11

Keamanan Database Oracle

By: Arif Basofii

Topik

- Menerapkan keutamaan tentang hak akses
- Mengatur user yang ada
- Menerapkan fungsi keamanan standard password
- Mengaudit aktivitas database

Industry Security Requirements

Security industri Oracle menerapkan:

- **Legal**: (dukungan lembaga luar)
  - Sarbanes-Oxley Act (SOX)
  - Health Information Portability and Accountability Act (HIPAA)
  - California Breach Law
  - UK Data Protection Act
- **Auditing**

  Menerapkan auditing dengan membedakan antara user biasa dan admin. (mana yg berhak/mana yg tidak)

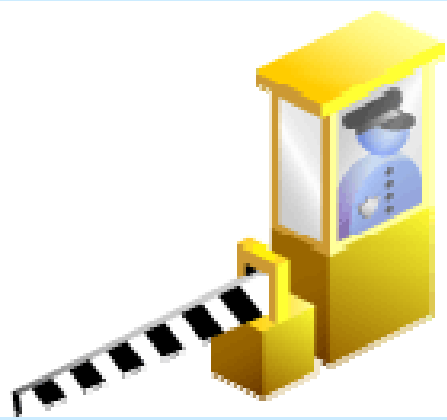
Separation of Responsibilities

- User dengan **privilege DBA** **harus terpercaya**
  - Tidak boleh adanya penyalahgunaan kepercayaan
  - Adanya **audit trails** untuk mengecek hasil auditnya
- Tanggung jawab DBA harus di share (tidak terpusat pd 1 orang, tp beberapa DBA)
- Accounts tidak pernah di share.
- **DBA** dan **system administrator** harus orang yang berbeda.
- Memisahkan antara tanggung jawab **operator** dan **DBA**.

Database Security

Ada 3 hal dalam keamana Database Oracle:

1. Membatasi akses ke data dan servis
2. Otentikasi users
3. Monitoring aktivitas yang mencurigakan



(1) Membatasi akses ke data dan servis: Principle of Least Privilege

1) Untuk membatasi akses ke data dan servis, dgn menerapkan penggunaan **privilege yg paling minim/kecil**.
   - Artinya: setiap orang jangan diberikan privilege yg **berlebihan**, tp diberikan privilege **secukupnya**.

Tindakan Prosedur:

- Install software yang dibutuhkan saja
- Aktifkan servis-servis yg dibutuhkan saja
- Berikan akses database dan OS pada orang-orang yg berhak
- Batasi akses ke account root atau administrator
- Batasi akses untuk penggunaan account SYSDBA dan SYSOPER
- Batasi akses user ke database objek tertentu yg berhubungan dgn pekerjaanya, jd jangan biarkan user dpt mengakses pada seluruh objek.

(1) Membatasi akses ke data dan servis: Applying the Principle of Least Privilege

2) Proteksi **data dictionary**:

```
O7_DICTIONARY_ACCESSIBILITY=FALSE
```

- Bentuk data dictionary: table &view
- Yg bisa lihat table: sys (krn yg punya)
- Bagaimana jika sys memberikan hak agar dpt di-select org lain? (boleh tidak? Ex: privilege : select any table)
- Maka, perlu di proteksi by parameter.

3) Mencabut privileges yg tidak diperlukan dari **PUBLIC**:

```
REVOKE EXECUTE ON UTL_SMTP, UTL_TCP, UTL_HTTP,
UTL_FILE FROM PUBLIC;
```

4) Membatasi akses direktori oleh users.

- Jika memiliki objek direktori jangan sembarang memberikan grant read/write.

(1) Membatasi akses ke data dan servis: Applying the Principle of Least Privilege (con't)

5) Membatasi user dgn privilege administrative.
6) Membatasi remote database authentication:
   - Otentifikasi ada 3: **password**, **external**, **global**
   - Yang harus dipikir: **external**
   - Agar user tidak bisa login by external, maka diset parameter.
     → keluar di ujian, apa artinya : False? / True?

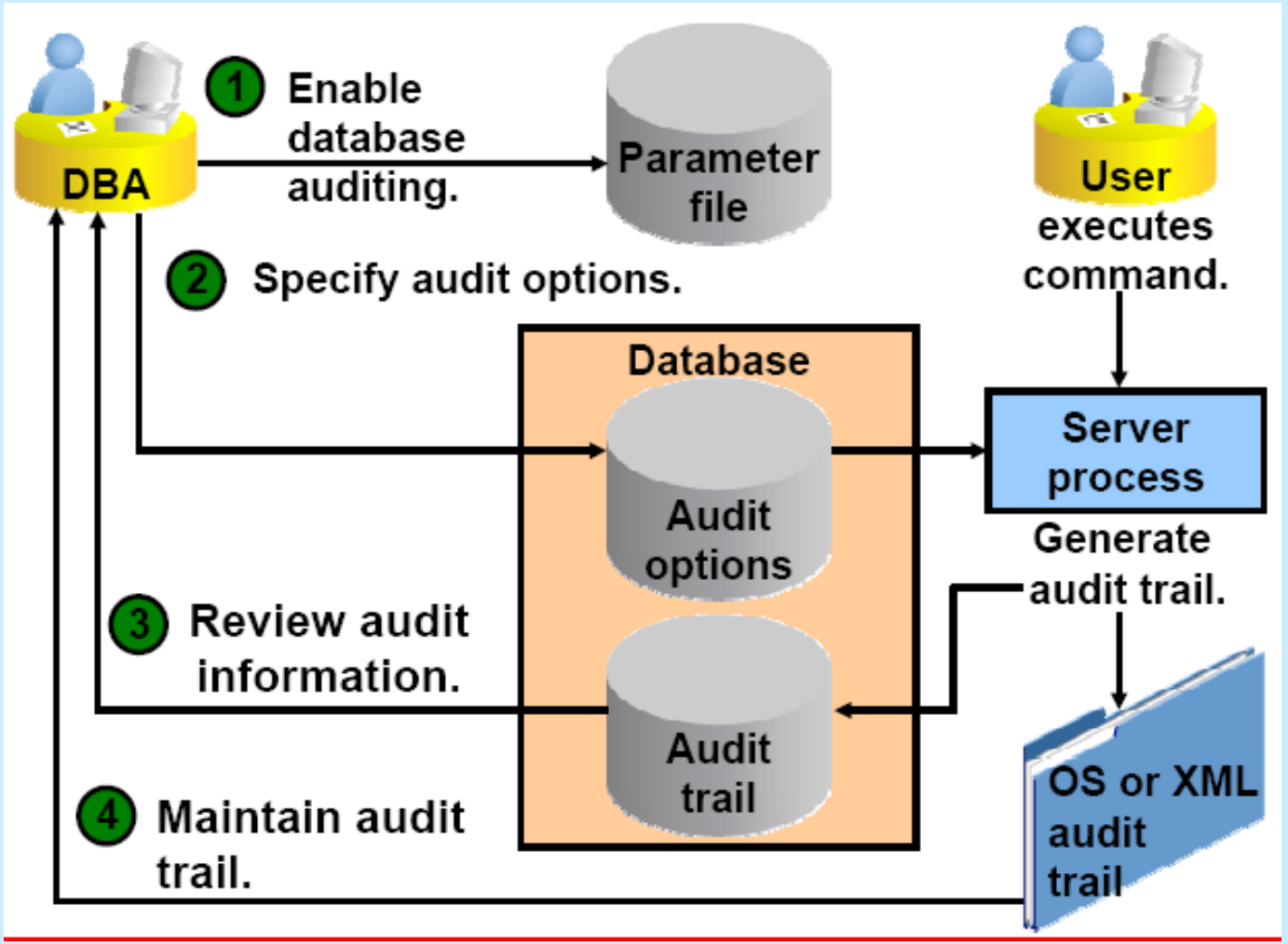
```
REMOTE_OS_AUTHENT=FALSE
```

TRUE : berarti tanpa password,
Why?
Krn diwakili saat login OS.
(BAHAYA!)

(2) Otentikasi Users:
Monitoring aktivitas yang Mencurigakan (Suspicious Activity)

- Otentikasi user menggunakan: global → bagus.
- **Monitoring** atau **auditing** aktivitas user yang **mencurigakan** dengan melakukan:
  1) **Mandatory auditing**
     - paling ringan, jarang digunakan
  2) **Standard database auditing**
     - Pada level database
  3) **Value-based auditing**
     - menggunakan trigger PL/SQL
  4) **Fine-grained auditing (FGA)**
     - menggunakan package
  5) **DBA auditing** → bagaimana kita mengaudit DBA, terutama sysdba dan sysoper, yg bisa bekerja saat sistemnya / instance shutdown.

Standard Database Auditing
(kadang disebut juga standar auditing)



Standard Database Auditing (kadang disebut juga standar auditing) – con't

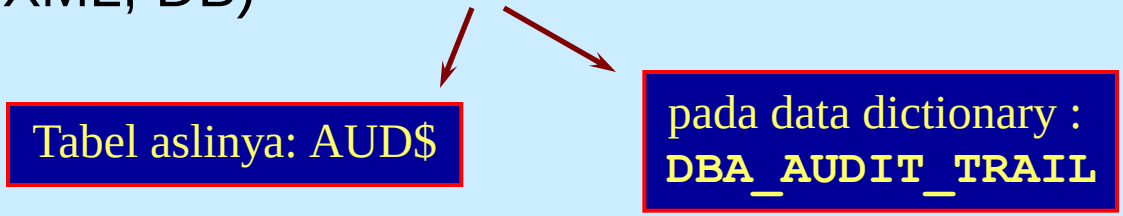
- Untuk mengaktifkan standar database auditing, sudah disediakan oleh **sistem Oracle**.
- Dengan **cara pengaktifkan** melalui **setting parameter** pada file: **audit_trail.**
  - Di versi 1, audit trail ada 3 demand (permintaan): **none**, **OS**, **DB.**
  - Di versi 2, audit trail ada tambahan: **XML**, dll.
- **Audit_trail** adalah tempat penyimpanan hasil audit.
- Tempat penyimpanannya, bisa **di:OS**, di dlm **DB** atau di file **XML**.

1

Standard Database Auditing (kadang disebut juga standar auditing) – con't

2

- Apa saja yang harus di audit, dapat di setting pada **audit optionnya**, lalu sistem akan melacak dalam **server process**.
  (*ingat! saat user terkoneksi user dibuatkan server process*)
- Jadi, ketika ada user yg melakukan aktivitas mencurigakan, maka langsung di audit, dan hasilnya pasti disimpan dalam **audit_trail** (penyimpanannya: OS, XML, DB)



  - Jadi, siapa saja yg menghapus data didalam **AUD$,** akan selalu diaudit.
  - Meskipun data dlm AUD$ dihapus, nama pelaku akan tetap muncul/tercatat.

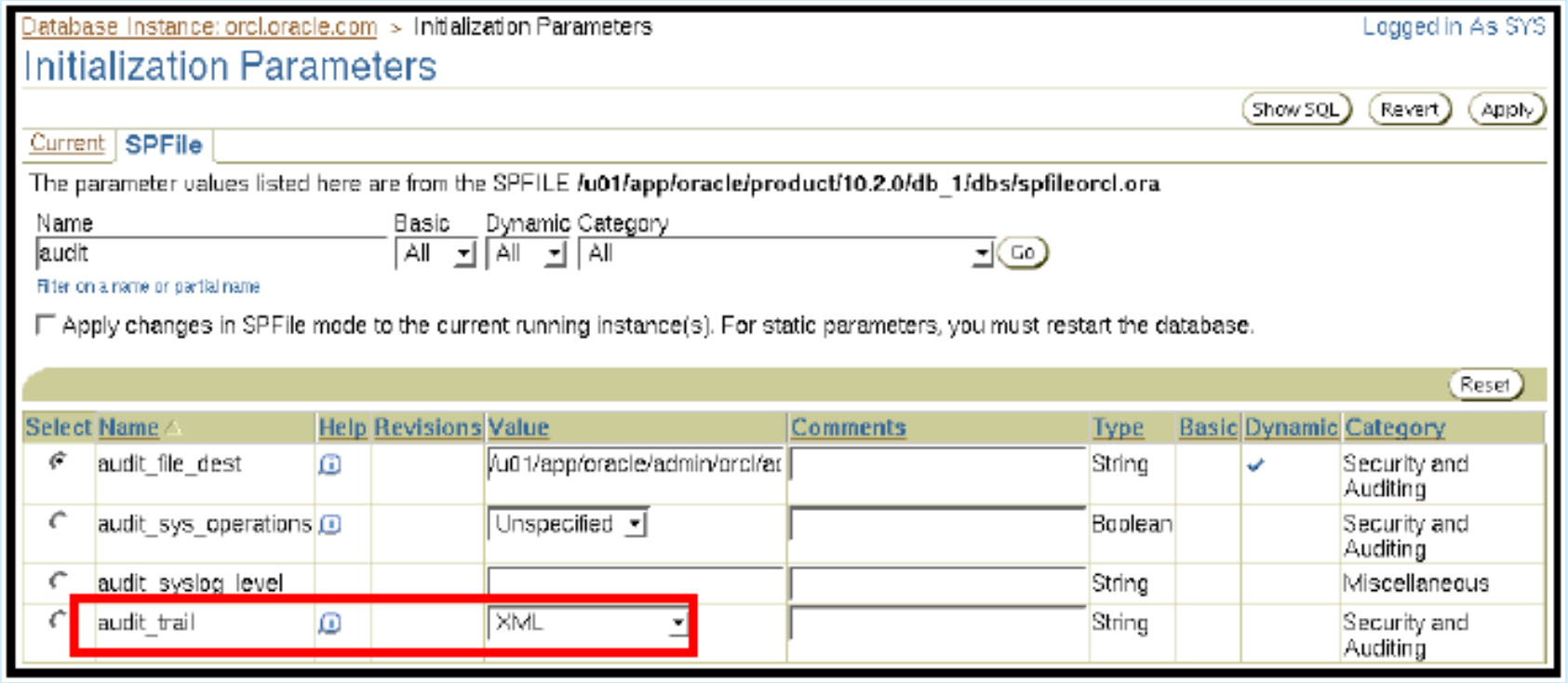
3

- Setelah itu DBA dapat melihat hasil informasi auditnya.

4

- Dan tindakan selanjutnya me-maintain audit trail.

Enabling Auditing

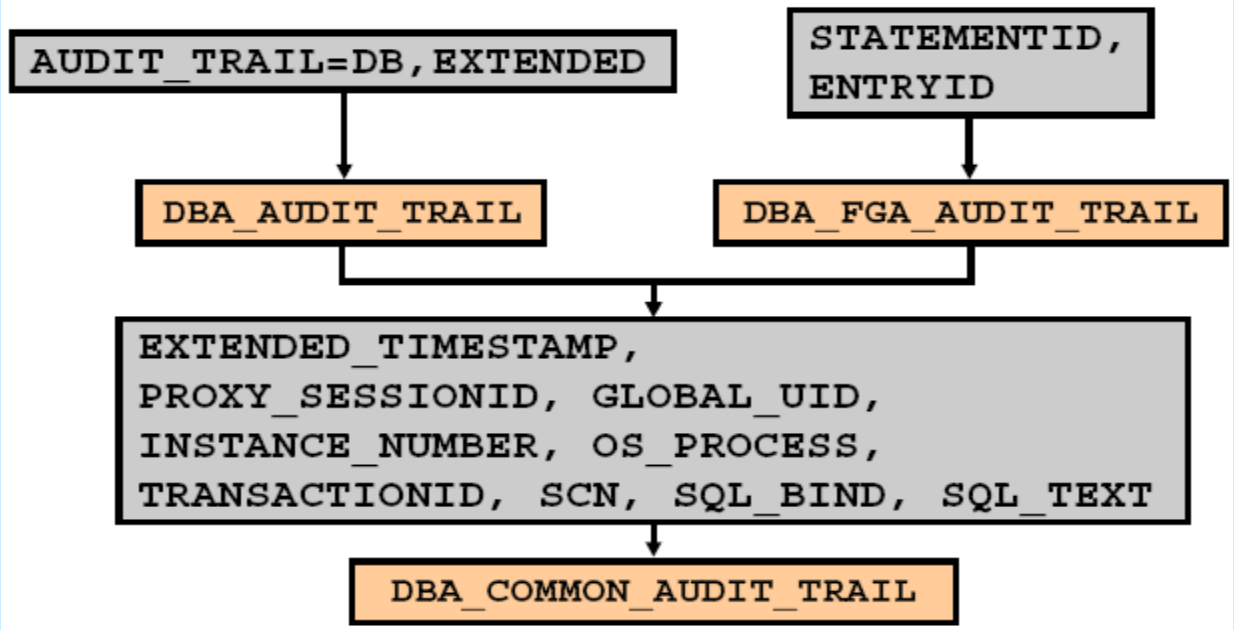



```
ALTER SYSTEM SET audit_trail="XML" SCOPE=SPFILE;
```

- Untuk meng-enable: set di **parameter file**.
- Cari **audit_trail** dan set nilai.
- **Restart** database setelah modifikasi static initialization parameter.

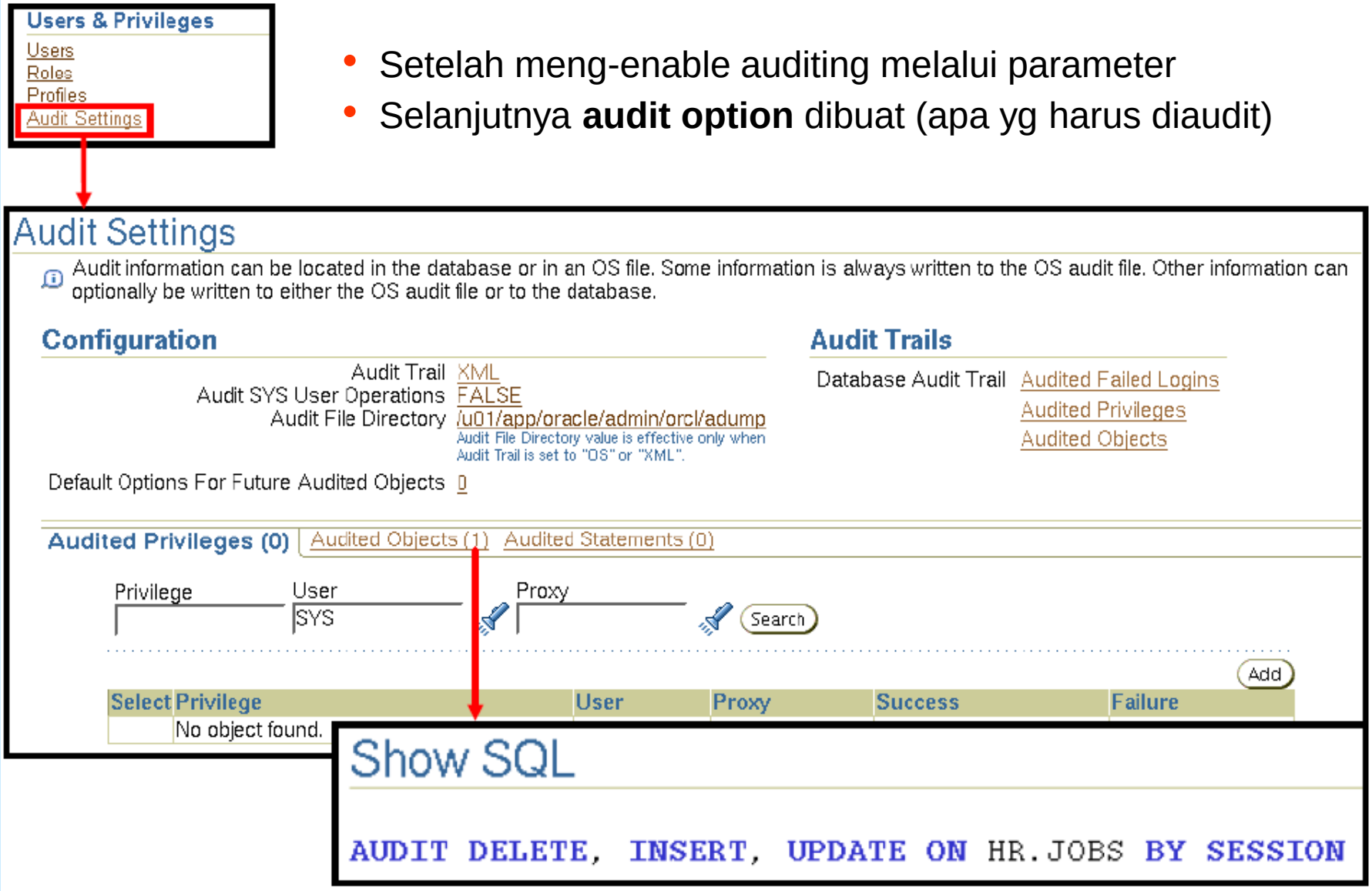
Uniform Audit Trails

Gunakan **AUDIT_TRAIL** untuk meng-enable database auditing



- Dari beberapa jenis audit, masing2 hasil audit akan di simpan pada audit_trail.
- Jenis audit FGA, yg paling kompleks: entry id, nomor instance, bahkan text perintah SQL-nya.
- Lalu untuk melihat **audit trail-**nya melalui **DBA_AUDIT_TRAIL (table asli: aud$).**
- Sedang yg **FGA**, dilihat melalui **DBA_FGA_AUDIT_TRAIL**.
- Bisa juga melihat pada **DBA_COMMON_AUDIT_TRAIL**.

Enterprise Manager Audit Page



- Setelah meng-enable auditing melalui parameter
- Selanjutnya **audit option** dibuat (apa yg harus diaudit)

Specifying Audit Options

1) SQL statement auditing:

```
AUDIT table;
```

Akan audit spt: create table, alter table

Di audit selama 1 session, sampai logout

2) System-privilege auditing (nonfocused and focused):

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

3) Object-privilege auditing (nonfocused and focused):

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

Setiap di akses selalu diaudit

Using and Maintaining Audit Information

Audited Objects

Filter Result | Return

Hide SQL

```
SELECT "OBJECT_SCHEMA", "OBJECT_NAME", "DB_USER", "STATEMENT_TYPE",
"EXTENDED_TIMESTAMP" FROM SYS.DBA_COMMON_AUDIT_TRAIL WHERE (action between 1 and 16) or
(action between 19 and 29) or (action between 32 and 41) or (action = 43) or (action between 51 and 99) or
(action = 103) or (action between 110 and 113) or (action between 116 and 121) or (action between 123 and 128)
or (action between 160 and 162)
```

Schema	Object Name	User Name	Action	Time (In Session's Time Zone)
HR	JOBS	AUDIT_USER	SESSION REC	2005-10-21 17.52.33.783793000 -7:0
HR	JOBS	HR	SESSION REC	2005-10-21 17.52.34.147582000 -7:0

Disable audit options if you are not using them.

Confirmation

No | Yes

Are you sure you want to remove the 3 selected audited objects?

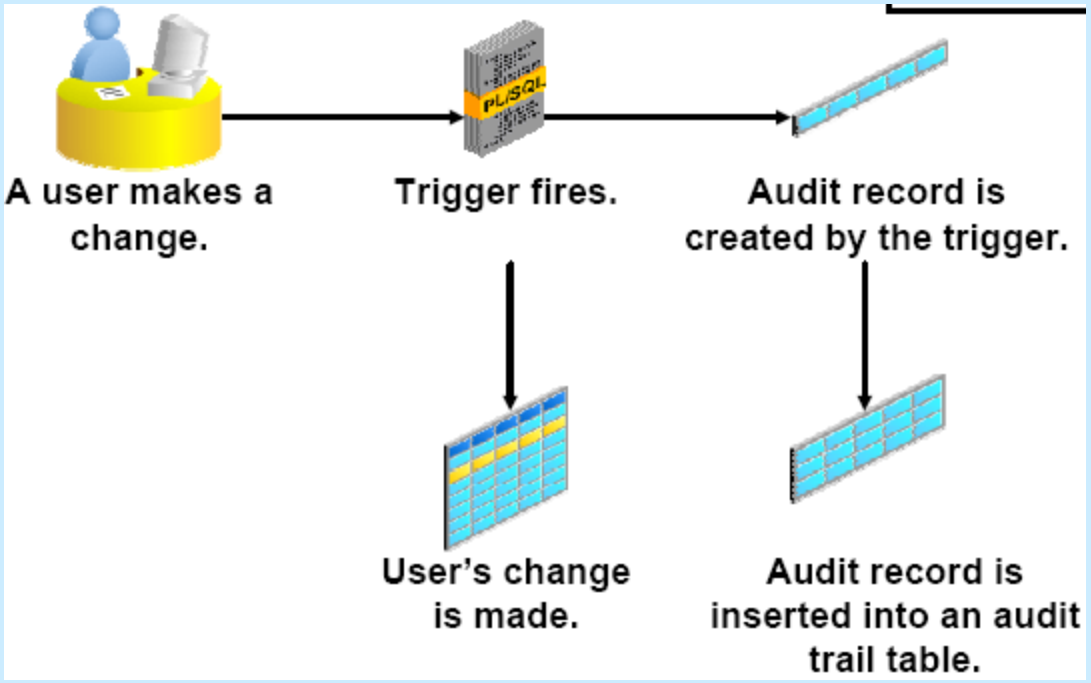
The audited statements you remove will no longer be audited on the objects.

Hide SQL

```
NOAUDIT DELETE ON HR.JOBS
NOAUDIT INSERT ON HR.JOBS
NOAUDIT UPDATE ON HR.JOBS
```

```
ALTER SYSTEM SET audit_trail = "NONE" SCOPE=SPFILE
```

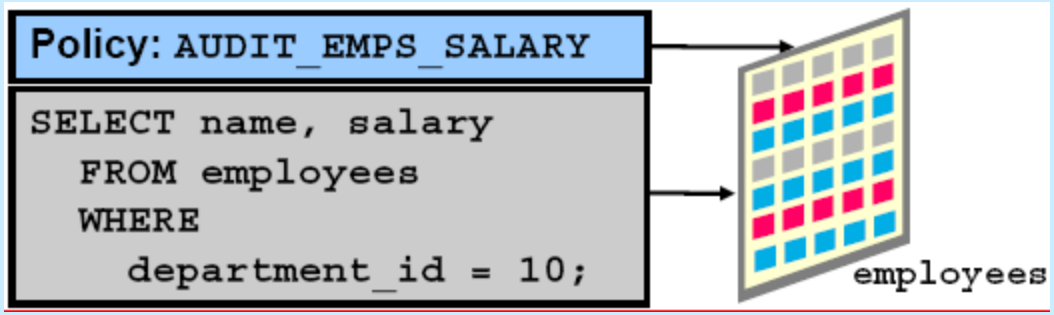
Value-Based Auditing



- Menggunakan **trigger.**
- Kelebihan trigger bisa mendapatkan: **data sebelum** dan **sesudah berubah**.
- **Artinya**: ketika melakukan auditing, kita bisa mendapatkan data sebelum berubah spt apa & data setelah berubah spt apa.
- Trigger akan dijalankan scr otomatis pd table. (misal. User melakukan perubahan table (update/delete) mk trigger akan jalan dan data dlm table data lama akan diambil bahkan dpt mengambil data baru)
- Didalam trigger ada recreate AS **NEW** & **ALL** (**ALL** utk ambil data lama/original, **NEW** mengambil data baru)
- Lalu disimpan dlm **audit_trail (**bukan pd db audit_trail), tp dibuat sendiri.
- Informasi apa yg disimpan? Tergantung programmer/yg buat trigger.
- Bahkan dgn trigger bisa mendapatkan: data sebelum & setelah berubah, IP address, terminal, s/w apa yg digunakan.

Fine-Grained Auditing

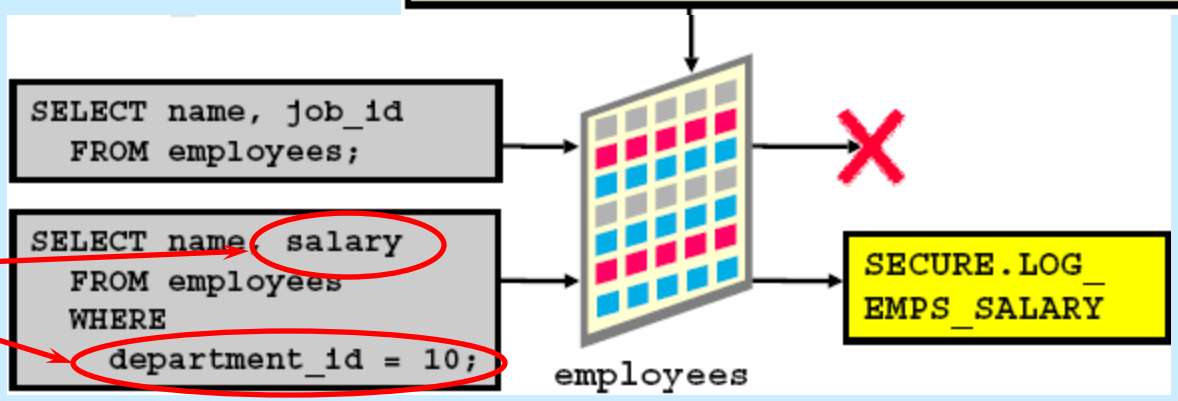
- Memonitor akses data
- Biasanya meng-audit ketika user melakukan operasi **DML** (**`SELECT, INSERT, UPDATE, DELETE`**, and **`MERGE`**)
- Bisa di-link utk audit table, view or per-masing[2] kolom.
- Bisa menjalankan procedure
- Utk menjalankannya menggunakan paket **`DBMS_FGA,`** disini memakai disebut **policy** (sbg nama auditnya)**.**



FGA Policy

- Menentukan dahulu:
  - Audit criteria
  - Audit action
- Dibentuk dgn membuat:
  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,
 statement_types => 'SELECT' );
```



Ke-duanya harus terpenuhi

- **ADD_POLICY** adalah procedure.
- Objek skema: HR (yg diaudit), …dst.
- Handle skema & module: menyatakan lokasi procedure
- procedurenya: “log_emps_salary”, adanya didlm skema “secure”
- Statement_type yg akan diaudit, bisa lebih dari 1 tp tanpa koma (,).

Select * from employees;
Di-audit tidak?

Audited DML Statement: Considerations

- Meng-audit perintah DML
- Me-Records yg ada didlm kolom dan kondisi.
- **DELETE** akan diaudit berdasar kolom
- **MERGE** diaudit dlm hubungannya dgn **INSERT** dan **UPDATE** .

```
UPDATE hr.employees
SET salary = 10
WHERE commission_pct = 90;
```

```
UPDATE hr.employees
SET salary = 10
WHERE employee_id = 111;
```

FGA Guidelines (wajib dibaca)

- Untuk audit semua statement, gunakan kondisi **null**.
- Nama policy harus unik
- Table/view yg diaudit harus sudah ada ketika membuat policy.
- Jika kondisi audit tdk valid muncul error **ORA-28112**.
- Jika kolom yg diaudit tdk ada, maka tdk ada baris yg diaudit.
- Jika event handler tdk ada, mk tdk ada error yg dikembalikan dan audit record tetap dibuat.

DBA Auditing

- **DBA Auditing**: sebenarnya meng-audit user **`SYSDBA`** atau **`SYSOPER`** yg bekerja ketika instance-nya down, dan bisa koneksi saat database closed.
- Untuk **Audit_trail** harus disimpan **diluar database** (minimal OS/XML).
- Utk mengaktifkan auditing SYSDBA or SYSOPER, gunakan **`audit_sys_operations`** dengan men-set parameter menjadi: **TRUE**.
- Hasil audit trail, jk disimpan dlm OS (diluar db) file disimpan dlm **`audit_file_dest`**.

Maintaining the Audit Trail

Audit trail harus dimaintain, karena makin lama makin besar.

Berikut cara terbaik:

- Review dan simpan record[2] yg lama.
- Mencegah problem storage.
- Menghindari kehilangan record data.

Applying Security Patches

- Use the Critical Patch Update process.
- Apply all security patches and workarounds.
- Contact the Oracle security products team.

Ringkasan

Pada bab ini, Anda seharusnya telah mempelajari bagaimana cara untuk:

- Menerapkan prinsip-prinsip hak akses
- Manajemen user account default
- Menerapkan standard keamanan standard
- Mengaudit aktivitas database

Latihan 1

Tugas :

- Mencegah penggunaan password yang sederhana
- Kemampuan account untuk mengunci dalam waktu 10 menit ketika terjadi kesalahan login
- Membebaskan aplikasi login server dari perubahan password
- Kegagalan audit untuk koneksi ke database

Latihan 2

Tugas :

- Audit SELECT pada kolom SALARY pada tabel EMPLOYEES
- Audit perubahan pada kolom SALARY dari tabel EMPLOYEES:
  - Nilai lama
  - Nilai baru
  - User yang membuat perubahan
  - Lokasi mana yang telah diubah dari yang dibuat