Information Systems Audit
Degree in Computer Engineering in Information Systems - Course 2021/2022

PRACTICE 2: ORACLE DBD AUDIT II

Objectives

- Mandatory and SYSDBA audit
- Enable standard auditing of the database
- Auditing attempted connections and access to database objects
- Good auditing practices

Content

1. Mandatory auditing and SYSDBA auditing

Oracle always audits certain database operations and writes them to the operating system audit files, regardless of the standard auditing configuration. This mandatory auditing includes logs about:

- **Database startup and shutdown.** An audit log is created indicating the OS user who is starting/stopping the instance, the terminal identifier and the date/time.
- **SYSDBA and SYSOPER logins.** All `SYSDBA` and `SYSOPER` logins are logged.

In addition to the mandatory audit, it is possible to audit DELETE, INSERT, UPDATE and MERGE operations performed by SYS users by enabling the AUDIT_SYS_OPERATIONS parameter. If SYS operations are audited, the initialization parameter AUDIT_FILE_DEST controls the location of the Audit records.

- AUDIT_FILE_DEST: specifies the location of the audit file in the operating system. By default, on Unix/Linux and Windows systems the default path is $ORACLE_BASE/admin/$ORACLE_SID/adump. Also, on Windows it is written to the system event viewer. The path should refer to a local disk on the machine where the database instance is installed so as not to impair system performance.
- AUDIT_SYS_OPERATIONS: enables the auditing of operations carried out by the SYS user, as well as that of any user connected with SYSDBA, SYSOPER privileges. The audit is written to the operating system audit file or to the one indicated in the AUDIT_FILE_DEST parameter. This parameter should be set to TRUE

To view the values of these parameters:

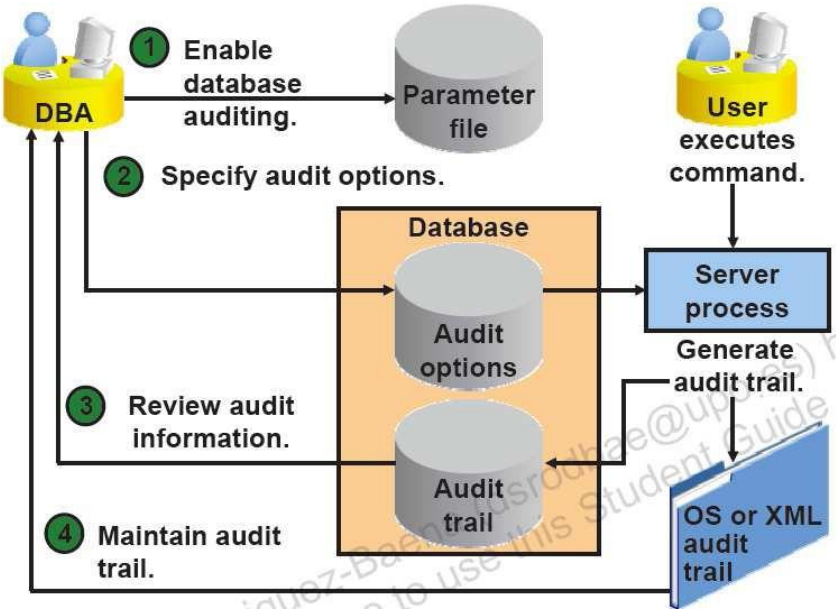
```
SHOW PARAMETER AUDIT_FILE_DEST
```

```
SHOW PARAMETER AUDIT_SYS_OPERATIONS
```

2. Standard Oracle database audit

In addition to mandatory auditing, Oracle provides a standard auditing mechanism that is enabled at the system level through the dynamic initialization parameter AUDIT_TRAIL. This parameter allows us to indicate the place where the audit trails will be stored.

After enabling database auditing and specifying the auditing options, the database starts to store auditing information. Specifically, standard auditing allows us to audit SQL statements, privileges, schema objects and network activity.



The possible values of the AUDIT_TRAIL parameter:

• **OS**: activates database auditing, writing the audited events to the operating system audit trail. By default, the operating system log files are located in the directory $ORACLE_BASE/admin/$ORACLE_SID/adump, both on Unix and Windows. On Windows systems it is also written to the Windows event viewer. The path to these files is configurable by means of the AUDIT_FILE_DEST parameter.
• **DB**: activate the audit and the data will be stored in the Oracle SYS.AUD$ taba.
• **DB, EXTENDED**: activates the audit and the data will be stored in the Oracle SYS.AUD$ table. In addition, the corresponding values will be written to the Sql Text and Sql Bind columns of the SYS.AUD$ table, thus collecting the SQL statement executed by the user and the variables used by the statement.
• **XML**: activates the database audit, the events will be written in XML files of the operating system.
• **XML, EXTENDED**: activates the database audit, the events will be written in the XML format of the operating system, also the Sql Text and Sql Bind values will be included. The V$XML_AUDIT_TRAIL view allows us to see all the XML files located in that directory.

```
SHOW PARAMETER AUDIT_TRAIL
```

When Oracle configures database auditing, the most commonly used SQL statements and privileges are audited by default. Specifically:

Privileges Audited by Default		
ALTER ANY PROCEDURE	CREATE ANY LIBRARY	GRANT ANY PRIVILEGE
ALTER ANY TABLE	CREATE ANY PROCEDURE	GRANT ANY ROLE
ALTER DATABASE	CREATE ANY TABLE	DROP ANY PROCEDURE
ALTER PROFILE	CREATE EXTERNAL JOB	DROP ANY TABLE
ALTER SYSTEM	CREATE PUBLIC DATABASE LINK	DROP PROFILE
ALTER USER	CREATE SESSION	DROP USER
AUDIT SYSTEM	CREATE USER	EXEMPT ACCESS POLICY
CREATE ANY JOB	GRANT ANY OBJECT PRIVILEGE	
Statements Audited by Default		
SYSTEM AUDIT BY ACCESS		
ROLE BY ACCESS		

The various auditing options are configured using the AUDIT command. The NOAUDIT command allows you to remove these audit settings.

AUDIT command
The AUDIT command allows you to initiate the following types of audits. This command can work even if database auditing is not enabled, but it will not leave a record until auditing is enabled.

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

- sql_statement_clause: activates auditing for a specific SQL statement.
- schema_object_clause: activates the audit for a specific object in the database.
- WHENEVER SUCCESSFUL: enables auditing only for SQL operations and statements on schema objects that complete successfully.
- WHENEVER NOT SUCCESSFUL: enable auditing only for SQL operations and statements on schema objects that cause errors.

.
The complete syntax can be found at:

https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_4007.htm

NOAUDIT command
The NOAUDIT command is used to stop the auditing activity that was previously activated with AUDIT.

Syntax:
NOAUDIT
{ sql_statement_clause | schema_object_clause | NETWORK}
[WHENEVER [NOT] SUCCESSFUL] ;

- sql_statement_clause: stops the auditing of a specific SQL statement.
- schema_object_clause: stops the audit for a specific object in the database.
- WHENEVER SUCCESSFUL: stops auditing only for SQL operations and statements on schema objects that complete successfully.
- WHENEVER NOT SUCCESSFUL: stops auditing only for SQL operations and statements on schema objects that cause errors.

The complete syntax can be found at:

https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9017.htm

3. Specifying audit options

Session Auditing. Allows to audit all database connections, both successful and failed. SQL> AUDIT

session;

It is also possible to audit the sessions of certain users: SQL>

AUDIT session by user;

Or just the successful or unsuccessful ones:

SQL>AUDIT session whenever successful;
SQL>AUDIT session whenever not successful;

SQL statement audits. The statement shown below allows auditing any data definition statement (DDL) that affects a table, including CREATE TABLE, DROP TABLE, TRUNCATE TABLE. The auditing of SQL statements can be filtered by user name and/or by success or failure of execution:

```
--Enable audit
SQL> AUDIT TABLE BY hr WHENEVER NOT SUCCESSFUL;

--Disable auditing
SQL> NOAUDIT TABLE BY hr WHENEVER NOT SUCCESSFUL;
```

Auditing of system privileges. Can be used to audit the exercise of some system privilege (such as DROP ANY TABLE). It can be filtered by user name and/or by success or failure of statement execution. By default, the audited is BY ACCESS. Each time an audited system privilege is exercised, an audit log is generated. We can group these records with the BY SESSION clause, so that only one record per session will be generated (thus, if a user executes several UPDATE statements on a table, only one record per session will be generated).

```
--Enable audit
SQL> AUDIT DROP ANY TABLE BY hr BY SESSION;

--Disable auditing
SQL> NOAUDIT DROP ANY TABLE BY hr BY SESSION;
```

Auditing privileges on objects. It can be used to audit actions on tables, views, procedures, sequences, directories, and user-defined data types. This type of auditing can be filtered by success or failure of statement execution and grouped by session or access. Unlike the previous auditing, the default grouping is by session.

```
--Enable audit
SQL> AUDIT UPDATE ON hr.employees BY ACCESS;

--Disable auditing
SQL> NOAUDIT UPDATE ON hr.employees BY ACCESS;
```

Network auditing. It can be used to audit unexpected errors in the network protocol or internal errors in the network layer. The types of errors discovered by network auditing are not connection failures, but may have other possible causes such as those related to configuration and data encryption processes.

```
SQL>AUDIT network;
```

4. Viewing audit information

Oracle provides a series of views that contain all the information related to the audit. Among these views we can highlight:

- ALL_DEF_AUDIT_OPTS: contains the default object audit options that apply when objects are created.
- DBA_AUDIT_EXISTS: shows the audit entries produced by AUDIT NOT EXISTS (this audit records all SQL statements that fail because the object does not exist).
- DBA_AUDIT_OBJECT: shows the audit logs of all database objects.
- DBA_AUDIT_SESSION: show all the records from audit logs CONNECT and DISCONNECT audit trails.
- DBA_AUDIT_STATEMENT: shows the audit records for all GRANT, REVOKE, AUDIT, NOAUDIT and ALTER SYSTEM statements in the database.
- DBA_AUDIT_TRAIL: standard audit trails
- DBA_OBJ_AUDIT_OPTS: collects the audit options of all objects.

- DBA_PRIV_AUDIT_OPTS: collects the system privileges that are currently audited per user.
- DBA_STMT_AUDIT_OPTS: collects the current system audit options per user.

A description of these can be found at: http://docs.oracle.com/cd/B28359_01/server.111/b28320/index.htm.

5. Good auditing practices

Oracle recommends these best practices when auditing a database:

- As a general rule, design the audit to gather the information you need to meet the audit requirements, focusing on the activities that cause the biggest security problems. Thus, auditing every table in the database is not practical, but auditing the columns of tables that have sensitive information is.
- Periodically archive and clean audit logs so as not to penalize database performance.

Bibliography

ORACLE 11GR2 Administration Workshop I Oracle University.

ORACLE DATABASE ONLINE DOCUMENTATION: https://docs.oracle.com/cd/E11882_01/index.htm

ORACLE DATABASE REFERENCE: https://docs.oracle.com/en/database/

Problems

P1. From SQL Developer using the AUDITOR user, query the values of the AUDIT_FILE_DEST, AUDIT_SYS_OPERATIONS and AUDIT_TRAIL parameters. Collect them in the Excel created in the first practice along with the meaning and recommendation on each.

show parameters;

Change the value of AUDIT_SYS_OPERATIONS to TRUE and AUDIT_TRAIL to DB in case they do not have these values.

Estimated time: 5 minutes

P2. Access the link [ORACLE DATABASE REFERENCE] where the detail of the views ALL_DEF_AUDIT_OPTS, DBA_AUDIT_EXISTS, DBA_AUDIT_OBJECT, DBA_AUDIT_SESSION, DBA_AUDIT_STATEMENT, DBA_AUDIT_TRAIL, DBA_OBJ_AUDIT_OPTS, DBA_PRIV_AUDIT_OPTS and DBA_STMT_AUDIT_OPTS, to see the description of these and understand the meaning of each field.

Estimated time: 30 minutes

P3. From SQL Developer or SQL Plus, enable session auditing. Then open a new connection with the HR user with the wrong password. Then try to connect with the correct password. From the AUDITOR user connection check the session audit. How do the two connection attempts differ?

HELP: You can use the following statements as BASE to achieve the requested information:

```
SELECT *
FROM DBA_AUDIT_SESSION
order by timestamp desc;

SELECT *
FROM DBA_AUDIT_TRAIL
order by timestamp desc;
```

TIP1: Review the different fields contained in the DBA_AUDIT_SESSION and DBA_AUDIT_TRAIL tables and display only those that are useful for this section.

TIP2: In SQLplus it is possible to format the output. Do some research on this possibility to make it easier to understand the output given by the executed statements.

Estimated time: 10 minutes

P4. From SQL Developer or SQL Plus, try to detect all failed connection attempts made in the last week. Do you detect any suspicious activity? What should you reflect and recommend in an audit report about it?

```
SELECT *
FROM DBA_AUDIT_SESSION
WHERE ACTION_NAME ='LOGON'
AND NVL(RETURNCODE,0) != 0
AND TIMESTAMP BETWEEN (SYSTIMESTAMP - 7) AND SYSTIMESTAMP
order by timestamp desc;
```

Estimated time: 10 minutes

P5. See the default object audit options. select * from

all_def_audit_opts;

Stop the audit activity for "alter", "grant", "insert", "update" or "delete".

NOAUDIT alter, grant, insert, update, delete ON default;

See now the default object auditing options.

Change these options so that from now on for created objects information is recorded whenever an "alter", "grant", "insert", "update" or "delete" occurs.

Please refer back to the default settings. What changes have occurred after each NOAUDIT/AUDIT run?

Estimated time: 5 minutes P6.

We activate several audits:

AUDIT ALTER TABLE;
AUDIT DELETE ANY TABLE;
AUDIT DROP ANY TABLE;
AUDIT AUDIT SYSTEM;
AUDIT ALL BY HR BY ACCESS;
AUDIT SELECT TABLE, UPDATE TABLE, DELETE TABLE, INSERT TABLE, EXECUTE PROCEDURE BY HR BY ACCESS;
AUDIT SELECT TABLE, DELETE TABLE, UPDATE TABLE, INSERT TABLE BY AUDITOR BY ACCESS;
AUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE, EXECUTE PROCEDURE BY ACCESS WHENEVER NOT SUCCESSFUL;

Check the enabled audit options:

select user_name user, audit_option option, success, failure from
DBA_STMT_AUDIT_OPTS order by user_name;

Then, from a connection to the HR user perform certain operations on the tables. The session must be created again.

select TABLE_NAME from USER_TABLES order by TABLE_NAME; --We see all user tables HR select *

from EMPLOYEES;

select * from COUNTRIES;

insert into COUNTRIES values('ES','Spain',1);

insert into JOB_HISTORY values(114,'01-OCT-2014','10-JAN-2023','AC_MGR',70);

delete from COUNTRIES where COUNTRY_ID='ES';

CREATE TABLE audit_test (CODE
  varchar2(3),
  DESCRIPTION varchar2(20)); --Create table

drop table audit_test; --Delete table

Review the audit records created in DBA_AUDIT_OBJECT as AUDITOR: select * from

DBA_AUDIT_OBJECT order by timestamp desc;

Check the audit options of all objects and system privileges that are audited: select * from

DBA_OBJ_AUDIT_OPTS;

select * from DBA_PRIV_AUDIT_OPTS;

Estimated time: 15 minutes

P7. Activate the NOT EXISTS audit and try to perform a query on a table that does not exist. Check the results in the DBA_AUDIT_EXISTS view.

HELP: You can use the following statement as a BASIS to achieve the requested information:

```
select *
from DBA_AUDIT_EXISTS
order by timestamp desc;
```

TIP1: review the various fields contained in the DBA_AUDIT_EXISTS tables and display only those that are useful for this section. Format the output for better visualization.

Estimated time: 5 minutes

P8. Check the content of the DBA_AUDIT_STATEMENT view, what information does it provide? HELP: You

can use the following statement as a BASIS to achieve the requested information:

```
select *
from DBA_AUDIT_STATEMENT
order by timestamp desc;
```

TIP1: Review the various fields contained in the DBA_AUDIT_STATEMENT tables and display only those that are useful for this section. Format the output for better visualization.

Estimated time: 5 minutes

P9. The DB shows the start date of the workers, as well as their job title. You have detected that an active worker is listed with a position that does not correspond to him. Could you investigate what could have happened and when? What controls could have failed?

select * from EMPLOYEES;

Estimated time: 10 minutes