28

Introduction to Auditing

Oracle Database provides the industry's most comprehensive auditing capability, enabling the capture of detailed information relating to who, what, when the action was performed, and the associated context with the activity which generated the audit record.

- What Is Auditing?
  Database auditing is the most accurate record of any database activity. Auditing tracks the use of privileges, activities of highly privileged users, access to sensitive data, actions performed on database objects and modifications made to database settings.
- Why Is Auditing Used?
  You typically use auditing to monitor user activity.
- Best Practices for Auditing
  You should follow best practices guidelines for auditing.
- Unified Auditing and Its Benefits
  Unified auditing was introduced in Oracle Database 12c with significant enhancements to auditing functionality.
- Who Can Perform Auditing?
  Oracle provides two roles for users who perform auditing: `AUDIT_ADMIN` and `AUDIT_VIEWER`, to enable separation of duties.
- Handling the Desupport of Traditional Auditing
  Traditional auditing is desupported, starting in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.
- Unified Auditing in a Multitenant Environment
  You can apply audit settings to individual PDBs or to the CDB, depending on the type of policy.
- Auditing in a Distributed Database
  Auditing is site autonomous in that a database instance audits only the statements issued by directly connected users.

Related Topics

- Guidelines for Auditing
  Oracle provides guidelines for auditing.

28.1 What Is Auditing?

Database auditing is the most accurate record of any database activity. Auditing tracks the use of privileges, activities of highly privileged users, access to sensitive data, actions performed on database objects and modifications made to database settings.

Database auditing has steadily increased in both capability and popularity over the past decade, and today is mandatory in most organizations. They need to audit not only to detect any unauthorized use, but also to ensure that they comply with different regulations, such as General Data Protection Regulation (GDPR), Payment Card Industry (PCI), California Consumer Privacty Act (CCPA), and other privacy regulations across the globe.

Database auditing is typically used for the following **use cases**:

- Monitoring activities of privileged database administrators
- Detecting unauthorized activity on sensitive assets
- Assisting with investigations of data breaches or other suspicious activity
- Providing proof of monitoring critical assets to auditors
- Providing reports on changes to the database environment to auditors

Database auditing is the most accurate record of any database activity, not just from connections happening over the wire but also through direct local logins, recursive SQLs, dynamic SQLs, and stored procedures.

An audit record gives you full execution context including details of the operation, type of SQL statement executed, use of powerful system privileges, operation performed, database object involved in the operation, and other session details that are useful for forensic analysis.

You can configure auditing for both successful and failed operations, however, parse or syntax errors are not audited. Additionally, you can include or exclude specific users from the audit. Auditing is independent of external connection factors like the network encryption, the access path, or the user, and is always available as a reliable source of actual events that have happened.

You can audit individual actions of the pluggable database (PDB) or individual actions in the entire multitenant container database (CDB). In addition to auditing the standard activities the database provides, auditing can include activities from Oracle Database Real Application Security, Oracle Automatic Storage Management, Oracle Recovery Manager, Oracle Data Pump, Oracle Machine Learning for SQL, Oracle Database Vault, Oracle Label Security, and Oracle SQL*Loader direct path events.

Oracle Database auditing has been enhanced with each successive release of the database. Traditional auditing was the historical database auditing approach in releases earlier than Oracle Database 12c. Unified auditing was introduced subsequently in Oracle Database 12c, where auditing functionality was significantly enhanced to provide a robust and highly customizable framework that can be fine-tuned to address specific security requirements. **Traditional auditing** is desupported in Oracle Database 23ai and Oracle recommends that you use **unified auditing**.

Oracle Database auditing (unified auditing) is enabled by default. Follow the below set of guidelines to ensure your database auditing requirements meet the most common security and compliance needs:

1. **Make the most of the *always-on mandatory audits*.** Certain security-sensitive database activities are mandatorily audited in the Oracle Database and cannot be disabled. Do not duplicate them.
2. **Use the *predefined unified audit policies*.** Oracle Database provides predefined unified audit policies that encompass the standard audit settings that most regulatory agencies require.
    a. The `ORA_SECURECONFIG` and `ORA_LOGIN_LOGOUT` pre-defined unified audit policies are automatically enabled in most deployments. Ensure to enable them if you have not done so already.
    b. Autonomous databases provides numerous predefined audit policies that are enabled by default.
    c. If you are using Oracle Data Safe or Oracle Audit Vault and Database Firewall (AVDF) to monitor the database activity across your enterprise, these products also offer a number of predefined audit policies to provision with a single click.

3. **Create *custom audit policies* for specialized use cases.** Oracle Database provides the flexibility to create and enable custom audit policies for your specific needs. You can either define unified audit policies or fine-grained audit policies for specialized needs.

Database auditing is frequently augmented with Database Activity Monitoring (DAM) solutions that collect and store the audit data for alert generation, analysis, and reporting. Oracle Database security products that offer DAM solutions include Oracle Data Safe, and Oracle Audit Vault and Database Firewall (AVDF).

Related Topics

- Activities That Are Mandatorily Audited
  Certain security sensitive database activities are always audited and such audit configurations cannot be disabled.
- Auditing Activities with the Predefined Unified Audit Policies
  Oracle Database provides predefined unified audit policies that cover commonly used security-relevant audit settings.
- Creating Custom Unified Audit Policies
  Oracle Database provides the flexibility to create and manage custom unified audit policies for your specialized needs.
- Value-Based Auditing with Fine-Grained Audit Policies
  Fine-grained auditing enables you to perform value-based auditing to audit access to certain rows based on values in specific columns.
- *Oracle Database PL/SQL Packages and Types Reference*
- Handling the Desupport of Traditional Auditing
  Traditional auditing is desupported, starting in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

28.2 Why Is Auditing Used?

You typically use auditing to monitor user activity.

Auditing can be used to accomplish the following:

- **Enable accountability for actions.** These include actions taken in a particular schema, table, or row, or affecting specific content.
- **Deter users (or others, such as intruders) from inappropriate actions based on their accountability.**
- **Investigate suspicious activity.** For example, if a user is deleting data from tables, then a security administrator can audit all connections to the database and all successful and unsuccessful deletions of rows from all tables in the database.
- **Notify an auditor of the actions of an unauthorized user.** For example, an unauthorized user could be changing or deleting data, or the user has more privileges than expected, which can lead to reassessing user authorizations.
- **Support post-incident investigations.**
- **Monitor and gather data about specific database activities.** For example, the database administrator can gather statistics about which tables are being updated, how many logical I/Os are performed, or how many concurrent users connect at peak times.
- **Detect problems with an authorization or access control implementation.** For example, you can create audit policies that you expect will never generate an audit record because the data is protected in other ways. However, if these policies generate audit records, then you will know the other security controls are not properly implemented.

- **Address auditing requirements for compliance.** Regulations such as the following have common auditing-related requirements:
  - Sarbanes-Oxley Act
  - Health Insurance Portability and Accountability Act (HIPAA)
  - International Convergence of Capital Measurement and Capital Standards: a Revised Framework (Basel II)
  - Japan Privacy Law
  - European Union Directive on Privacy and Electronic Communications

Oracle recommends that you audit your databases. Auditing is an effective method of enforcing strong internal controls so that your site can meet its regulatory compliance requirements. This enables you to monitor business operations, and find abnormal access patterns.

Auditing can not only monitor the database activity of database users, but also nondatabase users. "Nondatabase users" refers to the typical application service accounts and they are identified in the database using the `CLIENT_IDENTIFIER` attribute. To audit this type of user, you can use either unified audit or fine-grained audit policy, or Oracle Database Real Application Security.

28.3 Best Practices for Auditing

You should follow best practices guidelines for auditing.

- **As a general rule, design your auditing strategy to collect the amount of information that you need to meet compliance requirements, but focus on activities that cause the greatest security concerns.** For example, auditing every table in the database is not practical, but auditing tables with columns that contain sensitive data, such as salaries, is. With both unified and fine-grained auditing, there are mechanisms you can use to design audit policies that focus on specific activities to audit.
- **Periodically archive and purge the audit trail data.** You can use the `DBMS_AUDIT_MGMT` package to purge audit records in several different ways. You should regularly review the collected audit records and establish a system for collecting and retaining audit records based on your site's retention policies. In addition to `DBMS_AUDIT_MGMT`, Oracle Data Safe and Oracle Audit Vault and Database Firewall provide features that enable you manage the archiving and purging of audit trail data.
- **Oracle recommends that you configure a different tablespace for the unified audit trail.** You can use the `DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION` procedure. Take note of the fact that for Oracle Database Standard Edition and Express Edition, you can only associate the tablespace for unified auditing once. You should perform this association before you generate any audit records for the unified audit trail. After you have associated the tablespace, you cannot modify it because partitioning is only supported on Oracle Database Enterprise Edition. This limitation does not exist for Enterprise Edition.

Related Topics

- Guidelines for Auditing
  Oracle provides guidelines for auditing.
- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.
- *Oracle Database PL/SQL Packages and Types Reference*

28.4 Unified Auditing and Its Benefits

Unified auditing was introduced in Oracle Database 12c with significant enhancements to auditing functionality.

Unified auditing enables you to capture audit records from the following sources, and writes the audit records into a **single consolidated unified audit trail**:

- Audit records (including `SYS` audit records) from unified audit policies and `AUDIT` settings
- Fine-grained audit records from the `DBMS_FGA` PL/SQL package
- Oracle Database Real Application Security audit records
- Oracle Recovery Manager audit records
- Oracle Database Vault audit records
- Oracle Label Security audit records
- Oracle Machine Learning for SQL records
- Oracle Data Pump
- Oracle SQL*Loader Direct Load
- Oracle XML DB HTTP and FTP protocol messages

The unified audit trail, which resides in a read-only table in the `AUDSYS` schema in the `SYSAUX` tablespace, makes this information available in a uniform format in the `UNIFIED_AUDIT_TRAIL` data dictionary view, and is available in both multitenant and Oracle Database Real Application Clusters environments. The unified audit trail also normalizes the audit record format, using standardized column names and data types across all audit sources. The consolidated, normalized unified audit trail simplifies collection, analysis, and management of audit records generated by different audit sources. Consistent formatting simplifies reporting and analysis of the audit data.

Unified auditing offers a **high degree of integrity of audit trail** by not allowing users to tamper with the audit trail. The unified audit trail is stored in the `AUDSYS` schema and no one is allowed to log in to that schema in the database. `AUD$UNIFIED` is a specialized table which allows only `INSERT` activity. Any attempt to directly truncate, delete or update contents of the `AUD$UNIFIED` table fail, and will generate audit records. You can use the the built-in audit data management `DBMS_AUDIT_MGMT` package to manage audit data. Additionally, you can encrypt the audit tablespace with Transparent Data Encryption (TDE). You can protect the unified audit table with an Oracle Database Vault realm.

With unified auditing, **audit configuration is much simpler and focused for your needs**. You can create named audit policies once and enforce them in multiple dimensions (for example, on users and roles), giving you a lot more flexibility and simplicity. You can selectively audit to capture relevant activity with unified audit. Audit conditions can be based on application contexts, session contexts, and built-in functions. The `ONLY TOPLEVEL` clause of the `CREATE AUDIT POLICY` statement helps audit only the SQL statements that are directly issued by an end user, thus focusing only on end-user-initiated actions on sensitive tables. Such configuration flexibility in unified audit helps fine-tune audit policies to collect audit data that is targeted to your needs.

Unified auditing provides different roles for separation of duties to manage and view the audit data: `AUDIT_ADMIN` and `AUDIT_VIEWER`.

For typical use cases of auditing privileged users or auditing key database operations with unified auditing, **the performance impact is so low** that it cannot even be measured due to

low audit volume spread throughout the week. You could begin to see performance impact of 1 percent when the audit load increases to a few hundred audit events per second. For most use cases, you are not going to see overhead beyond this, but for cases where organizations want to audit application usage, it is best to tune the audit policies. Internal performance tests using a TPC-C mixed application workload show that with unified audit, you may see a CPU overhead in mid-single digit when auditing up to 360,000 audit records/hour. For extreme audit loads up to 1,800,000 audit records per hour, the additional overhead is still in a single digit.

> **Note:**
>
> 1. When the database is writeable, audit records are written to the unified audit trail. If the database is not writable (typically occurs when the database is closed or is read-only as in Oracle Data GuardADG), the Oracle Database writes audit records to external operating system spillover `.BIN` files in the `$ORACLE_BASE/audit/$ORACLE_SID` directory. The audit data present in the `.BIN` files is also surfaced in the `UNIFIED_AUDIT_TRAIL` data dictionary view.

Related Topics

- *Oracle Database Reference*

28.5 Who Can Perform Auditing?

Oracle provides two roles for users who perform auditing: `AUDIT_ADMIN` and `AUDIT_VIEWER`, to enable separation of duties.

The privileges that these roles provide are as follows:

- **AUDIT_ADMIN role.** This role enables you to create unified and fine-grained audit policies; enable,disable or drop the created unified audit and fine-grained audit policies; view audit data; and manage the audit trail administration. This role also enables you to change audit policies or modify the audit trail (including purging old audit data). Grant this role only to trusted users. Note that user `SYS` has this role.

  The list of privileges `AUDIT_ADMIN` provides is as follows:

  - `NOAUDIT` statement *
  - `AUDIT POLICY` statement
  - `NOAUDIT POLICY` statements
  - `CREATE AUDIT POLICY` statement
  - `ALTER AUDIT POLICY` statement
  - `DROP AUDIT POLICY` statement
  - `DBMS_FGA` PL/SQL package execution
  - `DBMS_AUDIT_MGMT` PL/SQL package execution
  - Selecting the following audit trail tables and views:
    * `SYS.AUD$` table *
    * `SYS.USER_AUDIT_TRAIL` data dictionary view *
    * `SYS.CDB_AUDIT_TRAIL` data dictionary view *

    - * SYS.FGA_LOG$ table *
    - * SYS.DBA_FGA_AUDIT_TRAIL data dictionary view *
    - * SYS.CDB_FGA_AUDIT_TRAIL data dictionary view *
    - * SYS.DBA_COMMON_AUDIT_TRAIL data dictionary view *
    - * SYS.CDB_COMMON_AUDIT_TRAIL data dictionary view *
    - * SYS.X$UNIFIED_AUDIT_TRAIL dynamic performance view
    - * SYS.V$UNIFIED_AUDIT_TRAIL dynamic performance view
    - * SYS.GV$UNIFIED_AUDIT_TAIL dynamic performance view
    - * AUDSYS.AUD$UNIFIED
    - * AUDSYS.UNIFIED_AUDIT_TRAIL data dictionary view
    - * AUDSYS.CDB_UNFIIED_AUDIT_TRAIL data dictionary view
  - Ability to change the following system parameters by using the ALTER SYSTEM statement:
    - * AUDIT_FILE_DEST *
    - * AUDIT_TRAIL *
    - * AUDIT_SYS_OPERATIONS *
    - * AUDIT_SYSLOG_LEVEL *
    - * UNIFIED_AUDIT_SYSTEMLOG
    - * UNIFIED_AUDIT_COMMON_SYSTEMLOG
  - Ability to change audit policies or modify the audit trail (including purging old audit data)
- **AUDIT_VIEWER role.** This role enables users to view and analyze audit data. It provides the EXECUTE privilege on the DBMS_AUDIT_UTIL PL/SQL package. The kind of user who needs this role is typically an external auditor. An auditor can view audit data after being granted the AUDIT_VIEWER role. If your users only need to query the views but not create audit policies, then grant them the AUDIT_VIEWER role. Note that user SYS has this role.

  The list of privileges AUDIT_VIEWER provides is as follows:
  - SYS.AUD$ table *
  - SYS.USER_AUDIT_TRAIL data dictionary view *
  - SYS.CDB_AUDIT_TRAIL data dictionary view *
  - SYS.FGA_LOG$ table *
  - SYS.DBA_FGA_AUDIT_TRAIL data dictionary view *
  - SYS.CDB_FGA_AUDIT_TRAIL data dictionary view *
  - SYS.DBA_COMMON_AUDIT_TRAIL data dictionary view *
  - SYS.CDB_COMMON_AUDIT_TRAIL data dictionary view *
  - SYS.X$UNIFIED_AUDIT_TRAIL dynamic performance view
  - SYS.V$UNIFIED_AUDIT_TRAIL dynamic performance view
  - SYS.GV$UNIFIED_AUDIT_TAIL dynamic performance view

  - AUDSYS.AUD$UNIFIED
  - AUDSYS.UNIFIED_AUDIT_TRAIL data dictionary view
  - AUDSYS.CDB_UNFIIED_AUDIT_TRAIL data dictionary view

* Deprecated; used in traditional auditing. Traditional auditing is desupported starting in Oracle Database 23ai, but if you still have traditional audit settings, they are accessible.

Related Topics

- Handling the Desupport of Traditional Auditing
  Traditional auditing is desupported, starting in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

28.6 Handling the Desupport of Traditional Auditing

Traditional auditing is desupported, starting in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

If you used traditional auditing in previous releases, when you upgrade to Oracle Database 23ai, the existing traditional audit settings will continue to be honored and audit records will continue to be generated into their respective audit trails. However, you cannot create new traditional audit settings or update existing traditional audit settings. You can only delete the existing traditional audit settings.

Oracle strongly recommends that you transition from traditional audit configurations to unified audit policies as soon as possible. In most cases, the transition is simple. Oracle Database has always-on mandatory audits to ensure security-sensitive database activities are always audited. Oracle Database also provides a set of predefined unified audit policies to help you get started. If you have upgraded your Oracle database installation from release 11g, then at a minimum, you should enable the following predefined policies, which address the most common security and compliance needs

- Secure configuration audit options (ORA_SECURECONFIG), such as audits of the ALTER ANY TABLE system privilege
- Logon and logoff failures (ORA_LOGIN_LOGOUT)

All new Oracle databases, created from release 12.2 and later, have the ORA_SECURECONFIG pre-defined unified audit policy enabled by default. Starting in release 23ai, the ORA_LOGIN_LOGOUT pre-defined unified audit policy is available and enabled by default. During database upgrades, these predefined unified audit policies are not enabled.

If you have highly customized traditional audit settings, then you have the following choices to transition them to unified audit policies:

- Create custom unified audit policies by using the rich features of unified audit to make your audit polices more conditional, selective, and focused. For example, you can create policies that audit actions on tables or databases, audit application context values, and filter the audit results to show only top level activities. You can create conditions to further filter the unified audit results. You can also create policies that are specific to many other Oracle features, such as SQL Firewall, Oracle Database Vault, Oracle Label Security, and so on.
- If you are unfamiliar with the syntax that is involved in creating unified audit policies, then use the syntax converter script that is available in My Oracle Support note 2909718.1. This creates .sql scripts to convert your current traditional audit configuration settings into syntactically correct unified audit policies. After you have completed the conversion, Oracle strongly recommends that you examine the policies and incorporate the various features of

unified auditing, such as creating conditions or auditing application context values, before you enable your policies.

After you have completed converting your traditional audit settings to unified audit policies, then carefully examine this generated script before you execute it to enable the unified audit policies and remove the existing traditional audit configurations.

For additional information about unified audit best practices, see the Oracle technical report Oracle Database Unified Audit: Best Practice Guidelines.

> **Note:**
>
> Unified auditing does not depend on the initialization parameters that were used by traditional auditing. See the Feature column in Considerations for Transitioning from Traditional to Unified Auditing for a list of these initialization parameters.

Related Topics

- Auditing Activities with the Predefined Unified Audit Policies
  Oracle Database provides predefined unified audit policies that cover commonly used security-relevant audit settings.
- Creating Custom Unified Audit Policies
  Oracle Database provides the flexibility to create and manage custom unified audit policies for your specialized needs.
- Syntax for Creating a Custom Unified Audit Policy
  To create a custom unified audit policy, you must use the `CREATE AUDIT POLICY` statement.
- Syntax for Creating a Custom Unified Audit Policy
  To create a custom unified audit policy, you must use the `CREATE AUDIT POLICY` statement.

28.7 Unified Auditing in a Multitenant Environment

You can apply audit settings to individual PDBs or to the CDB, depending on the type of policy.

Each PDB, including the root, has its own unified audit trail.

- **Unified audit policies created with the CREATE AUDIT POLICY and AUDIT statements:** You can create policies for both the root and individual PDBs.
- **Audit records written to the syslog:** On UNIX platforms, you can set the `UNIFIED_AUDIT_COMMON_SYSTEMLOG` initialization parameter in the CDB root to enable certain unified audit trail columns to be written to SYSLOG. On both Windows and UNIX, you can set the `UNIFIED_AUDIT_SYSTEMLOG` parameter in both the root and PDB level.
- **Fine-grained audit policies:** You can create policies for individual PDBs only, not the root.
- **Purging the audit trail:** You can perform purge operations for both the root and individual PDBs.

Related Topics

- Auditing in a Multitenant Deployment
  You can create unified audit policies for individual PDBs and in the root.
- Enabling SYSLOG and Windows Event Viewer Captures for the Unified Audit Trail
  You can write a subset of unified audit trail records to the UNIX SYSLOG or to the Windows Event Viewer.

- Creating Fine-Grained Audit Policies
  The `DBMS_FGA.ADD_POLICY` procedure creates a fine-grained audit policy.
- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.

28.8 Auditing in a Distributed Database

Auditing is site autonomous in that a database instance audits only the statements issued by directly connected users.

A local Oracle Database node cannot audit actions that take place in a remote database.