Database Auditing

Auditing gives the DBA the ability to track information within the database. It provides information on who performed a certain operation and when it was performed. Auditing is a reactive function. It gives the DBA information about an activity only after it has already occurred. This reactive information provides a snapshot of what occurred, depending on the level of detail being audited. It gives the DBA a basis for tracking changes within the database.

Because auditing causes additional rows to be added to the database for each operation, it is important to balance the auditing being done against constraints such as performance overhead and physical storage requirements.

To activate auditing for a database instance, the DBA must make certain that the AUDIT_TRAIL parameter of the INIT. ORA parameter file is set to DB or OS to indicate where the audit trail should be written. The default value for this parameter is NONE.

Statement Level Auditing

With Auditing that occurs at the statement level "sometimes called the privilege level", an audit record is written for each specific SQL statement that is issued. Depending on how specific the DBA makes this level of auditing, the audit information generated can be substantial.

In the following example, two audit options are set. One option logs CREATE TABLE activity within the database. The other option logs all CREATE SESSION activity done by USER1.

```
C:> sqlplus system
Password: ........
Connected.
SQL> audit create table by access whenever successful;
Statement processed.
SQL> audit create session by user1 by access whenever successful;
Statement processed.
```

Two important parameters appear in every SQL audit command:

- BY SESSION/BY ACCESS
- WHENEVER SUCCESSFUL/WHENEVER NOT SUCCESSFUL

BY SESSION/BY ACCESS determines how often audit records should be written. In a BY SESSION audit, the database writes a single audit record that sums all the times that an action took place during a given session. In a BY ACCESS audit, the database writes a single audit record for each SQL statement that was issued.

WHENEVER SUCCESSFUL/WHENEVER NOT SUCCESSFUL determines the conditions under which the audit records should be written. Audits that are WHENEVER SUCCESSFUL have information written only if they succeed. WHENEVER NOT SUCCESSFUL audits are written only if they do not succeed.

Object Level Auditing

It is possible to audit database information at the database object level, which enables you to trap operations done on a specific database object. The syntax is essentially the same as that for a statement level audit:

```
C:> sqlplus system
Password: ........
Connected.
SQL> audit delete on hr.payroll;
Statement processed.
```

The statement audit specifies a class of statements and, optionally, which user to audit for these statements. The object audit, on the other hand, points to a type of object operation and the name of an object.

The types of object level operations that can be performed are

ALTER
AUDIT
COMMENT
DELETE
EXECUTE

ALTER	AUDIT	COMMENT	DELETE	EXECUTE
GRANT	INDEX	INSERT	LOCK	RENAME
SELECT	UPDATE			

These object level operations can be performed on any of the following types of database objects:

- Tables
- Views
- Sequences
- Stored procedures, functions, or packages
- Snapshots

To deactivate object or privilege level auditing, bounce the database and set AUDIT_TRAIL to NONE, or specify the current audit options with the NOAUDIT command. For example,

```
C:> sqlplus system
Password: ........
Connected.
SQL> noaudit all;
Statement processed.
```

Audit Trail Location

The audit trails from Oracle's AUDIT option can be stored in either the database or the operating system. The location is determined at database startup, based on the value of the INIT.ORA parameter AUDIT_TRAIL.

Database

All audit information stored within the database is stored in the table SYS.AUD$, which by default is stored in the SYSTEM tablespace. In an audited database, it is important to make sure that audit trail information is not erased. To prevent that from happening, the DBA should limit the users who can actually write information to this table to SYS.

Operating System

By directing the database to archive its information at the operating system level, the DBA enables Oracle to store its audit trail information in the same location as the audit information generated by the operating system.

There are a few drawbacks to consider, though. Because the data is no longer in a table, non-database utilities are needed to access it. Likewise, depending on the amount of information being audited, the database can produce double, triple, or even further increase the amount of information that the operating system currently produces.

****