

Laboratorio 8

• Auditoría en Oracle



Auditoría

En general una auditoria es un proceso de inspección, de gestión dentro del entorno de una base de datos, el cual nos permite llevar un control de modificaciones, accesos y permisos a nuestra base de datos. Entre los principales autores de la creación de auditoria le corresponde al administrador de la base de datos.

Las auditorias en Oracle tienen la característica de poder auditar tres tipos de acciones, las cuales son: intentos de inicio de sesión, accesos a objetos y modificaciones a la base de datos en sí.

Todas estas informaciones de las auditorias son almacenadas en un diccionario de datos en la tabla conocida como **SYS.AUD\$**.

Basta con hacer un **SELECT * FROM AUD\$;** para ver todos los registros de las auditorías realizadas al acceso y modificación de la base de datos.

Tipos de auditoría

Dependiendo las reglas de negocio y de las necesidades de control que tenga la empresa, Oracle posee 5 tipos de auditorias

Auditoria estándar: este tipo es útil al momento de auditar sentencias SQL, privilegios, objetos de algún esquema, actividades de red o multitier. El comando para este tipo de auditoria es **AUDIT**.

Auditoría basada en valores: este tipo se implementa cada vez que es cambiada algún valor de sentencias DML es ejecutada en todas o en determinadas filas.

Auditoria de grano fino: está en un nivel más granular, donde las acciones auditadas se capturan basándose en el contenido accedido o modificado. Normalmente suele ocuparse este tipo cuando alguien trata de realizar acciones que respondan a condiciones específicas en la definición de la política.

Auditorías SYS: permite el monitorio de la actividad de un administrador del sistema, por ejemplo, todos aquellos usuarios que tengan el privilegio de poder acceder como **SYS**, serán guardados en un archivo del sistema operativo para evitar que sean borrados del a tabla **AUD\$** dentro de la base de datos. El parámetro de inicialización de esta auditoria es: **AUDIT_SYS_OPERATIONS**.

Auditoria mandataria: siempre está habilitada, que monitorea las operaciones que involucren el startup y shutdown de la base de datos, también audita todos los usuarios

que llegasen a utilizar los roles predefinidos del sistema como **SYSDBA**, **SYSASM** O **SYSOPER**. Este tipo de auditorías utiliza la palabra: **POLICY**.

Para el caso de las auditorías que veremos en esta práctica de laboratorio el alumno debe ingresar como **SYSDBA** en SQL Plus.

Para mejor visibilidad en las consultas en SQL plus se recomienda **SET PAGESIZE 1200** y **SET LINESIZE 1200**

Utilizando auditoría estándar

Oracle no trae activada por defecto activada este tipo de auditoría, entonces será necesario activarla. Para ver el estado de la auditoría es necesario escribir el comando: **SHOW PARAMETERS AUDIT;**

NAME	TYPE	VALUE
audit_file_dest	string	C:\OLOGS
audit_sys_operations	boolean	FALSE
audit_trail	string	NONE

Entendamos un poco mejor que es cada uno de estos parámetros.

audit_trail_dest: Especifica el directorio del sistema donde la auditoría trial es escrita, cuando los valores de los parámetros están en **os**, **xml** o **xml extended** (en la práctica se explicará más detalladamente).

audit_sys_operations: Activa o desactiva las auditorías de las operaciones emitidas por usuarios **SYS**, al igual que los usuarios que se conectan con privilegios **SYSDBA** O **SYSOPER**, el valor por defecto es **FALSE**.

audit_trail: Se ingresan valores para habilitar o deshabilitar auditorías a la base de datos, su valor por defecto es **NONE** dando a entender que no habrá ninguna auditoría.

Un cambio de cualquiera de estos se considera que es a nivel de sistema, y se debe hacer con el comando:

ALTER SYSTEM SET <nombre_parametro>=<valor>

El alcance (**SCOPE**) es un parámetro que se usa junto con el comando de **ALTER SYSTEM** cuando está cambiando cualquier parámetro de inicialización de un archivo.

Existen tres definiciones para **SCOPE**:

SCOPE=MEMORY: ORACLE realizara el cambio especificado con **ALTER SYSTEM**, pero únicamente para la sesión activa, la próxima vez que se reinicie la base de datos, este cambio se revertiría al valor por defecto

SCOPE=SPFILE: El cambio realizado con **ALTER SYSTEM** tendrá efectos la próxima vez que se reinicie la base de datos, no afectando la sesión actual.

SCOPE=BOTH: Si se desea que el cambio realizado por **ALTER SYSTEM** se ejecute inmediatamente y sea guardado para futuros reinicios en la base de datos.

Para realizar reinicios en la base de datos será necesario ingresar:

```
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1603411968 bytes
Fixed Size                  2176168 bytes
Variable Size               587205464 bytes
Database Buffers            1006632960 bytes
Redo Buffers                 7397376 bytes
Database mounted.
Database opened.
SQL> connect
Enter user-name: sys as sysdba
Enter password:
```

SQL>SHUTDOWN IMMEDIATE: Apagamos y desmontamos la base de datos de manera correcta.

SQL>STARTUP: Inicializamos la base de datos con todas sus instancias, servicios y listeners.

SQL>CONNECT: Nos conectamos a algún usuario que este en la base de datos

Cambiar la dirección de `audit_file_dest` a una carpeta '**C:\LogAuditorias<carnet>**'

```
SQL> ALTER SYSTEM SET audit_file_dest= 'C:\LogAuditorias00006715' SCOPE=SPFILE;
System altered.
```

Ahora debemos habilitar la opción de `audit_sys_operation` para auditar las acciones de usuarios SYS.

```
SQL> ALTER SYSTEM SET audit_sys_operations=TRUE SCOPE=SPFILE;
System altered.
```

Habilitamos la auditoría estándar

```
SQL> ALTER SYSTEM SET audit_trail=db SCOPE=SPFILE;
System altered.
```

El parámetro `audit_trail` puede tener distintos valores los cuales son:

- **none:** Auditoria desactivada.
- **db:** Auditoria activada y guarda todos los registros en SYS.AUD\$ en la tablespace SYSTEM.
- **os:** Auditoria activada, guarda todos los registros en el sistema operativo (un directorio y archivos concretos). Esta opción es recomendada en sistemas de alta seguridad.
- **db, extended:** auditoria activada, guarda los registros en SYS.AUD\$ además escribe valores en las columnas SQLTEXT y SQLBIND en la tabla SYS.AUD\$.
- **xml:** los registros de la auditoria serán escritos en ficheros XML, en el archivo destinado en **audit_trail_dest**.
- **xml, extended:** los registros de la auditoria serán escritos en ficheros XML, además incluirán valores como SQLTEXT Y SQLBIND.

En nuestro caso se dejará el parámetro db.

Una vez finalizado, este debe ser el resultado:

```
SQL> show parameter audit;
```

NAME	TYPE	VALUE
audit_file_dest	string	C:\LOGAUDITORIAS00006715
audit_sys_operations	boolean	TRUE
audit_trail	string	DB

Uso del comando AUDIT y NOAUDIT

Antes de empezar de lleno con esta sección, cree un usuario que tendrá una cuota de 2M en el tablespace por defecto **SYSTEM**, y cuyo tablespace temporal será **TEMP**.

Sintaxis del comando:

```
AUDIT
{sql_statement_clause | schema_object_clause | NETWORK}
[BY {SESSION | ACCESS}]
[WHENEVER [NOT] SUCCESSFUL];
```

Este permite iniciar los tipos de auditoría que a continuación se detallan:

Auditorías de inicio de sesión: cada intento de conexión con la base de datos por parte de un usuario (bien una aplicación externa o las aplicaciones del propio Oracle) puede ser auditado.

```
SQL> AUDIT SESSION BY testbenji;

Audit succeeded.
```

Al usuario creado, darle privilegios para que pueda iniciar sesión, y en una nueva línea de comandos iniciar sesión con él.

Ahora, en la línea de comandos con la conexión del usuario SYS, se hará una consulta para ver la auditoría de sesiones de su usuario.

```
SQL> SELECT Username, userhost, extended_timestamp, action_name
2 FROM dba_audit_session
3 WHERE username='BENJITEST';
```

USERNAME	USERHOST	EXTENDED_TIMESTAMP	ACTION_NAME
TESTBENJI	WORKGROUP\LAPTOP-1JK4A8N4	02-NOV-18 09.14.29.178000 PM -06:00	LOGON
TESTBENJI	WORKGROUP\LAPTOP-1JK4A8N4	02-NOV-18 09.16.37.856000 PM -06:00	LOGOFF
TESTBENJI	WORKGROUP\LAPTOP-1JK4A8N4	02-NOV-18 09.20.20.440000 PM -06:00	LOGON
TESTBENJI	WORKGROUP\LAPTOP-1JK4A8N4	02-NOV-18 09.20.23.023000 PM -06:00	LOGOFF

Aquí se puede observar el usuario que ingreso, la fecha, la hora y la acción que realizó.

Si se quieren ver todas las auditorias hechas para ese usuario simplemente sería cambiar el **FROM** de la consulta anterior a **dba_audit_trail**.

Auditorías de acción: cualquier acción que afecte a un objeto de la base de datos puede auditarse. Estas acciones pueden ser, por ejemplo: **create**, **alter**, **drop**, etc.

```
SQL> AUDIT CREATE TABLE BY TESTBENJI;

Audit succeeded.
```

Hoy, en la sesión de su usuario, cree una tabla con cualquier atributo de cualquier tipo, pues únicamente servirá de prueba.

Nuevamente, en el usuario SYS, realice la siguiente consulta

```
Connected.
SQL> SELECT username,owner,obj_name,action_name,priv_used,extended_timestamp
2 FROM dba_audit_object
3 WHERE username='TESTBENJI';
```

USERNAME	OWNER	ACTION_NAME	OBJ_NAME	PRIV_USED	EXTENDED_TIMESTAMP
TESTBENJI	TESTBENJI	CREATE TABLE	TEST	CREATE TABLE	02-NOV-18 09.50.32.823000 PM -06:00

Auditorías de 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** 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 que puede tomar alguno de los siguientes valores:

- **BY SESSION:** el registro de auditoría se escribirá una única vez por sesión
- **BY ACCESS:** el registro de auditoría se escribirá cada vez que se acceda al objeto auditado

Para la práctica se ocupará **BY ACCESS** ya que es una buena práctica que, por ejemplo, siempre que se haga un **SELECT** o un **INSERT** quede registrado. Sin embargo, hay que tener en cuenta que este puede afectar el rendimiento.

```
SQL> AUDIT SELECT TABLE, UPDATE TABLE, INSERT TABLE, DELETE TABLE BY TESTBENJI BY ACCESS;
Audit succeeded.
```

Desde su usuario, agregue, modifique y elimine filas, para que quede registro de las instrucciones.

Consulte el registro:

```
SQL> SELECT username, extended_timestamp, action_name, comment_text, priv_used
2 FROM dba_audit_trail
3 WHERE username='TESTBENJI';
```

```
TESTBENJI                                02-NOV-18 10.21.19.426000 PM -06:00
SELECT
```

```
TESTBENJI                                02-NOV-18 10.23.03.550000 PM -06:00
UPDATE
```

```
TESTBENJI                                02-NOV-18 10.23.03.550000 PM -06:00
DELETE
```

Para hacer una limpieza de los datos registrados, se debe ejecutar la siguiente consulta:

```
DELETE FROM SYS.AUD$;
```

Auditoría XML

El primer paso, es necesario modificar el valor del parámetro **audit_trail** a XML.

NAME	TYPE	VALUE
audit_file_dest	string	C:\LOGAUDITORIAS00006715
audit_sys_operations	boolean	TRUE
audit_trail	string	XML

Para probar este formato, se hará un nuevo audit cuando cualquier usuario realice una consulta a la tabla creada previamente.

```
SQL> audit select on TESTBENJI.test;
Audit succeeded.
```

Ahora, en cualquier usuario, realizar un select a dicha tabla. Una vez completada la consulta, dirigirse al directorio donde se especificó que se guardarán los archivos.

auditorias00006715

Compartir Vista

Guardar Cortar Copiar ruta de acceso Pegar acceso directo

Mover Copiar Eliminar Cambiar nombre Nueva carpeta

Nuevo elemento Fácil acceso Propiedades Abrir Seleccionar No seleccionar Invertir

Este equipo > Acer (C:) > LogAuditorias00006715

Nombre	Fecha de modifica...	Tipo	Tamaño
adx_orcl0067.txt	02/11/2018 23:05	Documento de tex	1 KB
orcl0067_ora_1420_1.xml	02/11/2018 23:00	Documento XML	1 KB
orcl0067_ora_6860_1.xml	02/11/2018 23:00	Documento XML	1 KB
orcl0067_ora_8068_1.xml	02/11/2018 22:55	Documento XML	1 KB
orcl0067_ora_8876_1.xml	02/11/2018 23:00	Documento XML	1 KB
orcl0067_ora_9124_1.xml	02/11/2018 23:05	Documento XML	1 KB
orcl0067_ora_10056_1.xml	02/11/2018 22:55	Documento XML	3 KB
orcl0067_ora_10932_1.xml	02/11/2018 22:55	Documento XML	2 KB
orcl0067_ora_11524_1.xml	02/11/2018 23:00	Documento XML	3 KB
orcl0067_ora_11584_1.xml	02/11/2018 23:00	Documento XML	1 KB
orcl0067_ora_14868_1.xml	02/11/2018 23:02	Documento XML	3 KB

ace

FORI

Como se puede observar hay muchos archivos XML, a pesar que se hizo únicamente la auditoria al momento de hacer el SELECT, estos demás archivos XML son generados gracias a las **auditorias mandatorias** las cuales son los registros de todas las operaciones que realiza en este caso el usuario **SYS** con privilegios **SYSDBA**.

```
DBID>
<Sql_Text>audit select on TESTBENJI.test</Sql_Text>
</AuditRecord>
<AuditRecord><Audit_Type>4</Audit_Type><Session_Id>4294967295</Session_Id><StatementId>2</StatementId><EntryId>5</EntryId><Extended_Timestamp>2018-11-03T05:02:37.008000Z</Extended_Timestamp><DB_User></DB_User><Ext_Name>LAPTOP-1JK4A8N4\darkg</Ext_Name><OS_User>LAPTOP-1JK4A8N4\darkg</OS_User><Userhost>WORKGROUP\LAPTOP-1JK4A8N4</Userhost><OS_Process>11220:14868</OS_Process><Instance_Number>0</Instance_Number><Returncode>0</Returncode><OSPrivilege>SYSDBA</OSPrivilege><DBID>3603998107</DBID>
<Sql_Text>select*from TESTBENJI.test</Sql_Text>
</AuditRecord>
```

En el último archivo se encuentra la consulta realizada a la tabla, dando por entendido que la auditoría fue un éxito.

Auditorías de conexión a la base de datos

Creación de la tabla auditoría, donde se guardarán los registros de inicio de la base de datos.

```
SQL> CREATE TABLE startup_audit (  
2     Event_type VARCHAR2(15),  
3     event_date DATE,  
4     event_time VARCHAR2(15),  
5     Usuario varchar(50),  
6     NombreBD varchar2(20)  
7 );
```

A continuación, se crearán dos triggers que llenen la tabla creada con la información correspondiente.

```
SQL> CREATE OR REPLACE TRIGGER inicio_audit  
2  AFTER STARTUP  ON DATABASE  
3  BEGIN  
4  INSERT INTO startup_audit VALUES  
5  (  
6  ora_sysevent,  
7  SYSDATE,  
8  TO_CHAR(sysdate, 'hh24:mm:ss'),  
9  ora_login_user,  
10 ora_database_name  
11 );  
12 END;  
13  
14 /
```



```
SQL> CREATE OR REPLACE TRIGGER shutdown_audit
  2  BEFORE SHUTDOWN  ON DATABASE
  3  BEGIN
  4    INSERT INTO startup_audit VALUES
  5    (
  6      ora_sysevent,
  7      SYSDATE,
  8      TO_CHAR(sysdate, 'hh24:mm:ss'),
  9      ora_login_user,
10      ora_database_name
11    );
12  END;
13  /
```

Luego de conectarse y desconectarse en la base de datos, al hacer la consulta a la tabla auditoría, se puede ver la operación que realizó el usuario y en qué base de datos la hizo.

```
connected.
SQL> select*from startup_audit;
```

EVENT_TYPE	EVENT_DAT	EVENT_TIME	
USUARIO			NOMBREBD
LOGOFF	02-NOV-18	23:11:45	
SYS			ORCL0067
LOGON	02-NOV-18	23:11:00	
SYS			ORCL0067