

PRACTICAS TEMA 9.

AUDITORÍA.

1. Ver la descripción de las vistas ALL_DEF_AUDIT_OPTS, DBA_AUDIT_OBJECT, DBA_AUDIT_SESSION, DBA_AUDIT_STATEMENT, DBA_AUDIT_TRAIL, DBA_OBJ_AUDIT_OPTS, DBA_PRIV_AUDIT_OPTS, y DBA_STMT_AUDIT_OPTS. Averiguar el significado de cada uno de los campos usando la documentación en línea.

SQL> desc ALL_DEF_AUDIT_OPTS

Nombre	¿Nulo?	Tipo
ALT		VARCHAR2(3)
AUD		VARCHAR2(3)
COM		VARCHAR2(3)
DEL		VARCHAR2(3)
GRA		VARCHAR2(3)
IND		VARCHAR2(3)
INS		VARCHAR2(3)
LOC		VARCHAR2(3)
REN		VARCHAR2(3)
SEL		VARCHAR2(3)
UPD		VARCHAR2(3)
REF		VARCHAR2(3)
EXE		VARCHAR2(3)
...		

2. Ver la descripción de la vista SYS.AUD\$.

SQL> desc sys.aud\$

Nombre	¿Nulo?	Tipo
SESSIONID	NOT NULL	NUMBER
ENTRYID	NOT NULL	NUMBER
STATEMENT	NOT NULL	NUMBER
TIMESTAMP#		DATE
USERID		VARCHAR2(30)
USERHOST		VARCHAR2(128)
TERMINAL		VARCHAR2(255)

<i>ACTION#</i>	<i>NOT NULL</i>	<i>NUMBER</i>
<i>RETURNCODE</i>	<i>NOT NULL</i>	<i>NUMBER</i>
<i>OBJ\$CREATOR</i>		<i>VARCHAR2(30)</i>
<i>OBJ\$NAME</i>		<i>VARCHAR2(128)</i>
<i>AUTH\$PRIVILEGES</i>		<i>VARCHAR2(16)</i>
<i>AUTH\$GRANTEE</i>		<i>VARCHAR2(30)</i>
<i>NEW\$OWNER</i>		<i>VARCHAR2(30)</i>
<i>NEW\$NAME</i>		<i>VARCHAR2(128)</i>
<i>SES\$ACTIONS</i>		<i>VARCHAR2(19)</i>
<i>SES\$TID</i>		<i>NUMBER</i>
<i>LOGOFF\$LREAD</i>		<i>NUMBER</i>
<i>LOGOFF\$PREAD</i>		<i>NUMBER</i>
<i>LOGOFF\$LWRITE</i>		<i>NUMBER</i>
<i>LOGOFF\$DEAD</i>		<i>NUMBER</i>
<i>LOGOFF\$TIME</i>		<i>DATE</i>
<i>COMMENT\$TEXT</i>		<i>VARCHAR2(4000)</i>
<i>CLIENTID</i>		<i>VARCHAR2(64)</i>
<i>SPARE1</i>		<i>VARCHAR2(255)</i>
<i>SPARE2</i>		<i>NUMBER</i>
<i>OBJ\$LABEL</i>		<i>RAW(255)</i>
<i>SES\$LABEL</i>		<i>RAW(255)</i>
<i>PRIV\$USED</i>		<i>NUMBER</i>
<i>SESSIONCPU</i>		<i>NUMBER</i>
<i>NTIMESTAMP#</i>		<i>TIMESTAMP(6)</i>
<i>PROXY\$SID</i>		<i>NUMBER</i>
<i>USER\$GUID</i>		<i>VARCHAR2(32)</i>
<i>INSTANCE#</i>		<i>NUMBER</i>
<i>PROCESS#</i>		<i>VARCHAR2(16)</i>
<i>XID</i>		<i>RAW(8)</i>
<i>AUDITID</i>		<i>VARCHAR2(64)</i>
<i>SCN</i>		<i>NUMBER</i>
<i>DBID</i>		<i>NUMBER</i>
<i>SQLBIND</i>		<i>CLOB</i>
<i>SQLTEXT</i>		<i>CLOB</i>

3. Habilitar la auditoría de base de datos modificando convenientemente el fichero de parámetros de inicialización de forma que se registren los apuntes de auditoría en base de datos.

Para comenzar a auditar debe modificarse el fichero de parámetros de inicialización, añadiendo el parámetro AUDIT_TRAIL y asignándole el valor DB.

4. Comprobar que usuarios de base de datos tienen asignados los privilegios AUDIT SYSTEM y AUDIT ANY.

SQL> connect system

Enter password:

Connected.

SQL> select * from dba_sys_privs where privilege ='AUDIT ANY';

GRANTEE	PRIVILEGE	ADM
DBA	AUDIT ANY	YES
IMP_FULL_DATABASE	AUDIT ANY	NO

SQL> select * from dba_sys_privs where privilege ='AUDIT SYSTEM';

GRANTEE	PRIVILEGE	ADM
DBA	AUDIT SYSTEM	YES

5. Auditar todas las conexiones exitosas y fallidas a la base de datos.

SQL> audit session;

Auditoria terminada correctamente.

6. Abrir una sesión a la base de datos como usuario "SYSTEM" y como usuario "prueba01". Comprobar que información se ha guardado en la auditoría como consecuencia de las operaciones anteriores (consultar la vista DBA_AUDIT_TRAIL).

SQL> select substr(os_username,1,10) usuario_so, substr(username,1,12) usuario, to_char(timestamp,'dd-mm-yyy hh24:mi:ss') tiempo_conexion, to_char(logoff_time,'dd-mm-yyyy hh24:mi:ss') tiempo_desconexion from dba_audit_trail order by username,timestamp,logoff_time;

USUARIO_SO	USUARIO	TIEMPO_CONEXION	TIEMPO_DESCONEXION
oracle	PRUEBA01	15-10-006 10:04:08	
oracle	SYSTEM	15-10-006 10:04:24	

7. Auditar las conexiones exitosas y fallidas a la base de datos de los usuarios “prueba01” y “prueba02”.

```
SQL> audit session by prueba01, prueba02;
```

Auditoria terminada correctamente.

8. Conectarse a la base de datos como usuario “prueba02” introduciendo una clave errónea. Comprobar el apunte realizado en la auditoría del sistema (consultar la vista DBA_AUDIT_SESSION).

```
SQL> select substr(os_username,1,10) usuario_so, substr(username,1,12)
usuario, to_char(timestamp,'dd-mm-yyy hh24:mi:ss') tiempo_conexion,
to_char(logoff_time,'dd-mm-yyyy hh24:mi:ss') tiempo_desconexion from
dba_audit_session where username='PRUEBA02' order by
username,timestamp,logoff_time
```

USUARIO_SO	USUARIO	TIEMPO_CONEXION	TIEMPO_DESCONEXION
oracle	PRUEBA02	15-10-006 10:08:43	

9. Auditar el uso de SELECT TABLE, DELETE TABLE, UPDATE TABLE e INSERT TABLE en cualquier caso (fallido o no) y por sentencia individual. Comprobar las opciones activas de auditoría (vista DBA_STMT_AUDIT_OPTS).

```
SQL> audit select table, delete table, update table, insert table by access;
```

Auditoria terminada correctamente.

```
SQL> select user_name usuario, audit_option opcion, success, failure from
DBA_STMT_AUDIT_OPTS order by user_name;
```

USUARIO	OPCION	SUCCESS	FAILURE
PRUEBA01	CREATE SESSION	BY ACCESS	BY ACCESS
PRUEBA02	CREATE SESSION	BY ACCESS	BY ACCESS
	INSERT TABLE	BY ACCESS	BY ACCESS
	DELETE TABLE	BY ACCESS	BY ACCESS
	SELECT TABLE	BY ACCESS	BY ACCESS

```
CREATE SESSION BY ACCESS BY ACCESS
UPDATE TABLE BY ACCESS BY ACCESS
```

7 filas seleccionadas.

10. Conectarse como usuario "prueba01". Comprobar si se ha creado la tabla "codigos_notas" en el "tablespace" ACADEMO, en caso contrario crearla.

```
CREATE TABLE CODIGOS_NOTAS
(CODIGO varchar2(3),
DESCRIPCION varchar2(20))
TABLESPACE ACADEMO
STORAGE (INITIAL 64K
NEXT 64K
MINEXTENTS 3
MAXEXTENTS 10);
```

Otorgar privilegios para consultar, insertar, actualizar y borrar registros de la tabla "codigos_notas" al usuario "prueba02".

Conectarse como usuario "prueba02" y realizar las siguientes operaciones:

- Seleccionar todos los registros de la tabla "codigos_notas".
- Insertar en la tabla "codigos_notas" los siguientes registros:

```
'0', 'NO PRESENTADO'
'1', 'APROBADO'
```

Consultar los registros de auditoría generados para las operaciones anteriores (vista DBA_AUDIT_OBJECT).

```
SQL> connect prueba01
```

```
Enter password:
```

```
Conectado.
```

```
SQL> CREATE TABLE CODIGOS_NOTAS
```

```
2 (CODIGO varchar2(3),
3 DESCRIPCION varchar2(20))
4 TABLESPACE ACADEMO
5 STORAGE (INITIAL 64K
6 NEXT 64K
7 MINEXTENTS 3
8* MAXEXTENTS 10)
```

Tabla creada.

```
SQL> grant select, insert, update, delete on codigos_notas to prueba02;
```

Grant succeeded.

```
SQL> connect prueba02
```

Enter password:

Conectado.

```
SQL> select * from prueba01.codigos_notas;
```

ninguna fila seleccionada

```
SQL> insert into prueba01.codigos_notas values ('0','NO PRESENTADO');
```

1 fila creada.

```
SQL> insert into prueba01.codigos_notas values ('1','APROBADO');
```

1 fila creada.

```
SQL> commit;
```

Confirmacion terminada.

```
SQL> connect system
```

Enter password:

Conectado.

```
SQL> select substr(os_username,1,10) usuario_so, substr(username,1,12) usuario,
to_char(timestamp,'dd-mm-yyy hh24:mi:ss') tiempo_conexion,
to_char(logoff_time,'dd-mm-yyyy hh24:mi:ss') tiempo_desconexion from
dba_audit_session where username='PRUEBA02' order by
username,timestamp,logoff_time;
```

USUARIO_SO	USUARIO	TIEMPO_CONEXION	TIEMPO_DESCONEXION
------------	---------	-----------------	--------------------

oracle	PRUEBA02	15-10-006 10:08:43	
oracle	PRUEBA02	15-10-006 10:24:43	15-10-2006 10:26:52

```
SQL> select substr(os_username,1,10) usuario_so, substr(username,1,12) usuario,
to_char(timestamp,'dd-mm-yyy hh24:mi:ss') tiempo_conexion, substr(owner,1,10)
propie, substr(obj_name,1,15) objeto, substr(action_name,1,10) accion from
```

dba_audit_object where username='PRUEBA02' order by os_username,timestamp, owner,obj_name,action_name;

USUARIO_SO	USUARIO	TIEMPO_CONEXION	PROPIE	OBJETO	ACCION
oracle	PRUEBA02	15-10-006 10:24:43	SYS	DUAL	SELECT
oracle	PRUEBA02	15-10-006 10:24:43	SYS	DUAL	SELECT
oracle	PRUEBA02	15-10-006 10:24:43	SYSTEM	PRODUCT_PRIVS	SELECT
oracle	PRUEBA02	15-10-006 10:24:43	SYSTEM	PRODUCT_PRIVS	SELECT
oracle	PRUEBA02	15-10-006 10:24:43	SYSTEM	SQLPLUS_PRODUCT	SELECT
oracle	PRUEBA02	15-10-006 10:24:43	SYSTEM	SQLPLUS_PRODUCT	SELECT
oracle	PRUEBA02	15-10-006 10:25:17	PRUEBA01	CODIGOS_NOTAS	SELECT
oracle	PRUEBA02	15-10-006 10:25:31	PRUEBA01	CODIGOS_NOTAS	INSERT
oracle	PRUEBA02	15-10-006 10:25:41	PRUEBA01	CODIGOS_NOTAS	INSERT

9 filas seleccionadas.

11.Consultar las opciones por defecto de auditoría de objetos (ALL_DEF_AUDIT_OPTS).

SQL> select * from all_def_audit_opts;

ALT	AUD	COM	DEL	GRA	IND	INS	LOC	REN	SEL	UPD	REF	EXE	FBK	REA
-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-

12.Especificar las opciones de auditoría por defecto para los objetos creados en un futuro de forma que se registre información siempre que se produzca un “alter”, “grant”, “insert”, “update” o “delete”.

SQL> audit alter, grant, insert, update, delete on default;

Auditoria terminada correctamente.

SQL> select * from all_def_audit_opts;

ALT	AUD	COM	DEL	GRA	IND	INS	LOC	REN	SEL	UPD	REF	EXE	FBK	REA
S/S	-/-	-/-	S/S	S/S	-/-	S/S	-/-	-/-	-/-	S/S	-/-	-/-	-/-	-/-

El significado de cada una de las columnas es:

-/-: No auditoria por defecto.

S/-: Auditado cuando sea exitosa la operación.

-/S: Auditado cuando sea fallida la operación.

ALT ... ALTER
 AUD ... AUDIT
 COM ... COMMENT
 DEL ... DELETE
 GRA ... GRANT
 IND ... INDEX
 INS ... INSERT
 LOC ... LOCK
 REN ... RENAME
 SEL ... SELECT
 UPD ... UPDATE
 REF ... REFERENCES
 EXE ... EXECUTE

13.Desactivar la auditoría de las conexiones a la base de datos.

SQL> select substr(user_name,1,12) usuario, privilege, success,failure from dba_priv_audit_opts order by user_name, privilege;

USUARIO	PRIVILEGE	SUCCESS	FAILURE
PRUEBA01	CREATE SESSION	BY ACCESS	BY ACCESS
PRUEBA02	CREATE SESSION	BY ACCESS	BY ACCESS
	CREATE SESSION	BY ACCESS	BY ACCESS

SQL> noaudit session;

No auditoria terminada correctamente.

SQL> select substr(user_name,1,12) usuario, privilege, success,failure from dba_priv_audit_opts order by user_name, privilege;

USUARIO	PRIVILEGE	SUCCESS	FAILURE
PRUEBA01	CREATE SESSION	BY ACCESS	BY ACCESS
PRUEBA02	CREATE SESSION	BY ACCESS	BY ACCESS

SQL> noaudit session by prueba01, prueba02;

No auditoria terminada correctamente.

SQL> select substr(user_name,1,12) usuario, privilege, success,failure from dba_priv_audit_opts order by user_name, privilege;

ninguna fila seleccionada

14.Desactivar la auditoría de uso de SELECT TABLE, DELETE TABLE, UPDATE TABLE e INSERT TABLE (activada en ejercicios anteriores).

SQL> select substr(user_name,1,12) usuario, substr(audit_option, 1,15) opcion, success,failure from dba_stmt_audit_opts order by user_name, audit_option;

USUARIO	OPCION	SUCCESS	FAILURE
	DELETE TABLE	BY ACCESS	BY ACCESS
	INSERT TABLE	BY ACCESS	BY ACCESS
	SELECT TABLE	BY ACCESS	BY ACCESS
	UPDATE TABLE	BY ACCESS	BY ACCESS

SQL> noaudit select table, delete table, update table, insert table;

No auditoria terminada correctamente.

SQL> select substr(user_name,1,12) usuario, substr(audit_option, 1,15) opcion, success,failure from dba_stmt_audit_opts order by user_name, audit_option;

ninguna fila seleccionada

15.Activar la auditoría para cualquier operación de consulta, inserción o borrado que se efectúe sobre la tabla “prueba01.codigos_notas”.

SQL> audit select, insert, delete on prueba01.codigos_notas;

Auditoria terminada correctamente.

```
SQL> select substr(owner,1,12),substr(OBJECT_NAME,1,12),
substr(object_type,1,10), DEL, INS, SEL, UPD from dba_obj_audit_opts where
owner='PRUEBA01' order by object_name, object_type;
```

SUBSTR(OWNER	SUBSTR(OBJEC	SUBSTR(OBJ	DEL	INS	SEL	UPD
PRUEBA01	CODIGOS_NOTA	TABLE	S/S	S/S	S/S	-/-

16. Conectado como usuario “prueba01” realizar una consulta del contenido de la tabla “codigos_notas”. Consultar los registros de auditoría generados (DBA_AUDIT_OBJECT).

```
SQL> connect prueba01
Introduzca su clave:
Conectado.
```

```
SQL> select * from codigos_notas;
```

COD	DESCRIPCION
0	NO PRESENTADO
1	APROBADO

```
SQL> insert into codigos_notas values ('3','NOTABLE');
```

1 fila creada.

```
SQL> commit;
```

Confirmacion terminada.

```
SQL> connect system
Introduzca su clave:
Conectado.
```

```
SQL> select substr(username,1,10) usuario, to_char(timestamp,'dd-mm-yyyy
hh24:mi:ss') fecha, substr(owner,1,10) prop, substr(obj_name,1,15) objeto,
ses_actions from dba_audit_object where owner='PRUEBA01' order by timestamp;
```

USUARIO	FECHA	PROP	OBJETO	SES_ACTIONS
PRUEBA02	15-10-2006 10:25:17	PRUEBA01	CODIGOS_NOTAS	
PRUEBA02	15-10-2006 10:25:31	PRUEBA01	CODIGOS_NOTAS	
PRUEBA02	15-10-2006 10:25:41	PRUEBA01	CODIGOS_NOTAS	
PRUEBA01	15-10-2006 11:03:58	PRUEBA01	CODIGOS_NOTAS	-----S--S-----

SES_ACTIONS Resumen de sesion, una cadena de 16 caracteres, uno por cada tipo de accion de la lista ordenada ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, y EXECUTE.

Las posiciones 14, 15, y 16 estan reservadas para usos futuros. Los caracteres son - para ninguno, S para éxito, F para fallo y B para ambos.

17. Desactivar la auditoría del ejercicio anterior.

```
SQL> noaudit select, insert, delete on prueba01.codigos_notas;
```

No auditoria terminada correctamente.

18. Desactivar todas las opciones de auditoría de objetos que existen por defecto.

```
SQL> noaudit all on default;
```

No auditoria terminada correctamente.

19. Comprobar las características de almacenamiento del “audit trail”. Borrar todos los registros de auditoría correspondientes a la tabla “prueba01.codigos_notas”.

```
SQL> select substr(owner,1,10) prop, substr(table_name,1,8)
tabla, substr(.tablespace_name,1,10) espacio, INITIAL_EXTENT,
NEXT_EXTENT, MIN_EXTENTS, MAX_EXTENTS from dba_tables where
table_name='AUD$';
```

PROP	TABLA	ESPACIO	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS
SYS	AUD\$	SYSTEM	65536	1		2147483645

20. Conectarse como usuario SCOTT a la bd y realizar una actualización de la tabla empleados, emp, modificando el salario, columna sal, de forma que el empleado número 7369 pase a cobrar 3000 euros. Comprobar qué apunte se ha generado en la vista FLASHBACK_TRANSACTION_QUERY.

SQL> connect scott

Introduzca la contraseña:

Conectado.

SQL> select empno,ename,sal from scott.emp;

EMPNO	ENAME	SAL
7369	SMITH	800
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7876	ADAMS	1100
7902	FORD	3000
7934	MILLER	1300

9 filas seleccionadas.

SQL> update emp set sal=3000 where empno=7369;

1 fila actualizada.

SQL> commit;

Confirmacion terminada.

SQL> connect / as sysdba;

Conectado.

La información en la vista FLASHBACK_TRANSACTION_QUERY incluye:

- *XID ... Identificador de transacción*
- *START_SCN NUMBER ... Número SCN de inicio de la transacción (system change number)*
- *START_TIMESTAMP DATE ... Momento de inicio de la transacción*
- *COMMIT_SCN NUMBER ... SCN de validación de la transacción (nulo para aquellas que estén activas)*

- *COMMIT_TIMESTAMP DATE ... Tiempo de validación de la transacción (nulo para aquellas que estén activas)*
- *LOGON_USER ... Usuario conectado*
- *UNDO_CHANGE# ... Undo SCN (1 o superior)*
- *OPERATION ... Operación realizada (D - Delete, I - Insert, U - Update o B - desconocida)*
- *TABLE_NAME ... Nombre de la tabla a la que se le aplica la operación DML*
- *TABLE_OWNER ... Propietario de la tabla*
- *ROW_ID ... Rowid de la fila modificada*
- *UNDO_SQL ... SQL para deshacer la operación DML indicada por OPERATION*

*SQL> select * from flashback_transaction_query where table_owner='SCOTT' order by start_scn;*

```

XID          START_SCN START_TI COMMIT_SCN COMMIT_T
-----
LOGON_USER          UNDO_CHANGE# OPERATION
-----
TABLE_NAME
-----
TABLE_OWNER          ROW_ID
-----
UNDO_SQL
-----
010023004D090000  4993228 22/10/06  4993235 22/10/06
SCOTT                1 UPDATE
EMP
SCOTT                AAAEtaAAEAAAAAgAAA
update "SCOTT"."EMP" set "SAL" = '800' where ROWID = 'AAAEtaAAEAAAAAgAAA';
    
```

21. Conectarse nuevamente como usuario SCOTT a la bd y realizar una actualización de la tabla empleados, emp, modificando el salario, columna sal, de forma que el empleado número 7902 pase a cobrar 20000 euros. No realizar la validación, “commit”. Comprobar -DESDE OTRA SESIÓN- si se ha modificado el valor de la tabla emp y si se ha generado algún apunte en la vista FLASHBACK_TRANSACTION_QUERY.

Desde la sesión donde se ha modificado el valor de la tabla emp, realizar “rollback”. Comprobar el valor de la tabla emp y si se ha generado algún apunte en la vista FLASHBACK_TRANSACTION_QUERY. ¿Qué ha sucedido?.

SQL> connect scott
Introduzca la contraseña:
Conectado.

SQL> select * from emp where empno=7902;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7902	FORD	ANALYST	7566	03/12/81	3000	

SQL> update emp set sal=20000 where empno=7902;

1 fila actualizada.

SQL> select * from emp where empno=7902;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7902	FORD	ANALYST	7566	03/12/81	20000	

Se abre una nueva sesión. Observamos que no se ha realizado el "commit".

SQL> connect / as sysdba;
Conectado.

SQL> select * from flashback_transaction_query where table_owner='SCOTT' order by start_scn;

010023004D090000	4993228	22/10/06	4993235	22/10/06		
SCOTT		1 UPDATE				
EMP						
SCOTT		AAAEtaAAEAAAAAgAAA				
update "SCOTT"."EMP" set "SAL" = '800' where ROWID = 'AAAEtaAAEAAAAAgAAA';						
060021004D080000	4993322	22/10/06		22/10/06		
SCOTT		1 UPDATE				
EMP						
SCOTT		AAAEtaAAEAAAAAgAAM				
update "SCOTT"."EMP" set "SAL" = '3000' where ROWID = 'AAAEtaAAEAAAAAgAAM';						

Desde la sesión donde se ha modificado el valor de la tabla se realiza un "rollback".

SQL> rollback;

Rollback terminado.

SQL> select * from emp where empno=7902;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7902	FORD	ANALYST	7566	03/12/81	3000	

Se comprueba el contenido de FLASHBACK_TRANSACTION_QUERY.

SQL> connect / as sysdba;
Conectado.

SQL> select * from flashback_transaction_query where table_owner='SCOTT' order by start_scn;

XID	START_SCN	START_TI	COMMIT_SCN	COMMIT_T
010023004D090000	4993228	22/10/06	4993235	22/10/06
SCOTT		1	UPDATE	
EMP				
SCOTT			AAAEtaAAEAAAAgAAA	
update "SCOTT"."EMP" set "SAL" = '800' where ROWID = 'AAAEtaAAEAAAAgAAA';				

22. Conectarse como usuario SCOTT a la bd y restaurar en la tabla empleados, emp, el valor original para el salario, columna sal, del empleado número 7369. Comprobar qué apunte se ha generado en la vista FLASHBACK_TRANSACTION_QUERY.

```
SQL> connect / as sysdba;
Conectado.
```

```
SQL> select * from flashback_transaction_query where table_owner='SCOTT'
order by start_scn;
```

```

XID          START_SCN START_TI COMMIT_SCN COMMIT_T
-----
LOGON_USER          UNDO_CHANGE# OPERATION
-----
TABLE_NAME
-----
TABLE_OWNER          ROW_ID
-----
UNDO_SQL
-----
010023004D090000  4993228 22/10/06  4993235 22/10/06
SCOTT                      1 UPDATE
EMP
SCOTT                      AAAEtaAAEAAAAAgAAA
update "SCOTT"."EMP" set "SAL" = '800' where ROWID = 'AAAEtaAAEAAAAAgAAA';

```

```
SQL> connect scott
Introduzca la contraseña:
Conectado.
```

```
SQL> update "SCOTT"."EMP" set "SAL" = '800' where ROWID =
'AAAEtaAAEAAAAAgAAA';
```

1 fila actualizada.

```
SQL> commit;
```

Confirmacion terminada.

```
SQL> select * from emp where empno=7369;
```

```

EMPNO ENAME  JOB          MGR HIREDATE          SAL  COMM
-----

```

DEPTNO

7369 SMITH CLERK 7902 17/12/80 800
20

SQL> select * from flashback_transaction_query where table_owner='SCOTT'
order by start_scn;

XID START_SCN START_TI COMMIT_SCN COMMIT_T

LOGON_USER UNDO_CHANGE# OPERATION

TABLE_NAME

TABLE_OWNER ROW_ID

UNDO_SQL

010023004D090000 4993228 22/10/06 4993235 22/10/06

SCOTT 1 UPDATE

EMP

SCOTT AAAEtAaAEAAAAAgAAA

update "SCOTT"."EMP" set "SAL" = '800' where ROWID = 'AAAEtaAAEAAAAAgAAA';

03000000030A0000 4993413 22/10/06 21/10/06

SCOTT 1 UPDATE

EMP

SCOTT AAAEtAaAEAAAAAgAAA

update "SCOTT"."EMP" set "SAL" = '3000' where ROWID =
'AAAEtaAAEAAAAAgAAA';

23. En el ejercicio se modifican distintos valores para la columna "sal" de la tabla "scott.emp", recuperando versiones anteriores haciendo uso de la característica "version query". Comprobar los apuntes generados en la tabla y los cambios producidos.

SQL> select * from emp;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

7369 SMITH CLERK 7902 17/12/80 800 20

7566 JONES MANAGER 7839 02/04/81 2975 20

7698 BLAKE MANAGER 7839 01/05/81 2850 30

```

7782 CLARK    MANAGER  7839    09/06/81  2450      10
7788 SCOTT    ANALYST  7566    19/04/87  3000      20
7839 KING     PRESIDENT 17/11/81  5000      10
7876 ADAMS    CLERK    7788    23/05/87  1290      20
7902 FORD     ANALYST  7566    03/12/81  3000      20
7934 MILLER   CLERK    7782    23/01/82  1300      10
    
```

9 filas seleccionadas.

Se realiza una modificación que afecta a varias filas.

```
SQL> update emp set sal=6500 where job='PRESIDENT' or job='ANALYST';
```

3 filas actualizadas.

```
SQL> select * from emp where job='PRESIDENT' or job='ANALYST';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	19/04/87	6500		20
7839	KING	PRESIDENT		17/11/81	6500		10
7902	FORD	ANALYST	7566	03/12/81	6500		20

Se consulta el valor anterior para la filas modificada. Hace 10 minutos y hace 5 segundos.

```
SQL> select empno,job,sal from emp AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '10' MINUTE ) where job='PRESIDENT' or job='ANALYST';
```

EMPNO	JOB	SAL
7788	ANALYST	3000
7839	PRESIDENT	5000
7902	ANALYST	3000

```
SQL> select empno,job,sal from emp AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '5' SECOND) where job='PRESIDENT' or job='ANALYST';
```

EMPNO	JOB	SAL
7788	ANALYST	6500

```
7839 PRESIDENT    6500
7902 ANALYST      6500
```

Se asigna el valor que tenía la columna “sal” para el presidente de la compañía hace 10 minutos.

```
SQL> update emp set sal = (select sal from emp AS OF TIMESTAMP
(SYSTIMESTAMP - INTERVAL '10' MINUTE ) where job='PRESIDENT') where
job='PRESIDENT';
```

1 fila actualizada.

```
SQL> select sal from emp where job='PRESIDENT';
```

```
SAL
-----
5000
```

24. Consultar el rango de valores para las filas modificadas en el ejercicio número 22. Averiguar la transacción que generó las modificaciones y la operación que se realizó.

```
SQL> SELECT versions_startscn, versions_starttime, versions_endscn,
versions_endtime, versions_xid, versions_operation ,empno,job,sal from emp
versions between timestamp (SYSTIMESTAMP -INTERVAL '15' MINUTE ) and
(SYSTIMESTAMP - INTERVAL '5' SECOND) where job='PRESIDENT' or job='ANALYST'
order by versions_starttime;
```

```
VERSIONS_STARTSCN VERSIONS_STARTTIME
-----
VERSIONS_ENDSCN VERSIONS_ENDTIME
-----
VERSIONS_XID  V  EMPNO JOB      SAL
-----
          5285781 29/10/06 15:58:54
08002C00AD070000 U    7902 ANALYST    6500

          5285781 29/10/06 15:58:54
08002C00AD070000 U    7839 PRESIDENT  6500

          5285781 29/10/06 15:58:54
08002C00AD070000 U    7788 ANALYST    6500
```

```

5285781 29/10/06 15:58:54
          7788 ANALYST      3000

5285781 29/10/06 15:58:54
          7839 PRESIDENT   5000

5285781 29/10/06 15:58:54
          7902 ANALYST      3000
    
```

6 filas seleccionadas.

Puede obtenerse información adicional acerca de la transacción implicada. El valor de la columna VERSIONS_XID de un “flashback version query” puede usarse para consultar la vista FLASHBACK_TRANSACTION_QUERY.

Nota. La función HEXTORAW convierte una cadena de caracteres que contiene dígitos hexadecimales a un valor tipo “raw”.

```
SQL> SELECT xid, operation, start_scn,commit_scn, logon_user, undo_sql
FROM flashback_transaction_query WHERE xid = HEXTORAW('08002C00AD070000');
```

```

XID          OPERATION          START_SCN COMMIT_SCN
-----
LOGON_USER
-----
UNDO_SQL
-----
08002C00AD070000 UPDATE          5285011  5285781
SCOTT
update "SCOTT"."EMP" set "SAL" = '6500' where ROWID =
'AAAEtaAAEAAAAAgAAM';

08002C00AD070000 UPDATE          5285011  5285781
SCOTT
update "SCOTT"."EMP" set "SAL" = '6500' where ROWID = 'AAAEtaAAEAAAAAgAAI';

08002C00AD070000 UPDATE          5285011  5285781
SCOTT
update "SCOTT"."EMP" set "SAL" = '6500' where ROWID =
'AAAEtaAAEAAAAAgAAH';
...
    
```

25. Auditar cualquier cambio que se realice en el “audit trail”.

```
SQL> connect / as sysdba;  
Conectado.
```

```
SQL> show user  
USER es "SYS"
```

```
SQL> audit insert, update, delete on sys.aud$ by access;
```

Auditoria terminada correctamente.

```
SQL> select substr(owner,1,12) prop,substr(OBJECT_NAME,1,12) objeto,  
substr(object_type,1,10) tipo, DEL, INS, SEL, UPD from dba_obj_audit_opts where  
owner='SYS' and OBJECT_NAME ='AUD$' order by object_name, object_type;
```

PROP	OBJETO	TIPO	DEL	INS	SEL	UPD
SYS	AUD\$	TABLE	A/A	A/A	-/-	A/A

26. Considere un caso hipotético donde quiere auditarse la base de datos en base a los siguientes hechos detectados:

- Se han modificado ciertas cuotas, asignación de espacios de almacenamiento, “tablespaces”, y contraseñas de usuario.
- Se están produciendo un número considerable de abrazos mortales debido a que se están adquiriendo bloqueos exclusivos sobre tablas.
- Se ha detectado el borrado de registros de la tabla SCOTT.EMP

Se sospecha que los usuarios “prueba01” y “prueba02” son los responsables de las acciones anteriores.

Determinar las distintas operaciones sobre las que habrá que efectuar una auditoría, así como consultar en las distintas vistas para obtener la información precisa que permita descartar o confirmar las sospechas que han dado lugar a la activación de la auditoría.

Al usuario "prueba02" se le han asignado todos los permisos posibles sobre scott.emp:

SQL> grant all on scott.emp to prueba02;

Grant succeeded.

Para activar la auditoría se ejecutara el siguiente orden de sentencias:

SQL> audit alter, index, rename on default by session;

Auditoria terminada correctamente.

SQL> audit session by prueba01, prueba02;

Auditoria terminada correctamente.

SQL> AUDIT ALTER USER;

Auditoria terminada correctamente.

SQL> AUDIT DROP USER;

Auditoria terminada correctamente.

SQL> AUDIT LOCK TABLE BY ACCESS WHENEVER SUCCESSFUL;

Auditoria terminada correctamente.

SQL> AUDIT DELETE ON scott.emp BY ACCESS WHENEVER SUCCESSFUL;

Auditoria terminada correctamente.

Supongamos que el usuario prueba01 ejecuta las sentencias:

SQL> connect prueba01

Enter password:

Connected.

SQL> ALTER USER prueba03 QUOTA 0 ON users;

ALTER USER prueba03 QUOTA 0 ON users

ERROR at line 1:

ORA-01031: privilegios insuficientes

```
SQL> DROP USER prueba04;  
DROP USER prueba04  
*  
ERROR at line 1:  
ORA-01031: privilegios insuficientes
```

Supongamos que el usuario prueba02 ejecuta las sentencias:

```
SQL> connect prueba02  
Enter password:  
Connected.
```

```
SQL> LOCK TABLE scott.emp IN EXCLUSIVE MODE;
```

Tabla(s) bloqueada(s).

```
SQL> DELETE FROM scott.emp WHERE mgr = 7698;
```

5 filas suprimidas.

```
SQL> CREATE INDEX scott.ename_index ON scott.emp (ename);  
CREATE INDEX scott.ename_index ON scott.emp (ename)  
*
```

```
ERROR at line 1:  
ORA-01031: privilegios insuficientes
```

```
SQL> DELETE FROM scott.emp WHERE empno = 8000;
```

0 filas suprimidas.

Recuperamos ahora información sobre los registros de auditoría:

- **Opciones de auditoría de sentencias fijadas.**

```
SQL> connect system  
Enter password:  
Conectado.
```

```
SQL> select user_name, audit_option, success, failure from  
sys.dba_stmt_audit_opts;
```

USER_NAME	AUDIT_OPTION	SUCCESS	FAILURE
	ALTER USER	BY ACCESS	BY ACCESS
	DROP USER	BY ACCESS	BY ACCESS
	LOCK TABLE	BY ACCESS	NOT SET
PRUEBA02	CREATE SESSION	BY ACCESS	BY ACCESS
PRUEBA01	CREATE SESSION	BY ACCESS	BY ACCESS

- **Opciones de auditoría de privilegios.**

```
SQL> select user_name, privilege, success, failure from
sys.dba_priv_audit_opts;
```

USER_NAME	PRIVILEGE	SUCCESS	FAILURE
	ALTER USER	BY ACCESS	BY ACCESS
	DROP USER	BY ACCESS	BY ACCESS
PRUEBA01	CREATE SESSION	BY ACCESS	BY ACCESS
PRUEBA02	CREATE SESSION	BY ACCESS	BY ACCESS

- **Opciones de auditoría para objetos del esquema SCOTT.**

```
SQL> SELECT * FROM sys.dba_obj_audit_opts WHERE owner = 'SCOTT' AND
object_name LIKE 'EMP%';
```

OWNER	OBJECT_NAME	OBJECT_TYPE	ALT	AUD	COM	DEL	GRA	IND
INS	LOC	REN	SEL	UPD	REF	EXE	CRE	REA
WRI	FBK							
SCOTT	EMP	TABLE	S/S	-/-	-/-	A/-	-/-	S/S
-/-	-/-	S/S	-/-	-/-	-/-	-/-	-/-	-/-

La información debe interpretarse de la siguiente forma:

- El carácter "-" indica que la opción de auditoría no esta fijada.
- EL carácter "S" indica que la opción de auditoría esta fijada "BY SESSION".
- El carácter "A" indica que la opción de auditoría esta fijada "BY ACCESS".

- Cada opción de auditoría puede ser para “WHENEVER SUCCESSFUL” y “WHENEVER NOT SUCCESSFUL”; las dos opciones están separadas por “/”.

- **Información de auditoría para la sentencia “AUDIT SESSION”.**

```
SQL> select substr(os_username,1,10) usuario_so, substr(username,1,12)
usuario, to_char(timestamp,'dd-mm-yyy hh24:mi:ss') tiempo_conexion,
to_char(logoff_time,'dd-mm-yyyy hh24:mi:ss') tiempo_desconexion from
dba_audit_session order by username,timestamp,logoff_time;
```

USUARIO_SO	USUARIO	TIEMPO_CONEXION	TIEMPO_DESCONEXION
oracle	PRUEBA01	15-10-006 10:04:08	15-10-2006 10:08:31
oracle	PRUEBA01	15-10-006 10:20:00	15-10-2006 10:22:14
oracle	PRUEBA01	15-10-006 10:22:22	15-10-2006 10:24:42
oracle	PRUEBA01	15-10-006 11:30:21	15-10-2006 11:33:41
oracle	PRUEBA02	15-10-006 10:08:43	
oracle	PRUEBA02	15-10-006 10:24:43	15-10-2006 10:26:52
oracle	PRUEBA02	15-10-006 11:33:42	
oracle	SYSTEM	15-10-006 10:04:24	15-10-2006 11:30:20
oracle	SYSTEM	15-10-006 10:26:53	15-10-2006 11:03:47

9 filas seleccionadas.

- **Registros de auditoria generados para objetos.**

```
SQL> select substr(username,1,12) usuario, to_char(timestamp,'dd-mm-yyy
hh24:mi:ss') tiempo_conexion, substr(owner,1,10) propie,
substr(obj_name,1,15) objeto, substr(action_name,1,15) accion from
dba_audit_object where owner='SCOTT' order by os_username,usuario,
timestamp, owner,obj_name,action_name;
```

USUARIO	TIEMPO_CONEXION	PROPIE	OBJETO	ACCION
PRUEBA02	15-10-006 12:15:55	SCOTT	EMP	LOCK
PRUEBA02	15-10-006 12:16:04	SCOTT	EMP	DELETE
PRUEBA02	15-10-006 12:16:29	SCOTT	EMP	DELETE

3 filas seleccionadas.

Debe tenerse en cuenta que la opción para auditar las sentencias alter, index, ... es posterior a la creación del objeto.

Si, explícitamente, se audita la tabla "emp".

```
SQL> connect system
Introduzca la contraseña:
Conectado.
```

```
SQL> audit alter, INDEX, RENAME ON scott.emp by session;
Auditoria terminada correctamente.
```

```
SQL> connect prueba02
Introduzca la contraseña:
Conectado.
```

```
SQL> alter table scott.emp pctfree 80;
```

Tabla modificada.

```
SQL> CREATE INDEX scott.ename_index ON scott.emp (ename);
CREATE INDEX scott.ename_index ON scott.emp (ename)
*
```

```
ERROR at line 1:
ORA-01031: insufficient privileges
```

```
SQL> connect system
Introduzca la contrase?a:
Conectado.
```

```
SQL> select substr(username,1,12) usuario, to_char(timestamp,'dd-mm-yyy
hh24:mi:ss') tiempo_conexion, substr(owner,1,10) propie,
substr(obj_name,1,15) objeto, substr(action_name,1,15) accion from
dba_audit_object where owner='SCOTT' order by os_username,usuario,
timestamp, owner,obj_name,action_name;
```

USUARIO	TIEMPO_CONEXION	PROPIE	OBJETO	ACCION
PRUEBA02	15-10-006 12:15:55	SCOTT	EMP	LOCK
PRUEBA02	15-10-006 12:16:04	SCOTT	EMP	DELETE
PRUEBA02	15-10-006 12:16:29	SCOTT	EMP	DELETE

PRUEBA02 15-10-006 12:43:28 SCOTT EMP SESSION REC

4 filas seleccionadas.

```
SQL> SELECT * FROM sys.dba_obj_audit_opts WHERE owner = 'SCOTT' AND
object_name LIKE 'EMP%';
```

OWNER	OBJECT_NAME	OBJECT_TYPE	ALT	AUD	COM	DEL	GRA	IND		
INS	LOC	REN	SEL	UPD	REF	EXE	CRE	REA	WRI	FBK
SCOTT	EMP	TABLE	S/S	-/-	-/-	A/-	-/-	S/S		
-/-	-/-	S/S	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-

Si se audita ahora por “acceso”.

```
SQL> audit alter, INDEX, RENAME ON scott.emp by access;
```

Auditoria terminada correctamente.

```
SQL> connect prueba02
Introduzca la contrase?a:
Conectado.
```

```
SQL> alter table scott.emp pctfree 90;
```

Tabla modificada.

```
SQL> CREATE INDEX scott.ename_index ON scott.emp (ename);
CREATE INDEX scott.ename_index ON scott.emp (ename)
```

*

```
ERROR en linea 1:
ORA-01031: privilegios insuficientes
```

```
SQL> connect system
Introduzca la contrase?a:
Conectado.
```

```
SQL> select substr(username,1,12) usuario, to_char(timestamp,'dd-mm-yyy
hh24:mi:ss') tiempo_conexion, substr(owner,1,10) propie,
```

```
substr(obj_name,1,15) objeto, substr(action_name,1,25) accion from
dba_audit_object where owner='SCOTT' order by os_username,usuario,
timestamp, owner,obj_name,action_name;
```

USUARIO	TIEMPO_CONEXION	PROPIE	OBJETO	ACCION
PRUEBA02	15-10-006 12:15:55	SCOTT	EMP	LOCK
PRUEBA02	15-10-006 12:16:04	SCOTT	EMP	DELETE
PRUEBA02	15-10-006 12:16:29	SCOTT	EMP	DELETE
PRUEBA02	15-10-006 12:43:28	SCOTT	EMP	SESSION REC
PRUEBA02	15-10-006 12:51:07	SCOTT	EMP	ALTER TABLE

5 filas seleccionadas.

Observemos que prueba02 no tiene permiso para crear índices.

```
SQL> connect system
Introduzca la contraseña:
Conectado.
```

```
SQL> select * from dba_sys_privs where grantee='PRUEBA02';
```

ninguna fila seleccionada

```
SQL> select * from dba_role_privs where grantee='PRUEBA02';
```

GRANTEE	GRANTED_ROLE	ADM DEF
PRUEBA02	CONEXION	NO YES

```
SQL> select * from dba_role_privs where grantee='CONEXION';
```

ninguna fila seleccionada

```
SQL> select * from dba_sys_privs where grantee='CONEXION';
```

GRANTEE	PRIVILEGE	ADM
CONEXION	CREATE SESSION	NO

Se le otorga permiso y se prueba a crear un índice como prueba02 sobre scott.emp. Aunque falle, registra el apunte.

SQL> grant create table to prueba02;

Concesion terminada correctamente.

SQL> connect prueba02
Introduzca la contraseña:
Conectado.

SQL> CREATE INDEX scott.ename_index ON scott.emp (kk);
CREATE INDEX scott.ename_index ON scott.emp (kk)

*

ERROR at line 1:
ORA-00904: "KK": identificador no valido

SQL> connect system
Introduzca la contrase?a:
Conectado.

SQL> select substr(username,1,12) usuario, to_char(timestamp,'dd-mm-yyy
hh24:mi:ss') tiempo_conexion, substr(owner,1,10) propie,
substr(obj_name,1,15) objeto, substr(action_name,1,25) accion from
dba_audit_object where owner='SCOTT' order by os_username,usuario,
timestamp, owner,obj_name,action_name;

USUARIO	TIEMPO_CONEXION	PROPIE	OBJETO	ACCION
PRUEBA02	15-10-006 12:15:55	SCOTT	EMP	LOCK
PRUEBA02	15-10-006 12:16:04	SCOTT	EMP	DELETE
PRUEBA02	15-10-006 12:16:29	SCOTT	EMP	DELETE
PRUEBA02	15-10-006 12:43:28	SCOTT	EMP	SESSION REC
PRUEBA02	15-10-006 12:51:07	SCOTT	EMP	ALTER TABLE
PRUEBA02	15-10-006 13:01:42	SCOTT	ENAME_INDEX	CREATE INDEX

6 filas seleccionadas.