Sicurezza delle basi di dati

corso di basi di dati e laboratorio

Prof. Alfio Ferrara

Anno Accademico 2020/2021

Indice

1 Politiche e modelli

1.1 Introduzione

Dispensa

Dispensa su Sicurezza delle basi di dati (scaricabile dal sito) tratta da: Castano, Fugini, Martella, Samarati, *Database Security*, Addison-Wesley, 1995 (Cap. 1, Cap. 2 (§§ 2.1,2.2,2.6,2.7), Cap. 4 (§ 4.2.2)

Obiettivi

Segretezza: protezione delle informazioni da letture non autorizzate

- Integrità: protezione dei dati da modifiche o cancellazioni non autorizzate
- Disponibilità: garanzia che non si verifichino casi in cui ad utenti legittimi venga negato l'accesso ai dati

Importanza assegnata a tali obiettivi varia a seconda del sistema considerato

Tecniche

- Autenticazione: meccanismi per verificare l'identità dell'utente che si connette al sistema
- Controllo dell'accesso: meccanismi che, per ogni richiesta di accesso ai dati, verificano che l'utente sia autorizzato a compiere l'accesso
- Crittografia: meccanismi che consentono di cifrare i dati in modo che possano essere decifrati solo da utenti autorizzati

Noi ci concentreremo sul controllo dell'accesso (responsabilità del DBMS).

Controllo dell'accesso

- Regola le operazioni che si possono compiere sulle informazioni e le risorse in una base di dati
- Lo scopo è limitare e controllare le operazioni che gli utenti effettuano, prevenendo azioni accidentali o deliberate che potrebbero compromettere l'integrità e la segretezza dei dati
- Le risorse sono costituite dai dati, memorizzati in oggetti a cui si vuole garantire protezione
- I soggetti sono agenti (utenti o programmi in esecuzione) che richiedono di poter esercitare privilegi (come lettura, scrittura o esecuzione) sui dati

Controllo dell'accesso

- Politiche di sicurezza: norme e principi che esprimono le scelte di fondo dell'organizzazione relativamente alla sicurezza dei propri dati
- Sono implementate mediante traduzione in un insieme di regole di autorizzazione che stabiliscono le operazioni ed i diritti che gli utenti possono esercitare sui vari oggetti del sistema
- Il **Reference Monitor** è un meccanismo di controllo che ha il compito di stabilire se l'utente può essere autorizzato (totalmente o parzialmente) a compiere l'accesso

1.2 Politiche di sicurezza

Politiche
La politica di sicurezza adottata dipende principalmente da fattori organizzativi, quali l'ambiente di installazione, le esigenze degli utenti, i regolamenti dell'organizzazione, o i vincoli di natura legale. Due classi fondamentali:

- Politiche per l'amministrazione della sicurezza
- Politiche per il controllo dell'accesso ai dati

Politiche per l'amministrazione
Stabiliscono chi concede e revoca i diritti di accesso

- Centralizzata: un unico autorizzatore, detto DBA, controlla l'intera base di dati.
- Decentralizzata: più autorizzatori responsabili del controllo di porzioni diverse della base di dati.

Ownership: l'utente che crea un oggetto (il proprietario) gestisce le autorizzazioni sull'oggetto

Controllo degli accessi
Le politiche per il controllo dell'accesso stabiliscono se e come i soggetti possono accedere a quali dati contenuti nel sistema, e se e come possono venire trasmessi i diritti di accesso

- Need-To-Know (minimo privilegio) Molto restrittiva, permette ad ogni utente l'accesso solo ai dati strettamente necessari per eseguire le proprie attività
- Maximized Sharing (massima condivisione) Consente agli utenti il massimo accesso alle informazioni nella base di dati, mantenendo comunque informazioni riservate

Caratteristiche

- NEED TO KNOW offre ottime garanzie di sicurezza ed è adatta a basi di dati con forti esigenze di protezione può portare ad un sistema eccessivamente protetto, negando accessi che non comprometterebbero la sicurezza del sistema
- MAXIMIZED SHARING soddisfa il massimo numero possibile di richieste di accesso viene utilizzata in ambienti in cui esiste una certa fiducia tra gli utenti ed in cui non è sentita una forte esigenza di protezione

Sistemi

- Sistema aperto l'accesso è permesso a meno che non sia esplicitamente negato le regole di autorizzazione indicano per ogni soggetto i diritti che egli non può esercitare sugli oggetti del sistema questi diritti sono i soli che gli saranno negati
- Sistema chiuso l'acceso è permesso solo se esplicitamente autorizzato le regole di autorizzazione indicano per ogni soggetto i diritti che egli può esercitare sugli oggetti del sistema questi diritti sono i soli che verranno accordati dal meccanismo di controllo
- Sistema aperto e chiuso: Un sistema chiuso implementa la politica del minimo privilegio, un sistema aperto implementa la politica della massima condivisione. Un sistema chiuso offre maggiori garanzie di sicurezza: una regola inavvertitamente cancellata o non inserita restringe ulteriormente l'accesso, mentre un sistema aperto permette accessi non autorizzati. La maggior parte delle basi di dati oggi esistenti sono sistemi chiusi.

Granularità

Granularità a cui il controllo dell'accesso deve essere effettuato
Requisito minimo: possibilità di specificare nelle regole di autorizzazione sugli oggetti a cui l'utente può accedere $\rightarrow$ nelle BD relazionali una relazione o attributi di relazione

Tipologie di controllo

- Controllo dipendente dal nome l'accesso è basato sul nome dell'oggetto
- Controllo dipendente dal contenuto l'accesso è subordinato al valore di uno o più attributi dell'oggetto (es., l'utente X può accedere ai dati degli impiegati il cui stipendio non supera una certa soglia).
- Controllo dipendente dal contesto l'accesso è subordinato al valore di variabili di sistema (es., data, tempo); es., i dati sugli impiegati possono essere acceduti solo in orario di lavoro.
- Controllo dipendente dalla storia degli accessi l'accesso è subordinato alla storia degli accessi eseguiti precedentemente (es., un utente può accedere ad un determinato dato solo se il numero di accessi da lui compiuti su quel dato in un determinato intervallo di tempo non supera una certa soglia)

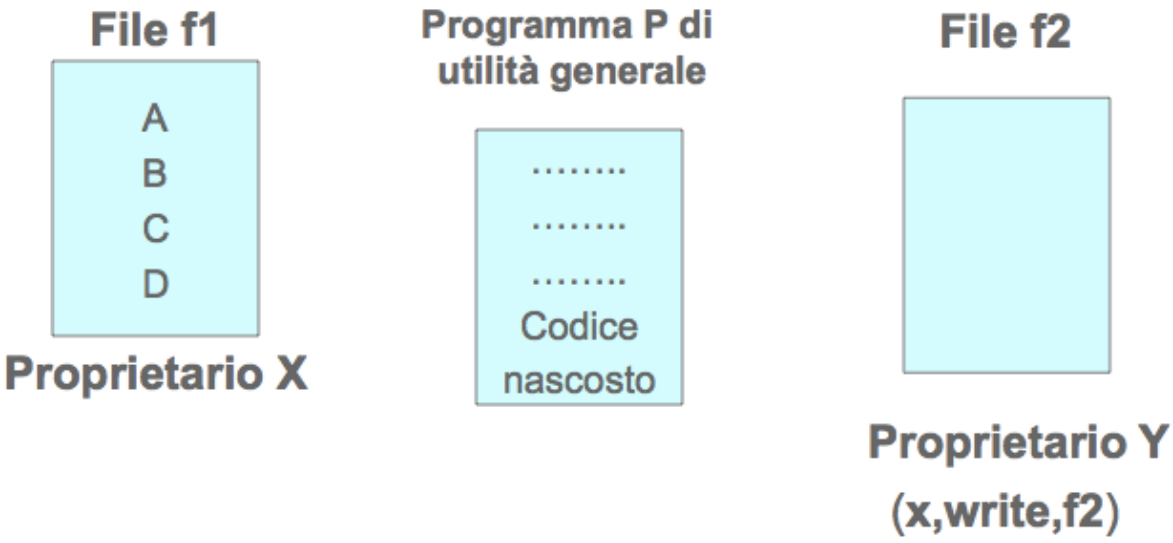
Politiche discrezionali

- Richiedono che vengano specificati i diritti che ogni soggetto possiede sugli oggetti del sistema, sottoforma di regole di autorizzazione. Il meccanismo di controllo esamina le richieste di accesso accordando solo quelle che sono autorizzate da una regola. Gli utenti possono a loro discrezione concedere o revocare i diritti di accesso sugli oggetti.
- Vantaggio: sono estremamente flessibili e adatte a numerosi contesti applicativi
- Svantaggio: non impongono restrizioni sull'uso che viene fatto del dato una volta acceduto ovvero non forniscono alcun controllo sul flusso di informazioni nel sistema Si ha un flusso tra un oggetto X e un oggetto Y quando si effettua una lettura del valore di X e una scrittura del valore in Y.

Politiche mandatorie

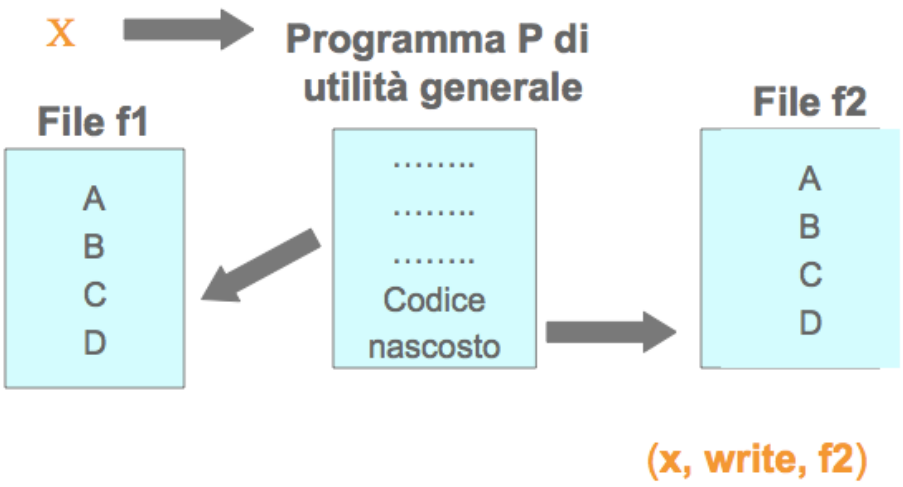
- Per basi di dati governative le politiche discrezionali non sono sufficienti
- Informazioni vitali, diversi livelli di sensitività, i controlli sul flusso di dati sono essenziali
- Attacchi sofisticati da parte di utenti determinati (es. Cavallo di Troia)

Cavallo di Troia (1)



Y concede ad X l'autorizzazione a scrivere su f2 (Y è proprietario di f2 e lo può fare)

Cavallo di Troia (2)



X esegue P (contenente del codice nascosto che gli fa leggere f1, file che Y non può leggere) ed avendo ricevuto da Y il diritto di scrivere su f2, trasferisce le informazioni contenute in f1 su f2 (file che Y può leggere).

Politiche mandatorie

Regolano l'accesso ai dati mediante la definizione di classi di sicurezza per i soggetti e gli oggetti del sistema

- Le classi di sicurezza sono ordinate parzialmente da una relazione d'ordine
- La classe di sicurezza assegnata ad un oggetto rappresenta il livello di sensitività dell'oggetto: maggiore è la classe assegnata ad un oggetto, più ingente sarà il danno derivante dal rilascio delle informazioni in esso contenute a soggetti non autorizzati
- La classe di sicurezza assegnata ad un soggetto è una misura del grado di fiducia che si ha nel fatto che tale soggetto non commetta violazioni

Controllo dell'accesso

Il controllo dell'accesso è regolato da una serie di assiomi di sicurezza che stabiliscono le relazioni (in base al modo di accesso considerato) che devono essere verificate fra la classe di un soggetto e quella di un oggetto affinché al primo sia concesso di esercitare un modo di accesso sul secondo Queste politiche sono applicate in ambienti, quali quello militare, dove la quantità di informazioni da proteggere è elevata, ci sono forti esigenze di protezione ed è possibile classificare rigidamente gli elementi del sistema I sistemi che adottano una politica mandatoria sono spesso indicati come sistemi multilivello

Politiche mandatorie

Possono essere classificate anche come politiche per il controllo del flusso, poiché evitano che le informazioni una volta accedute vengano trasferite verso oggetti con classificazione inferiore e quindi più accessibili (vedere esempio Cavallo di Troia) C'è un controllo completo sul sistema di autorizzazione La flessibilità è però ridotta e la circolazione di informazioni tra gli utenti è più difficile Le politiche mandatorie e discrezionali non sono mutuamente esclusive, possono cioè essere applicate insieme La politica mandatoria non controlla più le richieste di accesso ma le autorizzazioni che vengono assegnate ad un soggetto, mentre alla politica discrezionale è affidato il compito di controllare le richieste di accesso.

2 Modelli per il controllo dell'accesso

2.1 System R

Introduzione

- Il modello implementa una politica di tipo discrezionale e supporta il controllo dell'accesso in base sia al nome che al contenuto.
- Il sistema è un sistema chiuso: un accesso è concesso solo se esiste una esplicita regola che lo autorizza.
- L'amministrazione dei privilegi è decentralizzata mediante ownership: quando un utente crea una relazione, riceve automaticamente tutti i diritti di accesso su di essa ed anche la possibilità di delegare ad altri tali privilegi.

GRANT

```
GRANT Lista Privilegi | ALL [PRIVILEGES] ON Lista Relazioni | Lista Viste
TO Lista Utenti | PUBLIC [WITH GRANT OPTION]
```

Opzioni del GRANT

- Gli oggetti di protezione sono relazioni e viste
- I privilegi previsti sono: `INSERT, SELECT, UPDATE`
- Solo il proprietario può cancellare un oggetto

- Le parole chiave `ALL` (o `ALL PRIVILEGES`) consentono di concedere con un solo comando tutti i privilegi su una determinata relazione. Non possono essere utilizzate su viste.
- Con un unico comando di `GRANT` si possono concedere più privilegi su una stessa relazione e concedere privilegi sulla stessa relazioni a più utenti (in entrambi i casi l'ordine è irrilevante).
- Un comando di `GRANT` con soggetto `PUBLIC` è equivalente ad una concessione di privilegi a tutti gli utenti.

GRANT OPTION

- La delega dei privilegi avviene mediante la 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 concedere un privilegio su una determinata relazione solo se è il proprietario della relazione, o se ha ricevuto tale privilegio da altri con grant option.
- Se la clausola `WITH GRANT OPTION` non è specificata l'utente che riceve i privilegi non può concederli ad altri utenti.

I privilegi che ogni utente possiede sono divisi in:

- delegabili: privilegi concessi con grant option
- non delegabili: concessi senza grant option

Esempio (1)
`GRANT UPDATE(Stipendio, PremioP) ON Impiegato TO Rossi;` Rossi può modificare gli attributi Stipendio e PremioP delle tuple della relazione Impiegato
`GRANT SELECT, INSERT ON Impiegato TO Verdi, Gialli;` Verdi e Gialli possono selezionare ed inserire tuple nella relazione Impiegato
`GRANT ALL PRIVILEGES ON Impiegato TO Neri WITH GRANT OPTION;` Neri ha tutti i privilegi sulla relazione Impiegato e può delegare ad altri tali privilegi

Esempio (2)

- Bianchi: `GRANT SELECT, INSERT ON Impiegato TO Verdi WITH GRANT OPTION;`
- Bianchi: `GRANT SELECT ON Impiegato TO Rossi WITH GRANT OPTION;`

- Verdi: `GRANT SELECT, INSERT ON Impiegato TO Rossi;`

Rossi ha il privilegio di select (ricevuto sia da Bianchi che da Verdi) e insert (ricevuto da Verdi) sulla relazione Impiegato. Rossi può concedere ad altri utenti il privilegio di select (in quanto lo ha ricevuto da Bianchi con grant option), ma non quello di insert.

Implementazione

sysauth e **syscolauth**: Le regole di autorizzazione specificate dagli utenti sono memorizzate in due cataloghi di sistema di nome sysauth e syscolauth, implementati come relazioni. Una tupla di sysauth ha i seguenti attributi:

- id_utente: id dell'utente a cui sono concessi i privilegi;
- nome: nome della relazione su cui sono concessi i privilegi;
- creatore: utente che ha creato la relazione;
- tipo $\in \{R, V\}$: indica se l'oggetto è una relazione (tipo ='R') o una vista (tipo ='V');
- P $\in \{Y, N\}$: indica se l'oggetto ha (Y) o meno (N) il privilegio sulla relazione.

Sysauth contiene un attributo per ciascuno dei privilegi

- update $\in \{ALL, SOME, N\}$: indica se il soggetto ha il privilegio di update su tutte (ALL) alcune (SOME), o nessuna (N) colonna della relazione
- grantopt $\in \{Y, N\}$: indica se i privilegi sono delegabili (Y) o meno (N)

Esempio

id_utente	nome	creator	T	P	I	S	U	GO
bianchi	impiegato	bianchi	R	Y	Y	Y	all	Y
verdi	impiegato	bianchi	R	N	Y	Y	N	Y
rossi	impiegato	bianchi	R	N	N	Y	N	Y
rossi	impiegato	bianchi	R	N	Y	Y	N	N

SYSCOLAUTH

Le colonne su cui il privilegio di update può essere esercitato sono contenute nel catalogo SYSCOLAUTH SYSCOLAUTH contiene una tupla (id_utente, nome, colonna, grantopt) per ogni colonna della relazione su cui l'utente identificato da id_utente può esercitare il privilegio di update.

id_utente	nome	colonna	GO
bianchi	impiegato	imp	Y
bianchi	impiegato	mansione	Y
...	...	...	...

Uso del catalogo

- Quando un utente u esegue un comando di `GRANT`, il meccanismo di controllo accede ai cataloghi `SYSAUTH` e `SYSCOLAUTH` per determinare se u ha il diritto di delegare i privilegi specificati nel comando.
- L'insieme dei privilegi delegabili che l'utente u possiede è intersecato con l'insieme dei privilegi specificati nel comando di `GRANT`.
- Se l'intersezione è vuota, il comando non viene eseguito.
- Se l'intersezione coincide con i privilegi specificati nel comando, vengono concessi tutti i privilegi specificati.
- Altrimenti il comando viene eseguito parzialmente, cioè solo i privilegi contenuti dell'intersezione vengono accordati

Esempio

- Bianchi: `GRANT SELECT, INSERT ON Impiegato TO Gialli WITH GRANT OPTION;`
- Il comando viene eseguito (Bianchi è il proprietario della relazioni Impiegato)
- Verdi: `GRANT UPDATE ON Impiegato TO Gialli WITH GRANT OPTION;`
- Il comando non viene eseguito (Verdi non possiede il privilegio di update sulla relazione Impiegato)
- Rossi: `GRANT SELECT, INSERT ON Impiegato TO Neri;`
- Il comando viene eseguito parzialmente: Rossi ha i privilegi di select ed insert sulla relazione Impiegato ma non ha la grant option per insert quindi a Neri viene concesso solo il privilegio di select

REVOKE

`REVOKE Lista Privilegi | ALL [PRIVILEGES] ON Lista Relazioni | Lista Viste FROM Lista Utenti | PUBLIC`

- Un utente può revocare solo i privilegi che lui ha concesso.
- È possibile revocare più privilegi con un comando di REVOKE, ed un unico comando di REVOKE può essere utilizzato per revocare gli stessi privilegi sulla stessa relazione ad utenti diversi

Esempio (1)

- `REVOKE SELECT, INSERT ON Impiegato FROM Verdi, Gialli;`
- Vengono revocati a Verdi ed a Gialli i diritti di selezionare ed inserire tuple nella relazione Impiegato
- `REVOKE UPDATE ON Impiegato FROM Rossi;`
- Revoca a Rossi il diritto di modificare tuple della relazione Impiegato
- `REVOKE ALL ON Impiegato FROM Neri;`
- Revoca a Neri tutti i diritti che possedeva sulla relazione Impiegato

Quando si esegue una operazione di revoca, l'utente a cui i privilegi vengono revocati perde tali privilegi, a meno che essi non gli provengano anche da altre sorgenti indipendenti da quella che effettua la revoca.

Esempio (2)

- Bianchi: `GRANT SELECT ON Impiegato TO Verdi WITH GRANT OPTION;`
- Bianchi: `GRANT SELECT ON Impiegato TO Gialli WITH GRANT OPTION;`
- Verdi: `GRANT SELECT ON Impiegato TO Rossi;`
- Gialli: `GRANT SELECT ON Impiegato TO Rossi;`
- Verdi: `REVOKE SELECT ON Impiegato FROM Rossi;`

L'utente Rossi continua ad avere il privilegio di select sulla relazione Impiegato, anche se tale privilegio gli è stato revocato da Verdi, in quanto Rossi ha indipendentemente ottenuto tale privilegio da Gialli.

Revoca ricorsiva

- L'operazione di revoca di un privilegio è ricorsiva: è revocato il privilegio oggetto del comando di revoca e tutti i privilegi che non avrebbero potuto essere concessi se l'utente specificato nel comando di revoca non avesse ricevuto il privilegio revocato
- Un'operazione di revoca del privilegio **m** sulla relazione **R** all'utente **u1** da parte dell'utente **u2** ha l'effetto di far perdere a **u1** il privilegio **m** sulla relazione **R** (se **u1** non ha ottenuto tale privilegio da fonti indipendenti)
- Ha inoltre l'effetto di modificare il sistema portandolo in uno stato equivalente a quello in cui si sarebbe trovato se **u2** non avesse mai concesso a **u1** il privilegio di accesso **m** sulla relazione **R**.

Revoca ricorsiva

- Siano $G_1, \ldots, G_n$ una sequenza di operazioni di grant di un singolo privilegio sulla stessa relazione, tali che $\forall i, j = 1, \ldots, n$, se $i < j$, allora G_i è eseguita prima di G_j.
- Sia R_i la revoca del privilegio concesso con l'operazione G_i. La semantica della revoca ricorsiva impone che lo stato del sistema dopo l'esecuzione della sequenza:
- $G_1, \ldots, G_n, R_i$
- sia identico allo stato che si avrebbe dopo l'esecuzione della sequenza:
- $G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n$

Problema della revoca

- Necessità di determinare se un privilegio proviene da sorgenti indipendenti rispetto a quella specificata nel comando di revoke
- Sysauth e Syscolauth sono modificati per mantenere, per ogni privilegio, anche l'utente che ha concesso il privilegio, denominato grantor.

Revoca ricorsiva

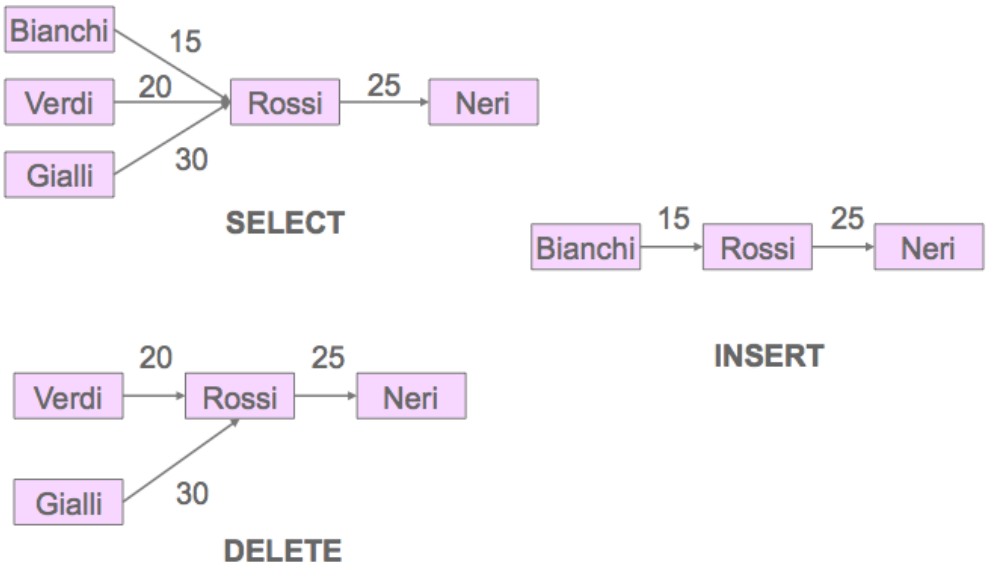
- Ogni colonna relativa ad un tipo di privilegio in Sysauth contiene (invece di 'Y' e 'N') un timestamp che denota il tempo in cui il privilegio è stato concesso.

- Il valore 0 indica che l'utente non ha quel privilegio
- Un valore $t \neq 0$ indica che privilegio è stato garantito all'utente al tempo t.
- Privilegi garantiti con lo stesso comando di GRANT hanno lo stesso timestamp.

Esempio

id_utente	nome	grantor	T	I	S	D	GO
rossi	impiegato	bianchi	R	15	15	0	Y
rossi	impiegato	verdi	R	0	20	20	Y
neri	impiegato	rossi	R	25	25	25	Y
rossi	impiegato	gialli	R	0	30	30	Y

Grafo delle autorizzazioni



Esempio (1)

- Al tempo 35 Verdi esegue il comando:
- `REVOKE ALL ON Impiegato FROM Rossi;`
- Si elimina la tupla (Rossi,Verdi,Impiegato,R,20,0,20,Y) dal catalogo Sysauth
- Si determinano quali privilegi non avrebbero potuto essere concessi se Rossi non avesse ricevuto da Verdi i privilegi revocati:

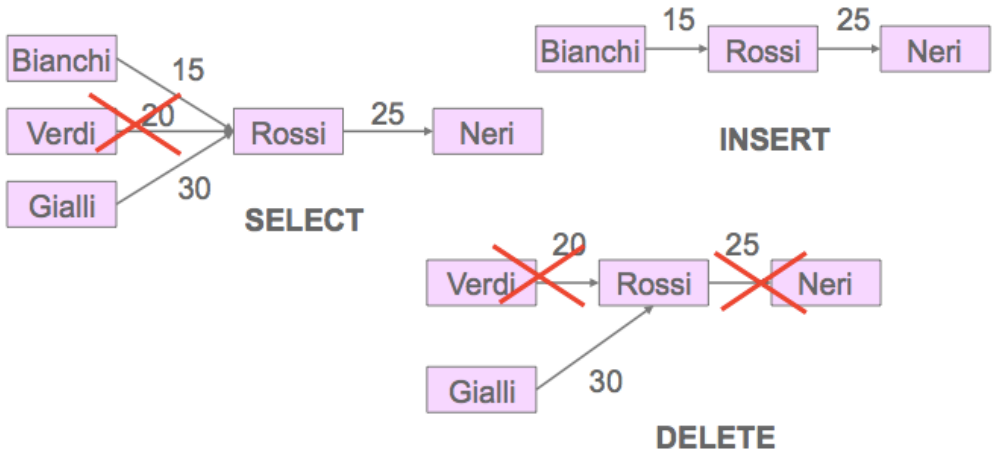
1. Si forma la lista dei timestamp dei privilegi delegabili rimanenti a Rossi, dopo che i privilegi conferiti da Verdi sono stati eliminati: DELETE=30, INSERT=15, SELECT15,30
2. Si forma la lista dei timestamp dei privilegi concessi da Rossi ad altri utenti (nell'esempio solo a Neri): DELETE=25, INSERT=25, SELECT=25
3. Un privilegio concesso da Rossi è revocato se Rossi non ha più il privilegio, oppure se ha ancora il privilegio ma SOLO con un timestamp maggiore
4. I passi 1, 2, 3 vengono eseguiti per ogni utente i cui privilegi sono stati modificati in seguito all'operazione di revoca

Esempio (2)

- Si revoca ricorsivamente il privilegio di delete concesso da Rossi a Neri (e quindi Neri perde tale privilegio) perché il timestamp 30 associato al privilegio di DELETE rimasto a Rossi è maggiore di 25, cioè del timestamp del privilegio di DELETE effettivamente concesso da Rossi a Neri
- Neri mantiene sia il privilegio di INSERT sia quello di SELECT

Esempio (3)

35: Verdi esegue REVOKE ALL ON Impiegato FROM Rossi;



2.2 Autorizzazioni si viste

Autorizzazioni su viste

- Le viste permettono di supportare il controllo dell'accesso basato su contenuto. Esempio: per autorizzare un utente a selezionare solo le tuple della relazione
- Impiegato relative ad Impiegato che non guadagnano più di 2000 euro, si definisce una vista che seleziona dalla relazione Impiegato le tuple che soddisfano tale condizione e si concede all'utente il privilegio di select sulla vista.
- Le viste permettono di:
- delegare privilegi su singole colonne di relazione: basta definire una vista come proiezione sulle colonne su cui si vogliono concedere i privilegi
- delegare privilegi statistici (media, somma, ecc.).

Autorizzazioni su viste

I privilegi che l'utente che crea una vista può esercitare sulla vista dipendono da:

- La semantica della vista, ovvero la sua definizione in termini della relazione o viste componenti
- Le autorizzazioni che l'utente possiede sulle relazioni o viste componenti
- Un privilegio sulla vista è delegabile solo se il creatore della vista ha il diritto di delegare tale privilegio su tutte le relazioni componenti.

Tipologie

- Vista V definita su una singola relazione R: Il proprietario di V ha su V gli stessi privilegi che ha su R ad eccezione dei privilegi che non si possono esercitare sulla vista a causa della sua semantica
- Vista V definita su più relazioni: Il proprietario di V ha su V l'intersezione dei privilegi che l'utente ha sulle relazioni componenti ad eccezione dei privilegi non eseguibili sulla vista

Esempio

- Rossi ha i privilegi di SELECT, INSERT e UPDATE, garantitegli da Bianchi, su Impiegato.
- Rossi esegue i comandi: `CREATE VIEW V1 AS SELECT Imp, Stipendio FROM Impiegato`
- Rossi può esercitare su V1 tutti i diritti che ha sulla relazione Impiegato.