DATABASE AUDITING: Oracle 8i/9i

by

Ashok Kapur

Hawkeye Technology, Inc.

http://www.hawkeyetechnology.com

Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS FOR EDUCATIONAL PURPOSES ONLY AND SHOULD NOT BE CONSTRUED AS LEGAL ADVICE. THE PROVIDER OF THIS INFORMATION EXPRESSLY DENIES LIABILITY AND UNDERTAKES NO RESPONSIBILITY FOR THE RELIANCE ON, OR CONSEQUENCES OF, USING THE INFORMATION CONTAINED HEREIN.

Agenda

- Intro to Auditing
- Auditing Feature Overview
- Audit Tables/Views
- How to Start and Stop Auditing
- Types of Auditing
- Other Auditing Methods
- Fine Grain Auditing
- Auditing Guidelines
- Questions

Introduction To Auditing

Define Auditing

- **au·dit***
  Pronunciation: 'o-d&t Function: *noun*
  1 a : a formal examination of an organization's or individual's accounts or financial situation **b :** the final report of an audit
  2 : a methodical examination and review
- **audit trail ***
  Function: *noun*
  Date: 1954
  : a record of a sequence of events (as actions performed by a computer) from which a history may be reconstructed
- Tracking and monitoring of selected database activity.

* From Merriam-Webster's Collegiate Dictionary

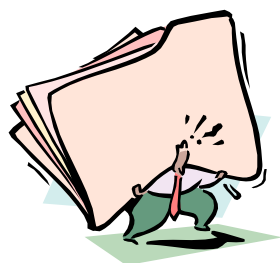
Why Audit?

- Company policies on auditing and logging information
- Log access to sensitive information
- Investigate suspicious or malicious activity
- Track schema changes

Auditing Feature Overview

Auditing Features

- Auditing Levels
  - Session auditing
  - Privilege auditing
  - Statement auditing
  - Object auditing



Auditing Features (contd.)

- Auditing Modifiers
  - When Success/When Failure
  - All users or selected users
  - All objects or selected objects
  - By session or by access
- Audit Trail Storage
  - Database
  - OS

Other Auditing Features

- Auto Auditing or Admin Auditing
  - Admin audit log
  - Alert Log
- Auditing via database triggers
- Redo logs and Archive logs

Things to Remember

- Audit records are generated during the EXECUTION phase of the statement execution.
- Audit record generation is done via autonomous transaction feature, thus independent of user's transaction
- No audit records are generated for sessions connected as SYSDBA or CONNECT INTERNAL

Audit Tables/Views

Auditing Tables and Views

- Audit trail table
  - SYS.AUD$
  - Owner: SYS Tablespace: SYSTEM
  - Move it or not to move it: That's the question?

- Audit Options views
  - ALL_DEF_AUDIT_OPTS
  - DBA_PRIV_AUDIT_OPTS
  - DBA_STMT_AUDIT_OPTS
  - DBA_OBJ_AUDIT_OPTS/USER_OBJ_AUDIT_OPTS

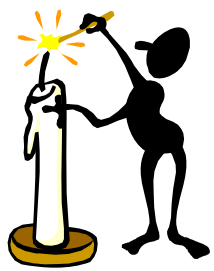
Audit Tables and Views(contd.)

- Audit Trail views
  - DBA_AUDIT_TRAIL/USER_AUDIT_TRAIL
  - DBA_AUDIT_SESSION/USER_AUDIT_SESSION
  - DBA_AUDIT_OBJECT/USER_AUDIT_OBJECT
  - DBA_AUDIT_STATEMENT/USER_AUDIT_STATEMENT
  - DBA_AUDIT_EXISTS
- Audit Code Lookup Views
  - AUDIT_ACTIONS
  - STMT_AUDIT_OPTION_MAP
  - SYSTEM_PRIVILEGE_MAP

How to Start and Stop Auditing

How to Start Auditing

1. Enable auditing feature (Turn on the master switch)
   - Init.ora parameter AUDIT_TRAIL=DB|OS
   - Bounce the database
2. Setup auditing options
   - AUDIT SQL statement



How to Stop Auditing

- Unset auditing options
  - NOAUDIT SQL statement
  - Note: NOAUDIT stmt syntax must match the AUDIT stmt syntax
- Disable Auditing feature (Turn off the master switch)
  - Init.ora parameter AUDIT_TRAIL=NONE
  - Bounce the database

Types of Auditing

Session Auditing

- Provides login and logout auditing
- Can audit login success or failure or both
- Some session stats are also recorded with logout information
- Must have AUDIT SYSTEM privilege

Session Auditing Example

- **AUDIT SESSION** [WHENEVER SUCCESSFUL | WHENEVER NOT SUCCESSFUL]

- **SELECT os_username, username, timestamp, logoff_time, action_name, returncode, logoff_lread, logoff_pread FROM dba_audit_session**

```
OS_USERNM USERNAME TIMESTAMP           LOGOFF_TIME         ACT_NM  RT_CD  LRD  PRD
--------- -------- ------------------- ------------------- ------- ----- ----- ---
AK837609  SYSTEM   01/09/2002 12:46:01 01/09/2002 12:53:16 LOGOFF      0  542   0
AK837609  SYS      01/09/2002 12:53:16                    LOGON    1017
AK837609  SYSTEM   01/09/2002 12:53:20                    LOGON       0
```

Privilege Auditing

- Provides auditing of system privilege uses
  - delete any table, alter table, create table, public synonym
- Audit record generated whenever an audited system privilege is used by the user.
- Audit option becomes active for subsequent connections
- Must have AUDIT SYSTEM privilege
- **Note: Object privileges are checked before system privileges.**
  - Delete priv. on a table takes precedence over DELETE ANY TABLE priv.

Privilege Auditing Example

- audit create table by akapur by access;

- **SELECT user_name, privilege, success, failure**
  FROM dba_priv_audit_opts;

```
USER_NAME  PRIVILEGE            SUCCESS    FAILURE
---------- -------------------- ---------- ----------
           CREATE SESSION       BY ACCESS  BY ACCESS
AKAPUR     CREATE TABLE         BY ACCESS  BY ACCESS
```

- **SELECT username,owner,obj_name,action_name,priv_used,timestamp**
  FROM dba_audit_object;

```
USERNAME  OWNER   OBJ_NAME  ACTION_NAME     PRIV_USED       TIMESTAMP
--------- ------- --------- --------------- --------------- -------------------
AKAPUR    AKAPUR  TEST      CREATE TABLE    CREATE TABLE    01/09/2002 13:43:12
SYSTEM    SYS     AUD$      TRUNCATE TABLE  DROP ANY TABLE  01/04/2002 14:43:28
```

Statement Auditing

- Provides DDL and DML statement auditing
  - table, index, procedure, profile, sequence, role,…
  - NOT EXISTS: audits all SQL stmts that fail due to “object does not exist” error.
- Can audit either success or failure and by session or by access
- Audit option becomes active for subsequent connections
- Must have AUDIT SYSTEM privilege

Statement auditing Example

- AUDIT index BY ACCESS WHENEVER SUCCESSFUL
- **SELECT user_name,audit_option,success,failure**
  FROM dba_stmt_audit_opts

```
USER_NAME  AUDIT_OPTION    SUCCESS    FAILURE
---------- --------------- ---------- ----------
           NOT EXISTS      BY ACCESS  BY ACCESS
           INDEX           BY ACCESS  NOT SET
AKAPUR     CREATE TABLE    BY ACCESS  BY ACCESS
```

- **SELECT username,owner,obj_name,action_name,priv_used,timestamp**
  FROM dba_audit_object

```
USERNAME  OWNER   OBJ_NAME   ACTION_NAME     PRIV_USED        TIMESTAMP
--------- ------- ---------- --------------- ---------------- -------------------
SYSTEM    AKAPUR  TEST_IDX1  CREATE INDEX    CREATE ANY INDEX 01/09/2002 14:17:59
AKAPUR    AKAPUR  TEST       CREATE TABLE    CREATE TABLE     01/09/2002 13:43:12
SYSTEM    SYS     AUD$       TRUNCATE TABLE  DROP ANY TABLE   01/04/2002 14:43:28
```

Object Auditing

- Provides capability to audit specific objects
  - AUDIT DELETE ON hr.dept
- Object must be in user's schema or user must have AUDIT ANY privilege
- Can audit tables, indexes, sequences, procedures and packages.
  - Note: Can not audit individual procedures within packages
- Audit changes for objects become effective immediately!

Object Auditing Example

- audit alter, grant on default by access whenever successful;

- **SELECT alt, gra FROM all_def_audit_opts**

```
ALT     GRA
------- -------
A/-     A/-
```

- **SELECT username,owner,obj_name,action_name,obj_privilege, grantee,timestamp FROM dba_audit_statement;**

```
USERNA OWNER  OBJ_NAME     ACTION_NAME  OBJ_PRIVILEGE    GRANTEE TIMESTAMP
------ ------ ------------ ------------ ---------------- ------- -----------------
AKAPUR AKAPUR TEST_DEFAULT GRANT OBJECT ---------Y------ SYSTEM 01/09/2002 8:00:45
```

- OBJ_PRIV flag values are Y/N for alter, audit, comment, delete, grant, index, lock, rename, select, update, references and execute. Last three are reserved.

Object Auditing Example (contd.)

- audit alter, delete, update on akapur.test by access;

- **SELECT owner,object_name,object_type,alt,del,upd**
  FROM dba_obj_audit_opts WHERE object_name='TEST'

```
OWNER      OBJECT_NAM OBJECT_TY ALT     DEL     UPD
---------- ---------- --------- ------- ------- -------
AKAPUR     TEST       TABLE     A/A     A/A     A/A
```

- **SELECT username,owner,obj_name,action_name,priv_used,timestamp**
  FROM dba_audit_object;

```
USERNAME OWNER   OBJ_NAME ACTION_NAME  PRIV_USED TIMESTAMP
-------- ------- -------- ------------ --------- ------------------
AKAPUR   AKAPUR  TEST     ALTER TABLE            01/09/2002 16:25:29
```

Other Auditing Methods

Auto Auditing

- Database startups, database shutdowns and administrator logins (as sysdba) are ALWAYS logged in OS audit logs.
  - For WINNT: System Event log
  - For Unix: $ORACLE_HOME/rdbms/audit
- Certain admin actions logged in ALERT LOG.
  - Create database, create controlfile, create tablespace,…
- NOTE: This auditing is independent of AUDIT_TRAIL init.ora parameter setting

Detailed Auditing via Triggers

- Database triggers can be used to augment auditing
- Triggers can be used to get column change information
- Triggers are more resource intensive than auditing but can provide more detailed auditing information
- Use AFTER triggers for auditing

Auditing using Redo/Arch Logs

- Since Redo/Arch logs contain detailed DML/DDL activity, Log Miner maybe used to either investigate foul play or to reverse an inadvertent action.
- Can also use Log Miner on a regular basis to monitor and log certain activity.
    - Why do it if Audit option is already available?
        - To log column value changes!

Fine-Grain Auditing

Fine-Grain Auditing

- Fine grain auditing (FGA) has been introduced in Oracle 9i.
- Built on Fine Grain Access control.
- Added FGA$ and FGA_LOG$ tables.
- Added several DBA_FGA% and USER_FGA% views.
- FGA allows auditing of SELECT statements.
- For example, audit SELECT on Salary column of EMP table for all users that do not have HR role

Fine-Grain Auditing (contd.)

- Administer FGA using DBMS_FGA package
  - DBMS_FGA.ADD_POLICY
  - DBMS_FGA.DROP_POLICY
  - DBMS_FGA.DISABLE_POLICY
  - DBMS_FGA.ENABLE_POLICY
- FGA ONLY works with cost based optimizer.
- ALWAYS ANALYZE the table being audited!!
- Auditing decision made during FETCH phase.

Fine Grain Auditing Example

- Audit SALARY column of EMP table for records where SALARY > 10000
  - **EXEC DBMS_FGA.ADD_POLICY(OBJECT_SCHEMA=> 'SYSTEM', -**
    OBJECT_NAME=> 'EMP', POLICY_NAME=> 'VIEW_SAL', -
    AUDIT_CONDITION => 'salary > 10000 ', AUDIT_COLUMN =>
    'SALARY');

- **select * from emp;**
- **select timestamp, db_user, object_schema, sql_text**
  from dba_fga_audit_Trail;
- TIMESTAMP DB_USER OBJECT_SCH SQL_TEXT

```
------------------- ---------- ----------   -----------------
02/12/2002 12:57:51 SYSTEM     SYSTEM       select * from emp
```

- **select * from emp where salary < 500;**
  - NOTE: No audit record generated.

Auditing Guidelines

Auditing Guidelines

- Always protect the audit table
- Audit with specific goal in mind.
- Try and not use AUDIT ALL!
- Trim the audit trail periodically. Instead of just deleting records, may want to archive them. (Copy records to another table and then export that table)

Protect Audit Trail

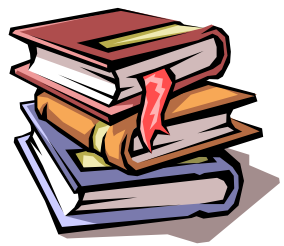
- Audit the audit trail table!
  - AUDIT UPDATE, DELETE ON SYS.AUD$ BY ACCESS
  - AUDIT DROP ANY TABLE BY ACCESS
- A non-privileged session that has delete access on AUD$ will not be able to delete these audit records.
- Keep SYSDBA, DROP ANY TABLE, DELETE on AUD$, UPDATE on AUD$ under strict control

What If…

- Audit table is full?
  - Users will not be able to logon if auditing sessions
  - Users will get warnings for each auditable action
- Move AUD$ to another TS?
  - This action is not supported by Oracle
  - Move it back before database upgrades and possibly patches

References

- Oracle 8i Concepts Manual
- Oracle 8i Administration Guide
- Oracle 9i Concepts Manual
- Oracle 9i Administration Guide
- Metalink Notes: 72460.1, 1019377.6, 1025314.6, 103964.1, 1068714.6



Questions?

Contact Information

Name: Ashok Kapur

Email: afkapur@hawkeyetechnology.com