Controllo dell'accesso

Sommario

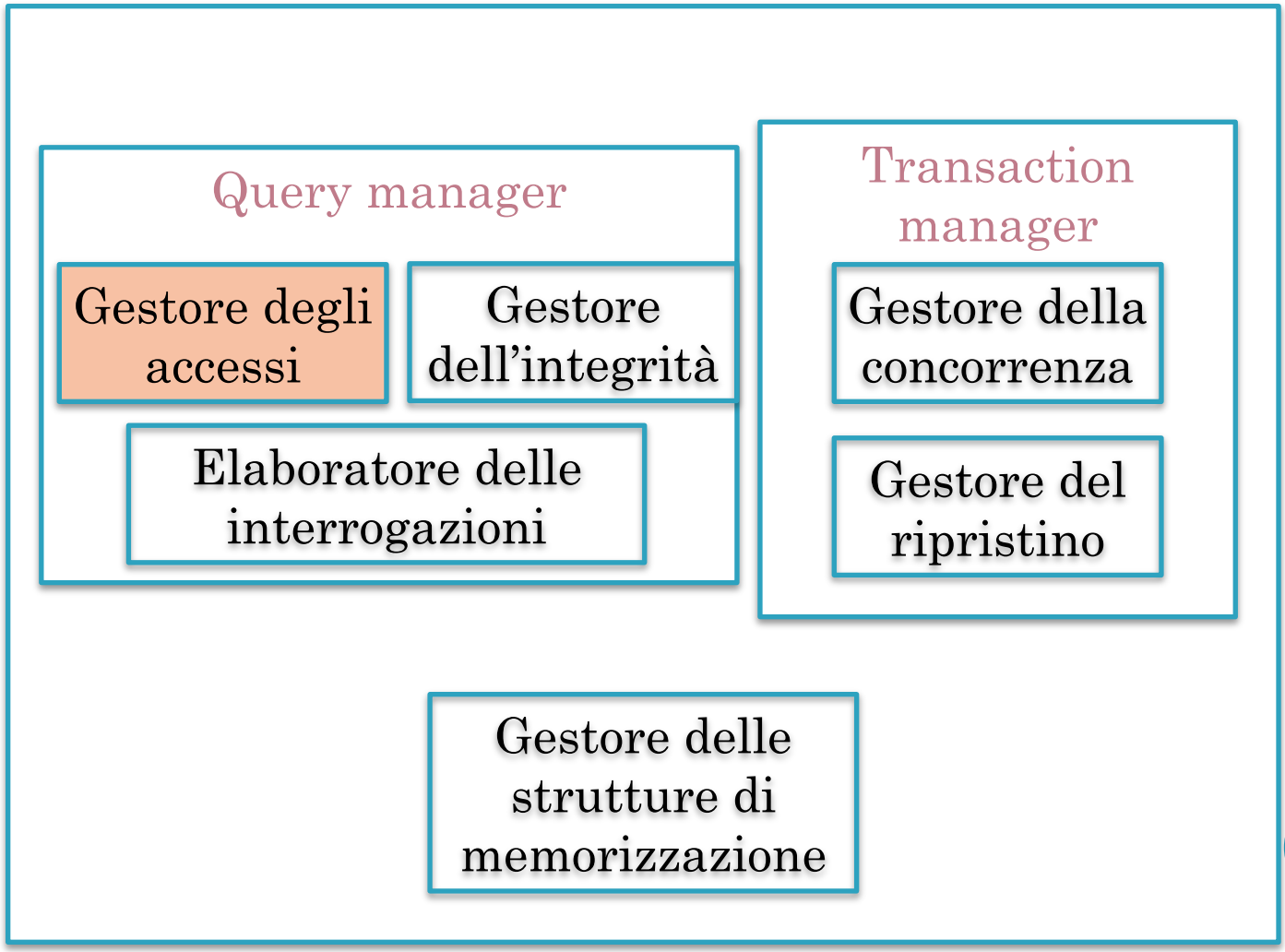
- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Sommario

- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Componenti di un DBMS

DBMS



Il problema

- Non tutti gli utenti di un sistema di basi di dati possono eseguire le stesse operazioni
  - Il gestore della videoteca può eseguire tutte le operazioni su tutte le tabelle della base di dati
  - Il cliente della videoteca può solo leggere le tabelle della base di dati
- Il controllo dell'accesso regola le operazioni che si possono compiere sui dati (e altre risorse) in una base di dati
- Scopo:
  - limitare e controllare le operazioni che gli utenti effettuano
  - prevenire azioni accidentali o deliberate che potrebbero compromettere la correttezza e la sicurezza dei dati

Il problema

- La gestione delle autorizzazioni può essere oltremodo complessa
- Il DBMS fornisce strumenti per realizzare le protezioni, che sono definite dall'amministratore della base dati (DBA)
- Il DBA ha il compito di conferire agli utenti i "giusti" privilegi, utilizzando particolari comandi SQL

- Data Control Language
  - Estende lo Storage Definition Language (SDL) con altri comandi utili all'amministrazione della base di dati, inclusa la gestione dei privilegi

Sommario

- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Concetti fondamentali

- Il controllo dell'accesso si basa sulla specifica di opportune politiche di sicurezza, che dipendono dal dominio considerato

- Politiche di sicurezza:
  - regole e principi secondo cui l'organizzazione vuole che siano protette le proprie informazioni
  - insieme di direttive ad alto livello che esprimono le scelte compiute da un organizzazione in merito alla protezione dei propri dati

- Esempio di politica di sicurezza:
  - "le valutazioni dei film possono essere viste solo dal responsabile della videoteca"

Concetti fondamentali

- La specifica delle politiche di sicurezza si basa su tre entità fondamentali:
  - oggetti risorse a cui vogliamo garantire protezione
  - soggetti entità “attive” che richiedono di poter accedere agli oggetti
  - privilegi che determinano le operazioni che i soggetti possono fare sugli oggetti

Concetti fondamentali - oggetti

- Qualsiasi componente dello schema di una base di dati è una risorsa e può essere considerata un oggetto
  - tabella
  - vista
  - attributo all'interno di una tabella o di una vista
  - dominio
  - procedura
  - ...

Concetti fondamentali - soggetti

- I soggetti possono essere classificati in varie tipologie:
  - utenti: singoli individui che si connettono al sistema
    - utente barbara
    - utente nino
    - utente 5367
  - gruppi: insiemi di utenti
  - ruoli: funzioni aziendali a cui assegnare un insieme di privilegi per lo svolgimento delle loro mansioni
    - gestore videoteca
    - cliente videoteca
  - …

Concetti fondamentali - privilegi

- I privilegi descrivono le azioni permesse sulle risorse
  - lettura
  - scrittura
  - aggiornamento
  - esecuzione
  - …
- Le azioni permesse su una certa risorsa dipendono dal tipo della risorsa
  - tabella: lettura, scrittura, aggiornamento,…
  - procedura: esecuzione,…

Concetti fondamentali - autorizzazioni

- Le politiche di sicurezza sono tradotte in un insieme di autorizzazioni che stabiliscono gli specifici diritti che i vari soggetti, abilitati ad accedere al sistema, possono esercitare sugli oggetti

- le autorizzazioni nel loro formato base possono essere rappresentate mediante una tupla (s,o,p) dove:
  - s è il soggetto a cui l'autorizzazione è concessa
  - o è l'oggetto sui cui è concessa l'autorizzazione
  - p è il privilegio che s può esercitare su o

- Si parla in questo caso di controllo dell'accesso discrezionale

- Le autorizzazioni sono dati di sistema e, come tali, vengono memorizzate nei cataloghi di sistema

Concetti fondamentali

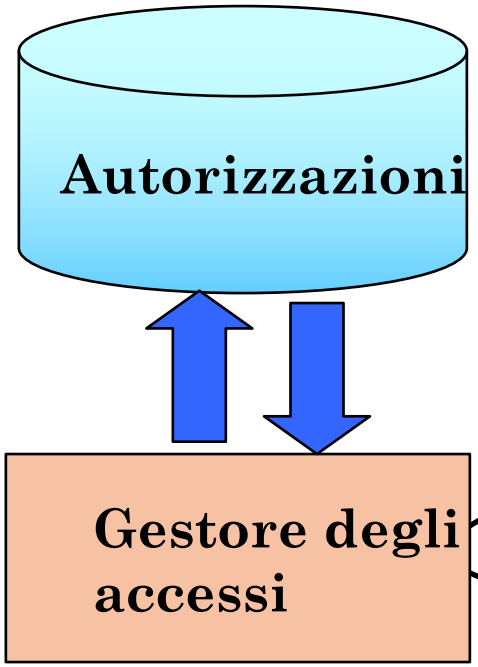
- Il controllo dell'accesso è effettuato mediante il gestore degli accessi, detto anche reference monitor
  - intercetta ogni comando inviato al DBMS
  - stabilisce, tramite l'analisi delle autorizzazioni, se il soggetto richiedente può essere autorizzato a compiere l'accesso richiesto
  - la concessione dell'autorizzazione può essere
    - totale: l'accesso richiesto può essere totalmente eseguito
    - parziale: sono una parte dell'accesso richiesto può essere eseguito (vedremo meglio in seguito)
    - negata: l'accesso richiesto non può essere eseguito

Concetti fondamentali

DBA

Politiche
di sicurezza

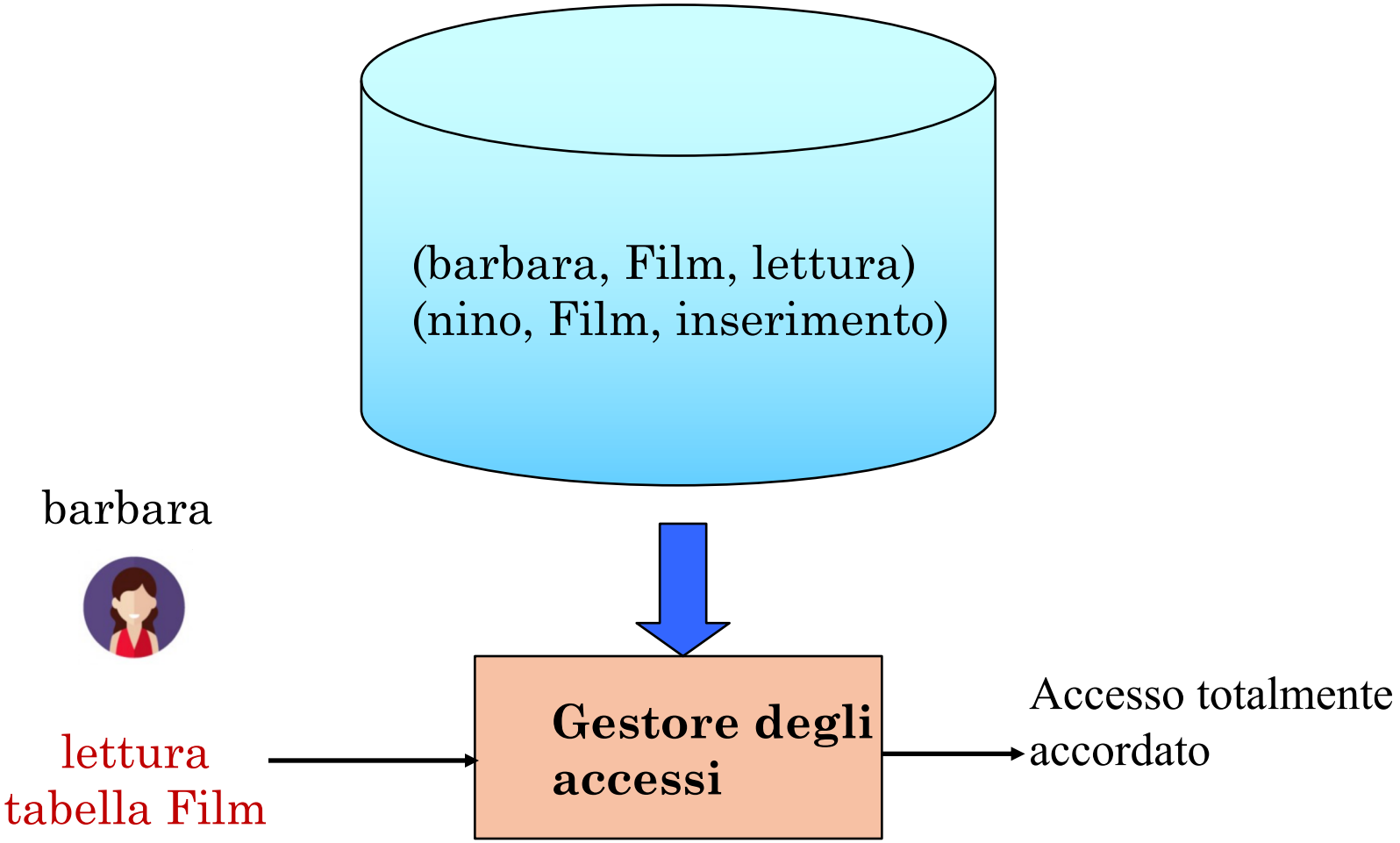
utente



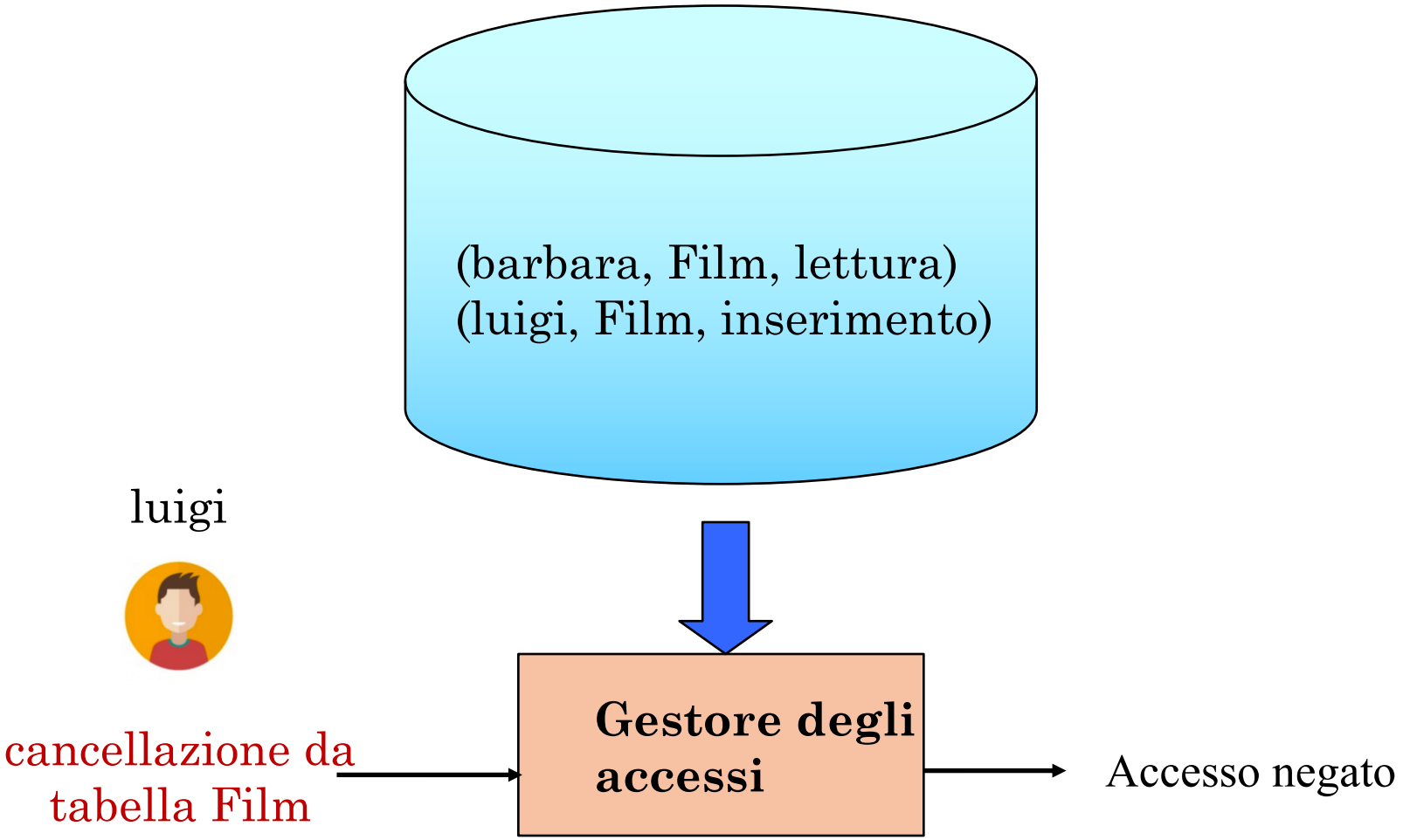
Accesso totalmente o parzialmente accordato

Accesso negato

ESEMPIO


(barbara, Film, lettura)
(nino, Film, inserimento)
barbara
lettura
tabella Film
Gestore degli accessi
Accesso totalmente accordato

ESEMPIO


(barbara, Film, lettura)
(luigi, Film, inserimento)
luigi
cancellazione da tabella Film
Gestore degli accessi
Accesso negato

Sommario

- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Principi generali

- Il modello realizza una politica di tipo discrezionale adottando il paradigma di sistema chiuso:
  - un accesso è concesso solo se esiste un'esplicita autorizzazione per esso
- L'amministrazione dei privilegi è decentralizzata mediante ownership:
  - l'utente che crea una relazione, riceve tutti i privilegi su di essa ed anche la possibilità di delegare ad altri tali privilegi

Principi generali

- La delega dei privilegi avviene mediante grant option
- Se un privilegio è concesso con grant option l'utente che lo riceve può non solo esercitare il privilegio, ma anche concederlo ad altri
- Un utente può quindi concedere un privilegio su una determinata relazione solo se è il proprietario della relazione o ha ricevuto tale privilegio con grant option

Principi generali

- Nei DBMS relazionali, le autorizzazioni possono essere specificate, quindi concesse, e revocate tramite comandi SQL
  - comando GRANT
    - concede privilegi su una risorsa a uno o più utenti
  - comando REVOKE
    - toglie a uno o più utenti i privilegi che erano stati loro concessi
- Ogni richiesta di esecuzione di comando SQL deve poi essere autorizzata, ovvero l'utente che esegue l'operazione deve avere i privilegi necessari
- 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
- Vengono utilizzati opportuni cataloghi per memorizzare le autorizzazioni e verificare gli accessi

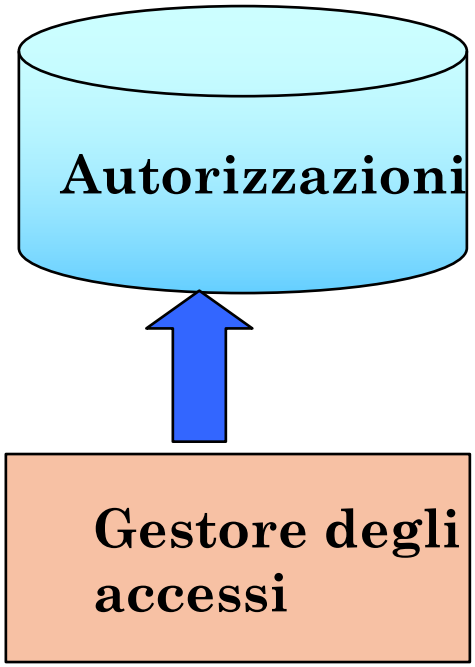
Sommario

- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Concessione e revoca di autorizzazioni

DBA

GRANT
REVOKE



Comando grant

- L'inserimento di una nuova autorizzazione nel sistema e quindi la concessione di nuovi privilegi avviene tramite il comando GRANT

COMANDO GRANT

<lista privilegi> indica l'insieme dei privilegi concessi con il comando GRANT

la parola chiave ALL PRIVILEGES indica tutti i privilegi previsti dal modello

<nome oggetto> indica il nome della risorsa della base di dati su cui sono concessi i privilegi (ad esempio, una tabella)

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

<lista utenti >indica l'insieme degli utenti a cui il comando si applica

la parola chiave PUBLIC consente di specificare le autorizzazioni implicate dal comando per tutti gli utenti del sistema

la clausola opzionale WITH GRANT OPTION consente la delega dell'amministrazione dei privilegi

Comando grant

- Privilegi specificabili (nel caso di oggetti che corrispondono a tabella) a
  - INSERT
    - permette di inserire una nuova tupla nella tabella
  - UPDATE
    - permette di aggiornare le tuple della tabella
  - DELETE
    - permette di eliminare tuple dalla tabella
  - SELECT
    - permette di utilizzare la tabella all'interno di un'interrogazione

- i privilegi INSERT, UPDATE e SELECT possono essere concessi selettivamente solo su alcune colonne di una relazione

- sono supportati altri privilegi:
  - references, che permette di utilizzare una colonna in un vincolo o asserzione
  - trigger, che permette di specificare trigger che operano su una certa relazione
  - execute, che permette l'esecuzione di una procedura o funzione
  - …

Privilegi del creatore della risorsa

- Alla creazione di una risorsa, il sistema concede tutti i privilegi su tale risorsa all'utente che ha creato la risorsa
  - I comandi di GRANT corrispondenti vengono eseguiti automaticamente dal sistema
- Solo il creatore della risorsa ha il privilegio di eliminare una risorsa (DROP) e modificarne lo schema (ALTER)
- il privilegio di eliminare e modificare una risorsa non può essere concesso a nessun altro utente

Privilegi del DBA

- L'amministratore della base di dati possiede tutti i privilegi su tutte le risorse

Esempio

- Supponiamo che tutte le tabelle della base di dati della videoteca siano state create dall'utente `luca`

- `luca: GRANT update(telefono) ON Clienti TO marco;`

- `luca: GRANT select ON Film TO barbara, giovanna WITH GRANT OPTION;`

- `giovanna: GRANT select ON Film TO matteo;`

- `luca: GRANT ALL PRIVILEGES ON Video, Film TO elena WITH GRANT OPTION;`

- `elena: GRANT insert, select ON Film TO barbara;`

Privilegi delegabili e non delegabili

- Un utente potrebbe ricevere lo stesso privilegio sulla stessa relazione da utenti diversi
  - questa possibilità ha ripercussioni sull'operazione di revoca dei privilegi

- Dato che i privilegi possono essere concessi con grant option, i privilegi che ogni utente possiede possono essere classificati in due categorie:
  - privilegi delegabili
  - privilegi non delegabili

Comando revoke

- Un utente può revocare solo i privilegi da lui stesso concessi
- È possibile revocare più privilegi con un unico comando di REVOKE
- Un unico comando di REVOKE può essere utilizzato per revocare gli stessi privilegi sulla stessa relazione ad utenti diversi

COMANDO REVOKE

<lista privilegi> indica l'insieme di privilegi oggetto del comando di revoca

REVOKE [GRANT OPTION FOR] <lista privilegi>

ON <nome oggetto>

FROM <lista utenti>

{RESTRICT | CASCADE};

<nome oggetto> indica il nome dell'oggetto della base di dati su cui sono revocati i privilegi

La clausola opzionale GRANT OPTION FOR serve per revocare la sola grant option, mantenendo il diritto ad esercitare i privilegi oggetto del comando di revoca

Comando revoke

- La revoca può essere richiesta con o senza cascata:
  - revoca senza cascata (RESTRICT): l'esecuzione del comando non viene concessa se questo comporta la revoca di altri privilegi, oppure la cancellazione di oggetti dello schema (ad esempio viste, lo vedremo in seguito)
    - valore di default
  - revoca con cascata (CASCADE): revoca anche tutti i privilegi che erano stati propagati, generando una reazione a catena ed eventuali elementi della base di dati che erano stati creati sfruttando questi privilegi

Esempio

- Comandi di REVOKE eseguiti da Luca

```
REVOKE update, insert ON Video FROM elena;

REVOKE update ON Clienti FROM marco;

REVOKE select ON Film FROM giovanna;
```

ESEMPIO

Comandi di REVOKE eseguiti da Luca

```
REVOKE update, insert ON Video FROM elena
CASCADE;

REVOKE update ON Clienti FROM marco CASCADE;

REVOKE select ON Film FROM giovanna CASCADE;
```

Sommario

- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Gestione dell'accesso

- Le informazioni sull'insieme di autorizzazioni correntemente presenti nel sistema sono memorizzate in cataloghi di sistema
- L'organizzazione di queste tabelle non segue uno standard, ogni sistema utilizza il proprio schema
- Il gestore dell'accesso
  - Durante l'esecuzione di comandi di GRANT e REVOKE, modifica opportunamente il contenuto dei cataloghi (inserendo, aggiornando o modificando tuple)
  - Durante l'esecuzione di altri comandi SQL, legge il contenuto dei cataloghi per stabilire se il comando può essere eseguito

GESTIONE DELL'ACCESSO

DBA

GRANT
REVOKE

Cataloghi di sistema

Gestore degli accessi

Accesso totalmente o parzialmente accordato

Accesso negato

Altro comando SQL

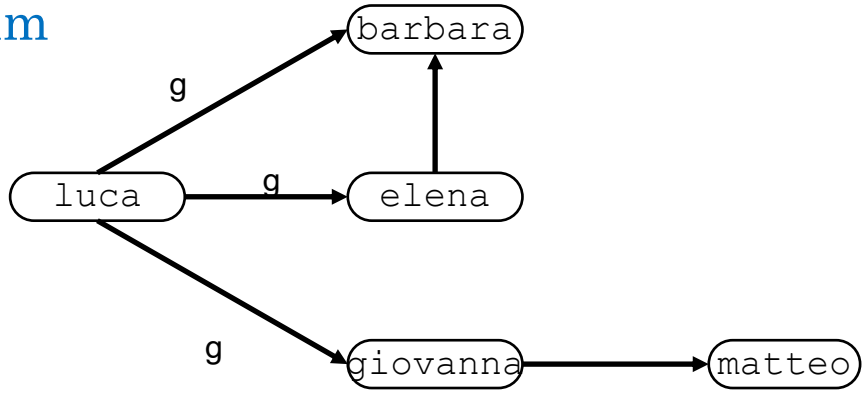
utente

Rappresentazione delle autorizzazioni

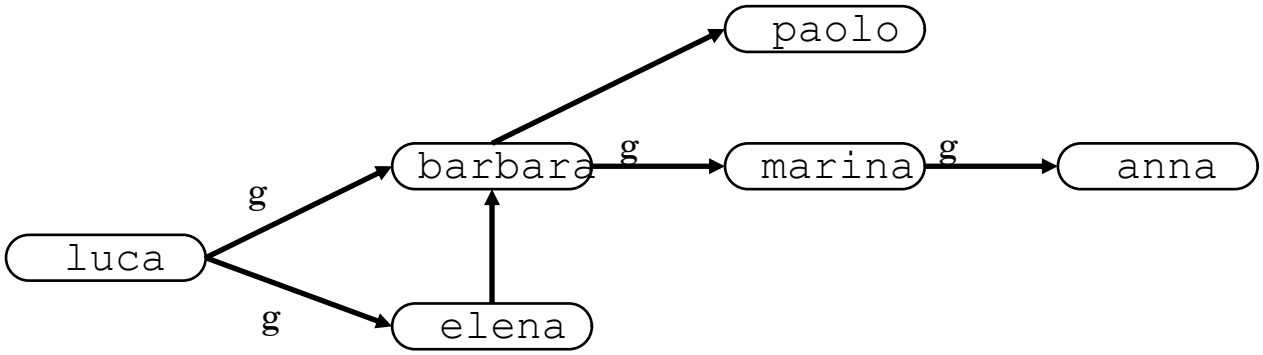
- Le informazioni contenute nei cataloghi possono essere rappresentate in astratto come un insieme di grafi, chiamati grafi delle autorizzazioni.
- Esiste un grafo per ogni privilegio su una certa tabella
- Un grafo delle interrogazioni per il privilegio p sulla tabella R contiene:
  - un nodo per ogni utente che possiede il privilegio p su R
  - un arco dal nodo u1 al nodo u2 se l'utente u1 ha concesso il privilegio p su R a u2
  - l'arco è etichettato con la lettera g se il privilegio è delegabile (concesso con GRANT OPTION)

Esempio

- Grafo delle autorizzazioni relativo al privilegio SELECT e alla relazione Film



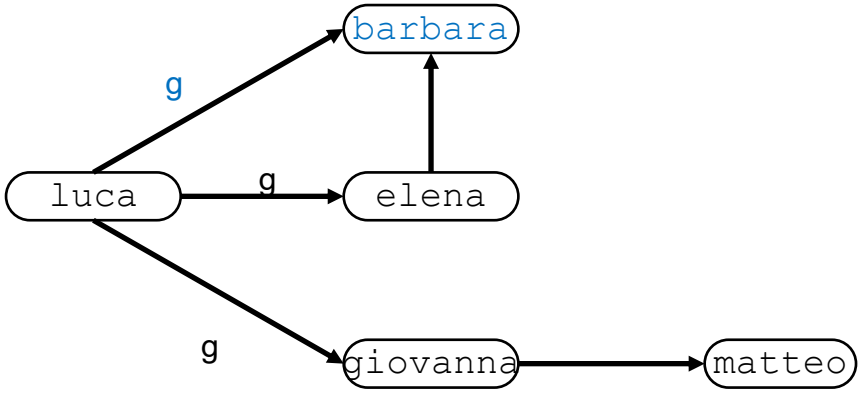
- Grafo delle autorizzazioni relativo al privilegio INSERT e alla relazione Film



Esecuzione dei comandi di GRANT

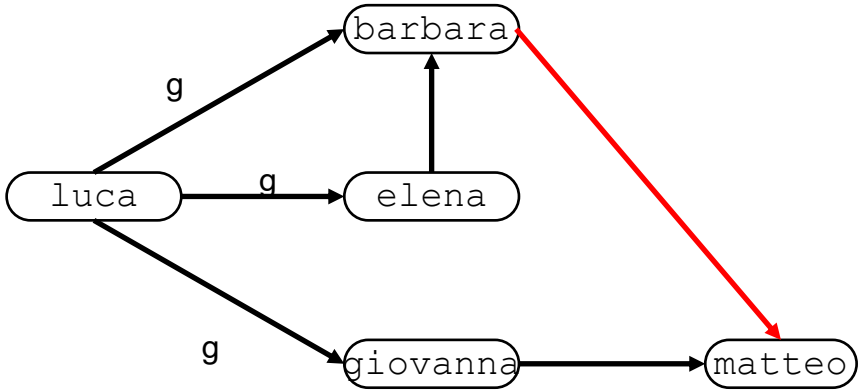
- Il gestore dell'accesso, durante l'esecuzione di un comando di GRANT:
  - verifica che l'utente che esegue il comando abbia i diritti per poterlo eseguire, leggendo i cataloghi (comando SELECT)
  - se l'utente può eseguire il comando di GRANT, i nuovi privilegi concessi vengono memorizzati nei cataloghi (comando INSERT), modificando di conseguenza i cataloghi
  - in alcuni casi il comando può essere eseguito solo parzialmente
    - solo alcune autorizzazioni tra quelle richieste vengono inserite

ESEMPIO

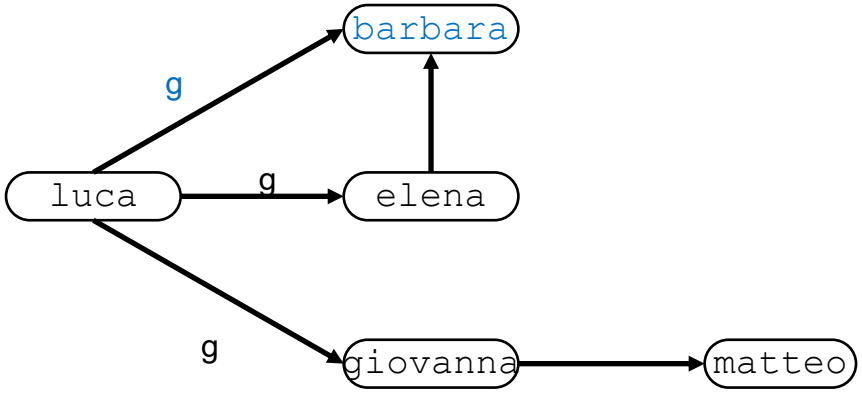



```
barbara:  GRANT select ON Film TO matteo;
```

Barbara ha acquisito il privilegio con grant option, quindi lo può delegare
Il nuovo grafo (che corrisponderà ad un nuovo contenuto delle tabelle dei catalogi è



ESEMPIO

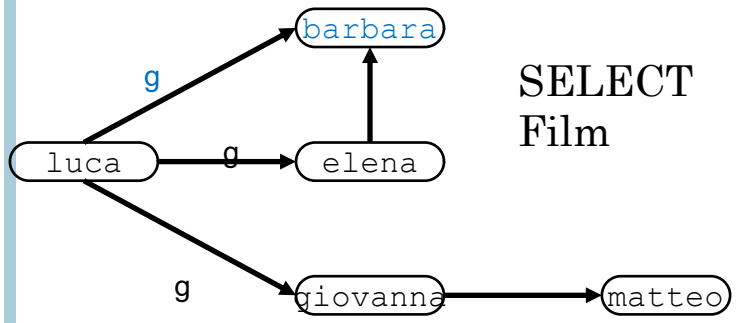



```
matteo:  GRANT select ON Film TO elena;
```

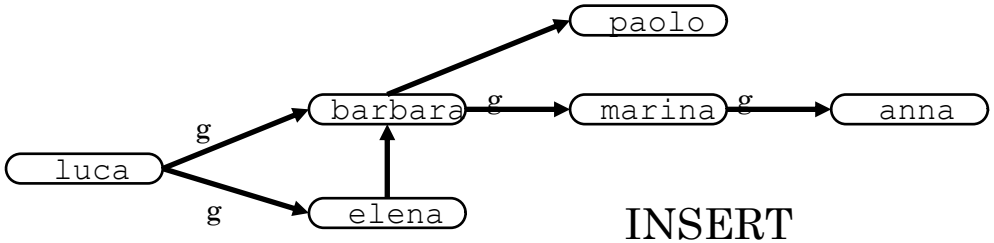
Matteo non ha acquisito il privilegio con grant option, quindi non lo può delegare

Il grafo non cambia

ESEMPIO



SELECT
Film

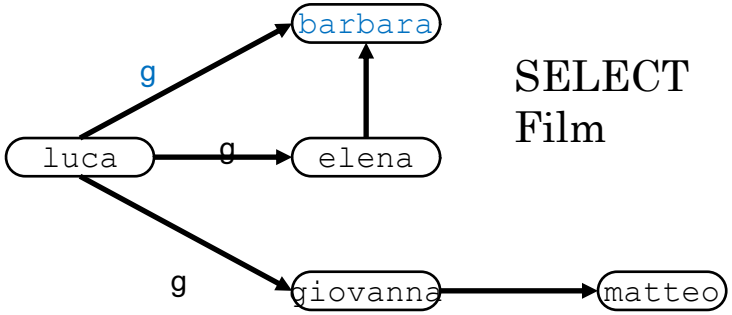


INSERT
Film

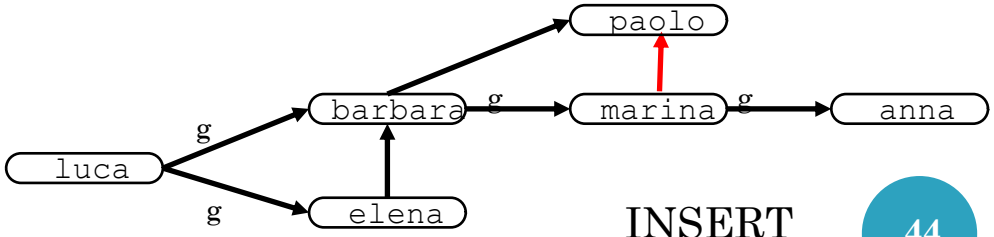
```
marina:  GRANT select, insert ON Film TO Paolo;
```

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

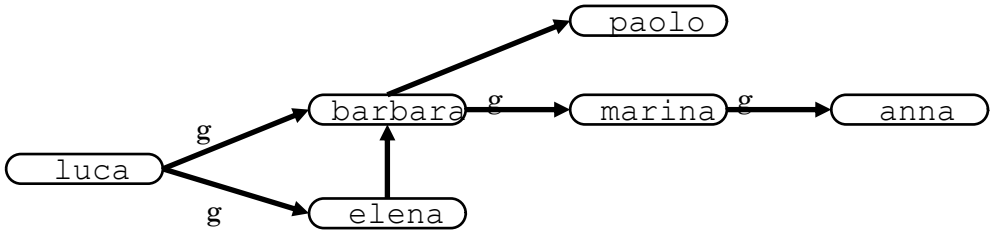


INSERT
Film

Esecuzione del comando REVOKE

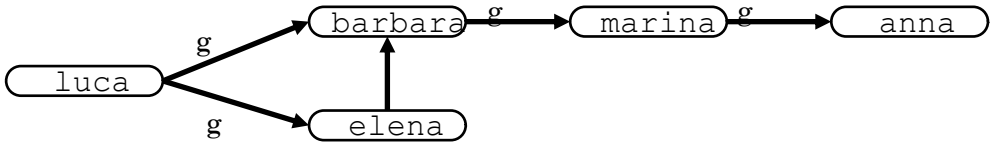
- Il gestore dell'accesso, durante l'esecuzione di un comando di REVOKE:
  - verifica che l'utente che esegue il comando abbia i diritti per poterlo eseguire, leggendo i cataloghi (comando SELECT)
  - se l'utente può eseguire il comando di REVOKE, le autorizzazioni revocate vengono cancellate dai cataloghi (comando DELETE o UPDATE)
  - in alcuni casi il comando può essere eseguito solo parzialmente
    - solo alcune autorizzazioni tra quelle richieste vengono modificate o cancellate
    - dipende anche anche dalla modalità di REVOKE (RESTRICT/CASCADE)

ESEMPIO

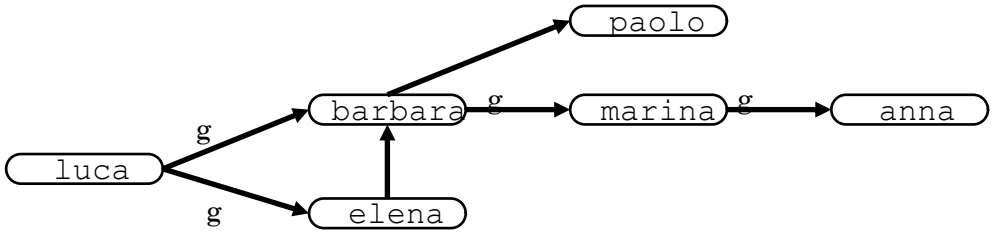



```
barbara:  REVOKE insert ON Clienti FROM paolo RESTRICT;
```

Poiché il privilegio di select era stato concesso a Paolo senza GRANT OPTION, Paolo non può averlo delegato ad altri e quindi l'arco tra Barbara e Paolo si può rimuovere

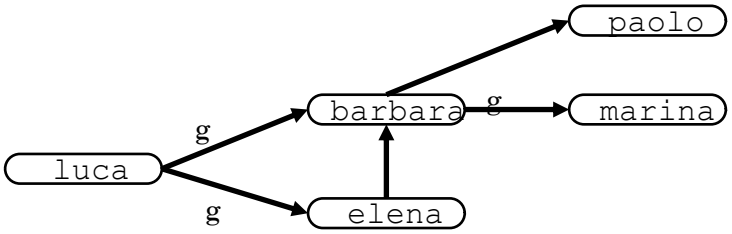


ESEMPIO

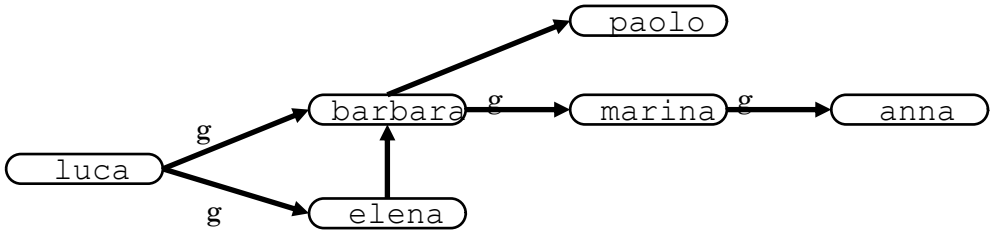



```
marina:  REVOKE insert ON Clienti FROM anna RESTRICT;
```

Questa volta il privilegio è stato concesso ad Anna da Marina con GRANT OPTION ma Anna non lo ha delegato ad altri, quindi anche in questo caso si può rimuovere l'arco

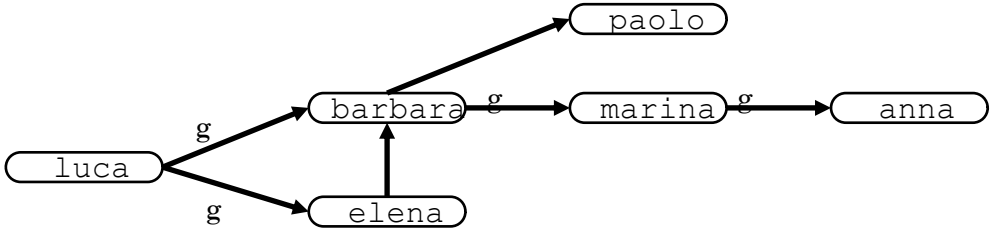


ESEMPIO

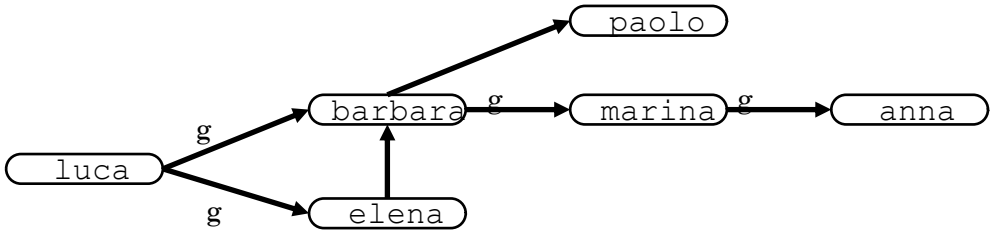



```
barbara:  REVOKE insert ON Clienti FROM marina RESTRICT;
```

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

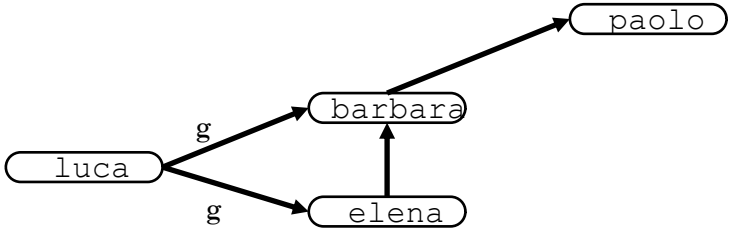


ESEMPIO




```
barbara:  REVOKE insert ON Clienti 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:

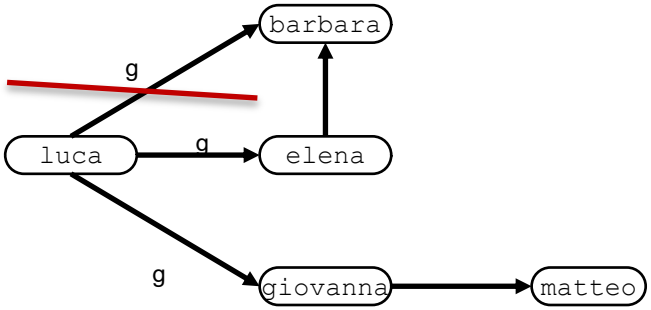


Esecuzione del comando REVOKE

- L'esecuzione del comando di revoca richiede attenzione

1. Un utente potrebbe avere acquisito il privilegio da parte di più utenti
   - In questo caso, l'utente potrà continuare ad esercitare il privilegio anche dopo l'operazione di revoca nel caso in cui lo abbia ricevuto da altre fonti indipendenti dal soggetto che effettua la revoca
2. Inoltre, come abbiamo già discusso, dato che i privilegi possono essere concessi con GRANT OPTION, la revoca di un privilegio potrebbe comportare ulteriori revoche (revoca ricorsiva con REVOKE CASCADE)

ESEMPIO – FONTI INDIPENDENTI

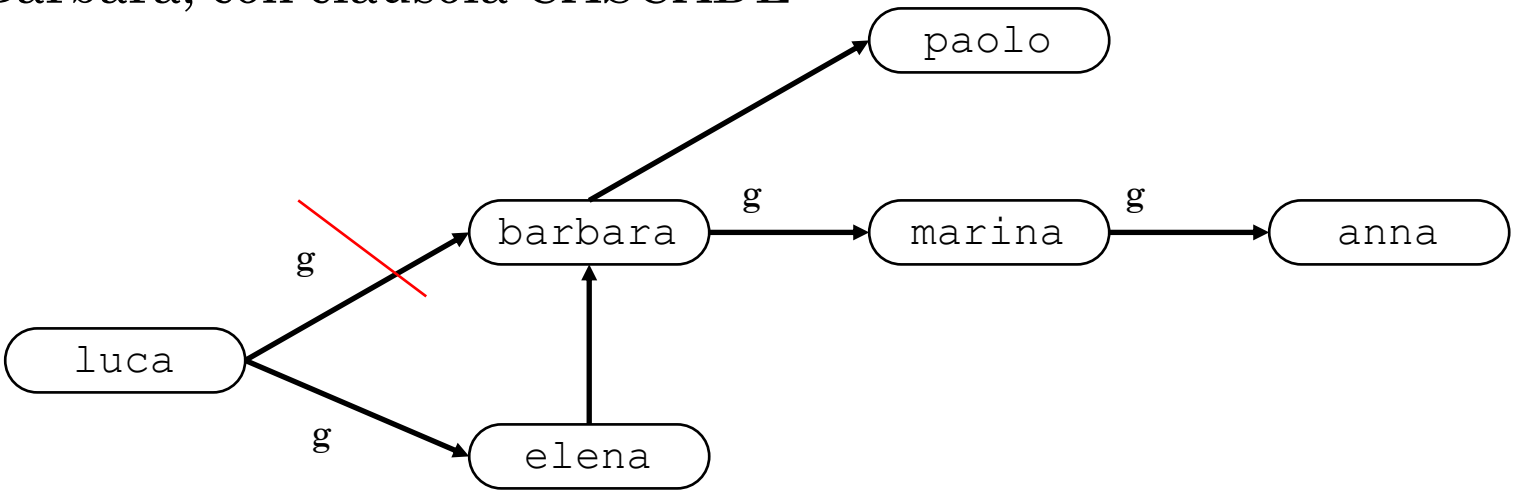


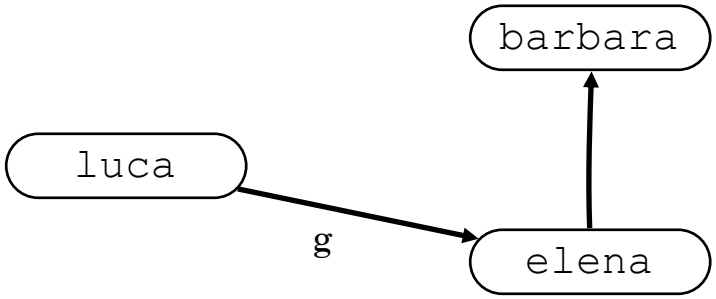

```
luca: REVOKE  select ON  Film  FROM  barbara;
```

- Dopo la revoca Barbara mantiene ancora tale privilegio, grazie all'autorizzazione concessale da Elena (arco entrante in Barbara ancora esistente dopo revoca)
- Barbara non potrà però più concedere a terzi il privilegio select su Film (l'autorizzazione acquisita da Elena non prevede grant option)

ESEMPIO – REVOCA IN CASCATA

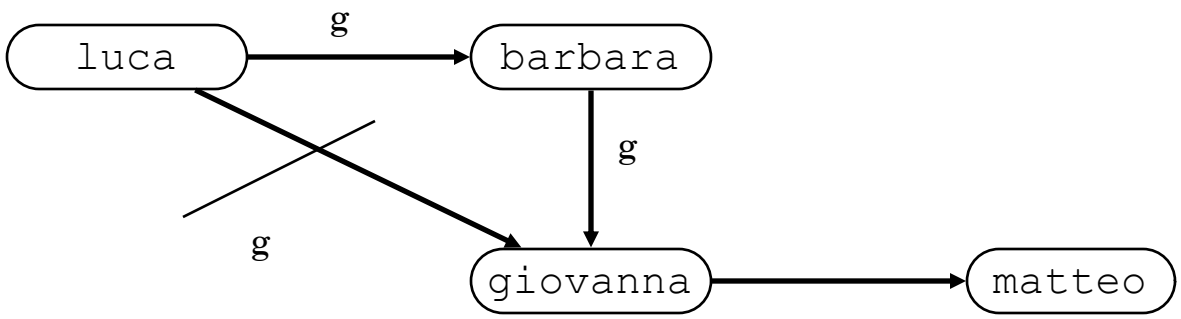
- Supponiamo che Luca revochi il privilegio select concesso a Barbara, con clausola CASCADE





Barbara perde grant option e quindi in cascata vengono cancellati tutti gli altri privilegi concessi da Barbara

Esempio



- Supponiamo che Luca revochi il privilegio SELECT concesso a Giovanna, con modalità CASCADE
- Questo non comporterebbe la revoca del privilegio concesso da Giovanna a Matteo, in quanto Giovanna continua ad avere il privilegio SELECT con grant option sulla relazione Film concessole da Barbara

Revoca in cascata

1. Si individua nel grafo l'arco corrispondente all'autorizzazione da revocare
2. Sia (u_1, l, u_2) l'arco considerato
   - u_1 utente corrispondente al nodo di partenza (utente che revoca privilegio)
   - l etichetta arco tra u_1 e u_2 (grant option)
   - u_2 utente corrispondente al nodo di arrivo (utente a cui il privilegio viene revocato)
3. Si cancella arco (u_1, l_{12}, u_2)
4. Se esistono altri archi entranti in u_2 con etichetta g (grant option) ➔ nessuna altra cancellazione è necessaria
5. Se non esistono altri archi entranti in u_2 con etichetta g
   - Si rimuovono tutti gli archi uscenti da u_2
   - Per ogni arco uscente (u_2,l_{23} u_3), si riparte dal passo (2)
   - (quindi ricorsivamente si valuta la cancellazione di altri archi)

Sommario

- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Controllo dell'accesso basato sui ruoli

- Nel modello di controllo dell'accesso di SQL, i soggetti possono rappresentare ruoli
  - RBAC - Role-based Access Control
- Un ruolo rappresenta una funzioni che gli utenti ricoprono all'interno della realtà organizzativa in cui operano
  - esempi di ruolo per il dominio della videoteca:
    cliente
    commesso
    direttoreVideoteca
- Gli utenti sono abilitati a ricoprire uno o più ruoli, in base alle mansioni che devono svolgere

Controllo dell'accesso basato sui ruoli

- I privilegi possono essere concessi ai singoli utenti ma anche ai ruoli
- Le autorizzazioni specificate per un ruolo sono quelle necessarie per esercitare le funzioni connesse al ruolo stesso
- Ogni utente che ricopre un ruolo acquisisce tutte le autorizzazioni ad esso connesse
- Vantaggi
  - controllo dell'accesso più flessibile
  - possibilità che un utente ricopra ruoli diversi in momenti diversi
  - semplificazione dell'attività di amministrazione
  - facilità nella definizione del profilo di nuovi utenti

CREAZIONE RUOLI

- Creazione di un ruolo:
  - CREATE ROLE <nome ruolo>;
- Eliminazione di un ruolo:
  - DROP ROLE <nome ruolo>;
- Un utente in momenti diversi può ricoprire ruoli diversi
- Associazione dinamica di un ruolo all'utente della sessione attiva
  - SET ROLE NomeRuolo

COMANDO GRANT ESTESO AI RUOLI

- Il comando GRANT può essere utilizzato anche per
  1. concedere privilegi non solo ad utenti ma anche a ruoli
  2. autorizzare un utente non solo all'esercizio di privilegi ma anche a ricoprire uno o più ruoli
  3. definire gerarchie tra i ruoli: autorizzare un ruolo a ricoprirne un altro

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

COMANDO GRANT ESTESO AI RUOLI

1. Concessione privilegi ai ruoli

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

la parola chiave PUBLIC consente di specificare tutti gli utenti/ruoli del sistema

<lista ruoli> indica l'insieme dei ruoli a cui vengono concessi i privilegi

Esempio

1. Concessione privilegi ai ruoli

CREATE ROLE direttoreVideoteca;

GRANT delete ON Clienti TO direttoreVideoteca
WITH GRANT OPTION;

COMANDO GRANT ESTESO AI RUOLI

- **2. Autorizzare un utente a ricoprire un ruolo**

GRANT <lista ruoli concessi>
TO {<lista utenti> | PUBLIC}
[WITH ADMIN OPTION];

<lista utenti> indica l'insieme degli utenti per cui vengono abilitati i ruoli concessi con il comando

<lista ruoli concessi> indica l'insieme dei ruoli concessi con il comando GRANT

la parola chiave PUBLIC consente di abilitare i ruoli specificati nel comando per tutti gli utenti/ruoli del sistema

La clausola opzionale WITH ADMIN OPTION è l'analogo per i ruoli della grant option:

- vengono ereditate tutte le autorizzazioni specificate per quel ruolo
- è anche possibile concedere a terzi la possibilità di ricoprire il ruolo

Esempio

2. Autorizzare un utente a ricoprire un ruolo

GRANT direttoreVideoteca TO Luca WITH ADMIN OPTION;

COMANDO GRANT ESTESO AI RUOLI – GERARCHIE TRA RUOLI

- SQL permette anche di definire i ruoli gerarchicamente

- Una gerarchia sui ruoli definisce un ordinamento parziale $\geq$ tra di essi:
  - induce un'ereditarietà dei privilegi tra ruoli nella gerarchia
  - stabilisce una relazione tra gli utenti abilitati a ricoprire i vari ruoli

Gerarchicomando GRANT esteso ai ruoli – gerarchie tra ruoli

- Dati due ruoli r1 ed r2, $r1 \geq r2$ implica che
  - r1 eredita tutti i privilegi assegnati ad r2
  - quindi tutti gli utenti associati ad r1 sono anche utenti associati ad r2
- Il principio di ereditarietà è motivato dal fatto che un ruolo (r1) dovrebbe poter effettuare sugli oggetti del sistema tutte le operazioni che possono essere effettuate da ruoli corrispondenti a funzioni più in basso (r2) nella gerarchia
- Se $r1 \geq r2$, possiamo anche dire che gli utenti con ruolo r1 sono abilitati a rivestire anche il ruolo r2

Esempio

- Ruoli:
  `direttoreVideoteca`
  `commesso`

- `direttoreVideoteca` ≥ `commesso`
  - I privilegi attribuiti al `commesso` vengono ereditati dal `direttoreVideoteca` (il direttore può svolgere tutte le attività che svolge un commesso ma il viceversa non è vero)
  - agli utenti con ruolo `direttoreVideoteca` viene quindi implicitamente attribuito anche il ruolo `commesso`

COMANDO GRANT ESTESO AI RUOLI

- **3. Gerarchie tra ruoli**

<lista ruoli> indica l'insieme dei ruoli per cui vengono abilitati i ruoli concessi con il comando

GRANT <lista ruoli concessi>
TO {<lista ruoli> | PUBLIC}
[WITH ADMIN OPTION];

la parola chiave PUBLIC consente di abilitare i ruoli specificati nel comando per tutti gli utenti/ruoli del sistema

<lista ruoli concessi> indica l'insieme dei ruoli concessi con il comando GRANT

Per ogni ruolo r_2 in <lista ruoli concessi> e ogni ruolo r_1 in <lista ruoli> si vuole quindi imporre $r_1 \geq r_2$

COMANDO GRANT ESTESO AI RUOLI

- La clausola opzionale WITH ADMIN OPTION è l'analogo per i ruoli della grant option
- Quando si è abilitati a ricoprire un ruolo con admin option non solo vengono ereditate tutte le autorizzazioni specificate per quel ruolo, ma è anche possibile concedere ad altri utenti o ruoli la possibilità di ricoprire il ruolo

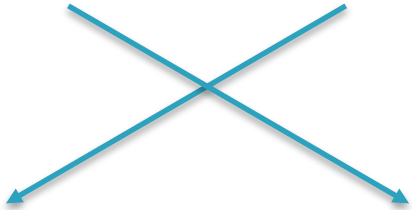
ESEMPIO

`direttoreVideoteca ≥ commesso`

3. Gerarchie tra ruoli

CREATE ROLE direttoreVideoteca;
CREATE ROLE commesso;

GRANT commesso to direttoreVideoteca;



Quindi direttoreVideoteca ≥ commesso

COMANDO REVOKE ESTESO AI RUOLI

- Il comando REVOKE può essere utilizzato:
  1. per la revoca di privilegi ai ruoli
  2. per la revoca di ruoli (e per eliminare relazioni gerarchiche)

COMANDO REVOKE ESTESO AI RUOLI

- **1. Revoca di privilegi a ruoli**

REVOKE [GRANT OPTION FOR] <lista privilegi>

ON <nome oggetto>

FROM {<lista utenti>|<lista ruoli>}

{RESTRICT | CASCADE};

<lista ruoli> indica l'insieme dei ruoli a cui sono revocati i privilegi

Esempio

1. Revoca di privilegi a ruoli

REVOKE delete ON Clienti FROM direttoreVideoteca;

COMANDO REVOKE ESTESO AI RUOLI

- **2. Revoca di ruoli a utenti o ruoli**

<lista ruoli revocati> indica l'insieme di ruoli revocati con il comando

REVOKE [ADMIN OPTION FOR] <lista ruoli revocati>

FROM {<lista utenti>|<lista ruoli>}

{RESTRICT | CASCADE};

<lista utenti> indica l'insieme degli utenti a cui viene revocata l'abilitazione a ricoprire i ruoli oggetto della revoca

la clausola opzionale ADMIN OPTION FOR serve per revocare la sola admin option, mantenendo il diritto a ricoprire i ruoli oggetto del comando di revoca

<lista ruoli> indica l'insieme dei ruoli a cui viene revocata l'abilitazione a ricoprire i ruoli oggetto della revoca (quindi si toglie un arco dalla gerarchia)

Esempio

- **2. Revoca di ruoli a utenti o ruoli**

- REVOKE ADMIN OPTION FOR direttoreVideoteca
  FROM Luca;

- REVOKE direttoreVideoteca FROM Luca;

- REVOKE direttoreVideoteca FROM amministratoreVideoteca;

Vantaggi uso di ruoli

- 10 utenti
- A ciascun utente deve essere concesso il privilegio di lettura su tabella T, di proprietà di utente U
- Quante nuove autorizzazioni sono necessarie?
- Quanti comandi di GRANT?

- Per ciascun utente, U esegue 1 grant per concedere il privilegio di SELECT su tabella T
- Totale 10 nuove autorizzazioni a utenti
- Si può utilizzare un solo comando di GRANT

Vantaggi uso di ruoli

- 10 utenti
- Tutti rivestono lo stesso ruolo r
- A ciascun utente deve essere concesso il privilegio di lettura su tabella R, di proprietà di utente U
- Quante nuove autorizzazioni sono necessarie?
- Quanti comandi di GRANT?

- U esegue 1 grant per concedere il privilegio di SELECT su tabella T al ruolo r
- Automaticamente tutti gli utenti acquisiscono il privilegio
- Totale 1 grant
- Totale 1 nuova autorizzazione a ruolo

Vantaggio gerarchie

- **tempo t_0**: nessuno (tranne l'amministratore) può cancellare tuple dalla tabella Video
- **tempo t_1**: la politica aziendale cambia e sia il commesso sia il direttore della videoteca possono cancellare tuple dalla tabella Video

- Senza gerarchie, sono necessarie due autorizzazioni (concesse eventualmente nel contesto di un singolo comando)
- Se `direttoreVideoteca` ≥ `commesso`, ne basta una sola (a chi?)

Sommario

- Introduzione
- Il modello per il controllo degli accessi
- Controllo degli accessi in SQL
  - Principi generali
  - Comandi
  - Gestione dell'accesso
  - I ruoli
  - Controllo dell'accesso basato sul contenuto

Autorizzazioni su viste

- Le viste sono un importante meccanismo attraverso cui è possibile fornire forme più sofisticate di controllo dell'accesso
- La sintassi del comando GRANT non consente di concedere privilegi solo su alcune tuple di una tabella
- È però sufficiente definire una vista che selezioni le tuple di interesse e autorizzare l'accesso alla vista invece che alla relazione di base

Autorizzazioni su viste

- Le viste permettono di concedere anche privilegi statistici
  - ad esempio, un utente potrebbe non essere autorizzato a vedere i titoli dei film noleggiati da ciascun cliente, ma solo il numero di noleggi effettuati
- Tramite le viste, è possibile soddisfare questo requisito di sicurezza:
  - basta definire una vista che computa il numero di noleggi effettuati da ogni cliente e concedere all'utente l'accesso alla vista invece che alle relazioni di base
- Le viste consentono di realizzare il così detto controllo dell'accesso in base al contenuto
- Ovvero, permettono di autorizzare l'accesso solo a specifiche tuple di una relazione, sulla base dei valori dei loro attributi

Autorizzazioni su viste

- Chi può creare una vista?
  - Un utente può creare una vista solo se ha il privilegio select sulle relazioni/viste su cui è definita
- L'owner della vista quali privilegi può esercitare sulla vista?
  - L'insieme dei privilegi esercitabili dipende da
    - (i) le autorizzazioni che l'utente possiede sulle relazioni/viste su cui la vista è definita
    - (ii) la semantica della vista, ovvero la sua definizione in termini delle relazioni o viste componenti
- I privilegi esercitabili sulla vista sono quelli contenuti nell'intersezione di (i) e (ii)

Esempio

- Barbara ha il privilegio SELECT su Film, con GRANT OPTION

- Barbara esegue il comando:

```
CREATE VIEW Commedie AS
SELECT * FROM Film
WHERE genere = 'commedia';
```

- L'esecuzione del comando di creazione viene autorizzata

Autorizzazioni su viste

- Se la vista è definita su una singola relazione (o vista)
  - i privilegi che l'utente che crea la vista ha su di essa sono gli stessi che ha sulla relazione o vista componente
  - i privilegi sulla vista saranno delegabili o meno, a seconda di come sono definiti per la relazione o vista componente

ESEMPIO

- Barbara ha solo il privilegio SELECT su Film, con GRANT OPTION
- Barbara esegue il comando:

```
CREATE VIEW Commedie AS
SELECT * FROM Film
WHERE genere = 'commedia';
```

- Privilegi esercitabili da Barbara sulla vista Commedie
  - (i) Barbara possiede su Film il privilegio di SELECT con GRANT OPTION
  - (ii) La definizione della vista ammette l'esecuzione di operazioni di SELECT, INSERT, DELETE, UPDATE (vista definita su singola relazione), quindi Barbara, come creatore della vista, potrebbe esercitare tutti questi privilegi (con GRANT OPTION)
  - L'intersezione di (i) e (ii) coincide sempre con (i) per viste definite su singole tabelle/viste quindi contiene solo SELECT
  - Barbara può concedere a terzi tale privilegio, in quanto possiede GRANT OPTION

Autorizzazioni su viste

- Se la vista è definita su più relazioni (o viste)
  - i privilegi che l'utente che crea la vista ha su di essa sono ottenuti intersecando i privilegi che l'utente ha su tutte le relazioni (o viste) componenti con le operazioni eseguibili sulla vista
  - un privilegio sulla vista è delegabile solo se il creatore della vista ha il diritto di delegare tale privilegio su tutte le relazioni o viste componenti

Autorizzazioni su viste

- Se la vista è definita su più relazioni (o viste)
  - SQL pone alcune restrizioni sulle operazioni che possono essere effettuate su una vista
    - ad esempio, una vista che computa delle statistiche non può essere aggiornata
  - queste restrizioni si riflettono anche sulle autorizzazioni che un utente che crea una vista ha sulla vista stessa

Esempio

- Elena ha i privilegi SELECT, INSERT e UPDATE sulla relazione Noleggi, con GRANT OPTION
- Elena esegue il comando:

```
CREATE VIEW NumNoleggi AS
SELECT codCli, COUNT(*) AS NumNol
FROM Noleggi
GROUP BY codCli;
```

- L'esecuzione del comando viene autorizzata e la vista creata
- Privilegi esercitabili da Elena sulla vista
  - (i) SELECT, INSERT, UPDATE
  - (ii) SELECT
  - L'intersezione contiene solo SELECT, quindi questo è l'unico privilegio di Elena sulla vista
  - Elena può concedere a terzi tale privilegio, in quanto possiede GRANT OPTION

Autorizzazioni su viste

- La concessione di privilegi su una vista è molto simile a quella su relazioni di base:
  - i privilegi che un utente può concedere ad altri su una vista sono quelli che possiede con grant option
- Le operazioni di revoca sono, invece, più complicate, in quanto è necessario stabilire cosa succede ad una vista se un privilegio SELECT su una delle relazioni o viste componenti è revocato
  - in accordo alla semantica della revoca ricorsiva, si cancella la vista se il privilegio revocato era l'unico utile per la sua definizione

ESEMPIO

```
barbara>
CREATE VIEW Commedie  AS
SELECT * FROM  Film
WHERE  genere = 'commedia';
```

```
elena>
CREATE VIEW  NumNoleggi  AS
SELECT  codCli,
        COUNT(*) AS NumNol
FROM  Noleggi
GROUP BY  codCli;
```

- Se Luca revoca ad Elena (Barbara) il privilegio SELECT sulla relazione Noleggi e Elena (Barbara) non ha acquisito da altri utenti lo stesso privilegio, la vista NumNoleggi (Commedie) viene a sua volta cancellata

- In caso contrario la vista viene mantenuta