Il linguaggio SQL: autorizzazioni

Sistemi Informativi T

Versione elettronica: 04.6.SQL.autorizzazioni.pdf

Autorità e privilegi

- Nei DBMS SQL ogni operazione deve essere autorizzata, ovvero l'utente che esegue l'operazione deve avere i privilegi necessari.
- I privilegi vengono concessi e revocati per mezzo delle istruzioni **GRANT** e **REVOKE**
- Un principio fondamentale è che un utente che ha ricevuto un certo privilegio può a sua volta accordarlo ad altri utenti solo se è stato esplicitamente autorizzato a farlo
- Mediante **GRANT** e **REVOKE** si controllano anche le autorità, ovvero il diritto ad eseguire azioni amministrative di un certo tipo
  - Ad esempio, se si ha l'autorità **SYSADM** (che include anche quella di **DBADM**) è possibile passare ad altri utenti l'autorità **DBADM** (Database Administrator Authority):

    ```
    GRANT DBADM ON DATABASE TO Pippo WITH GRANT OPTION;
    ```

    in cui la clausola **WITH GRANT OPTION** autorizza l'utente Pippo a passare l'autorità ad altri utenti

GRANT: privilegi per SCHEMI

- Il formato dell'istruzione GRANT per assegnare privilegi su schemi è:

```
GRANT < lista di privilegi >
ON SCHEMA <schema name>
TO { <lista di utenti e gruppi> | PUBLIC }
[ WITH GRANT OPTION ]
```

- I privilegi possibili sono

CREATEIN: per creare oggetti (tables,views) nello schema

ALTERIN: per modificare la struttura di tables dello schema

DROPIN: per eliminare oggetti dallo schema

```
GRANT CREATEIN, ALTERIN
ON SCHEMA B16884
TO USER S00125, S00126
```

GRANT: privilegi per TABLES e VIEWS

- Il formato per assegnare privilegi su tables e views è:

```
GRANT { ALL | < lista di privilegi > }
ON [ TABLE ] <table name>
TO { <lista di utenti e gruppi> | PUBLIC }
[ WITH GRANT OPTION ]
```

- I privilegi possibili includono quello “master” di **CONTROL** (posseduto automaticamente da chi ha creato l’oggetto) e quelli di **ALTER**, **DELETE**, **INSERT**, **SELECT**, **INDEX**, **REFERENCES** e **UPDATE**
- Per **REFERENCES** e **UPDATE** si può anche specificare una lista di attributi
- **ALL** conferisce tutti i privilegi che chi conferisce il privilegio può passare ad altri, ma in ogni caso non **CONTROL**
- **PUBLIC** concede i privilegi specificati a tutti gli utenti, inclusi quelli futuri

GRANT: dettagli sui privilegi

- CONTROL: comprende tutti i privilegi (su una view sono solo SELECT, INSERT, DELETE e UPDATE). Inoltre permette di conferire tali privilegi ad altri utenti; può essere conferito solo da qualcuno che ha autorità SYSADM o DBADM
- ALTER: attribuisce il diritto di modificare la definizione di una tabella
- DELETE: attribuisce il diritto di cancellare righe di una tabella
- INDEX: attribuisce il diritto di creare un indice sulla tabella
- INSERT: attribuisce il diritto di inserire righe nella tabella
- REFERENCES: attribuisce il diritto di definire foreign keys in altre tabelle che referenziano la tabella
- SELECT: attribuisce il diritto di eseguire query sulla tabella/vista e di definire VIEW
- UPDATE: attribuisce il diritto di modificare righe della tabella/vista
- Per eseguire una query, è necessario avere il privilegio di SELECT o di CONTROL su tutte le table e le view referenziate dalla query

GRANT: privilegio REFERENCES

- L'unico privilegio che non agisce "direttamente" sulla tabella in oggetto è REFERENCES
- Il motivo per cui tale privilegio va concesso esplicitamente è legato alle politiche di reazione associabili a una foreign key
- Ad es., la definizione:

```
CREATE TABLE Esami (
    Matricola char(5)   NOT NULL,
    CodCorso     int    NOT NULL,
    ...
    FOREIGN KEY CodCorso REFERENCES Corsi
    ON DELETE NO ACTION          -- cancellazioni non permesse
```

di fatto permetterebbe all'utente che crea la relazione ESAMI di bloccare ogni cancellazione sulla relazione CORSI

GRANT: esempi (1)

- Paperino autorizza Pippo e Topolino a leggere la relazione Employee e a modificare i valori di Salary; inoltre concede loro di passare questo privilegio ad altri utenti:

```
Paperino>  GRANT SELECT, UPDATE(Salary)
           ON TABLE Employee TO USER Pippo, USER Topolino
           WITH GRANT OPTION
```

- … e Pippo ne approfitta subito:

```
Pippo>     GRANT SELECT
           ON TABLE Employee TO USER Pluto
```

- Pluto può eseguire query su Employee, ma non aggiornamenti; inoltre non può passare lo stesso privilegio ad altri

GRANT: esempi (2)

- Se ora Topolino esegue:

```
Topolino>  GRANT UPDATE(Salary)
           ON TABLE Employee TO USER Pluto
           WITH GRANT OPTION
```

  allora Pluto può anche modificare i valori di Salary e passare lo stesso privilegio ad altri

- Quindi:

```
Pluto>     GRANT ALL
           ON TABLE Employee TO USER Minnie
```

  trasferisce a Minnie (ma senza GRANT OPTION) il solo privilegio sulla modifica dei valori di Salary

GRANT: esempi (3)

- Pippo crea una vista su Employee:

```
Pippo>      CREATE VIEW NomiEmp(NOME,COGNOME)
            AS SELECT LASTNAME,FIRSTNME
               FROM EMPLOYEE
```

e permette a Orazio di fare query su tale vista:

```
Pippo>      GRANT SELECT
            ON TABLE NomiEmp TO USER Orazio
```

- Quindi ora Orazio può interrogare **`NomiEmp`**, ma non **`Employee`**!

REVOKE

- Il formato dell'istruzione REVOKE per revocare privilegi su tables e views è:

```
REVOKE { ALL | < lista di privilegi > }
ON [ TABLE ] <table name>
FROM { <lista di utenti e gruppi> | PUBLIC }
```

- A differenza del GRANT, per eseguire REVOKE bisogna avere l'autorità **SYSADM** o **DBADM**, oppure il privilegio di **CONTROL** sulla relazione
- Il REVOKE non agisce a livello di singoli attributi; pertanto non si possono revocare privilegi di **UPDATE** solo su un attributo e non su altri (per far ciò è quindi necessario revocarli tutti e poi riassegnare solo quelli che si vogliono mantenere)

REVOKE: esempi

- Se Pippo, che non ha autorità DBADM o SYSADM, né CONTROL su Employee, prova ad eseguire:

```
Pippo>      REVOKE SELECT
            ON TABLE Employee FROM Pluto
```

  si verifica un errore

- Viceversa, se Paperino ha autorità **DBADM** ed esegue

```
Paperino>   REVOKE SELECT
            ON TABLE Employee FROM Pippo, Topolino
```

  né Pippo né Topolino possono più eseguire query su Employee, ma continuano a poter aggiornare Salary

- La vista NomiEmp definita da Pippo su Employee diventa “inoperativa”, ovvero non più utilizzabile (e quindi Orazio non può più interrogarla)
- Si noti che Pluto mantiene il privilegio SELECT su Employee
- Lo standard SQL prevede una gestione del REVOKE più complessa, che include anche effetti di revoca dei privilegi “in cascata” (in cui quindi Pluto perderebbe il privilegio di SELECT)

Riassumiamo:

- Mediante le istruzioni **GRANT** e **REVOKE** è possibile, rispettivamente, concedere e revocare privilegi e autorità
- E' inoltre possibile, se se ne ha il diritto, concedere il diritto di propagare tali privilegi/autorità, specificando la clausola **WITH GRANT OPTION**