

11

Oracle Database Security

Objectives

After completing this lesson you should be able to do the following:

- **Apply the principal of least privilege**
- **Manage default user accounts**
- **Implement standard password security features**
- **Audit database activity**

Database Security

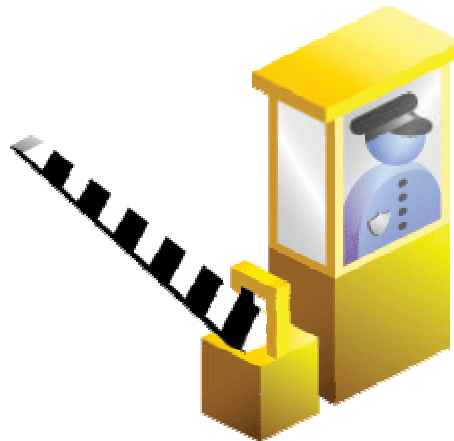
A secure system ensures the confidentiality of the data it contains. There are several aspects of security:

- **Restricting access to data and services**
- **Authenticating users**
- **Monitoring for suspicious activity**



Apply the Principle of Least Privilege

- **Protect the data dictionary**
- **Revoke unnecessary privileges from PUBLIC**
- **Restrict the directories accessible by users**
- **Limit users with administrative privileges**
- **Restrict remote database authentication**



Protect the Data Dictionary

- Protect the data dictionary by ensuring the following initialization parameter is set to **FALSE**:

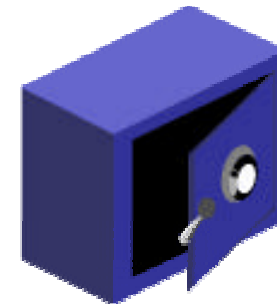
```
O7_DICTIONARY_ACCESSIBILITY = FALSE
```

- This configuration prevents users with **ANY TABLE** system privileges from accessing data dictionary base tables.
- A **FALSE** setting also prevents user **SYS** from logging in as anything other than **SYSDBA**
- The default value of this parameter is **FALSE**. If you find it set to **TRUE**, ensure there is a good business reason.

Revoke Unnecessary Privileges from PUBLIC

- Revoke all unnecessary privileges and roles from the database server user group PUBLIC.
- Many built-in packages grant `EXECUTE` to PUBLIC.
- Execute on the following packages should usually be revoked from PUBLIC:

- `UTL_SMTP`
- `UTL_TCP`
- `UTL_HTTP`
- `UTL_FILE`
- `DBMS_OBFUSCATION_TOOLKIT`



- **Example:**

```
SQL> REVOKE execute ON utl_file FROM PUBLIC;
```

Restrict the Operating System Directories Accessible by the User

The UTL_FILE_DIR configuration parameter:

- Designates which directories are available for PL/SQL file I/O
- Enables database users to read or write from the listed directories on the database server

Initialization Parameters

Current **SPFile** Show SQL Revert Apply

The parameter values listed here are from the SPFILE `/u01/app/oracle/product/10.1.0/dbs/spfileorcl.ora`

Filter Go
Filter on a name or partial name

☐ Apply changes in SPFile mode to the current running instance(s). For static parameters, you must restart the database.

Reset

Select	Name	Help	Revisions	Value	Type	Basic	Dynamic	Category
<input checked="" type="radio"/>	utl_file_dir			<code> '/oracle/stage1','/oracle/stage2','/oracle/stage3'</code>	String			PL/SQL

Limit Users with Administrative Privileges

- **Restrict the following types of privileges:**
 - Grants of system and object privileges
 - SYS-privileged connections: SYSDBA and SYSOPER
 - DBA-type privileges, such as DROP ANY TABLE
 - Run-time permissions
- **Example: List all users with the DBA role:**

```
SQL> SELECT grantee FROM dba_role_privs
      2 WHERE granted_role = 'DBA';
GRANTEE
-----
SYS
SYSTEM
```


Disable Remote Operating System Authentication

- Remote authentications should be used only when you trust all clients to appropriately authenticate users.
- Remote authentication process:
 - The database user is authenticated externally.
 - The remote system authenticates the user.
 - The user logs on to the database without further authentication.
- To disable, ensure that the following instance initialization parameter is at its default setting:

```
REMOTE_OS_AUTHENT = FALSE
```

Manage Default User Accounts

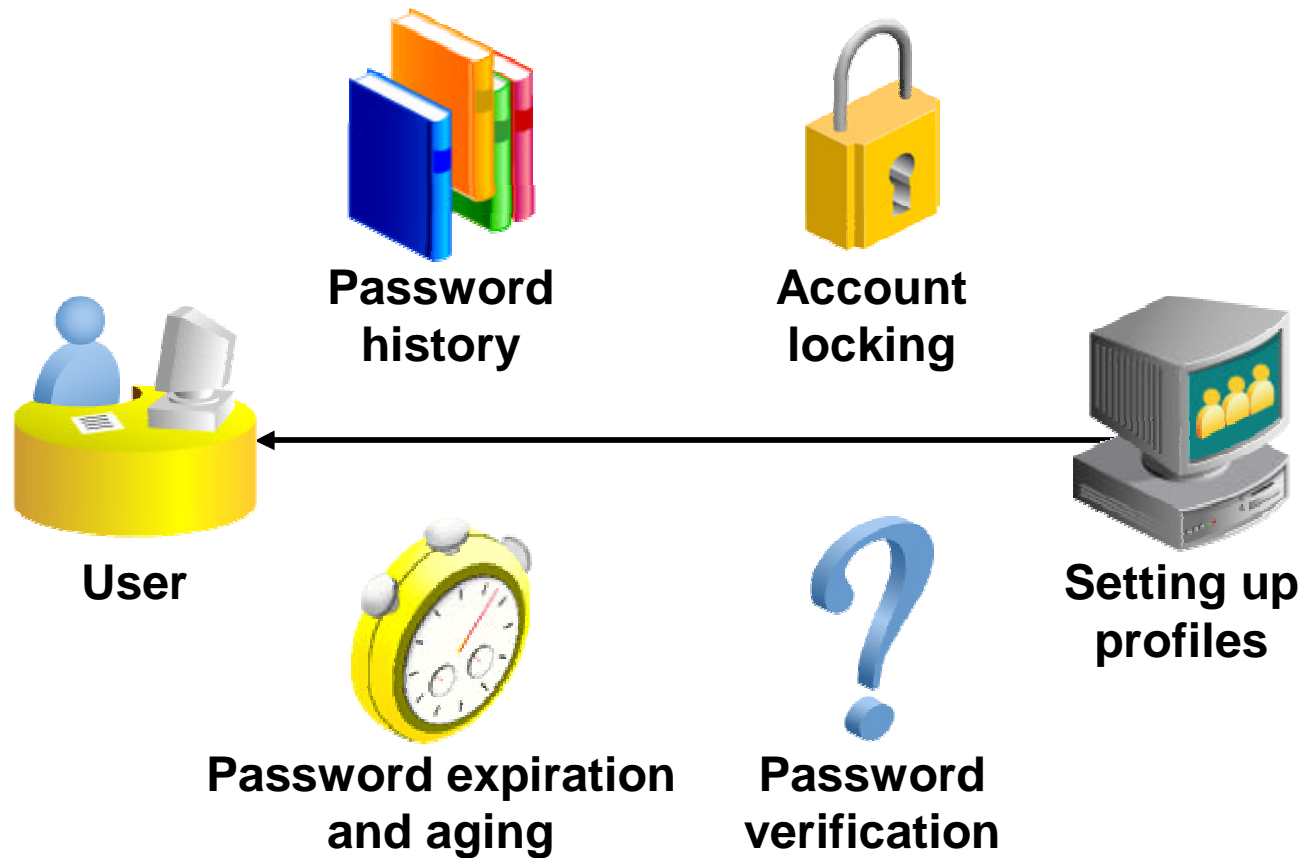
- **DBCA expires and locks all accounts, except:**
 - SYS
 - SYSTEM
 - SYSMAN
 - DBSNMP
- **For a manually created database, lock and expire any unused accounts.**

The screenshot shows the 'Edit User: CTXSYS' dialog box with the following fields and options:

- Name:** CTXSYS
- Profile:** DEFAULT (dropdown menu)
- Authentication:** Password (dropdown menu)
- * Enter Password:** [Redacted with dots]
- * Confirm Password:** [Redacted with dots]
- Password Status:** Expired
Enter and confirm a password to un-expire the password
- * Default Tablespace:** SYSAUX (with a key icon)
- Temporary Tablespace:** TEMP (with a key icon)
- Status:** ☒ Locked ☐ Unlocked

At the top right of the dialog are buttons for 'Show SQL', 'Revert', and 'Apply'. Below the title bar are tabs for 'General', 'Roles', 'System Privileges', 'Object Privileges', and 'Quotas'.

Implement Standard Password Security Features



Password Account Locking

Parameter	Description
FAILED_LOGIN_ATTEMPTS	Number of failed login attempts before lockout of the account
PASSWORD_LOCK_TIME	Number of days the account is locked after the specified number of failed login attempts



Password Expiration and Aging

Parameter	Description
<code>PASSWORD_LIFE_TIME</code>	Lifetime of the password in days after which the password expires
<code>PASSWORD_GRACE_TIME</code>	Grace period in days for changing the password after the first successful login after the password has expired



Password History

Parameter	Description
<code>PASSWORD_REUSE_TIME</code>	Number of days before a password can be reused
<code>PASSWORD_REUSE_MAX</code>	Number of password changes required before the current password can be reused



Password Verification

Parameter	Description
<code>PASSWORD_VERIFY_ FUNCTION</code>	A PL/SQL function that makes a password complexity check before a password is assigned

Password verification functions must:

- Be owned by the `sys` user
- Return a Boolean value (true or false)



Supplied Password Verification Function:

VERIFY_FUNCTION

The supplied password verification function enforces password restrictions where the:

- Minimum length is four characters
- Password cannot be equal to username
- Password must have at least one alphabetic, one numeric, and one special character
- Password must differ from the previous password by at least three letters



Creating a Password Profile

Create Profile

Show SQLCancelOK

GeneralPassword

Password

Expire in (days)90

Lock (days past expiration)10

History

Number of passwords to keepUNLIMITED

Number of days to keep for120

Complexity

Complexity functionVERIFY_FUNCTION

Failed Login

Number of failed login attempts to lock after3

Number of days to lock for5/1440

Assigning Users to a Password Profile

Edit User: NGREENBERG

[Show SQL](#) [Revert](#) [Apply](#)

[General](#) [Roles](#) [System Privileges](#) [Object Privileges](#) [Quotas](#) [Consumer Groups](#) [Proxy Users](#)

Name **NGREENBERG**


Profile **CUSTOMPROFILE** ▼


Authentication **Password** ▼

* Enter Password

* Confirm Password

☐ Expire Password now

* Default Tablespace 

Temporary Tablespace 

Status ☐ Locked ☒ Unlocked

Monitoring for Suspicious Activity

Monitoring or auditing should be an integral part of your security procedures.

Oracle's built-in audit tools include:

- **Database auditing**
- **Value-based auditing**
- **Fine-grained auditing (FGA)**

Audit Tool Comparisons

Type of Audit	What Is Audited?	What Is in the Audit Trail?
Standard database auditing	Privilege use including object access	Fixed set of data
Value-based auditing	Data changed by DML statements	Administrator defined
Fine-grained auditing (FGA)	SQL statements (insert, update, delete, and select) based on content	Fixed set of data including the SQL statement

Standard Database Auditing

Enabled through the `AUDIT_TRAIL` parameter

- **NONE: Disables collection of audit records**
- **DB: Enables auditing with records stored in the database**
- **OS: Enables auditing with records stored in the operating system audit trail**

Can audit:

- **Login events**
- **Exercise of system privileges**
- **Exercise of object privileges**
- **Use of SQL statements**

Specifying Audit Options

- **SQL statement auditing**

```
AUDIT table;
```

- **System privilege auditing (nonfocused and focused)**

```
AUDIT select any table, create any trigger;  
AUDIT select any table BY hr BY SESSION;
```

- **Object privilege auditing (nonfocused and focused)**

```
AUDIT ALL on hr.employees;  
AUDIT UPDATE,DELETE on hr.employees BY ACCESS;
```

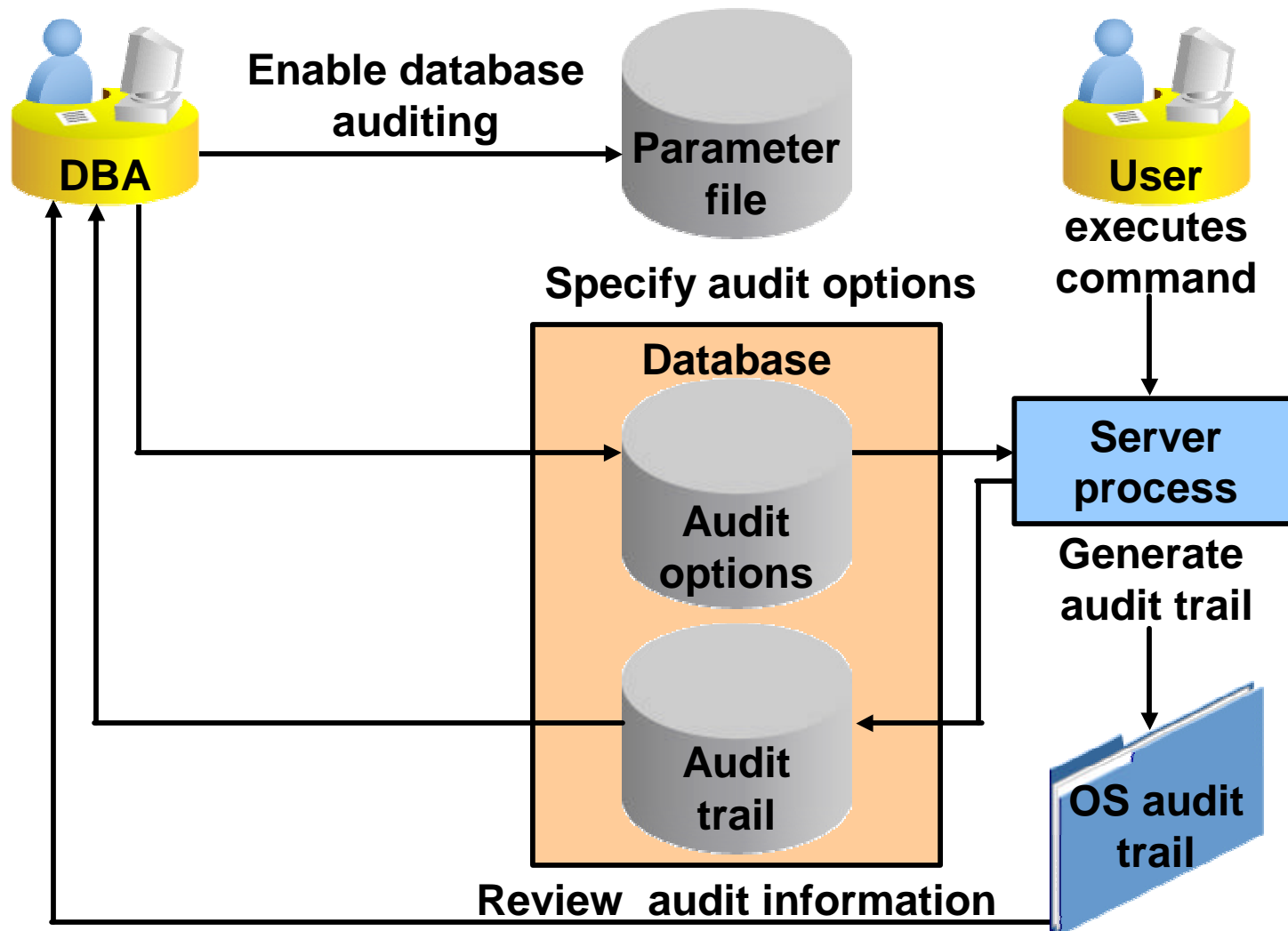
- **Session auditing**

```
AUDIT session whenever not successful;
```

Viewing Auditing Options

Data Dictionary View	Description
ALL_DEF_AUDIT_OPTS	Default audit options
DBA_STMT_AUDIT_OPTS	Statement auditing options
DBA_PRIV_AUDIT_OPTS	Privilege auditing options
DBA_OBJ_AUDIT_OPTS	Schema object auditing options

Standard Database Auditing

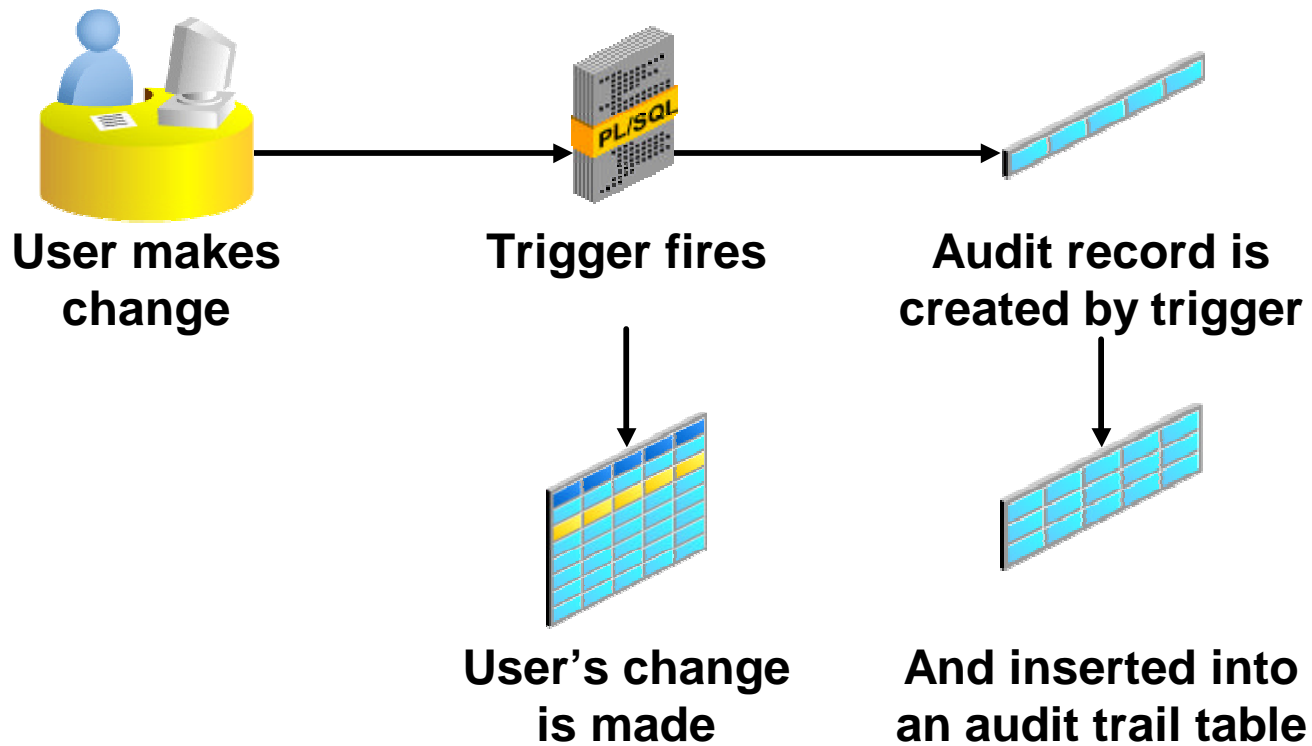


ORACLE

Viewing Auditing Results

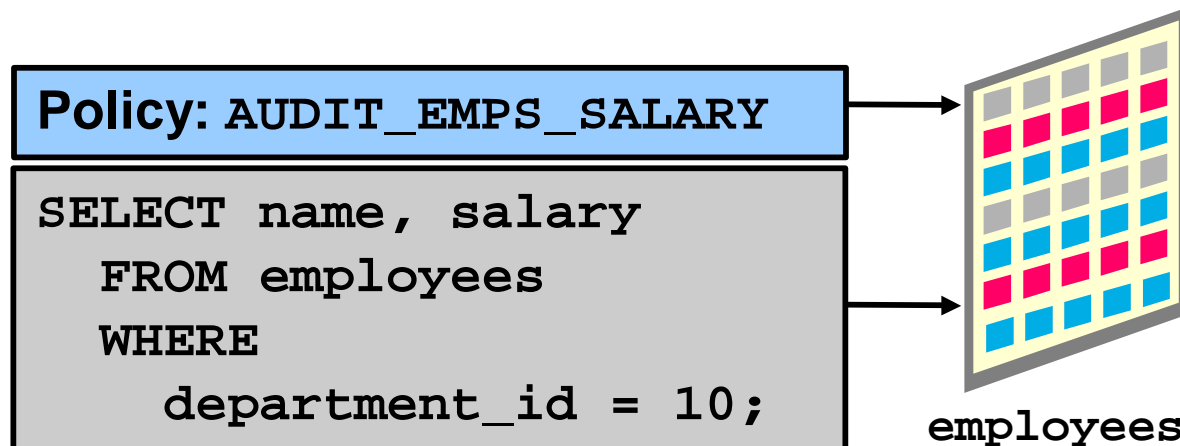
Audit Trail View	Description
DBA_AUDIT_TRAIL	All audit trail entries
DBA_AUDIT_EXISTS	Records for AUDIT EXISTS/NOT EXISTS
DBA_AUDIT_OBJECT	Records concerning schema objects
DBA_AUDIT_SESSION	All connect and disconnect entries
DBA_AUDIT_STATEMENT	Statement auditing records

Value-Based Auditing



Fine-Grained Auditing (FGA)

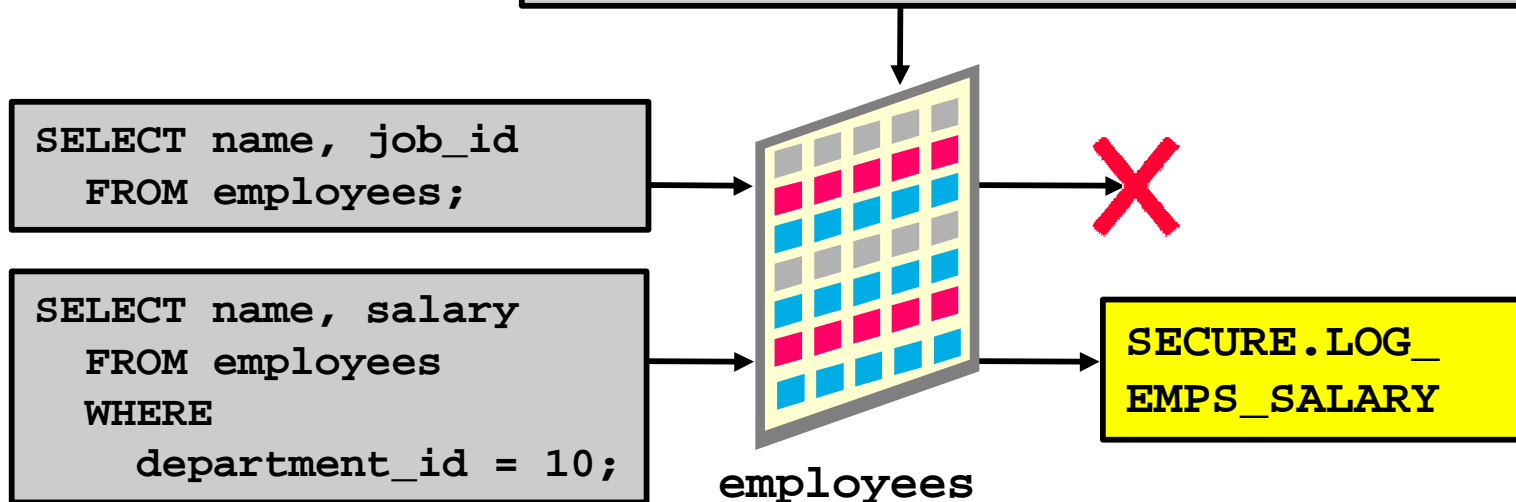
- Monitors data access based on content
- Audits `SELECT` or `INSERT`, `UPDATE`, `DELETE`
- Can be linked to a table or view
- May fire a procedure
- Is administered with the `DBMS_FGA` package



FGA Policy

- **Defines:**
 - Audit criteria
 - Audit action
- **Is created with**
DBMS_FGA
.ADD_POLICY

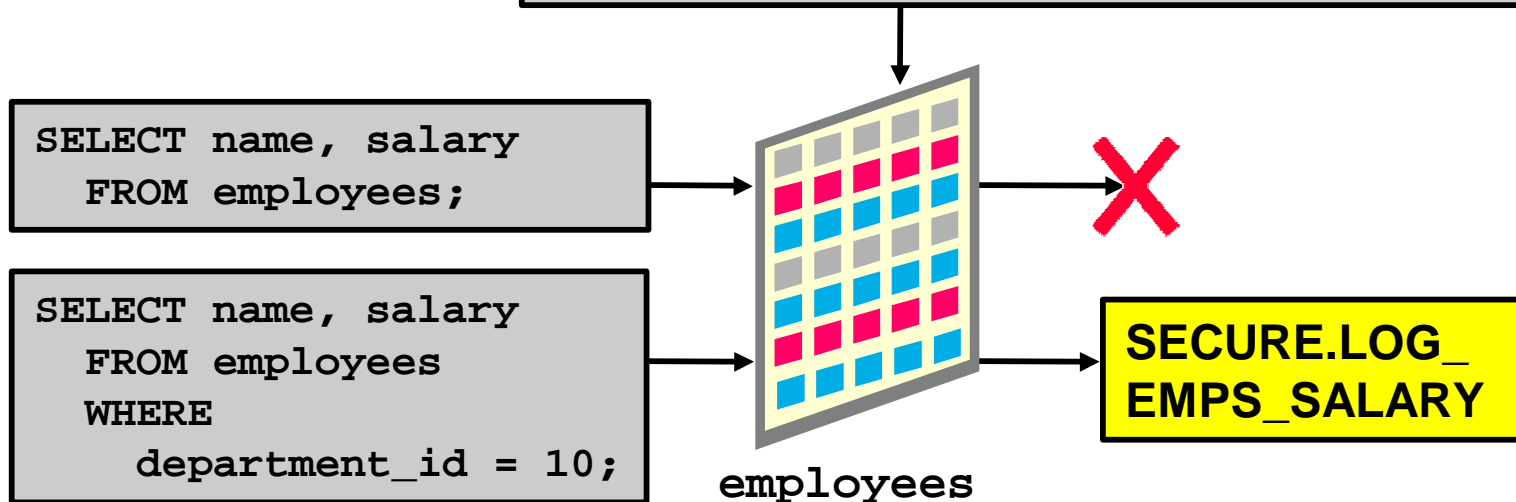
```
dbms_fga.add_policy (  
  object_schema => 'hr',  
  object_name   => 'employees',  
  policy_name   => 'audit_emps_salary',  
  audit_condition=> 'dept_id=10',  
  audit_column  => 'salary',  
  handler_schema=> 'secure',  
  handler_module=> 'log_emps_salary',  
  enable        => TRUE,  
  statement_types=> 'select' );
```



FGA Policy

- **Defines**
 - Audit criteria
 - Audit action
- **Is created with**
DBMS_FGA
.ADD_POLICY

```
dbms_fga.add_policy (  
  object_schema => 'hr',  
  object_name   => 'employees',  
  policy_name   => 'audit_emps_salary',  
  audit_condition  
    => 'department_id = 10',  
  audit_column  => 'salary',  
  handler_schema=> 'secure',  
  handler_module=> 'log_emps_salary',  
  enable       => TRUE );
```



DBMS_FGA Package

Subprogram	Description
ADD_POLICY	Creates an audit policy using the supplied predicate as the audit condition
DROP_POLICY	Drops an audit policy
ENABLE_POLICY	Enables an audit policy
DISABLE_POLICY	Disables an audit policy

Enabling and Disabling an FGA Policy

- **Enable a policy:**

```
dbms_fga.enable_policy (  
  object_schema => 'hr',  
  object_name   => 'employees',  
  policy_name   => 'audit_emps_salary' );
```

- **Disable a policy:**

```
dbms_fga.disable_policy (  
  object_schema => 'hr',  
  object_name   => 'employees',  
  policy_name   => 'audit_emps_salary' );
```

Dropping an FGA Policy

```
SQL> EXEC dbms_fga.drop_policy ( -  
> object_schema => 'hr', -  
> object_name    => 'employees', -  
> policy_name    => 'audit_emps_salary');
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```


Triggering Audit Events

- The following SQL statements cause an audit:

```
SELECT  count(*)  
  FROM hr.employees  
 WHERE department_id = 10  
    AND salary > v_salary;
```

```
SELECT salary  
  FROM hr.employees;
```

- The following statement does *not* cause an audit:

```
SELECT last_name  
  FROM hr.employees  
 WHERE department_id = 10;
```

Data Dictionary Views

View Name	Description
DBA_FGA_AUDIT_TRAIL	All FGA events
ALL_AUDIT_POLICIES	All FGA policies for objects the current user can access
DBA_AUDIT_POLICIES	All FGA policies in the database
USER_AUDIT_POLICIES	All FGA policies for objects in the current user schema

DBA_FGA_AUDIT_TRAIL

```
SQL> SELECT to_char(timestamp, 'YMMDDHH24MI')
2          AS timestamp,
3          db_user,
4          policy_name,
5          sql_bind,
6          sql_text
7  FROM dba_fga_audit_trail;
```

TIMESTAMP	DB_USER	POLICY_NAME	SQL_BIND

SQL_TEXT			

0201221740	SYSTEM	AUDIT_EMPS_SALARY	#1(4):1000
SELECT count(*)			
FROM hr.employees			
WHERE department_id = 10			
AND salary > :b1			

DBA_FGA_AUDIT_TRAIL

```
SQL> SELECT to_char(timestamp, 'YYMMDDHH24MI')
2          AS timestamp,
3          db_user,
4          policy_name,
5          sql_bind,
6          sql_text
7  FROM dba_fga_audit_trail;
```

TIMESTAMP	DB_USER	POLICY_NAME	SQL_BIND

SQL_TEXT			

0201221740	SYSTEM	AUDIT_EMPS_SALARY	#1(4):1000
SELECT count(*)			
FROM hr.employees			
WHERE department_id = 10			
AND salary > :b1			

FGA Guidelines

- To audit all statements, use a `null` condition.
- If you try to add a policy that already exists, error ORA-28101 is raised.
- The audited table or view must already exist when you create the policy.
- If the audit condition syntax is invalid, an ORA-28112 is raised when the audited object is accessed.
- If the audit column does not exist in the table, no rows are audited.
- If the event handler does not exist, no error is returned and the audit records is still created.

Auditing SYSDBA and SYSOPER Users

User's with SYSDBA or SYSOPER privileges can be connecting with the database closed.

- Audit trail must be stored outside of the database.
- Connect as SYSDBA or SYSOPER is always audited.
- Enable additional auditing of SYSDBA or SYSOPER actions with `audit_sys_operations`.
- Control audit trail with `audit_file_dest`. Default is:
 - `$ORACLE_HOME/rdbms/audit` (UNIX/Linux)
 - Windows Event Log (Windows)

Security Updates

- Oracle posts security alerts on the Oracle Technology Network Web site at:
<http://otn.oracle.com/deploy/security/alerts.htm>
- Oracle database administrators and developers can also subscribe to be notified about critical security alerts via e-mail by clicking the “Subscribe to Security Alerts Here” link.

Summary

In this lesson you should have learned how to:

- **Apply the principal of least privilege**
- **Manage default user accounts**
- **Implement standard password security features**
- **Audit database activity**

Practice 11-1 Overview: Database Security (Part 1)

Tasks:

- **Prevent the use of simple passwords**
- **Force accounts to lock for 10 minutes after four failed login attempts**
- **Exempt the application server login from forced password changes**
- **Audit unsuccessful attempts to connect to the database**

Practice 11-2 Overview: Database Security (Part 2)

Tasks:

- **Audit select on the SALARY column of the EMPLOYEES table**
- **Audit changes to the SALARY column of the EMPLOYEES table, capture:**
 - Old value
 - New value
 - User who made the change
 - What location the change was made from