

PERTEMUAN 14

MENGONTROL AKSES USER

Tujuan Pembelajaran :

- Membuat User dan Privilege
- Pengaturan Role
- Penggunaan statement GRANT dan REVOKE untuk mengatur object privileges

TEORI DAN PERCOBAAN

14.1. Mengontrol Akses User

Pada lingkungan dengan banyak user, perlu dipelihara keamanan data (security) untuk mengakses dan menggunakan database.

Dengan security dari Oracle Server maka kita bisa :

- Mengontrol akses database
- Memberikan akses terhadap object spesifik yang ada dalam database
- Mengkonfirmasi pemberian privilege (hak akses) dalam data dictionary
- Membuat synonym untuk object database

Database security dapat dikelompokkan menjadi 2 (dua) hal :

- System security
- Data security

System security meliputi akses dan penggunaan database pada level system, semisal username dan password, ruang disk yang dialokasikan ke user, dan operasi system yang diperbolehkan pada user. **Database security** meliputi akses dan penggunaan database object dan perlakuan yang bisa diberikan oleh user terhadap object dari database.

14.2. Privileges

- **Privilege** adalah hak atas sesuatu.
- **System privilege** adalah hak akses terhadap database

- **Object privilege** adalah hak untuk memanipulasi isi dari database object
- **Schema (skema)** adalah kumpulan object, semisal table, view dan sequence.

Ada lebih dari 80 lebih privilege yang tersedia. DBA memiliki level privilege tertinggi yang bisa melakukan :

- Pembuatan user baru : CREATE USER
- Menghapus user : DROP USER
- Menghapus table : DROP ANY TABLE
- Membackup table : BACKUP ANY TABLE

14.3. Pembuatan User

Untuk membuat user baru digunakan perintah CREATE USER.

Percobaan 1 : Buat user baru dengan nama scott2 dan password macan

```
SQL> CREATE USER scott2
2 IDENTIFIED BY macan;

User created.
```

Sekali user dibuat, maka DBA dapat memberikan system privilege yang spesifik terhadap user tersebut, yaitu :

- CREATE SESSION : untuk berhubungan dengan database
- CREATE TABLE : untuk membuat table dalam skema user
- CREATE SEQUENCE : untuk membuat sequence
- CREATE VIEW : untuk membuat view
- CREATE PROCEDURE : untuk membuat procedure

Percobaan 2 : Berikan privilege kepada user SCOTT2 untuk membuat table, sequence dan view.

```
SQL> GRANT CREATE TABLE, CREATE SEQUENCE, CREATE VIEW
2 TO scott2;

Grant succeeded.
```

14.4. Role

Role adalah nama dari sekumpulan privilege yang saling berelasi dan diberikan pada user. Role dibuat untuk mempermudah proses pemberian dan pelepasan privilege. Hal pertama yang dikerjakan sehubungan dengan role adalah : pertama role dibuat dengan perintah CREATE ROLE nama_role. Kemudian DBA dapat menandai privilege apa saja yang diberikan kepada role. Setelah itu baru ditentukan user mana saja yang boleh memiliki role tersebut.

Percobaan 3 : Buat role *manager*

```
SQL> CREATE ROLE manager;
```

```
Role created.
```

Percobaan 4 : Berikan privilege untuk membuat table dan view pada role *manager*.

```
SQL> GRANT create table, create view  
2 TO manager;
```

```
Grant succeeded.
```

Percobaan 5 : Berikan role *manager* ke BLAKE dan CLARK.

```
SQL> GRANT manager to BLAKE,CLARK;
```

```
Grant succeeded.
```

14.5. Perubahan Password

Setiap user dapat merubah sendiri passwordnya dengan perintah ALTER USER.

Percobaan 6 : Ubah password dari user SCOTT2 menjadi singa.

```
SQL> ALTER USER scott2  
2 IDENTIFIED BY singa;
```

```
User altered.
```

14.6. Object Privileges dan Pembuatannya (Grant privileges)

Object privilege adalah privilege yang sesuai untuk membentuk perlakuan terhadap object database. Tabel berikut menjelaskan ada tidaknya object privilege terhadap suatu object dalam database.

Object privilege	Table	View	Sequence	Procedure
ALTER	ada		Ada	
DELETE	ada	Ada		
EXECUTE				Ada
INDEX	Ada			
INSERT	ada	Ada		
REFERENCES	Ada			
SELECT	ada	ada	Ada	
UPDATE	ada	Ada		

Perintah untuk memberikan object privilege :

```
GRANT      object_priv [(columns)]  
ON        object  
TO        [user|role|PUBLIC]  
[WITH GRANT OPTION];
```

Percobaan 7 : Berikan object privilege SELECT pada table EMP, untuk user BLAKE dan CLARK.

```
SQL> GRANT select  
2 ON emp  
3 TO BLAKE, CLARK;  
  
Grant succeeded.
```

Percobaan 8 : Berikan object privilege UPDATE pada kolom DNAME dan LOC pada table DEPT, untuk user BLAKE dan CLARK.

```
SQL> GRANT UPDATE(dname,loc)  
2 ON dept  
3 TO BLAKE, CLARK;  
  
Grant succeeded.
```

14.7. Penggunaan WITH GRANT OPTION dan PUBLIC

Dengan penambahan **WITH GRANT OPTION** pada saat memberikan suatu object privilege, maka akan membuat user yang diberi object privilege dapat memberikan hak yang diberikan atasnya itu kepada user yang lain.

Percobaan 9 : Berikan object privilege SELECT dan INSERT pada table DEPT kepada user SCOTT2, dan berikan keyword WITH GRANT OPTION, agar user SCOTT2 bisa memberikan object privilege yang serupa pada user yang lain.

```
SQL> GRANT SELECT,INSERT  
2 ON DEPT  
3 TO SCOTT2  
4 WITH GRANT OPTION;
```

Grant succeeded.

Jika pada saat pemberian suatu object privilege diberikan keyword PUBLIC, maka akan menyebabkan object privilege yang didefinisikan diberikan ke semua user yang ada dalam system.

Percobaan 10 : Berikan object privilege SELECT pada table DEPT yang dimiliki oleh SCOTT kepada semua user yang ada dalam system.

```
SQL> GRANT SELECT  
2 ON SCOTT.DEPT  
3 TO PUBLIC;
```

Grant succeeded.

14.8. Pemeriksaan Privileges

Untuk memeriksa keberadaan dari privilege, dapat dilihat pada data dictionary berikut :

Data dictionary	Keterangan
ROLE_SYS_PRIVS	System privilege yang diberikan pada role
ROLE_TAB_PRIVS	Table privilege yang diberikan pada role
USER_ROLE_PRIVS	Role yang bisa diakses oleh user
USER_TAB_PRIVS_MADE	Object privilege yang diberikan pada objectnya user
USER_TAB_PRIVS_RECD	Object privilege yang diberikan pada user
USER_COL_PRIVS_MADE	Object privilege atas kolom yang dipunyai user
USER_COL_PRIVS_RECD	Object priv. Yang diberikan pada user pada kolom yg spesifik.

14.9. Revoke (menghapus) Object Privileges

Untuk menghapus privilege, digunakan perintah **REVOKE**. Dengan perintah REVOKE, privilege yang diberikan ke user yang lain melalui WITH GRANT OPTION juga akan dihapus.

Sintak umum dari REVOKE :

```
REVOKE      {privilege }, privilege ...]|ALL}
ON          object
FROM {user[, user ...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```

Percobaan 11 : Hapus privilege SELECT dan INSERT yang diberikan pada user BLAKE dan CLARK pada table DEPT.

```
SQL> REVOKE select,insert
2 ON DEPT
3 FROM BLAKE,CLARK;

Revoke succeeded.
```

TAMBAHAN ROLE

MEMBUAT ROLE

Role adalah hak yang memperbolehkan user melakukan beberapa fungsi dalam database

Syntax:

GRANT role TO *nama_user* [WITH ADMIN OPTION];

Jika WITH ADMIN OPTION dipakai, maka user dapat meng-grant role kepada user lain.

MENGHAPUS ROLE

Untuk menghapus role, dapat dilakukan dengan perintah REVOKE dengan syntax:

REVOKE role FROM *nama_user*;

Macam-macam role yang dikenal Oracle adalah **Connect**, **Resource** dan **DBA**.

A. ROLE CONNECT

User dengan role jenis Connect dapat melakukan berbagai kegiatan terhadap database.

Sebagai contoh kita memberikan role Connect pada user bernama Angel:

```
Enter statements:
--CREATE USER ANGEL IDENTIFIED BY PANGEL;
GRANT CONNECT TO ANGEL;
```

Execute Save Script Clear Screen Cancel

Grant succeeded.

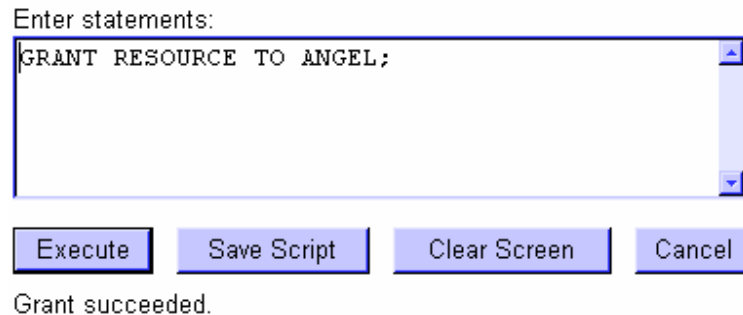
Maka, dengan role ini, user Angel dapat melakukan Selct, Insert, Update dan Delete dari table yang dimiliki oleh user lain setelah role di-grant. User juga dapat membuat table, view, sequence, cluster dan synonym.

B. ROLE RESOURCE

Role jenis Resource, user dapat mengakses lebih dalam lagi kedalam database Oracle.

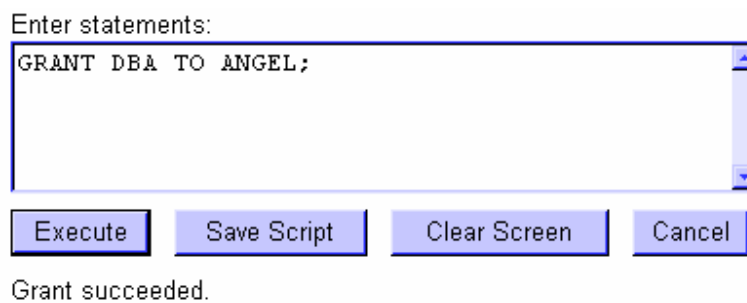
Selain role yang sama dengan Connect, role Resource juga diberi hak untuk membuat

Procedure, Trigger dan Index. Contoh, akan memberikan role Resource pada user Angel.

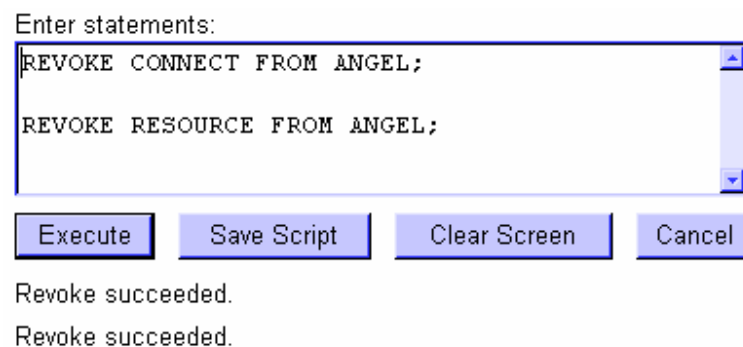


C. ROLE DBA

Untuk jenis role DBA, merupakan role tertinggi yang dapat melakukan apa saja kedalam sistem database. Sebaiknya jumlah user yang diberi role DBA dibatasi jumlahnya untuk mengurangi kerawanan pada sistem database kita.



Setelah melihat contoh diatas, user Angel telah memiliki role Connect, Resource dan DBA. Maka role Connect dan Resource pada user Angel sudah tidak diperlukan lagi karena redundant dengan role DBA (role DBA yang paling tinggi), sehingga dapat dihapus.



Sekarang user Angel dapat melakukan apa saja dengan role DBA-nya.

LATIHAN BAB 14

1. Buatlah sebuah table dengan nama Gaji dengan struktur table:

Nama : Varchar2(30)

Gaji : Number

Umur : Number

2. Buat user dengan nama Kirana dan Karana.
3. Berikan role CONNECT kepada user Kirana.
4. Berikan role CONNECT dan RESOURCE kepada user Karana.
5. Jelaskan apa arti role dan perbedaannya yang diberikan pada kedua user diatas.
6. Masukkan data berikut kedalam tabel Gaji:

NAMA	GAJI	UMUR
KIRANA	850000	30
KARANA	950000	30
LARA	850000	30

7. Karena user Kirana hanya diberikan role CONNECT, berikan privilege SELECT pada user Kirana.
8. Berikan role SELECT dan UPDATE pada user Karana sehingga dapat meng-update table Gaji pada field gaji.
9. Masuklah sebagai user Kirana, kemudian UPDATE-lah table Gaji. Apa yang terjadi?
10. Masuklah sebagai user Karana, kemudian INSERT-lah data baru pada table Gaji. Apa yang terjadi?
11. Masuklah sebagai user Karana, kemudian UPDATE-lah data baru pada kolom Umur table Gaji. Apa yang terjadi?
12. Masuklah sebagai user Karana, kemudian UPDATE-lah data baru pada kolom Gaji table Gaji. Apa yang terjadi?

NAMA	GAJI	UMUR
KIRANA	850000	30
KARANA	1000000	30
LARA	850000	30