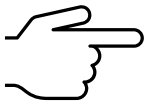
# AUDITING DATABASES

DAILY, WEEKLY AND MONTHLY

Asfaw Gedamu Haileselasie



Caution: Please use the commands with care, try them on test environments first.

## A. Using SQL Scripts

These scripts will give you a good overview of the activities that have taken place in your database on a daily, weekly, and monthly basis. You can use this information to identify any potential security risks or performance issues.

You can modify these scripts to fit your specific needs. For example, you can add filters to the scripts to only include certain users, tables, or stored procedures. You can also use the scripts to generate reports that you can share with other members of your team.

Here are some additional tips for auditing database activities:

- Run the audit scripts regularly, such as once a day, once a week, or once a month.
- Save the results of the audit scripts in a secure location.
- Review the results of the audit scripts for any suspicious activity.
- Take action to address any suspicious activity that you find.

By following these tips, you can help to protect your database from unauthorized access and malicious activity.

## Daily audit

```
-- Get a list of all database users who have logged in today

SELECT
    user_name,
    login_time
FROM sys.dm_exec_sessions
WHERE login_time >= CURRENT_DATE - 1
ORDER BY login_time DESC;

-- Get a list of all tables that have been modified today

SELECT
```

```
    table_name,
    last_update
FROM information_schema.tables
WHERE last_update >= CURRENT_DATE - 1
ORDER BY last_update DESC;
-- Get a list of all stored procedures that have been executed
today

SELECT
    procedure_name,
    last_execution
FROM sys.procedures
WHERE last_execution >= CURRENT_DATE - 1
ORDER BY last_execution DESC;

-- Get a list of all logins that have failed today

SELECT
    login_name,
    failed_login_attempts
FROM sys.logins
WHERE failed_login_attempts > 0
ORDER BY failed_login_attempts DESC;
```

## Weekly audit

```
-- Get a list of all database users who have logged in this week

SELECT
    user_name,
    login_time
FROM sys.dm_exec_sessions
WHERE login_time >= CURRENT_DATE - 7
ORDER BY login_time DESC;

-- Get a list of all tables that have been modified this week

SELECT
    table_name,
    last_update
FROM information_schema.tables
WHERE last_update >= CURRENT_DATE - 7
ORDER BY last_update DESC;

-- Get a list of all stored procedures that have been executed
this week
```

```
SELECT
    procedure_name,
    last_execution
FROM sys.procedures
WHERE last_execution >= CURRENT_DATE - 7
ORDER BY last_execution DESC;

-- Get a list of all logins that have failed this week

SELECT
    login_name,
    failed_login_attempts
FROM sys.logins
WHERE failed_login_attempts > 0
ORDER BY failed_login_attempts DESC;
```

## Monthly audit

```
-- Get a list of all database users who have logged in this
month

SELECT
    user_name,
    login_time
FROM sys.dm_exec_sessions
WHERE login_time >= CURRENT_DATE - 30
ORDER BY login_time DESC;

-- Get a list of all tables that have been modified this month

SELECT
    table_name,
    last_update
FROM information_schema.tables
WHERE last_update >= CURRENT_DATE - 30
ORDER BY last_update DESC;

-- Get a list of all stored procedures that have been executed
this month

SELECT
    procedure_name,
    last_execution
FROM sys.procedures
WHERE last_execution >= CURRENT_DATE - 30
```

```
ORDER BY last_execution DESC;

-- Get a list of all logins that have failed this month

SELECT
    login_name,
    failed_login_attempts
FROM sys.logins
WHERE failed_login_attempts > 0
ORDER BY failed_login_attempts DESC;
```

## B. Using Shell Scripts

Altornatively, you may use the following shell scripts on `sqlplus`.

### Daily audit

```
# Get a list of all database users who have logged in today
echo "Getting a list of all database users who have logged in
today..."

sqlplus -S user/password @daily_audit.sql

# Get a list of all tables that have been modified today
echo "Getting a list of all tables that have been modified
today..."

sqlplus -S user/password @daily_audit_tables.sql

# Get a list of all stored procedures that have been executed
today
echo "Getting a list of all stored procedures that have been
executed today..."

sqlplus -S user/password @daily_audit_procedures.sql

# Get a list of all logins that have failed today
echo "Getting a list of all logins that have failed today..."

sqlplus -S user/password @daily_audit_logins.sql
```

### Weekly audit

```
# Get a list of all database users who have logged in this week
echo "Getting a list of all database users who have logged in
this week..."

sqlplus -S user/password @weekly_audit.sql

# Get a list of all tables that have been modified this week
echo "Getting a list of all tables that have been modified this
week..."

sqlplus -S user/password @weekly_audit_tables.sql

# Get a list of all stored procedures that have been executed
this week
echo "Getting a list of all stored procedures that have been
executed this week..."

sqlplus -S user/password @weekly_audit_procedures.sql

# Get a list of all logins that have failed this week
echo "Getting a list of all logins that have failed this
week..."

sqlplus -S user/password @weekly_audit_logins.sql
```

## Monthly audit

```
# Get a list of all database users who have logged in this month
echo "Getting a list of all database users who have logged in
this month..."

sqlplus -S user/password @monthly_audit.sql

# Get a list of all tables that have been modified this month
echo "Getting a list of all tables that have been modified this
month..."

sqlplus -S user/password @monthly_audit_tables.sql

# Get a list of all stored procedures that have been executed
this month
echo "Getting a list of all stored procedures that have been
executed this month..."
```

```
sqlplus -S user/password @monthly_audit_procedures.sql

# Get a list of all logins that have failed this month
echo "Getting a list of all logins that have failed this
month..."

sqlplus -S user/password @monthly_audit_logins.sql
```

These scripts will run the same SQL scripts that I mentioned earlier, but they will be run from a Linux shell script. This allows you to run the scripts from a cron job or other scheduling mechanism.

To run the scripts, you will need to save them as `daily_audit.sql`, `weekly_audit.sql`, and `monthly_audit.sql`. You will also need to create a user account with the appropriate permissions to run the scripts.

Once you have done that, you can run the scripts by running the following commands:

Bash

```
# Run the daily audit script
./daily_audit.sh

# Run the weekly audit script
./weekly_audit.sh

# Run the monthly audit script
./monthly_audit.sh
```

The scripts will output the results of the audit to the console. You can also save the results to a file by redirecting the output of the script to a file.

For example, to save the results of the daily audit to a file called `daily_audit.log`, you would run the following command:

Bash

```
./daily_audit.sh > daily_audit.log
```