Controllo dell'Accesso

Lorenzo Vaccarecci

10 Maggio 2024

1 Introduzione

Il problema è che **non tutti gli utenti possono eseguire le stesse operazioni** (*es. gestore e cliente*). Lo scopo del controllo dell'accesso è quello di **limitare e controllare** le operazioni degli utenti e **prevenire** azioni che potrebbero compromettere la correttezza e la sicurezza del sistema. Le operazioni vengono catturate prima di essere eseguite.

2 Politiche di sicurezza

- Regole e principi: per la protezione delle informazioni
- Insieme di direttive ad alto livello

Es. "le valutazioni dei film possono essere viste solo dal responsabile della videoteca".
Tre entità principali:

- Oggetti: sono le risorse a cui vogliamo garantire protezione (le tabelle)
- Soggetti: sono le entità (gli utenti)
- Privilegi: descrivono quali operazioni possono essere eseguite dai soggetti sugli oggetti

3 Controllo degli Accessi in SQL

- Sistema chiuso: un accesso è concesso solo se è stato esplicitamente permesso
- Amministrazione decentralizzata (mediante **ownership**): l'utente che crea una relazione, riceve privilegi di controllo su di essa e può concedere o revocare privilegi
- `GRANT`: se un privilegio è concesso con grant option l'utente che lo riceve può non solo esercitare il privilegio, ma anche concederlo ad altri
- `REVOKE`: toglie i privilegi a uno o più utenti

3.1 `GRANT`

L'inserimento di una nuova autorizzazione avviene tramite il comando `GRANT`:

```
GRANT {<lista privilegi> | ALL PRIVILEGES}
ON <nome oggetto>
TO {<lista utenti> | <lista ruoli> | PUBLIC}
[WITH GRANT OPTION];
```

La lista dei privilegi è composta da: `SELECT`, `INSERT`, `UPDATE`, `DELETE`, ecc.
`WITH GRANT OPTION` (opzionale) consente la delega dell'amministrazione dei privilegi. Alla creazione di una risorsa, il creatore riceve automaticamente tutti i privilegi su di essa quindi **i comandi di `GRANT` corrispondenti vengono eseguiti automaticamente** dal sistema. Possiamo concedere il privilegio anche su singole colonne. Esempi di `GRANT` (p.20).

3.2 `REVOKE`

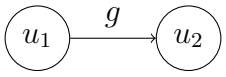
La revoca di privilegi avviene tramite il comando `REVOKE`:

```
REVOKE [GRANT OPTION FOR] <lista privilegi>
ON <nome oggetto>
FROM <lista utenti>
[CASCADE | RESTRICT];
```

La clausola opzionale `GRANT OPTION FOR`, se presente, revoca il privilegio di concedere il privilegio ad altri utenti. **Un utente può revocare solo i privilegi concessi da lui.**

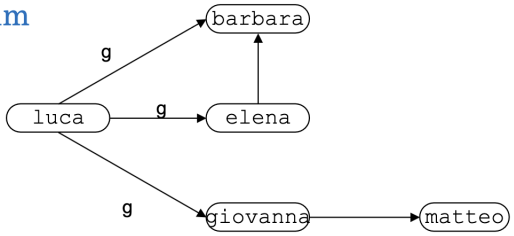
4 Rappresentazione delle autorizzazioni

In modo astratto le informazioni contenute nei cataloghi possono essere rappresentate in astratto come un insieme di grafi (grafi delle autorizzazioni). Esiste un grafo per ogni privilegio p su una certa tabella. Un grafo delle interrogazioni per il privilegio sulla tabella R contiene:

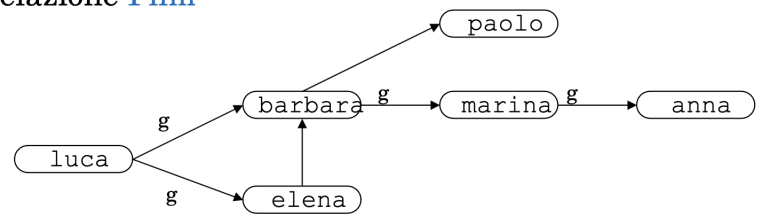


Un nodo per ogni utente che ha il privilegio p sulla tabella R. L'arco uscente da u_1 verso u_2 indica che u_1 ha concesso il privilegio a u_2 ed è etichettato g se è concesso con `GRANT OPTION`.

Grafo delle autorizzazioni relativo al privilegio SELECT e alla relazione Film

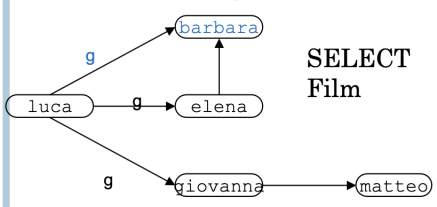


Grafo delle autorizzazioni relativo al privilegio INSERT e alla relazione Film

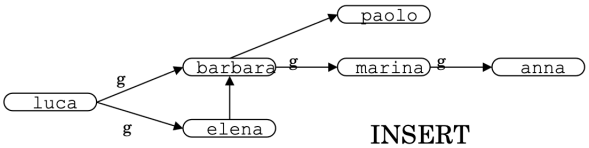


Un comando di `GRANT` può essere eseguito parzialmente in alcuni casi:

ESEMPIO



SELECT
Film

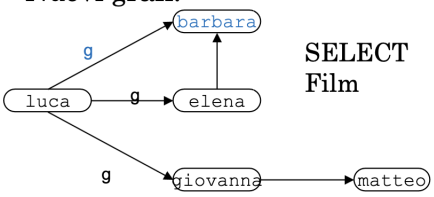


INSERT
Film

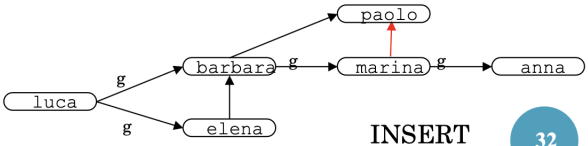
```
marina:  GRANT select, insert ON Film TO Paolo;
```

Marina ha acquisito il privilegio INSERT con grant option, quindi lo può delegare, ma non ha acquisito il privilegio di SELECT, quindi non lo può delegare (esecuzione parziale del comando)

Nuovi grafi:



SELECT
Film

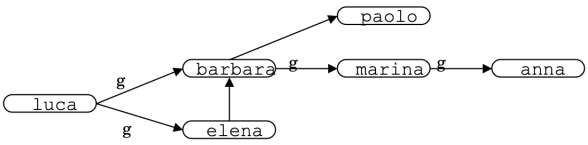


INSERT
Film

32

Il comando `REVOKE`:

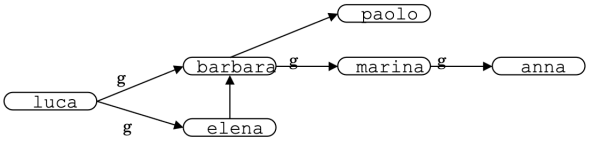
ESEMPIO



INSERT
Film

```
barbara:  REVOKE insert ON Film FROM marina RESTRICT;
```

Il privilegio è stato concesso a Barbara a Marina con GRANT OPTION e Marina lo ha a sua volta concesso ad Anna. Quindi sarebbe necessario rimuovere due archi dal grafo, ma poiché la REVOKE è RESTRICT, questo non è possibile e il grafo non cambia

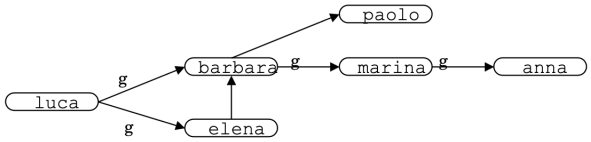


INSERT
Film

36

`RESTRICT`: non elimina altri nodi oltre quello designato
`CASCADE`: elimina anche tutti i nodi che dipendono da quello designato, quindi se un nodo è puntato da due nodi, viene eliminato solo l'arco che parte dal nodo chiamante (però i figli vengono eliminati).

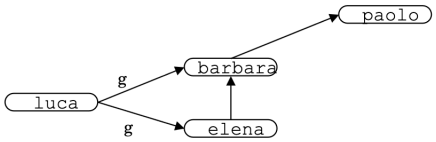
ESEMPIO



INSERT
Film

```
barbara:  REVOKE insert ON Film FROM marina CASCADE;
```

Il privilegio è stato concesso a Barbara a Marina con GRANT OPTION e Marina lo ha a sua volta concesso ad Anna. Quindi è necessario rimuovere due archi dal grafo, poiché la REVOKE è CASCADE, questo è possibile e il grafo diventa:



INSERT
Film

Se non si specifica niente, il comportamento di default è `RESTRICT`.

5 Ruoli

I ruoli sono funzioni svolte in ambito di un'organizzazione. Gli utenti sono abilitati a ricoprire uno o più ruoli. I privilegi possono essere concessi anche ai ruoli, le autorizzazioni specificate per un ruolo sono quelle necessarie per esercitare le funzioni connesse al ruolo stesso. Semplificano l'attribuzione dei privilegi ai soggetti.

```
1    CREATE ROLE <nome ruolo>;
2    DROP ROLE <nome ruolo>;
3    SET ROLE <nome ruolo>; -- Associazione dimanica di un ruolo all'utente
   della sessione attiva
```

Nel comando `GRANT` mettiamo

```
1    GRANT <lista ruoli>
2    TO {<lista utenti> | PUBLIC}
3    [WITH ADMIN OPTION]; -- Analogo a GRANT OPTION
```

E' possibile dare una gerarchia ai ruoli:

```
1    GRANT <ruolo_figlio>
2    TO <ruolo_padre>;
```

I privilegi del figlio sono acquisiti dal padre ma non viceversa.
Esempio:

```
1    CREATE ROLE direttoreVideoteca;
2    CREATE ROLE commesso;
3
4    GRANT SELECT ON film TO commesso;
5    GRANT INSERT, DELETE ON film TO direttoreVideoteca;
6    GRANT commesso TO direttoreVideoteca;
```

Di solito prima si danno i privilegi(specifici) ai figli e poi al padre.

5.1 REVOKE

```
REVOKE [ADMIN OPTION FOR]
<lista ruoli>
FROM <lista utenti>
```

6 Controllo dell'accesso basato sul contenuto

6.1 Autorizzazione su Viste

Le viste consentono di realizzare il così detto controllo dell'accesso in base al contenuto. Permettono di concedere anche privilegi statistici (es. una vista che computa il numero di noleggi effettuati da ogni cliente e concedere all'utente l'accesso alla vista invece che alla tabella).
Chi può creare una vista?
Gli utenti che hanno il privilegio di `SELECT` su tutte le tabelle utilizzate dalla vista.
Il proprietario della vista quali privilegi può esercitare sulla vista?
Bisogna considerare:

- Le autorizzazioni sulle tabelle coinvolte ($P1$)
- Regole per formare le autorizzazioni $P2$ (p. 26)

Il proprietario ha i privilegi che ha sono l'intersezione tra $P1$ e $P2$.