Użytkownicy, uprawnienia, role, obserwacja bazy danych

Zarządzanie użytkownikami

- Mechanizmy uwierzytelniania
  - baza danych, system operacyjny, serwer autoryzacji
- Przywileje
  - systemowe, obiektowe
- Role - nazwane zbiory przywilejów
- Limity na zajmowaną przestrzeń dyskową
- Limity na wykorzystanie zasobów
  - CPU, I/O, czas bezczynności, liczba sesji
- Zarządzanie kontem
  - stopień złożoności hasła,
  - czas życia hasła i konta,
  - blokowanie konta
- Obserwacja działań użytkowników

Schematy i użytkownicy

- **Każda baza danych Oracle zawiera SCHEMATY**
- **Schemat jest zbiorem obiektów, takich jak: tabele, perspektywy, procedury PL/SQL, pakiety**
- **Każda baza danych Oracle zawiera UŻYTKOWNIKÓW**
- **W celu przyłączenia się do bazy danych, użytkownik musi dostarczyć systemowi Oracle swoją nazwę i hasło dostępu**
- **Schematy tworzone są automatycznie dla każdego z tworzonych użytkowników**
  - **nazwa schematu jest równa nazwie użytkownika**
- **Hasło dostępu może być zmieniane przez użytkownika**
- **Podczas instalacji bazy danych tworzonych jest dwóch wyróżnionych użytkowników: SYS i SYSTEM**

Tworzenie nowych użytkowników

```
CREATE USER <nazwa użytkownika>
IDENTIFIED BY <hasło dostępu>
[DEFAULT TABLESPACE <przestrzeń tabel>]
[TEMPORARY TABLESPACE <przestrzeń tabel>]
[PROFILE <nazwa profilu>]
[QUOTA <rozmiar> | UNLIMITED ON <przestrzeń tabel>]
[PASSWORD EXPIRE]
[ACCOUNT <LOCK | UNLOCK>]
```

```
CREATE USER scott
IDENTIFIED BY tiger
DEFAULT TABLESPACE data
TEMPORARY TABLESPACE temp
QUOTA 1M ON data
QUOTA 100K ON system
PROFILE default;
```

Modyfikacja i usuwanie użytkownika

```
ALTER USER <nazwa użytkownika>
[IDENTIFIED BY <hasło dostępu>]
[DEFAULT TABLESPACE <przestrzeń tabel>]
[TEMPORARY TABLESPACE <przestrzeń tabel>]
[PROFILE <nazwa profilu>]
[QUOTA <rozmiar> ON <przestrzeń tabel>]
```

```
ALTER USER scott
IDENTIFIED BY lion;
```

```
DROP USER <nazwa użytkownika>
[CASCADE]
```

```
DROP USER scott CASCADE;
```

Informacje o użytkownikach

ALL_USERS

USER_USERS

DBA_USERS

DBA_TS_QUOTAS

USER_TS_QUOTAS

```
SELECT * FROM all_users;
```

```
SELECT * FROM user_ts_quotas;
```

Rozłączanie sesji użytkownika

```
SQL> SELECT sid, serial#, username
FROM v$session;

SID         SERIAL#         USERNAME
---------- --------------- ---------------
         1               1
         7            2126 SCOTT
         8             786 SYSTEM

SQL> ALTER SYSTEM KILL SESSION '7, 2126';
```

Uwierzytelnianie użytkowników

- **Uwierzytelnienie przez bazę danych Oracle**
  - **w celu przyłączenia się do bazy danych wymagane jest wprowadzenie nazwy i hasła użytkownika Oracle**
- **uwierzytelnienie przez system operacyjny**
  - **użytkownicy przyłączają się do bazy danych bez wprowadzania swoich nazw i haseł,**
- **Uwierzytelnienie przez serwer autoryzacji**
  - **scentralizowane zarządzanie użytkownikami w środowisku rozproszonym**

Uwierzytelnienie przez system operacyjny

```
CREATE USER <nazwa użytkownika>
IDENTIFIED EXTERNALLY
[DEFAULT TABLESPACE <przestrzeń tabel>]
[TEMPORARY TABLESPACE <przestrzeń tabel>]
[PROFILE <nazwa profilu>]
[QUOTA <rozmiar> ON <przestrzeń tabel>]
```

Utworzenie użytkownika uwierzytelnianego przez system operacyjny

```
SQL> show parameter prefix
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------
os_authent_prefix                    string      ops$
```

initSID.ora

```
os_authent_prefix = abc$
```

```
SQL> create user abc$adm1
identified externally;
SQL> grant connect to abc$adm1;
```

```
adm1@miner:~/admin > sqlplus /
```

Uprawnienia użytkowników i role

- **Uprawnienia SYSTEMOWE zezwalają użytkownikowi na wykonywanie w bazie danych operacji określonego typu**
- **Uprawnienia OBIEKTOWE zezwalają użytkownikowi na wykonanie określonych operacji na konkretnym obiekcie bazy danych**
- **Uprawnienia mogą być kontrolowane przy pomocy RÓL, które stanowią nazwane zbiory uprawnień**

Uprawnienia systemowe - przykłady

CREATE SESSION - zezwala użytkownikowi na logowanie się do bazy danych

CREATE TABLE - zezwala użytkownikowi na tworzenie tabel w jego własnym schemacie

DROP ANY VIEW - zezwala użytkownikowi na usuwanie dowolnych perspektyw

EXECUTE ANY PROCEDURE - zezwala użytkownikowi na wykonywanie dowolnych procedur PL/SQL

ALTER SYSTEM - zezwala użytkownikowi na wydawanie poleceń ALTER SYSTEM

Nadawanie uprawnień systemowych

```
GRANT <uprawnienie|rola> [, <uprawnienie|rola>...]
TO <nazwa użytkownika|rola|PUBLIC>
[WITH ADMIN OPTION]
```

```
GRANT CREATE SESSION
TO jan
WITH ADMIN OPTION;
```

Informacje o aktualnych uprawnieniach

```
SQL> SELECT * FROM dba_sys_privs
GRANTEE    PRIVILEGE                 ADM
------     --------------------      ----
SYS        DELETE ANY TABLE          NO
SYS        INSERT ANY TABLE          NO
SYS        SELECT ANY TABLE          YES
SYS        UNLIMITED TABLESPACE      YES
SYS        UPDATE ANY TABLE          NO
SYSTEM     UNLIMITED TABLESPACE      YES
```

Odbieranie uprawnień systemowych

```
REVOKE <uprawnienie|rola> [, <uprawnienie|rola>...]
FROM <nazwa użytkownika|rola|PUBLIC>
```

```
REVOKE CREATE SESSION
FROM jan;
```

- **Odebranie użytkownikowi uprawnienia systemowego nadanego z opcją ADMIN nie powoduje kaskadowego odebrania uprawnień użytkownikom, którym uprawnienie to zostało przez niego przekazane**

Typy uprawnień obiektowych

SELECT - wydawanie poleceń SELECT FROM dla obiektu (tabela, perspektywa, migawka, sekwencja)

UPDATE - wydawanie poleceń UPDATE dla obiektu (tabela, perspektywa)

INSERT - wydawanie poleceń INSERT dla obiektu (tabela, perspektywa)

ALTER - wydawanie poleceń ALTER dla obiektu (tabela, perspektywa, sekwencja) i CREATE TRIGGER ON dla obiektu (tabela)

DELETE - wydawanie poleceń DELETE FROM dla obiektu (tabela, perspektywa)

EXECUTE - wydawanie poleceń EXECUTE dla obiektu (procedura, funkcja)

INDEX - wydawanie poleceń CREATE INDEX ON dla obiektu (tabela)

REFERENCES - wydawanie poleceń CREATE i ALTER TABLE definiujących klucz obcy dla obiektu (tabele)

Nadawanie uprawnień obiektowych

```
GRANT <uprawnienie|ALL> [(<atrybut> [,<atrybut>])]
[, <uprawnienie|ALL> [(<atrybut> [,<atrybut>])]...]
ON <obiekt>
TO <nazwa użytkownika|rola|PUBLIC>
[WITH GRANT OPTION]
```

```
GRANT SELECT ON pracownicy TO jan, scott;
```

```
GRANT SELECT, INSERT, UPDATE (PLACA_POD)
ON pracownicy TO PUBLIC;
```

- **Aby móc nadać/odebrać dany przywilej obiektowy należy spełniać jeden z poniższych warunków:**
  - **być właścicielem obiektu**
  - **posiadać to uprawnienie z opcją GRANT**
  - **posiadać uprawnienie systemowe GRANT ANY OBJECT PRIVILEGE**

Informacje o aktualnych uprawnieniach

DBA_TAB_PRIVS
DBA_COL_PRIVS
USER_TAB_PRIVS
USER_TAB_PRIVS_MADE
USER_TAB_PRIVS_RECD
USER_COL_PRIVS_MADE
USER_COL_PRIVS_RECD

```
SELECT * FROM
user_tab_privs;
```

Odbieranie uprawnień obiektowych

```
REVOKE <uprawnienie|ALL> [(<atrybut> [,<atrybut>])]
[, <uprawnienie|ALL> [(<atrybut> [,<atrybut>])]...]
ON <obiekt>
FROM <nazwa użytkownika|rola|PUBLIC>
[CASCADE CONSTRAINTS]
```

```
REVOKE SELECT ON pracownicy FROM scott;
```

```
REVOKE ALL ON pracownicy FROM PUBLIC;
```

- **Odebranie użytkownikowi uprawnienia obiektowego nadanego z opcją GRANT powoduje kaskadowe odebranie tego uprawnienia użytkownikom, którym uprawnienie to zostało przez niego przekazane**

Role

- **Rola jest nazwanym zbiorem uprawnień zarówno systemowych, jak i obiektowych**
- **Role upraszczają zarządzanie uprawnieniami dla dużych zbiorów użytkowników o podobnej charakterystyce (np. pracownicy działu kadr, zarząd)**
- **Zmieniając uprawnienia należące do roli, zmienia się uprawnienia wszystkich użytkowników do niej przypisanych**
- **W bazie danych zdefiniowane są zawsze role standardowe: CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE, IMP_FULL_DATABASE**
- **Typowy zestaw uprawnień można przydzielić użytkownikowi przydzielając mu role CONNECT i RESOURCE**
  - **należy jednak mieć świadomość, że nadanie roli RESOURCE powoduje niejawne nadanie przywileju UNLIMITED TABLESPACE (!)**

Informacje o rolach

ROLE_SYS_PRIVS
ROLE_TAB_PRIVS
ROLE_ROLE_PRIVS
SESSION_ROLES
USER_ROLE_PRIVS
DBA_SYS_PRIVS
DBA_TAB_PRIVS
DBA_ROLES

```
SELECT * FROM dba_roles;
```

Tworzenie nowych ról

```
CREATE ROLE <nazwa roli>
[NOT IDENTIFIED | IDENTIFIED BY <hasło> |
IDENTIFIED EXTERNALLY]
```

```
CREATE ROLE kadry
NOT IDENTIFIED;
```

Modyfikowanie ról

```
ALTER ROLE <nazwa roli>
[NOT IDENTIFIED | IDENTIFIED BY <hasło> |
IDENTIFIED EXTERNALLY]
```

```
ALTER ROLE kadry
IDENTIFIED BY styczeń;
```

Włączanie i wyłączanie ról

```
SET ROLE <nazwa roli> [IDENTIFIED BY <hasło>]
[, <nazwa roli> [IDENTIFIED BY <hasło>]...]

SET ROLE ALL [EXCEPT <nazwa roli> [, <nazwa roli>...]

SET ROLE NONE
```

```
SET ROLE kadry
IDENTIFIED BY styczeń;
```

```
SET ROLE NONE;
```

Ustawianie ról domyślnych dla użytkownika

```
ALTER USER <nazwa użytkownika>
DEFAULT ROLE <nazwa roli> [, <nazwa roli>...]

ALTER USER <nazwa użytkownika>
DEFAULT ROLE ALL [EXCEPT<nazwa roli>[,<nazwa roli>...]]

ALTER USER <nazwa użytkownika>
DEFAULT ROLE NONE
```

```
ALTER USER scott
DEFAULT ROLE kadry;
```

Profile

- **Profile są wykorzystywane do kontroli użytkowania zasobów systemowych**
- **Zasobami systemowymi są: czas procesora, operacje I/O, czas bezczynności, czas trwania sesji, równoczesne sesje**
- **Kiedy przekroczony zostanie limit zasobów określony przez profil, wtedy: aktualne polecenie jest wycofywane, dozwolone jest COMMIT i ROLLBACK, praca w sesji musi zostać zakończona**
- **Weryfikacja limitów systemowych włączana jest parametrem inicjalizacyjnym**

  RESOURCE_LIMIT=TRUE

  lub poleceniem

  ALTER SYSTEM
  SET RESOURCE_LIMIT=TRUE

Tworzenie nowego profilu

```
CREATE PROFILE <nazwa profilu> LIMIT
[SESSIONS_PER_USER <limit|UNLIMITED|DEFAULT>]
[CPU_PER_SESSION <limit|UNLIMITED|DEFAULT>]
[CPU_PER_CALL <limit|UNLIMITED|DEFAULT>]
[CONNECT_TIME <limit|UNLIMITED|DEFAULT>]
[IDLE_TIME <limit|UNLIMITED|DEFAULT>]
[LOGICAL_READ_PER_SESSION <limit|UNLIMITED|DEFAULT>]
[LOGICAL_READ_PER_CALL <limit|UNLIMITED|DEFAULT>]
[COMPOSITE_LIMIT <limit|UNLIMITED|DEFAULT>]
[PRIVATE_SGA <limit K | limit M | UNLIMITED | DEFAULT>]
```

```
CREATE PROFILE student
LIMIT SESSIONS_PER_USER 2;
```

Tworzenie nowego profilu

```
CREATE PROFILE <nazwa profilu> LIMIT
[FAILED_LOGIN_ATTEMPTS wartość]
[PASSWORD_LIFE_TIME wartość]
[{PASSWORD_REUSE_TIME|PASSWORD_REUSE_MAX} wartość]
[PASSWORD_LOCK_TIME wartość]
[PASSWORD_GRACE_TIME wartość]
[PASSWORD_VERIFY_FUNCTION {nazwa f. | NULL|DEFAULT}]
```

```
CREATE PROFILE hasła LIMIT
PASSWORD_LIFE_TIME 50
PASSWORD_GRACE_TIME 3;
```

Blokowanie i odblokowanie konta

```
ALTER USER <nazwa użytkownika>
[ACCOUNT LOCK]
[ACCOUNT UNLOCK]
[PASSWORD EXPIRE]
```

```
ALTER USER scott ACCOUNT LOCK;
```

```
ALTER USER scott ACCOUNT UNLOCK;
```

```
ALTER USER jones PASSWORD EXPIRE;
```

Obserwacja bazy danych (auditing)

- **Wykrywanie podejrzanych operacji, np. prób włamania się do systemu**
- **Monitorowanie i zbieranie informacji o aktywności systemu**
- **Obserwacja:**
  - **poleceń (DDL lub DML)**
  - **obiektów**
  - **uprawnień**

Włączanie obserwacji

- **Do pliku parametrów należy dodać:**
  AUDIT_TRAIL = xx
- **Należy zrestartować bazę danych**
- **AUDIT_TRAIL określa miejsce, gdzie będą zapisywane wyniki monitorowania:**
  - **wartość ustawiona na DB lub TRUE – obserwacje będą zapisywane do tabeli systemowej SYS.AUD$,**
    - **tabela SYS.AUD$ i perspektywy używane do obserwacji instalowane są przez skrypt cataudit.sql**
  - **wartość ustawiona na OS – wyniki obserwacji będą zapisywane w dzienniku systemu operacyjnego (o ile system posiada taki dziennik)**
- **Wyłączanie obserwacji: AUDIT_TRAIL = NONE**

Obserwacja poleceń i uprawnień

```
AUDIT <polecenie|uprawnienie> [, <polecenie|uprawnienie>...]
[BY <nazwa użytkownika> [, <nazwa użytkownika>...]]
[BY SESSION | BY ACCESS]
[WHENEVER [NOT] SUCCESSFUL]
```

```
NOAUDIT <polecenie|uprawnienie> [, <polecenie|uprawnienie>...]
[BY <nazwa użytkownika> [, <nazwa użytkownika>...]]
[WHENEVER [NOT] SUCCESSFUL]
```

```
AUDIT SELECT, INSERT BY scott
WHENEVER NOT SUCCESSFUL;
```

Obserwacja obiektów

```
AUDIT <operacja> [, <operacja>...]
ON <nazwa obiektu|DEFAULT>
[BY SESSION | BY ACCESS]
[WHENEVER [NOT] SUCCESSFUL]
```

```
NOAUDIT <operacja> [, <operacja>...]
ON <nazwa obiektu>
[WHENEVER [NOT] SUCCESSFUL]
```

```
AUDIT EXECUTE
ON usun_pracownika BY SESSION;
```

Opcje obserwacji obiektów - operacje

ALTER, AUDIT

SELECT

INSERT, UPDATE, DELETE

EXECUTE

GRANT

INDEX

LOCK

RENAME, COMMENT

Informacje o obserwacji

- **Perspektywa systemowa DBA_STMT_AUDIT_OPTS zawiera charakterystykę obserwacji poleceń**
- **Perspektywa systemowa DBA_PRIV_AUDIT_OPTS zawiera charakterystykę obserwacji uprawnień**
- **Perspektywa systemowa DBA_OBJ_AUDIT_OPTS zawiera charakterystykę obserwacji obiektów**
- **Pozostałe perspektywy systemowe związane z obserwacją: STMT_AUDIT_OPTION_MAP, AUDIT_ACTIONS, ALL_DEF_AUDIT_OPTS, USER_OBJ_AUDIT_OPTS, DBA_AUDIT_TRAIL, USER_AUDIT_TRAIL, DBA_AUDIT_SESSION, USER_AUDIT_SESSION, DBA_AUDIT_STATEMENT, USER_AUDIT_STATEMENT, DBA_AUDIT_OBJECT, USER_AUDIT_OBJECT, DBA_AUDIT_EXISTS**

Dziennik obserwacji

- **Dziennikiem obserwacji jest tabela systemowa SYS.AUD$**
- **Każda krotka dziennika zawiera informacje o:**
  - **użytkowniku**
  - **kodzie polecenia i przywileju**
  - **obiekcie**
  - **czasie**
- **W celu usunięcia z dziennika wszystkich krotek należy wydać polecenie:**
  TRUNCATE TABLE sys.aud$
- **W celu wyświetlenia wyników obserwacji obiektów wygodnie jest korzystać z perspektywy systemowej DBA_AUDIT_OBJECT:**

```
SELECT username, obj_name,
       action_name, ses_actions
FROM sys.dba_audit_object;
```

Wyzwalacze

- **Uruchamiane przez zdarzenia:**
  - **DML: INSERT, UPDATE, DELETE**
  - **DDL: ALTER, AUDIT, CREATE, DROP, GRANT, NOAUDIT, RENAME, REVOKE**
  - **systemowe: LOGON, LOGOFF, STARTUP, SHUTDOWN, SERVERERROR**
- **Mogą dotyczyć:**
  - **TABLE/VIEW**
  - **DATABASE**
  - **SCHEMA**

Przykłady wyzwalaczy

```
CREATE OR REPLACE TRIGGER drop_trigger
        BEFORE DROP ON scott.SCHEMA
        BEGIN
                RAISE_APPLICATION_ERROR (
                        num => -20000,
                        msg => 'Nie mozna usunac tabeli');
        END;
/
```

```
CREATE OR REPLACE TRIGGER logoff_trigger
        BEFORE LOGOFF ON DATABASE
        BEGIN
          DBMS_OUTPUT.PUT_LINE( 'Do widzenia' );
        END;
/
```