Administración de Bases de Datos

Tema 5
(parte 3 de 3)

Seguridad en las Bases de Datos

5.7. Auditoría

- Oracle utiliza diversos métodos de auditoría para monitorizar qué tipos de privilegios se están usando y a qué objetos se está accediendo
- Mediante la auditoría se pueden investigar actividades maliciosas (borrado de tablas, etc.) o recopilar datos sobre actividades concretas (tablas que se actualizan, fechas de inserciones, etc.)
- Tipos de auditoría:
  - **De instrucciones**. Audita sentencias SQL independientemente de los objetos físicos a los que se esté accediendo. Por ejemplo, **AUDIT TABLE** activa una auditoría sobre todas las operaciones relacionadas con las tablas: CREATE TABLE, ALTER TABLE, DELETE TABLE, etc.
  - **De privilegios**. Se pueden activar auditorías sobre el uso de sentencias que necesitan privilegios de sistema. Por ejemplo, **AUDIT CREATE ROLE**, **AUDIT CREATE USER**
  - **De objetos de un esquema**. Especifica una auditoría sobre un objeto específico. Por ejemplo, **AUDIT SELECT ON usu1.tabla1**. La auditoría de objetos siempre se aplica a todos los usuarios de la BD
  - **De granularidad fina**. Audita los accesos a tabla y los privilegios basándose en el contenido de los objetos a los que se está accediendo. Utiliza el paquete **DBMS_FGA**

- Es necesario inicializar el parámetro **AUDIT_TRAIL** en el fichero de parámetros inicial. Este parámetro NO es dinámico

```
AUDIT_TRAIL = [DB | OS | NONE | XML]
```

- **NONE** indica que la auditoría no está activada
- El valor **DB** (valor por defecto) indica que la auditoría se almacenará en base de datos excepto para los valores que siempre se envían al S.O. ("arranque" y "parada" de la instancia, conexiones a la base de datos con privilegios de administrador)
- El valor **OS** indica que la auditoría se almacenará en el S.O. El parámetro **AUDIT_FILE_DEST** indica el lugar en el que se guardan los ficheros, así como los registros de auditoría para SYS (por defecto $ORACLE_BASE\admin\$ORACLE_SID\adump y, en segundo lugar, $ORACLE_HOME\rdbms\audit)
- El valor **XML** indica que los registros se escriben como ficheros xml en el S.O. Pueden consultarse mediante la vista **V$XML_AUDIT_TRAIL**

- En la auditoría básica (sentencias, privilegios, esquemas) los registros se almacenan en la tabla del diccionario de datos **SYS.AUD$** ("*db audit trail*") o en ficheros de sistema operativo ("*operating system audit trail*"). En el primer caso, existen diferentes vistas que permiten usar la información almacenada. Por ejemplo: **DBA_AUDIT_TRAIL**

- Al crear la BD con el *Database Configuration Assistant* (DBCA), ésta se configura para auditar las sentencias SQL y privilegios más comunes.

- Dependiendo de los objetos auditados se recoge distinto tipo de información. En cualquier caso, siempre se recoge la siguiente:
  - Usuario que realizó una determinada operación
  - Sesión
  - Terminal
  - El objeto o los objetos a los que accedió el usuario
  - Fecha y hora del proceso
  - Código de la acción
- Sin embargo, no se refleja la información que se modificó (por ejemplo con un UPDATE)
- **¡¡ IMPORTANTE !!**. Puesto que la tabla **SYS.AUD$** es propiedad del usuario SYS, es recomendable crear un espacio de tablas exclusivo para la información de auditoría para no llenar, en exceso, el espacio de tablas SYSTEM
- Para auditar una sentencia SQL o privilegio se debe tener el privilegio de sistema **AUDIT SYSTEM**. Para auditar operaciones sobre un objeto, debe pertenecer al esquema o tener privilegio **AUDIT ANY**

- El parámetro (estático) de inicialización **AUDIT_SYS_OPERATIONS** permite especificar la auditoría de aquellas sesiones de usuarios conectados como **SYS** (privilegio SYSDBA)

- Los registros generados son escritos en el "*audit trail*" de sistema operativo. No depende del valor del parámetro AUDIT_TRAIL

- Si su valor es **TRUE** (**AUDIT_SYS_OPERATIONS = TRUE**), se auditan dichas operaciones

- Si su valor es **FALSE** (valor por defecto), las operaciones no son auditadas

Auditoría de instrucciones

- Sintaxis:

```
AUDIT {<sentencia1, sentencia2, ...> | ALL}
[BY <usuario1, usuario2,...>]
[BY SESSION | BY ACCESS]
[WHENEVER [NOT] SUCCESSFUL]
```

- Para cada sentencia auditada, Oracle genera un registro de auditoría que contiene, entre otra, la siguiente información:
  - El usuario que ejecuta la operación
  - El tipo de operación
  - El objeto que interviene en la operación
  - Fecha y hora de la operación

- **BY <usuario>:** se puede especificar una lista de usuarios a los que se quiere auditar (por defecto se audita a todos)
- **BY SESSION:** se escribe una sola línea por cada sentencia del mismo tipo que se audita en la misma sesión (por defecto)
- **BY ACCESS:** se escribe una línea por cada sentencia. Esta opción puede aumentar mucho el log de auditoría, por lo que sólo se debe utilizar para objetos críticos y en un periodo determinado de tiempo en el que se busque algo concreto. Si se auditan sentencias de definición de datos del lenguaje (LDD) siempre se audita por acceso
- **WHENEVER SUCCESSFUL:** sólo se auditan las sentencias que han tenido éxito
- **WHENEVER NOT SUCCESSFUL:** sólo se auditan las sentencias fallidas
- Si se omiten las dos opciones anteriores, se realiza la auditoría independientemente del éxito o fallo de la sentencia

Nota: Para cualquier tipo de auditoría (sentencia, privilegio o sistema), si se opta por auditar **NOT SUCCESSFUL**, se generan registros sólo si el fallo se produce por alguna razón relacionada con la opción auditada. Por ejemplo, no se produce registro si una sentencia falla por no tener cuota estando auditando CREATE TABLE

- Ejemplos:

```
AUDIT ALL;

AUDIT SESSION WHENEVER NOT SUCCESSFUL;

AUDIT TABLE;
```

- La segunda sentencia audita los inicios de sesiones fallidos. La vista para consultar esta auditoría es **DBA_AUDIT_SESSION**

```
SELECT * FROM DBA_AUDIT_SESSION;
```

- La tercera sentencia audita operaciones sobre tablas (creación, borrado, etc.). La vista para consultar este tipo de auditoría es **DBA_AUDIT_OBJECT**

```
SELECT * FROM DBA_AUDIT_OBJECT;
```

Opcion	Sentencias SQL auditadas.
Database link	Create database link / drop database link
Index	Create index / alter index /drop index
Not exists	Todas las sentencias SQL que fallan por no existir un determinado objeto
Procedure	Create function / create package / create package body / create procedure / drop function / drop package / drop procedure
Public database link	Create public database link / drop public database link
Public synonym	Create public synonym / Drop public synonym
Role	Create role / alter role / drop role / set role

Opción	Sentencias SQL auditadas.
Rollback Statement	Create rollback segment/ alter rollback segment / drop rollback segment
Sequence	Create sequence / drop sequence
Session	Conexiones - *valor por defecto y único BY SESSION* -
Synonym	Create synonym / drop synonym
System audit	Audit sentencias_sql / Noaudit sentencias_sql
System grant	Grant y revoke privilegios_sistema y roles
Table	Create table / drop table / truncate table
Tablespace	Create tablespace / drop tablespace / alter tablespace

Opción	Sentencias SQL auditadas.
Trigger	Create trigger / drop trigger /alter trigger
User	Create user / alter user /drop user
View	Create view /drop view
Alter table	Alter table
Delete table	Delete from <tabla>, <vista>
Grant procedure	Grant / revoke <privilegio> on <procedimiento, funcion, paquete>
Grant sequence	Grant / revoke <privilegio> on <secuencia>
Grant table	Grant / revoke <privilegio> on <tabla, vista, vista materializada>

Opción	Sentencias SQL auditadas.
Insert table	Insert into <tabla, vista>
Lock table	Lock table <tabla, vista>
Select sequence	Cualquier sentencia que contenga sequence.CURRVAL o sequence.NEXTVAL
Select table	Select from <tabla, vista, vista materializada>
Update table	Update <tabla, vista>

Auditoría de privilegios

- Sintaxis:

```
AUDIT {<priv_sistema1, priv_sistema2, ...> | ALL PRIVILEGES}
[BY <usuario1, usuario2,...>]
[BY SESSION | BY ACCESS]
[WHENEVER [NOT] SUCCESSFUL]
```

- La cláusula **ALL PRIVILEGES**, indica que debe auditarse todo los privilegios de sistema

Privilegios auditables

ALTER DATABASE	*CREATE PROFILE*
ALTER SYSTEM	*ALTER PROFILE*
AUDIT SYSTEM	*DROP PROFILE*
CREATE DATABASE LINK	*CREATE ROLE*
CREATE PUBLIC DATABASE LINK	*ALTER ANY ROLE*
DROP PUBLIC DATABASE LINK	*DROP ANY ROLE*
CREATE PROCEDURE	*CREATE ROLLBACK SEGMENT*
CREATE ANY PROCEDURE	*ALTER ROLLBACK SEGMENT*
ALTER ANY PROCEDURE	*DROP ROLLBACK SEGMENT*
DROP ANY PROCEDURE	
EXECUTE ANY PROCEDURE	*CREATE SESSION*
	ALTER SESSION

Privilegios auditables

CREATE ANY TABLE / ANY INDEX *ALTER ANY TABLE / ANY INDEX* *DELETE ANY TABLE* *DROP ANY TABLE / ANY INDEX* *INSERT ANY TABLE* *UPDATE ANY TABLE* *SELECT ANY TABLE*	*CREATE USER* *ALTER USER* *DROP USER* *CREATE VIEW* *CREATE ANY VIEW* *DROP ANY VIEW* *ANALYZE ANY* *AUDIT ANY* *COMMENT ANY TABLE*

Auditoría de objetos de un esquema

- Sintaxis:

```
AUDIT {<sentencia_objeto1, sentencia_objeto2, ...> | ALL}
ON {[esquema.]objeto_auditado | DEFAULT}
[BY SESSION | BY ACCESS]
[WHENEVER [NOT] SUCCESSFUL]
```

- La cláusula **ALL** indica todas las opciones posibles sobre un tipo de objeto concreto
- Mediante la cláusula **ON DEFAULT** se establece por defecto las opciones indicadas para todo objeto creado en adelante. Una vez se han establecido estas opciones de auditoría predeterminadas, cualquier objeto que se cree con posterioridad se auditará automáticamente con esas opciones
- Pueden cambiarse indicando explícitamente el objeto

- Ejemplos:

```
AUDIT INSERT ON pepe.NOMINAS;

AUDIT ALL ON anselmo.EMPLEADOS;

AUDIT DELETE ON luisa.VENTAS BY SESSION;
```

- La primera sentencia audita las operaciones de inserción que se realizan sobre la tabla NOMINAS
- La segunda sentencia audita todas las operaciones de LMD que se hagan sobre la tabla EMPLEADOS
- La tercera sentencia audita las operaciones de borrado que se hagan sobre la tabla VENTAS, pero almacenando un único registro por sesión
- La vista para consultar este tipo de auditoría es DBA_AUDIT_OBJECT

```
SELECT * FROM DBA_AUDIT_OBJECT;
```

- Para desactivar la auditoría de una sentencia es necesario poseer el privilegio de sistema **AUDIT SYSTEM**
- Para detener la auditoría sobre un objeto, debe pertenecer al esquema o tener el privilegio **AUDIT ANY**
- Se utiliza la sentencia **NOAUDIT**. La sintaxis de NOAUDIT es similar a la de AUDIT (para cada sentencia de auditoría es necesaria una sentencia NOAUDIT que la deshabilite)
- Mediante la sentencia NOAUDIT sólo se indican deshabilitar opciones de auditoría. Para deshabilitar completamente la auditoría debe modificarse el parámetro de inicialización **AUDIT_TRAIL**
- La gestión del "*audit trail*", en modo DB, los registros son grabados en el objeto de SYS (SYS.AUD$), el cual es directamente modificable. Pueden borrarse registros con la sentencia DELETE:

```
DELETE FROM SYS.AUD$;

DELETE FROM SYS.AUD$ WHERE OBJ$NAME= <nombre_objeto>;
```

- No es conveniente auditar todos los movimientos de la base de datos, ya que las vistas serían difícil de manejar e interpretar
- Recomendaciones:
  - usar la auditoría donde y cuando se necesite
  - auditar inicios de sesión y operaciones administrativas
  - auditar objetos críticos con datos críticos
  - controlar el tamaño de la tabla

- Protección de la pista de auditoría:

```
AUDIT ALL ON SYS.AUD$ BY ACCESS;
```

Auditoría de grano fino

- Para llevarla a cabo NO es preciso habilitar la auditoria básica ("*audit trail*")
- Permite monitorizar accesos a datos basándose en su contenido y auditar sentencias SELECT y LMD en tablas y vistas mediante la creación de una política de auditoría al efecto
- Se usa el paquete **DBMS_FGA** y sus procedimientos asociados, generándose apuntes en el "*audit trail*" de grano fino. Se accede a través de la vista **DBA_FGA_AUDIT_TRAIL**
- Este tipo de auditoría permite especificar columnas, especialmente relevantes, que precisan ser auditadas (Ejemplo: dni, sueldos, datos de salud, ...)
- Es posible establecer acciones a llevar a cabo si se produce el acceso conforme a la política de auditoría implementada
- Proporciona las siguientes funcionalidades:
  - Usar diferentes políticas para INSERT, UPDATE, DELETE y SELECT, así como tener varias de ellas asociadas a cada tabla
  - Activar la auditoria sólo cuando es necesaria y sólo si es referenciada una columna concreta

- Ejemplo:

```
CREATE TABLE emp (
 empno     NUMBER(4) NOT NULL,
 ename     VARCHAR2(10),
 job       VARCHAR2(9),
 mgr       NUMBER(4),
 hiredate  DATE,
 sal       NUMBER(7,2),
 comm      NUMBER(7,2),
 deptno    NUMBER(2)
);

INSERT INTO emp (empno, ename, sal) VALUES (9999, 'Tim', 1);
INSERT INTO emp (empno, ename, sal) VALUES (9998, 'Larry', 50001);
```

```
BEGIN
  DBMS_FGA.add_policy(
    object_schema   => 'AUDIT_TEST',
    object_name     => 'EMP',
    policy_name     => 'SALARY_CHK_AUDIT',
    audit_condition => 'SAL > 50000',
    audit_column    => 'SAL');
END;
```

```
SELECT sal FROM emp WHERE ename = 'Tim';
SELECT sal FROM emp WHERE ename = 'Larry';
```

```
SELECT sql_text
FROM dba_fga_audit_trail;


SQL_TEXT
----------------------------------------
SELECT sal FROM emp WHERE ename = 'Larry'
```

sólo se registra la consulta donde el sueldo es mayor que 50000

- Procedimiento **ADD_POLICY**:

```
DBMS_FGA.ADD_POLICY(
   object_schema      VARCHAR2,
   object_name        VARCHAR2,
   policy_name        VARCHAR2,
   audit_condition    VARCHAR2,
   audit_column       VARCHAR2,
   handler_schema     VARCHAR2,
   handler_module     VARCHAR2,
   enable             BOOLEAN,
   statement_types    VARCHAR2,
   audit_trail        BINARY_INTEGER IN DEFAULT,
   audit_column_opts  BINARY_INTEGER IN DEFAULT);
```

- Procedimiento **ADD_POLICY**:

object_schema:	The schema of the object to be audited. (If NULL, the current log-on user schema is assumed.)
object_name:	The name of the object to be audited
policy_name:	The unique name of the policy
audit_condition:	A condition in a row that indicates a monitoring condition. NULL is allowed and acts as TRUE
audit_column:	The columns to be checked for access. These can include hidden columns. The default, NULL, causes audit if any column is accessed or affected
handler_schema:	The schema that contains the event handler. The default, NULL, causes the current schema to be used
handler_module:	The function name of the event handler; includes the package name if necessary. This function is invoked only after the first row that matches the audit condition in the query is processed. If the procedure fails with an exception, the user SQL statement will fail as well
enable:	Enables the policy if TRUE, which is the default
statement_types:	The SQL statement types to which this policy is applicable: INSERT, UPDATE, DELETE, or SELECT only
audit_trail:	Destination of fine grained audit records. Also specifies whether to populate LSQLTEXT and LSQLBIND in fga_log$
audit_column_opts:	Establishes whether a statement is audited when the query references any column specified in the audit_column parameter or only when all such columns are referenced

- Procedimiento **DISABLE_POLICY**:

```
DBMS_FGA.DISABLE_POLICY(
   object_schema   VARCHAR2,
   object_name     VARCHAR2,
   policy_name     VARCHAR2 );
```

object_schema:	The schema of the object to be audited. (If NULL, the current log-on user schema is assumed.)
object_name:	The name of the object to be audited
policy_name:	The unique name of the policy

- Procedimiento **ENABLE_POLICY**:

```
DBMS_FGA.ENABLE_POLICY(
   object_schema   VARCHAR2,
   object_name     VARCHAR2,
   policy_name     VARCHAR2,
   enable          BOOLEAN);
```

object_schema:	The schema of the object to be audited. (If NULL, the current log-on user schema is assumed.)
object_name:	The name of the object to be audited
policy_name:	The unique name of the policy
enable:	Defaults to TRUE to enable the policy

- Procedimiento **DROP_POLICY**:

```
DBMS_FGA.DROP_POLICY(
   object_schema   VARCHAR2,
   object_name     VARCHAR2,
   policy_name     VARCHAR2 );
```

object_schema:	The schema of the object to be audited. (If NULL, the current log-on user schema is assumed.)
object_name:	The name of the object to be audited
policy_name:	The unique name of the policy

- Vistas sobre la tabla SYS.AUD$
  - **AUDIT_ACTIONS:** Tabla con tipos de acción de la auditoría (acción, nombre)
  - **DBA_AUDIT_TRAIL:** Todo los registros
  - **DBA_AUDIT_OBJECT:** Todos los registros de los objetos auditables del sistema
  - **DBA_AUDIT_EXISTS:** Registros de auditoría producidos por AUDIT NOT EXISTS
  - **DBA_AUDIT_SESSION:** Registros relativos a conexiones y desconexiones
  - **DBA_AUDIT_STATEMENT:** Registros para las sentencias GRANT, REVOKE, AUDIT, NOAUDIT, y ALTER SYSTEM
  - **ALL_DEF_AUDIT_OPTS:** Opciones por defecto de auditoría de objetos que serán aplicadas al crearlos
  - **DBA_STMT_AUDIT_OPTS:** Opciones actuales de auditoría por sentencia
  - **DBA_PRIV_AUDIT_OPTS:** Privilegios de sistema auditados
  - **DBA_OBJ_AUDIT_OPTS:** Opciones de auditoría para todos los objetos
  - **DBA_FGA_AUDIT_TRAIL:** Registros de auditoría grano fino
  - **DBA_AUDIT_POLICIES:** Políticas de auditoría de grano fino en el sistema