Securitatea bazelor de date – master anul 2
Laborator 2

Auditarea activităților în baza de date

Cuvinte cheie: • auditare • *database audit trail* • *operating system audit trail*	• *trigger*-i pentru audit • politici de audit • pachetul DBMS_FGA

- *Auditarea* activității pe baza de date are două componente: *monitorizarea și înregistrarea* persistentă a unei mulţimi de activități și evenimente, stabilită a-priori, din baza de date.
- Obiectivele auditării activităților pe baza de date cuprind: non-repudierea, investigarea activităților suspecte, detectarea problemelor generate de configurările curente privind autorizarea (accesul la resurse), complianța cu legislaţia în vigoare, controlul.

1. Auditare standard

1.1 Ce activități auditam?

Pornirea și oprirea bazei de date, conectarea administratorului la baza de date	***Sunt auditate implicit de către sistemul Oracle; datele sunt stocate automat în OS***

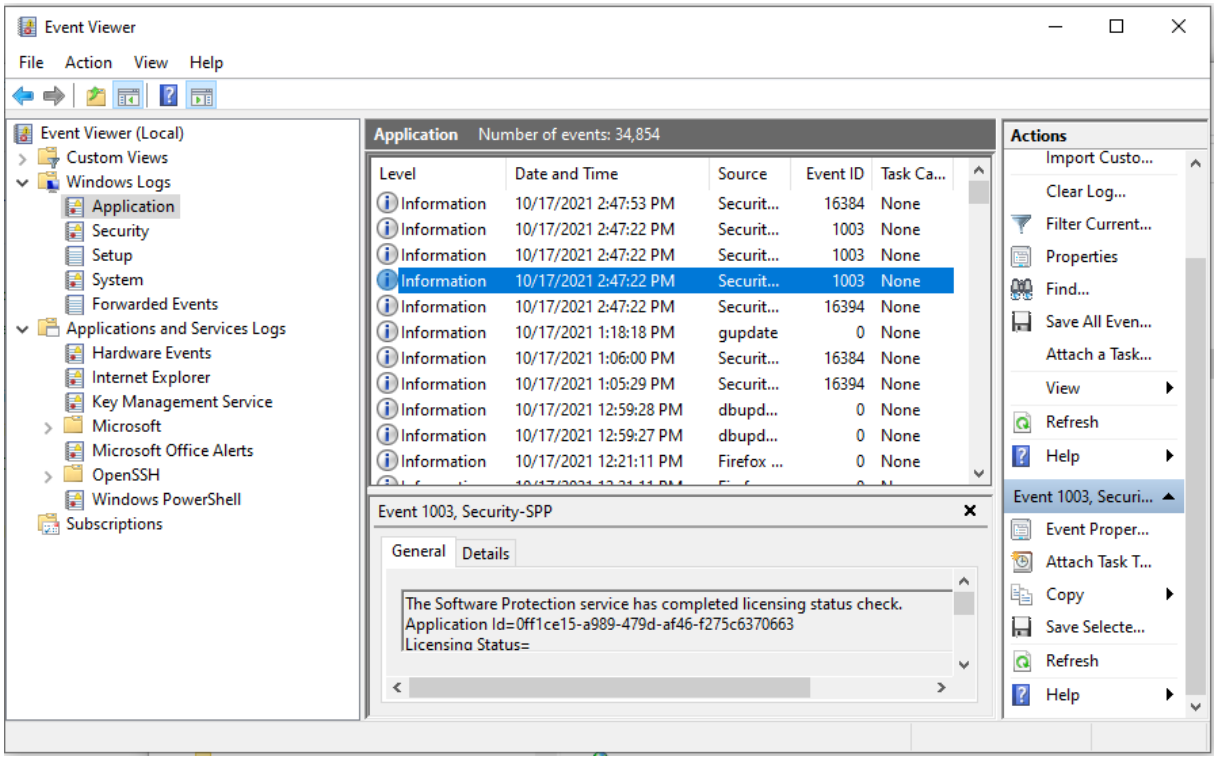
	Pentru toţi utilizatorii	Pentru utilizatorul Tom
Comenzi SQL		
*- **LDD :** toate comenzile CREATE TABLE și DROP TABLE*	***AUDIT TABLE***	***AUDIT TABLE BY Tom***
*- **LMD :** toate comenzile INSERT, UPDATE, respectiv DELETE*	***AUDIT INSERT TABLE*** ***AUDIT DELETE TABLE*** ***AUDIT UPDATE TABLE***	***AUDIT INSERT TABLE BY Tom*** ***s.a.m.d.***
*- **SELECT:** toate interogările pe toate tabelele și toate vizualizările*	***AUDIT SELECT TABLE***	***AUDIT SELECT TABLE BY Tom***
Comenzi SQL pe un obiect specificat (schema.obiect) al bazei de date		
- doar atunci când comanda eșuează	***AUDIT SELECT, INSERT, UPDATE,*** ***DELETE ON Tom.employees BY ACCESS WHENEVER NOT SUCCESSFUL;***	***AUDIT SELECT, INSERT, UPDATE,*** ***DELETE ON Tom.employees BY Tom WHENEVER NOT SUCCESSFUL;***
- oricând	***AUDIT SELECT, INSERT, UPDATE,*** ***DELETE ON Tom.employees;***	***AUDIT SELECT, INSERT, UPDATE,*** ***DELETE ON Tom.employees BY Tom;***
-audit implicit pentru obiectele ce vor fi create	***AUDIT ALTER, GRANT, INSERT, UPDATE, DELETE ON DEFAULT;***	-

Activitatea în rețea	***AUDIT NETWORK***	-
Exercitare privilegii - *de fiecare dată când este utilizat un privilegiu pentru efectuarea unei acțiuni pe baza de date*	***Exemple:*** ***AUDIT CREATE ANY VIEW (in orice schema)*** ***AUDIT CREATE VIEW (în schema proprie)***	***AUDIT CREATE ANY VIEW BY Tom*** ***s.a.m.d.***
Sesiune de lucru pe baza de date	***AUDIT SESSION***	***AUDIT SESSION BY Tom***

1.2 Unde înregistrăm informaţiile monitorizate?

- În baza de date – *database audit trail*:
  → ***audit_trail = DB*** (tabelul *SYS.AUD$*, vizualizările *DBA_AUDIT_TRAIL*, *DBA_COMMON_AUDIT_TRAIL*)
  ```
  alter system set audit_trail=db scope=spfile;
  ```
  → ***audit_trail = DB,EXTENDED*** **(**același tabel și aceleași vizualizări, dar se stochează și textul comenzilor în câmpul *SQLTEXT* de tip *CLOB***)**

- Extern bazei de date – *operating system audit trail*. Variante :
  → ***audit_trail = OS*** (sub *Windows*, *Control Panel* → *Administrative Tools* → *Event Viewer* → zona „Application” din *Windows Event Viewer*)
  ```
  alter system set audit_trail=os scope=spfile;
  ```



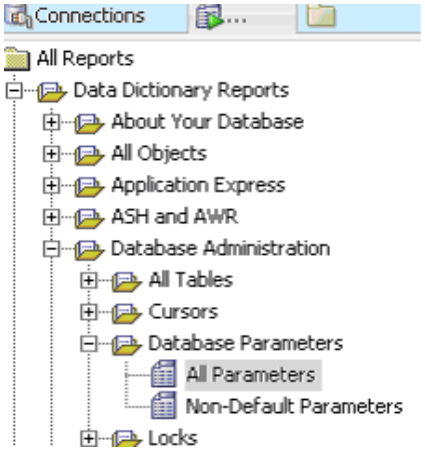
→ ***audit_trail = XML, AUDIT_FILE_DEST = calea catre fişier*** **(**implicit este $ORACLE_BASE/admin/$ORACLE_SID/adump.**)**
```
alter system set audit_trail=xml scope=spfile;
```

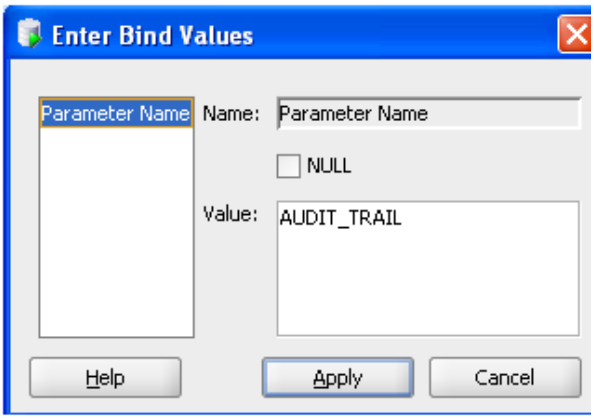
1.3 Pornire și oprire audit standard

- Pentru a afla configuraţia curentă privind locul stocării datelor monitorizate (cu *lowercase*!):
  - → comanda *SELECT*:

    ```
    select value from v$parameter where name='audit_trail';
    ```
  - → sau, din SQLPlus:

    ```
    show PARAMETER audit_trail
    ```
  - → sau, din SQLDeveloper: meniul *View* → *Reports* → *All Reports* → *Data Dictionary Reports* → *Database Administration* → *Database Parameters* → *All Parameters* → se selectează conexiunea de utilizat, iar in fereastra de dialog se înscrie valoarea AUDIT_TRAIL , apoi se apasă butonul „Apply”





- ***Pornirea auditului pentru activitatea X (vezi tabelul de pe prima pagină): AUDIT x***
- ***Oprirea auditului pentru activitatea X: NOAUDIT x***
- ***Oprirea în masă a auditului pentru toate comenzile SQL nelegate de un obiect specific: NOAUDIT ALL***
- ***Oprirea în masă a auditului pentru exercitarea privilegiilor: NOAUDIT ALL PRIVILEGES***
- ***Oprirea în masă a auditului pentru toate comenzile SQL legate de un obiect specific OBJ:***
  NOAUDIT ALL ON obj
- ***Oprirea în masă a auditului configurat implicit, pentru toate comenzile SQL legate de obiecte:***
  NOAUDIT ALL ON DEFAULT

1.4 Ştergerea informaţiilor de monitorizare, după ce le arhivam

- În funcţie de numărul activităților auditate, de frecvența lor zilnică, volumul datelor de monitorizare poate deveni foarte mare și ocupă astfel spațiu util pe disc. De aceea se recomandă arhivarea periodică a datelor monitorizate și ştergerea lor din sistemul de producţie.

- Dacă se realizează înregistrarea datelor în baza de date (*database audit trail*), atunci se pot utiliza comenzi de ştergere (reţinem, după arhivarea datelor in prealabil!):

```
DELETE FROM SYS.AUD$;
```

- Se poate opta pentru ştergerea informaţiilor monitorizate pentru un anumit obiect al bazei de date, de exemplu pentru tabelul *EMPLOYEES*:

```
DELETE FROM SYS.AUD$ WHERE OBJ$NAME='EMPLOYEES';
```

2. *Trigger*-i pentru auditare

2.1 Recapitulare *trigger*-i

- Ne reamintim de la cursul de SGBD că „*un declanşator (trigger) este un bloc PL/SQL sau apelul CALL al unei proceduri PL/SQL care se execută automat ori de câte ori are loc un anumit eveniment declanşator*”.
- *Trigger*-ii sunt de două tipuri: la nivelul bazei de date (operaţii pe baza de date) și la nivel de aplicaţie (de exemplu, apăsarea unui buton pe un formular în *Oracle Forms*). Categoria de interes pentru noi în acest material este cea a *trigger*-ilor la nivelul bazei de date.
- ***Trigger*-ii la nivelul bazei de date (*database triggers*)** se clasifică la rândul lor în 3 categorii:
  - → *Trigger*-i LMD – declanșați de comenzi LMD pe un tabel. Pot fi executaţi o singură dată la nivelul unei comenzi indiferent de numărul de înregistrări afectate (*trigger*-i la nivel de instrucţiune) sau pot fi executaţi *FOR EACH ROW* (*trigger*-i la nivel de înregistrare). Le corespund tipurile de *trigger*-i *BEFORE STATEMENT*, *AFTER STATEMENT*, *BEFORE EACH ROW*, *AFTER EACH ROW*;
  - → *Trigger*-i *INSTEAD OF* – declanșați de comenzi LMD pe o vizualizare;
  - → *Trigger*-i *SYSTEM* – declanșați de evenimente precum pornirea/oprirea bazei de date, comenzi LDD, conectare/deconectare utilizator. Le corespund tipurile de *trigger*-i *AFTER EVENT*, *BEFORE EVENT*.
- Interogarea tabelului ***SYS.TRIGGER$*** sau a vizualizării ***ALL_TRIGGERS*** oferă informaţii despre toţi *trigger*-ii de la nivelul bazei de date.

```
SELECT DISTINCT TRIGGER_TYPE FROM ALL_TRIGGERS;
```

```
TRIGGER_TYPE
----------------
BEFORE STATEMENT
BEFORE EACH ROW
AFTER EACH ROW
BEFORE EVENT
AFTER STATEMENT
AFTER EVENT
INSTEAD OF

7 rows selected.
```

- View-ul ***DBA_TRIGGERS*** oferă informaţii despre *trigger*-ii creaţi de produsele *Oracle* automat la instalare. Imediat după crearea unei baze de date regăsim 617 triggeri DBA. Pentru a afla informaţii despre *trigger*-ii *SYSTEM* (de tip '***BEFORE EVENT***' **şi** '***AFTER EVENT***'**)** creaţi automat la instalare, executăm următoarea interogare:

```
SELECT SUBSTR(OWNER,1,20) OWNER ,
       SUBSTR(TRIGGER_NAME,1,30) TRIGGER_NAME,
       SUBSTR(TRIGGERING_EVENT,1,30) TRIGGERING_EVENT,
       TRIGGER_TYPE
FROM DBA_TRIGGERS
WHERE TRIGGER_TYPE='BEFORE EVENT' OR TRIGGER_TYPE='AFTER EVENT'
ORDER BY TRIGGER-TYPE DESC;
```

OWNER	TRIGGER_NAME	TRIGGERING_EVENT	TRIGGER_TYPE
SYS	XDB_PI_TRIG	DROP OR TRUNCATE	BEFORE EVENT
SYS	CDC_ALTER_CTABLE_BEFORE	ALTER	BEFORE EVENT
MDSYS	SDO_ST_SYN_CREATE	CREATE	BEFORE EVENT
MDSYS	SDO_TOPO_DROP_FTBL	DROP	BEFORE EVENT
EXFSYS	EXPFIL_RESTRICT_TYPEEVOLVE	CREATE OR ALTER	BEFORE EVENT
EXFSYS	EXPFIL_DROPOBJ_MAINT	DROP	BEFORE EVENT
SYS	CDC_DROP_CTABLE_BEFORE	DROP	BEFORE EVENT
MDSYS	SDO_GEOR_BDDL_TRIGGER	DDL	BEFORE EVENT
SYS	CDC_CREATE_CTABLE_BEFORE	CREATE	BEFORE EVENT
EXFSYS	RLMGR_TRUNCATE_MAINT	TRUNCATE	BEFORE EVENT
WMSYS	NO_VM_DDL	CREATE OR ALTER OR DROP OR REN	BEFORE EVENT
OWNER	TRIGGER_NAME	TRIGGERING_EVENT	TRIGGER_TYPE
SYS	OLAPISHUTDOWNTRIGGER	SHUTDOWN	BEFORE EVENT
MDSYS	SDO_NETWORK_DROP_USER	DROP	AFTER EVENT
MDSYS	SDO_GEOR_ADDL_TRIGGER	DDL	AFTER EVENT
MDSYS	SDO_DROP_USER	DROP	AFTER EVENT
SYS	OLAPISTARTUPTRIGGER	STARTUP	AFTER EVENT
MDSYS	SDO_GEOR_ERR_TRIGGER	ERROR	AFTER EVENT
EXFSYS	EXPFIL_DROPUSR_MAINT	DROP	AFTER EVENT
EXFSYS	EXPFIL_ALTEREXPTAB_MAINT	ALTER OR RENAME	AFTER EVENT
SYS	CDC_CREATE_CTABLE_AFTER	CREATE	AFTER EVENT
WMSYS	NO_VM_DROP_A	DROP	AFTER EVENT
SYS	AW_REN_TRG	RENAME	AFTER EVENT
OWNER	TRIGGER_NAME	TRIGGERING_EVENT	TRIGGER_TYPE
SYSMAN	MGMT_STARTUP	STARTUP	AFTER EVENT
SYS	AW_DROP_TRG	DROP	AFTER EVENT
SYS	AW_TRUNC_TRG	TRUNCATE	AFTER EVENT

25 rows selected

- Tot la instalare, în mod automat sunt creaţi *trigger*-i LMD în schema utilizatorului HR:

```
SELECT SUBSTR(TABLE_NAME,1,20) TABLE_NAME,
SUBSTR(TRIGGER_TYPE,1,30) TRIGGER_TYPE,TRIGGER_BODY FROM
DBA_TRIGGERS
WHERE  OWNER='HR';
```

TABLE_NAME	TRIGGER_TYPE	TRIGGER_BODY
EMPLOYEES	BEFORE STATEMENT	BEGIN secure_dml; END secure_employees;
EMPLOYEES	AFTER EACH ROW	BEGIN add_job_history(:old.employee_id, :old.hire_date, sysdate,

2.2 Utilizarea *trigger*-ilor în auditare

- Pentru auditare, putem crea *trigger*-i personalizaţi care să înregistreze anumite informaţii de interes. În general, vom crea un tabel special pentru stocarea informaţiilor monitorizate.

- Triggerii construiţi de noi se regăsesc la interogarea tabelului *TRIGGER*$ şi a *view*-urilor *ALL_TRIGGERS*, *USER_TRIGGERS*.

- Câteva recapitulări utile referitoare la procesarea *trigger*-ilor, utile în auditare:

1) **Trebuie ca *trigger*-ii pe care îi construim să nu influenţeze activitatea normală din baza de date**. Scopul auditului este să monitorizeze pasiv și să înregistreze activitatea pentru analiza ulterioară. Prin urmare NU vom defini *trigger*-i *INSTEAD OF* care să deturneze rezultatele din tabelele vizate către tabela de audit!

2) ***Trigger*-ii LMD la nivel de instrucţiune și la nivel de înregistrare pot coexista. Vor fi apelaţi în ordinea următoare:**

Trigger BEFORE instrucţiune
Pentru fiecare înregistrate afectată
  Trigger BEFORE înregistrare
  Operaţie LMD propriu-zisă
  Trigger AFTER înregistrare
Trigger AFTER instrucţiune

Din perspectiva auditului, trebuie decisă cu atenţie granularitatea monitorizării, pentru că scopul nu este să clonăm tabelele de bază, ci să înregistrăm activitatea pe ele.

3) ***Trigger*-ii definiţi de utilizatori vor fi executaţi doar dacă din punct de vedere al sistemului Oracle instrucţiunea este corectă și poate avea loc.** Pentru o instrucţiune LMD greşit construită sau care încalcă unele constrângeri, de exemplu, nu se va ajunge până la *trigger*-ul definit de utilizator, ci eroarea va fi returnată înainte.

În concluzie, pentru audit sunt adecvaţi în special *trigger*-ii LMD la nivel de instrucţiune.

3. Politici de auditare

- Cea de-a treia modalitate de audit se referă la *Fine Grain Audit* prin politici de auditare. Structura unei politici de auditare este următoarea:
  - → specificarea obiectului (schemă, nume obiect, coloane) supus monitorizării;
  - → specificarea acţiunilor monitorizate asupra obiectului (*SELECT*, *INSERT*, *UPDATE*, *DELETE*); implicit este *SELECT*;
  - → specificarea condiţiilor sub care se înregistrează informaţiile monitorizate; este corespondentul clauzei *WHEN* din *trigger*-i si este opţional;
  - → un *event handler* care să trateze suplimentar evenimentul; acesta este opţional.
- O politică de auditare poate fi activă (status *ENABLED*) sau inactivă (status *DISABLED*). Nu pot fi definite mai mult de 256 de politici de auditare la nivelul unui obiect al bazei de

date.

Lista politicilor de auditare active se obţine prin interogarea vizualizării *ALL_AUDIT_POLICIES*, astfel:

```
SELECT POLICY_TEXT,ENABLED FROM ALL_AUDIT_POLICIES
WHERE OBJECT_NAME='DEPARTMENTS';
```

- Pentru gestionarea politicilor de auditare avem la dispoziţie pachetul *DBMS_FGA* (este necesar să acordaţi privilegiu pentru utilizatorii ce vor scrie cod PL/SQL care să folosească acest pachet: *grant execute on dbms_fga to nume_utilizator;*).

- **Sintaxa:**

```
DBMS_FGA.ADD_POLICY (
        object_schema=>'nume schema',
        object_name=>'obiect auditat',
        policy_name=>'nume unic de politica',
        audit_column=>'col1,col2,.. din obiectul auditat',
        enable=>false,
        statement_types=>'select,insert,update,delete'
        handler_schema=>'schema ce contine handler'
        handler_module=>'nume handler');
```

- Se impune ca modulul *handler* să fie o procedură PL/SQL cu următoarea signatură: **CREATE OR REPLACE PROCEDURE <fname> (object_schema VARCHAR2, object_name VARCHAR2, policy_name VARCHAR2) AS ...**

- Rezultatele auditului pot fi obţinute din tabelul *SYS.FGA_LOG$* și din vizualizarea ***dba_fga_audit_trail***

- Pentru activarea sau dezactivarea unei politici de auditare:

```
DBMS_FGA.ENABLE_POLICY / DBMS_FGA.DISABLE_POLICY (
    object_schema=>'nume schema de care apartine obiectul',
    object_name=>'obiect auditat',
    policy_name=>'nume unic de politica');
```

Observaţie: Acţiunile administratorului (*as SYSDBA*) nu sunt auditate (modificare în ini.ora)!

4. Exerciţii

1. Configuraţi baza de date pentru audit standard cu stocarea datelor monitorizate în cadrul bazei de date.
Se vor monitoriza toate activitățile de interogare efectuate în baza de date, cu stocarea textului cererilor efectuate de utilizatori.

Să se afișeze un raport al acestor activități pentru tabelele date anterior. Opriți auditul configurat.

2. Configuraţi baza de date pentru audit standard cu stocarea datelor monitorizate în cadrul unui fișier XML în calea standard.
Se vor monitoriza toate comenzile LMD pe tabelul *HR.EMPLOYEES* care eșuează.

Să se consulte fișierele XML rezultate. Opriţi auditul configurat.

3. Cu scopul auditarii, creaţi *trigger*(i) care să înregistreze într-un tabel de audit (TAB_AUDIT_EMP) informaţii despre operaţiile LMD de ștergere pe tabelul *EMPLOYEES* și numărul de înregistrări afectate.

4. Cu scopul auditării, creaţi un *trigger* care să înregistreze într-un tabel de audit (*TAB_AUDIT_EMP*) informaţii despre operaţiile LMD care stabilesc salarii peste plafonul de 20000.

5. Creaţi o politică de auditare astfel încât să fie înregistrate instrucţiunile LMD de modificare a şefilor departamentelor (*MANAGER_ID*) pe tabelul *DEPARTMENTS*.