

## Controllo dell'accesso

Consente il gestore degli accessi al DB, per rendere controllati e sicuri gli accessi.

Nessun utente di un DB possono eseguire le stesse operazioni  
→ il gestore può eseguire tutto  
→ il cliente può leggere le tabelle (magari alcune)  
ma non modificarle

DCL

Il Data Control Language permette di gestire dati sensibili in SQL,  
estende la storage definition language per gestire i privilegi sul DB.

Il controllo dell'accesso si basa su **politiche di sicurezza**  
(regole e direttive)

Lo specifica di delle politiche → basa su 3 entità:

- oggetti → risorse a cui vogliamo fornire protezione
- soggetti → coloro che richiedono l'accesso agli oggetti  
(utenti, gruppi, ruoli)
- privilegi → operazioni che i soggetti possono eseguire sugli oggetti

L'autorizzazione è una tripla:  $(S, O, P)$

S può accedere a O con certi P

Le autorizzazioni sono memorizzate in cataloghi

Il gestore puo' scegliere di negare l'accesso o forza eseguire in modo parziale

es. (borbora, film, SELECT)

Borbora puo' leggere ma non inserire / modificare se non presentate un'altra forma nei cataloghi che lo specifichi

Il modello usa una politica discrittiva adottando un sistema di ruoli



L'amministrazione dei privilegi e decentralizzata (multi ownership), ovvero esistono piu' soggetti autorizzati.

Chi crea l'oggetto ha tutti i privilegi su esso e puo' scegliere quali fornire ad altri.

La delega dei privilegi avviene mediante **grant option**

- chi riceve tali privilegi puo' a sua volta fornirlo ad altri
- se non con grant option non puo' fornirlo ad altri

In SQL

- GRANT concede privilegi

- REVOKE *(oggi è privilegio)*

Note: per eseguire tali comandi bisogna avere il permesso

GRANT {<listo privilegi> | ALL PRIVILEGES} *p*  
ON < nome oggetto > *s*  
TO {<listo utenti> | PUBLIC} *s*  
[WITH GRANT OPTION]

Note: solo chi crede la risorsa può fare DROP o ALTER  
permesso non delegabile ad altri *↳ cambia lo schema*

c). Iva: GRANT update (telefno) ON Clienti TO marco, giovanni;

↓  
ha i<sup>e</sup> privilegi = *P* ↓ modifica solo *O* ↓ *S*  
solo *in una colonna* *sorano*  
sera → riconosce solo le grant option  
*z triple diverse*

• REVOKE [GRANT OPTION FOR] <listo privilegi>

ON < nome oggetto >

FROM <listo utenti>

{ RESTRICT | CASCADE }

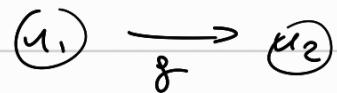
es. REVOKE update, insert ON Video FROM elena;

Le triple vengono rappresentati nei cataloghi tramite graph delle autorizzazioni.

Un graph A privilegi su una certa tabella.

Un grafo delle interrogazioni per p sulla tabella R

- un nodo x ogni utente che ha p >= R

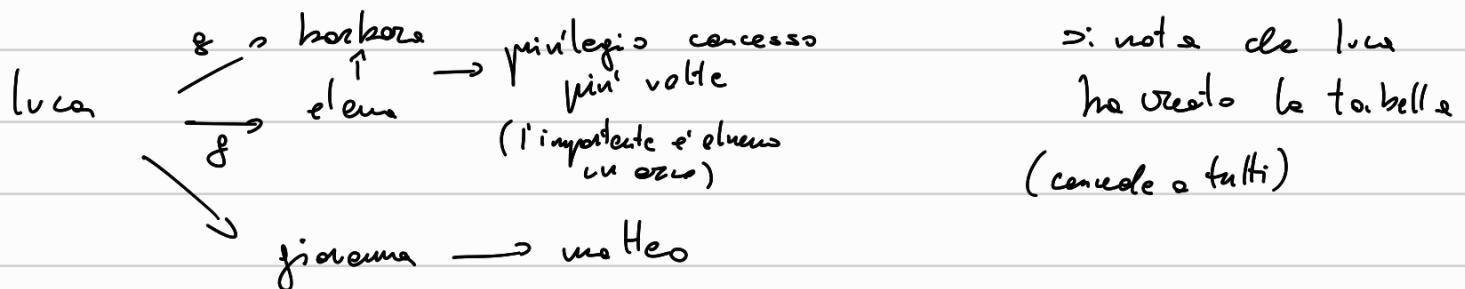


- un arco se  $u_1$  ha concesso p a  $u_2$

↓

etichettato con g se concesso con GRANT OPTION

es grafo su p select più la relazione Film



Quando viene eseguito GRANT il gestore asforna il grafo per vedere se l'utente <sup>che vuole fare</sup> ha un arco entrante con g (come bene può fornire e sue volte) osservando intanto se l'utente è nel grafo (cioè ha p)

L'esecuzione parallela di un comando è possibile, ad esempio se un utente ha p in insert ma non in SELECT e il GRANT viene chiesto in entrambi i privilegi (viste delle sole x insert)

→ REVOKE cancella l'arco del grafo (ed eventualmente il nodo)  
(g)

REVOKE con RESTRICT cancella l'arco solo se non vi sono  
altre autorizzazioni dipendenti da esso  
(oppure elimina l'arco ma il privilegio rimane da altri)

REVOKE con CASCADE cancella l'orso e tutte le autorizzazioni date dal nodo in cascata

Nota: se un nodo ha più ordini entranti, una REVOKE non cancella il privilegio necessariamente, avendo il nodo resto ancora collegato da altri (può togliere però il GRANT)

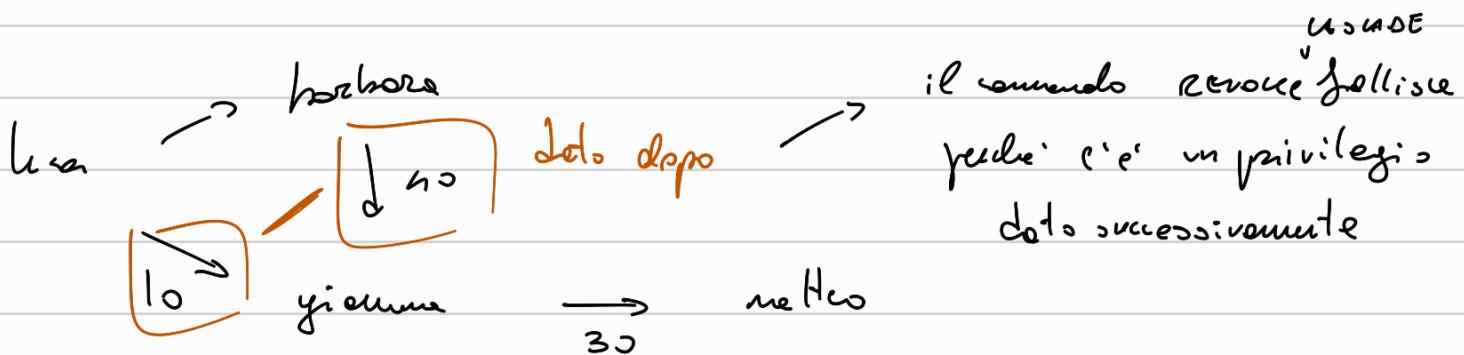


Lo REVOKE può togliere  
più autorizzazioni a uno  
o più utenti

e quindi le autorizzazioni concesse  
cedono

(x concedere il nodo deve avere un  
ordine con g entrante)

Nota: considerando i timestamp di quando sono state fornite  
le autorizzazioni per il meccanismo di rimozione in cascata,  
le implementazioni non garantiscono



Si puo' controllare l'eccesso su tuple attraverso formule sui viste (restrizioni di tuple), altrimenti il GRANT non lo permette

/16-05

## Autorizzazioni con viste

Le viste permettono anche di visualizzare dati statistici

- Chi puo' creare una vista? Chi ha il privilegio di select sulle relazioni che la vista coinvolge (cioe' potervi accedere)

- Quali privilegi puo' dare sulla vista, essendo owner?

Dipende da

- P1 → privilegi su tutte le relazioni su cui la vista e' definita  
P2 → operazioni che possono essere eseguite sulla vista

I privilegi consentibili sono P1 ∩ P2

Tali privilegi possono essere delegati con i medesimi permessi  
(pur avendo il privilegio di GRANT OPTION)

- Bebbere ha solo il priv. di SELECT acquisito con GRANT OPTION su film

Bebbere puo' → creare la vista su film vedra' la il select su film

$$P1 = \{ \text{SELECT} \}$$

$$P2 = \{ \text{SELECT, INSERT, UPDATE, DELETE} \} \rightarrow \text{e' owner della vista, ha tutti i permessi}$$

puo' delegare? si ha GRANT OPTION

Caso? → P1 ∩ P2 aveva solo il SELECT

Se ha una non univocita' (es. join, group by) puo eseguire  
solo select sulle viste (il group by aggrega, non se ne ha come espedire  
il "mediatore")

Se ha tutti i permessi sulla tabella base e un owner della  
vista su quella selezione non concede il permesso di accesso  
a quelle viste, non puo accedere alle viste