE ANY TABLE                         BY ACCESS  BY ACCESS

FRAN
CREATE TABLE                             BY ACCESS  BY ACCESS


USER_NAME                      PROXY_NAME
------------------------------ ------------------------------
PRIVILEGE                                SUCCESS    FAILURE
---------------------------------------- ---------- ----------
FRAN
CREATE SESSION                           BY ACCESS  BY ACCESS
```


```
SQL> Select * from dba_stmt_audit_opts where user_name='FRAN';

USER_NAME                      PROXY_NAME
------------------------------ ------------------------------
AUDIT_OPTION                             SUCCESS    FAILURE
---------------------------------------- ---------- ----------
FRAN
INSERT TABLE                             BY ACCESS  BY ACCESS

FRAN
DELETE TABLE                             BY SESSION BY SESSION

FRAN
GRANT ANY OBJECT PRIVILEGE               BY ACCESS  NOT SET


USER_NAME                      PROXY_NAME
------------------------------ ------------------------------
AUDIT_OPTION                             SUCCESS    FAILURE
---------------------------------------- ---------- ----------
FRAN
GRANT ANY PRIVILEGE                      BY ACCESS  NOT SET

FRAN
SELECT ANY TABLE                         BY ACCESS  NOT SET

FRAN
DROP ANY TABLE                           BY ACCESS  NOT SET


USER_NAME                      PROXY_NAME
------------------------------ ------------------------------
AUDIT_OPTION                             SUCCESS    FAILURE
---------------------------------------- ---------- ----------
FRAN
CREATE ANY TABLE                         BY ACCESS  BY ACCESS

FRAN
CREATE TABLE                             BY ACCESS  BY ACCESS

FRAN
CREATE SESSION                           BY ACCESS  BY ACCESS
```

Ahora veremos cómo queda registrado la creación de la tabla cuando auditamos el create table para el usuario en concreto Fran.

Select username,owner,obj_name,action_name,priv_used,timestamp FROM dba_audit_object where username='FRAN';

```
ACTION_NAME                    PRIV_USED                                TIMESTAM
------------------------------ ---------------------------------------- --------
CREATE TABLE                   CREATE TABLE                             06/03/11

FRAN                             FRAN
COMPRAS
CREATE TABLE                   CREATE TABLE                             06/03/11

FRAN                             FRAN

USERNAME                       OWNER
------------------------------ ------------------------------
OBJ_NAME
--------------------------------------------------------------------------------

ACTION_NAME                    PRIV_USED                                TIMESTAM
------------------------------ ---------------------------------------- --------
PROVEEDORES
CREATE TABLE                   CREATE TABLE                             06/03/11
```


Auditoria de un Objeto

Además de las acciones a nivel de sistema sobre objetos, también es posible auditar las acciones de manipulación de datos sobre objetos.
Se pueden auditar operaciones de *select*, *insert*, *update* y *delete* (selección, inserción, modificación y borrado) sobre tablas. Este tipo de auditoría es similar a la anterior de auditoría de acción, la única diferencia es que el comando "Audit" incorpora un parámetro nuevo "by session" (el registro de auditoría se Estudio e Implantación de Audit Vault escribirá una única vez por sesión) o "by access" (el registro de auditoría se escribirá cada vez que se acceda al objeto auditado).
Por ejemplo, para auditar los "insert" realizados sobre la tabla PROVEEDORES" por acceso, el comando será:

audit insert on PROVEEDORES by access;

Nota: al indicar "by Access" hay que tener cuidado pues registrará un suceso de auditoría por cada insert, esto puede afectar al rendimiento. De ser así siempre será mejor optar por "by session" que sólo registrará un suceso de auditoría por sesión, aunque es menos exhaustivo.
Otro ejemplo, para auditar todas las acciones realizadas en la tabla "PROVEEDORES" por sesión utilizaremos el siguiente comando:

audit all onPROVEEDORES by session;

El comando anterior auditará todas las acciones realizadas sobre la tabla FACTURACION (select, insert, update,delete), pero sólo un registro de auditoría por cada sesión.
Otro ejemplo, para auditar las eliminaciones de registros de la tabla "nóminas":

audit delete PROVEEDORES by access;

Vamos a ver como realizamos la auditoria de acciones sobre cada objeto.En este caso lo vamos a hacer sobre un insert y un delete. Para ello como siempre primero activamos las auditorias en el usuario sys:

Audit insert on Fran.proveedores by Access;

En el usuario Fran insertamos una fila

```
SQL> insert into proveedores
  2  values('P007','c.carretero','sevilla');

1 fila creada.
```

Ahora nos vamos al usuario sys y con la siguiente sentencia ya podemos ver que la acción ha quedado registrada.

Select username, userhost, timestamp, owner, obj_name, action_name from dba_audit_object where username='FRAN';

```
USERNAME
------------------------------
USERHOST
----------------------------------------------------------------

TIMESTAM OWNER
-------- ------------------------------
OBJ_NAME
----------------------------------------------------------------

ACTION_NAME
----------------------------
FRAN
WORKGROUP\PIER-PC
06/03/11 FRAN

USERNAME
------------------------------
USERHOST
----------------------------------------------------------------

TIMESTAM OWNER
-------- ------------------------------
OBJ_NAME
----------------------------------------------------------------

ACTION_NAME
----------------------------
PROVEEDORES
INSERT
```

Para auditar el delete realizamos la misma operación primero en el usuario sys activamos la auditoria, realizamos la acción en el usuario Fran y volvemos a sys para realizar la comprobación de que ha quedado registrada auditoria.

Primero como sys: Audit delete on fran.proveedores by Access;

Vamos al usuarion fran y mediante un delete proveedores borramos las filas de la tabla.

Ahora nos vamos de nuevo a sys

Select username, userhost, timestamp, owner, obj_name, action_name from dba_audit_object where username='FRAN'; y ya tenemos de Nuevo registrada la accion delete.

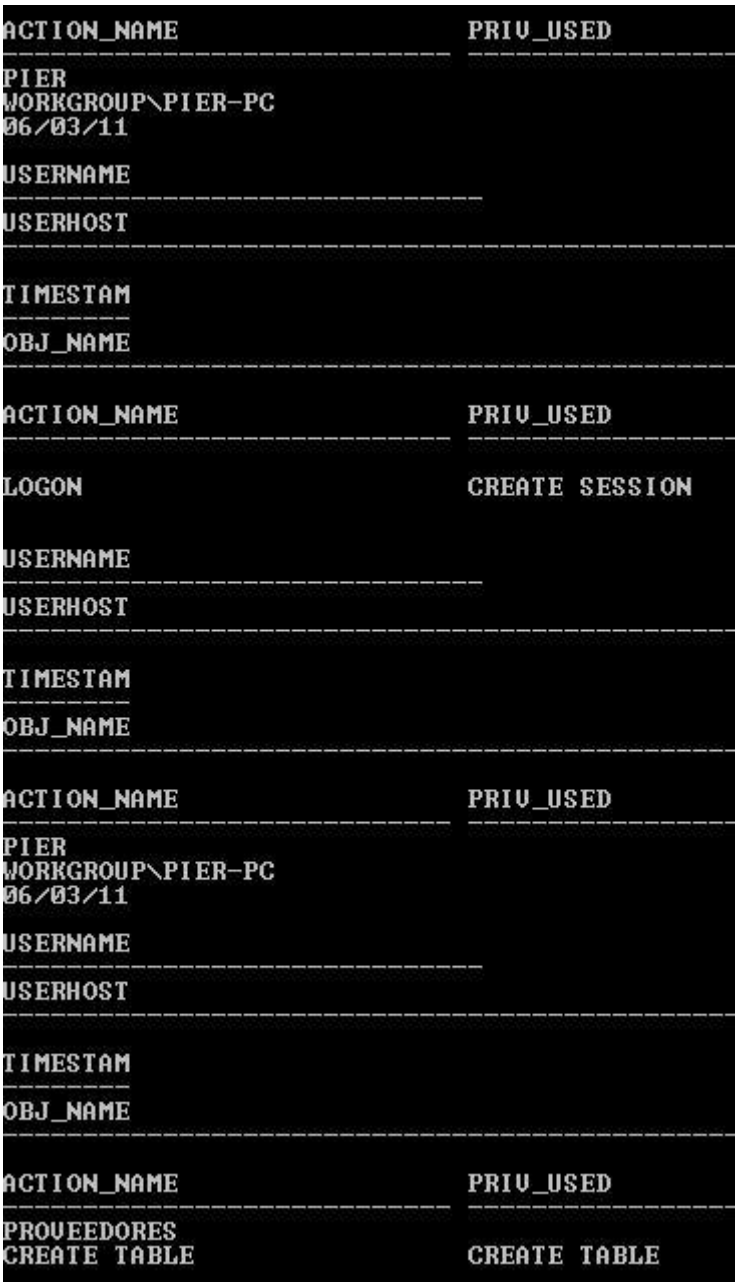
```
USERNAME                       OWNER
------------------------------ ------------------------------
OBJ_NAME
--------------------------------------------------------------------------------

ACTION_NAME                    PRIV_USED                                TIMESTAM
------------------------------ ---------------------------------------- --------
FRAN                           MDSYS
SDO_GEOR_DDL__TABLE$$
DELETE                                                                  06/03/11

FRAN                           FRAN
PROVEEDORES
DELETE                                                                  06/03/11

USERNAME                       OWNER
------------------------------ ------------------------------
```

Auditar acciones de cada usuario

Mediante el comando Audit all; podemos realizar la auditoria de cada usuario.Para ello lo realizaremos con varios usuarios en éste caso vamos a usar Pier y Fran1.

En el usuario Pier creamos una tabla (proveedores). Si nos vamos a sys con el siguiente comando podemos ver el resultado guardado.

Select username, userhost, timestamp, obj_name, action_name, priv_used from dba_audit_trail where username='PIER';



Nos vamos a conectar ahora como Fran1

```
SQL> Select username, userhost, timestamp, obj_name, action_name, priv_used from
 dba_audit_trail where username='FRAN1';

USERNAME
------------------------------
USERHOST
--------------------------------------------------------------------------------

TIMESTAM
--------
OBJ_NAME
--------------------------------------------------------------------------------

ACTION_NAME                    PRIV_USED
---------------------------- ----------------------------------------

FRAN1
WORKGROUP\PIER-PC
06/03/11

USERNAME
------------------------------
USERHOST
--------------------------------------------------------------------------------

TIMESTAM
--------
OBJ_NAME
--------------------------------------------------------------------------------

ACTION_NAME                    PRIV_USED
---------------------------- ----------------------------------------

LOGON                          CREATE SESSION
```

Creamos una tabla que para el ejemplo será la misma proveedores e insertaremos unos registros para hacer select. Una vez hecho esto vamos a sys y con el mismo comando vemos el registro de la creación de la tabla.

```
ACTION_NAME                    PRIV_USED
---------------------------- ----------------------------------------

FRAN1
WORKGROUP\PIER-PC
06/03/11

USERNAME
------------------------------
USERHOST
--------------------------------------------------------------------------------

TIMESTAM
--------
OBJ_NAME
--------------------------------------------------------------------------------

ACTION_NAME                    PRIV_USED
---------------------------- ----------------------------------------

PROVEEDORES
CREATE TABLE                   CREATE TABLE
```

Y así sucedivamente si vamos realizando acciones sobre el usuario en cuestión ya sea insert, drop....

Auditar Privilegio del Sistema:

En éste caso vamos a auditar el delete en las tablas que hemos ido creando en los distintos usuarios.

Para que quede constancia de ello es necesario activar la auditoria siguiente:

Audit delete any table by Access;

En los distintos usuarios borramos los registros y volvemos al usuario sys y con la siguiente sentencia

```
SQL> SELECT ACTION_NAME, USERNAME,TIMESTAMP,OBJ_NAME FROM dba_audit_trail WHERE
ACTION_NAME='DROP TABLE'   ;
```

quedan registrados el borrar de tablas.

```
ACTION_NAME                    USERNAME                       TIMESTAM
------------------------------ ------------------------------ --------
OBJ_NAME
--------------------------------------------------------------------------------

DROP TABLE                     FRAN1                          06/03/11
PROVEEDORES

DROP TABLE                     FRAN                           06/03/11
ALQUILERES
```

Gestionar registros de auditoría (proteger, destruir, etc.

Proteger

Los registros de la tabla SYS.AUD$ pueden ser objeto susceptible de ser eliminados ya que pueden reflejar acciones no autorizadas en la BD.Podemos obtener cualquier acción que se haga sobre ésta tabla ejecutando el siguiente comando.

audit all on sys.aud$ by access;

Conviene tener registrado todos los movimientos que se realizan sobre ésta tabla y quien tiene acceso a ella el supuesto de haber otro usuario aparte del dba con permiso para acceder, éste control podemos realizarlo por medio de la auditoria

Audit insert, update, delete on sys.aud$ by access:

Eliminar

Para una mayor seguridad podemos eliminar la tabla sys.aud$ una vez que hayamos analizado los datos para que no queden guardados, se haría:

Delete from sys.aud$;

Para la completa destrucción de los registros de la auditoria tales como vistas, tablas etc tenemos que ejecutar un script llamado ***catnoaud.sql*** que se encuentra en C:\app\Usuario\product\11.2.0\dbhome_1\RDBMS\ADMIN\catnoaud.sql

Analizar los redo logs con LogMiner

El Logminer es una herramienta de las bases de datos Oracle, que se utiliza para extraer sentencias DML directamente de los archivos de redo log. De estos es posible extraer la sentencia sql original que realizó la transacción y la sentencia que puede ser utilizada para deshacerla. Ayuda de una manera sencilla y comoda a interpretar los reso logs.

Para ver el valor inicial del log miner lo hacemos de la siguiente manera:

```
SQL> show parameter utl

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
create_stored_outlines               string
utl_file_dir                         string
SQL>
```

Captura parámetro show parameter utl

Para ver nuestro fichero de log basta con ejecutar el siguiente comando

```
SQL> select member from v$logfile;

MEMBER
--------------------------------------------------------------------------------

C:\APP\USUARIO\ORADATA\ORCL\REDO03.LOG
C:\APP\USUARIO\ORADATA\ORCL\REDO02.LOG
C:\APP\USUARIO\ORADATA\ORCL\REDO01.LOG

SQL>
```

Captura del parámetro select member from v$logfile

Para activar el log Miner tenemos que pasarle una ruta para el achivo que vayamos a usar en el parámetro "*utl_file_dir*". La ruta para éste ejemplo será c:\app\Usuario\logminer

```
SQL> alter system set utl_file_dir='c:app\Usuario\logminer' scope=spfile;

Sistema modificado.

SQL>
```

alter system set utl_file_dir='c:app\Usuario\logminer' scope=spfile;

Ahora al reiniciar la base de datos y la sesión de LogMiner será la del usuario sys y creamos el directorio anterior.

A continuación creamos el diccionario que va a usar LogMiner, el único aspecto a tener en cuenta es que debe estar en el mismo directorio que hemos creado anteriormente, lo hacemos mediante el siguiente comando.

exec DBMS_LOGMNR_D.BUILD(DICTIONARY_FILENAME =>'logminer.ora', DICTIONARY_LOCATION => 'C:\app\Usuario\logminer');

```
SQL> alter system set utl_file_dir='C:\app\Usuario\logminer' scope=spfile;

Sistema modificado.

SQL> shutdown
Base de datos cerrada.
Base de datos desmontada.
Instancia ORACLE cerrada.
SQL> startup
Instancia ORACLE iniciada.

Total System Global Area 1071333376 bytes
Fixed Size                  1375792 bytes
Variable Size             343933392 bytes
Database Buffers          721420288 bytes
Redo Buffers                4603904 bytes
Base de datos montada.
Base de datos abierta.
SQL> show parameter utl

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
create_stored_outlines               string
utl_file_dir                         string      C:\app\Usuario\logminer
SQL> exec DBMS_LOGMNR_D.BUILD( DICTIONARY_FILENAME =>'logminer.ora', DICTIONARY_
LOCATION => 'c:\app\Usuario\logminer');

Procedimiento PL/SQL terminado correctamente.

SQL>
```

Ya podemos ver el funcionamiento de logminer para ello le especificaremos a logminer los ficheros a estudiar (lo haremos en base a los anteriores).

Vamos a añadir los tres ficheros de log

```
SQL> exec DBMS_LOGMNR.add_logfile('c:\app\Usuario\oradata\orcl\REDO03.log');

Procedimiento PL/SQL terminado correctamente.

SQL> exec DBMS_LOGMNR.add_logfile('c:\app\Usuario\oradata\orcl\REDO02.log');

Procedimiento PL/SQL terminado correctamente.

SQL> exec DBMS_LOGMNR.add_logfile('c:\app\Usuario\oradata\orcl\REDO01.log');

Procedimiento PL/SQL terminado correctamente.
```

Y vamos a borrar el fichero de los 2 y 3

```
SQL> exec DBMS_LOGMNR.remove_logfile('c:\app\Usuario\oradata\orcl\REDO01.log');

Procedimiento PL/SQL terminado correctamente.

SQL> exec DBMS_LOGMNR.remove_logfile('c:\app\Usuario\oradata\orcl\REDO02.log');

Procedimiento PL/SQL terminado correctamente.

SQL>
```

Para comprobar su funcionamiento ejecutaremos la siguiente sentencia y veremos que el fichero correspondiente al redo3.log es el que se ha añadido a la carpeta que creamos para logminer.

exec DBMS_LOGMNR.START_LOGMNR(DICTFILENAME=>'c:\app\Usuario\logminer\logminer.ora') ;

```
SQL> exec DBMS_LOGMNR.START_LOGMNR(DICTFILENAME=>'c:\app\Usuario\logminer\logmin
er.ora') ;

Procedimiento PL/SQL terminado correctamente.

SQL>
```

Veamos un ejemplo práctico: Damos primero formato para la select de salida.

```
SQL> set pages 100
SQL> set lines 150
SQL> colum sql_redo format a50
SQL> column sql_undo format a50
```

La select es la siguiente:

select sql_redo, sql_undo from **v$logmnr_contents** where rownum<10;

Obteniendo el siguiente resultado

```
SQL> select sql_redo, sql_undo from v$logmnr_contents where rownum<10;

SQL_REDO                                                 SQL_UNDO
-------------------------------------------------------- --------------------------------
---------------------
set transaction read write;
update "SYS"."UNDO$" set "NAME" = '_SYSSMU1_151854 update "SYS"."UNDO$" set "NAM
E" = '_SYSSMU1_151854
8437$', "USER#" = '1', "FILE#" = '3', "BLOCK#" = ' 8437$', "USER#" = '1', "FILE#
" = '3', "BLOCK#" = '
128', "SCNBAS" = '5698896', "SCNWRP" = '0', "XACTS 128', "SCNBAS" = '5672058', "
SCNWRP" = '0', "XACTS
QN" = '1764', "UNDOSQN" = '700', "INST#" = '0', "S QN" = '1759', "UNDOSQN" = '68
4', "INST#" = '0', "S
TATUS$" = '2', "TS#" = '2', "SPARE1" = '2' where " TATUS$" = '3', "TS#" = '2', "
SPARE1" = '2' where "
NAME" = '_SYSSMU1_15185484437$' and "USER#" = '1' a NAME" = '_SYSSMU1_15185484437$
' and "USER#" = '1' a
nd "FILE#" = '3' and "BLOCK#" = '128' and "SCNBAS" nd "FILE#" = '3' and "BLOCK#"
 = '128' and "SCNBAS"
 = '5672058' and "SCNWRP" = '0' and "XACTSQN" = '1  = '5698896' and "SCNWRP" = '
0' and "XACTSQN" = '1
759' and "UNDOSQN" = '684' and "INST#" = '0' and " 764' and "UNDOSQN" = '700' an
d "INST#" = '0' and "
STATUS$" = '3' and "TS#" = '2' and "SPARE1" = '2'  STATUS$" = '2' and "TS#" = '2
' and "SPARE1" = '2'
and ROWID = 'AAAAAPAABAAAADhAAB';                        and ROWID = 'AAAAAPAABAAAADhA
AB';
```

La vista más importante del diccionario de datos para la herramienta logminer es v$logmnr_contents, pero existen estas otras que nos pueden ser de gran ayuda:

V$LOGMNR_CONTENTS
V$LOGMNR_DICTIONARY
V$LOGMNR_LOGS
V$LOGMNR_PARAMETERS

Por último para cerrar la session de LogMiner podernos hacerlo mediante:

exec DBMS_LOGMNR.END_LOGMNR;

o directamente saliendo del usuario sys.

37768972

Auditoria de Grano Fino

En la auditoría genérica, se guarda qué usuarios realizaron qué operación sobre un objeto de la base de datos. Sin embargo, a veces esto no es suficiente. A veces es necesario saber, qué consulta ejecutó un usuario sobre una tabla en un momento determinado o qué datos fueron borrados, modificados o insertados por parte del usuario.

Este tipo de auditoría se llama **Auditoría de grano fino**, y está disponible en Oracle desde su versión 9i. Es capaz de auditar no sólo qué objeto fue consultado por un usuario, sino que también puede auditar qué información obtuvo, en el caso de haber hecho consulta y qué información introdujo, borró o modificó en el caso de haber hecho una modificación.

La Auditoría de grano fino surgió por la necesidad de capturar acciones fruto del uso indebido de un privilegio por parte de un usuario. El *Log Miner* nos permite recuperar la información de los ficheros de Log, y uniendo sus fuerzas con la Auditoría general, podríamos recopilar la información de las inserciones, borrado o modificaciones pero aun así no sería tan potente como una auditoria de grano fino.

La clave está en que todas las sentencias SQL ejecutadas por el usuario sobre sus datos, o sobre el diccionario de datos, son grabadas en los archivos Redo Log con el objetivo de que una posible recuperación de la base de datos pueda llevarse a cabo. Con ello es posible realizar una reconstrucción de una tabla en un momento determinado usando dichas sentencias en el mismo orden en que fueron ejecutadas.

Hay que tener en cuenta que si se audita la inserción, en caso de que ésta no tenga éxito no se guarda el dato de auditoría. Es necesario indicarlo explícitamente usando auditoria genérica.

Veamos un ejemplo de auditoría de grano fino

Lo primero que vamos a hacer es crear una tabla en el usuario Fran

```
SQL> create table cursos
  2  (
  3    nombreCurso  varchar(50),
  4    codigoCurso varchar(50) constraint PK_CURSOS primary key,
  5    profesor varchar(9),
  6    maxAlumnos number,
  7    num_horas number
  8  );
```

Y la llenamos de registros

```
SQL> INSERT INTO cursos
  2   VALUES ('linux' , '001 ', 'jose',20,450);

1 fila creada.

SQL>
SQL> INSERT INTO cursos
  2   VALUES ('windows' , '002 ', 'alberto',30,120);

1 fila creada.

SQL> INSERT INTO cursos
  2   VALUES ('bbdd' , '003 ', 'jose',15,500);

1 fila creada.

SQL> INSERT INTO cursos
  2   VALUES ('office' , '004 ', 'isabel',35,150);

1 fila creada.

SQL> INSERT INTO cursos
  2   VALUES ('diseño' , '005 ', 'pedro',32,200);

1 fila creada.

SQL>
```

Ahora creamos la política que verificará los cursos sonde el máximo de alumnos es mayor a 25

Para que quede constancia de los registros haremos select sobre los datos introducidos.

select maxAlumnos from cursos where nombreCurso='linux';

select maxAlumnos from cursos where nombreCurso='office';

select maxAlumnos from cursos where nombreCurso='windows';

Ahora vamos al usuario sys y mediante el siguiente comando quedara registrada la sentencia que hemos realizado y que cumple la política de grano fino.

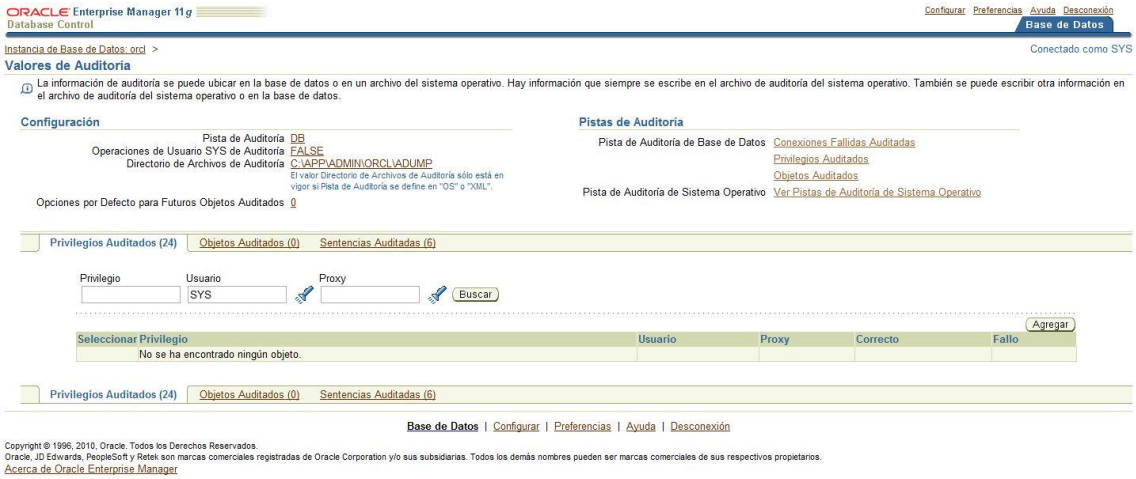
select sql_text from dba_fga_audit_trail;

La auditoria de grano fino es mucho mas exacta y permite mayoe detalle que la auditoria normal pero hay que tener en cuenta que el rendimiento de la base de datos disminuye considerablemente mas aun cuando la tenemos permanentemente activada, por eso tenemos que tener en cuenta que se debe realizar en contadas ocasiones y con el uso de la base de datos que sea mínimo para que no baje tanto su rendimiento

Auditar con la herramienta Enterprise Manager:

Enterprise Manager consiste en un conjunto de herramientas de administración que facilitan la configuración, el control y la administración de la base de datos Oracle.Está dirigido a los administradores y ofrece el acceso a múltiples sitios y sistemas desde una única interfaz de forma sencilla y amigable.En ningún caso sustituye a las acciones que se pueden hacer con la consola de comandos es más con ella podemos analizar todo con mucho mas detalle.

Para acceder a la Auditoria en Enterprise Manager tenemos que acceder en el panel de control a Servidores y una vez allí buscar Valores de auditoria.

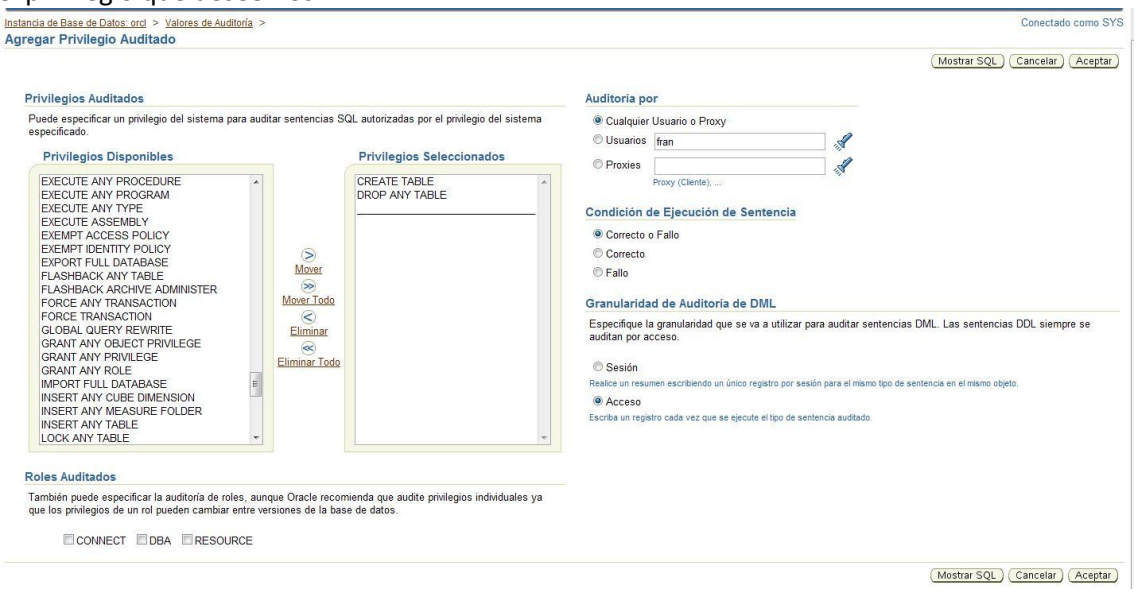


La información que muestra es referente a si la auditoria está activada, cuales es el directorio de la auditoria asi como los usuarios que tienen privilegios se encuentran auditados.

Tal y como lo hemos hecho antes lo primero que vamos a hacer es auditar pero es éste caso por la interfaz web.

Auditar privilegio para Fran:

Vamos a agregar privilegio y nos encontramos con la siguiente pantalla donde podemos añadir el privilegio que deseemos.



Tras rellenar el formulario, aceptamos y como en el caso de la consola tenemos que ir al usuario fran y ejecutar el privilegio que le hayamos dado.

Para comprobar que la auditoria ha quedado registrada en la interfaz principal vamos a privilegios auditados y nos aparecerá la sigueitne línea:

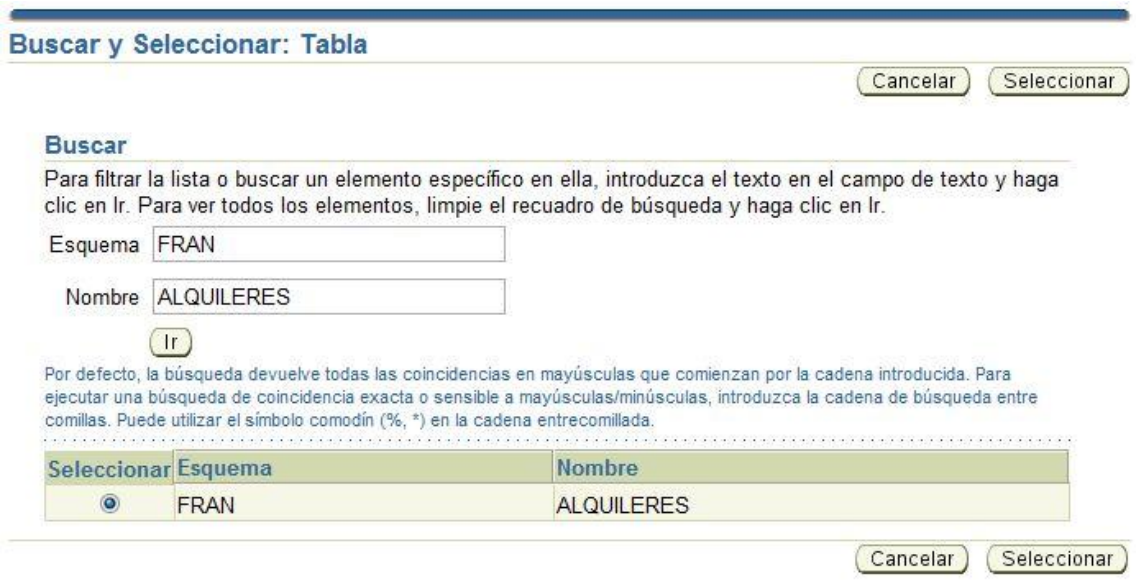
Privilegio	Nombre de Usuario	Acción	Hora
CREATE TABLE	FRAN	CREATE TABLE	2011-03-05 21:11:33.0

Auditar Objeto para Fran:

Siguiendo el mecanismo de antes ahora en la pantalla principal en lugar de privilegio, cliqueamos sobre objeto auditado

Al igual que antes reyenamos el formulario con el objeto que queramos auditar y aceptamos. Igualmente nos vamos al usuario a ejecutar la instrucción para que quede guardada en la auditoria.

En éste caso para auditar un objeto es necesario especificarle el usuario y la tabla para ello nos ponemos bre tabla de objetos y damos al icono de la derecha, se nos deplegará la siguiente ventana



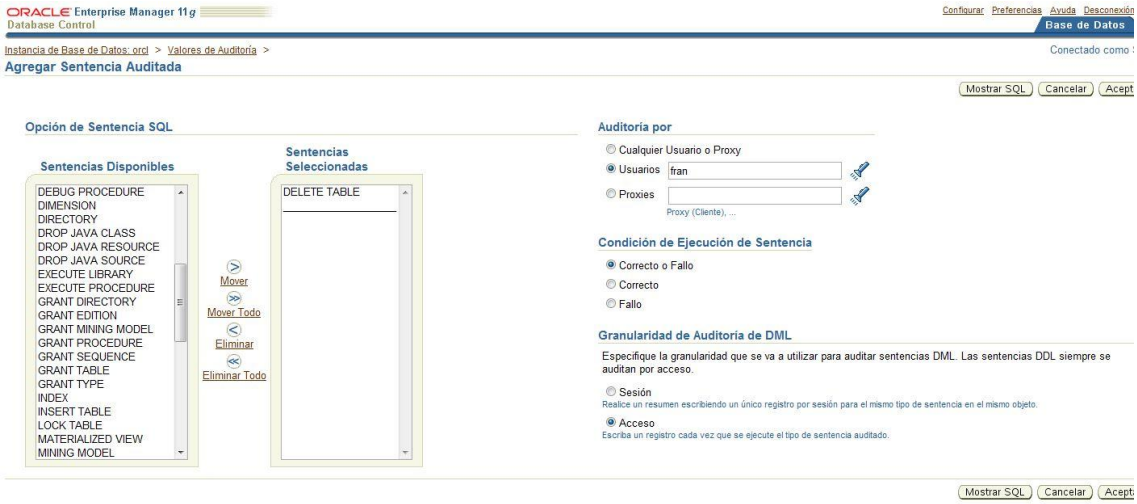
Donde buscamos el esquema y el nombre de la tabla sobre la que vamos a auditar.

Una vez aceptada la auditoria basta con irse de nuevo a objetos auditados y veremos la siguiente línea:

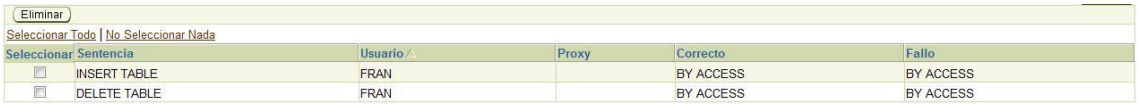
Esquema	Nombre del Objeto	Nombre de Usuario	Acción	Hora
FRAN	ALQUILERES	FRAN	INSERT	2011-03-05 22:55:00.0

Auditar una sentencia para Fran:

Volvemos a repetir el procedimiento anterior eligiendo la sentencia y rellenando el formulario.



Cuando aceptemos nos vamos a objetos auditados.



Enterprise Manager es una herramienta estupenda, amigable y sencilla tiene aquello que debe tener sin florituras, se agradece algo así de forma que veamos las cosas un poco mas claras que en la pantalla de comandos aunque creo que la interfaz está limitada y el nivel de detalle que tenemos con la “pantallita negra” no lo conseguimos con el Manager.