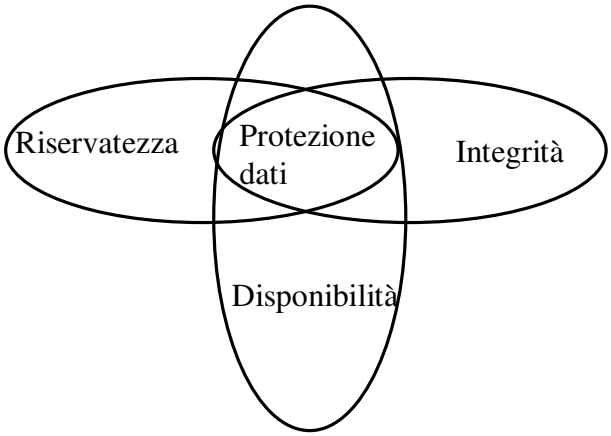
Protezione dei dati

Proprietà di sicurezza



Proprietà di sicurezza

- *Riservatezza* (o confidenzialità) – assicura la protezione dei dati da letture non autorizzate
  - nel caso di dati personali si usa il temine *privatezza* (*privacy)*
- *Integrità* – assicura la protezione dei dati da modifiche non autorizzate
- *Disponibilità* – assicura che ai soggetti con le necessarie autorizzazioni non sia negato l'accesso ai dati

Proprietà di sicurezza: Esempi

- Database degli stipendi, si deve assicurare che:
  - L'ammontare dello stipendio degli impiegati non sia rivelato a tutti gli utenti del database
  - Gli stipendi siano modificati solo dagli impiegati con la necessaria autorizzazione
  - Ogni dipendente possa accedere ai dati relativi al suo stipendio

Proprietà aggiuntive

- *Autenticità* – chi riceve informazioni è sicuro della fonte (requisito fondamentale in ambito web)
- *Completezza* – ogni soggetto riceve tutte le informazioni per cui ha le necessarie autorizzazioni

Proprietà di sicurezza – Come?

- Bisogna distinguere tra:
  - Sicurezza delle informazioni durante la trasmissione
  - Sicurezza delle informazioni mentre risiedono nel sistema informativo

Proprietà di sicurezza – Come?

- La confidenzialità è assicurata da:
  - Meccanismo di controllo dell'accesso
  - Tecniche di cifratura
- L'integrità è assicurata da:
  - Meccanismo di controllo dell'accesso
  - Vincoli di integrità
  - Tecniche di cifratura

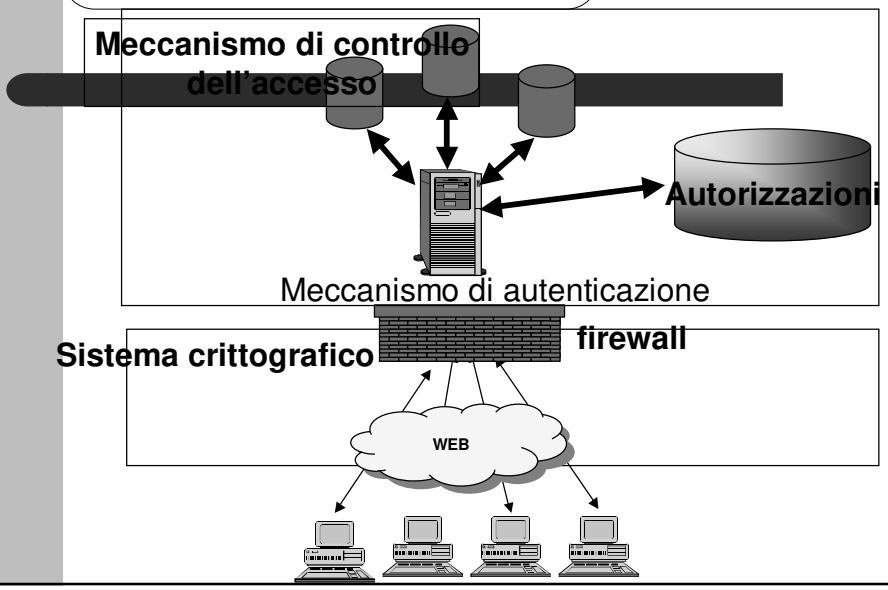
Proprietà di sicurezza – Come?

- La disponibilità è assicurata da:
  - Tecniche di recovery
  - Meccanismo di controllo della concorrenza
  - Sistemi anti denial of service
- L'autenticità è assicurata dalle tecniche di firma digitale

Proprietà di sicurezza – Come?

- Firewall:
  - Si frappone fra la rete interna ed Internet e controlla tutto il traffico in entrata/uscita
  - Il firewall deve essere un sistema fidato (trusted)

Protezione dati: una soluzione completa



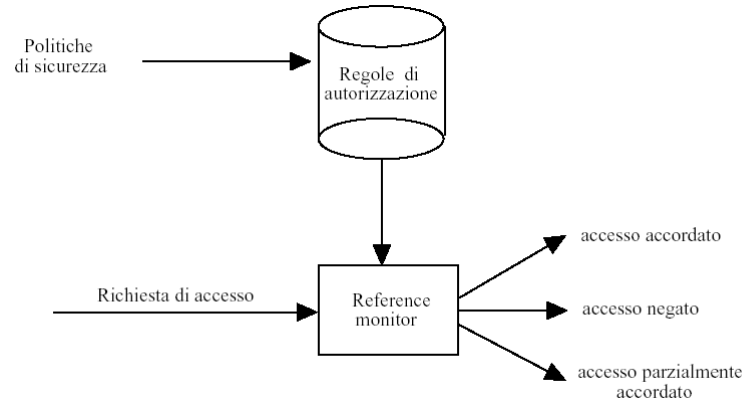
Controllo dell'accesso - concetti fondamentali

- 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 accidentali o deliberate azioni che potrebbero compromettere la correttezza e la sicurezza dei dati

Controllo dell'accesso - concetti fondamentali

- Tre componenti:
  - **soggetti**: sono gli agenti (entità attive) che richiedono di esercitare i modi di accesso sui dati
  - **oggetti**: (entità passive) ciò a cui si vuole garantire protezione (dati, risorse, ecc.)
  - **modi di accesso**: modalità di utilizzo degli oggetti da parte dei soggetti (read, select, print, ecc.)

Controllo dell'accesso - concetti fondamentali



Controllo dell'accesso - concetti fondamentali

- **Politiche di sicurezza**: leggi e principi secondo cui l'organizzazione vuole che siano gestite e protette le informazioni
  - insieme di direttive ad alto livello che esprimono le scelte di fondo dell'organizzazione relative alla sicurezza dei propri dati
- implementate mediante traduzione in un insieme di **regole di autorizzazione**: stabiliscono le operazioni e i diritti che i soggetti possono esercitare sui vari oggetti del sistema
- **reference monitor:** modulo software che ha il compito di stabilire se un soggetto può essere autorizzato (totalmente o parzialmente) a compiere un accesso

Politiche di sicurezza

- 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 della sicurezza

- Chi concede e revoca i diritti di accesso
- **Centralizzata**:
  - un unico autorizzatore (o gruppo), detto DBA, controlla l'intera base di dati
- **Decentralizzata**:
  - autorizzatori (o gruppi) diversi controllano porzioni differenti della base di dati
- **Ownership**:
  - l'utente che crea un oggetto gestisce le autorizzazioni sull'oggetto

Politiche per l'amministrazione della sicurezza

- **Gerarchica**:
  - il DBA può delegare ad altri utenti la possibilità di concedere o revocare diritti su specifici oggetti
- **Cooperativa**:
  - le autorizzazioni non possono essere concesse da un singolo utente, ma richiedono il consenso di più utenti

Politiche per il controllo dell'accesso

- Devono stabilire quando e come i soggetti possono accedere agli oggetti nel sistema, e se e come possono venire trasmessi i diritti di accesso
- le politiche per il controllo degli accessi si suddividono in:
  - **politiche discrezionali**
  - **politiche mandatorie**

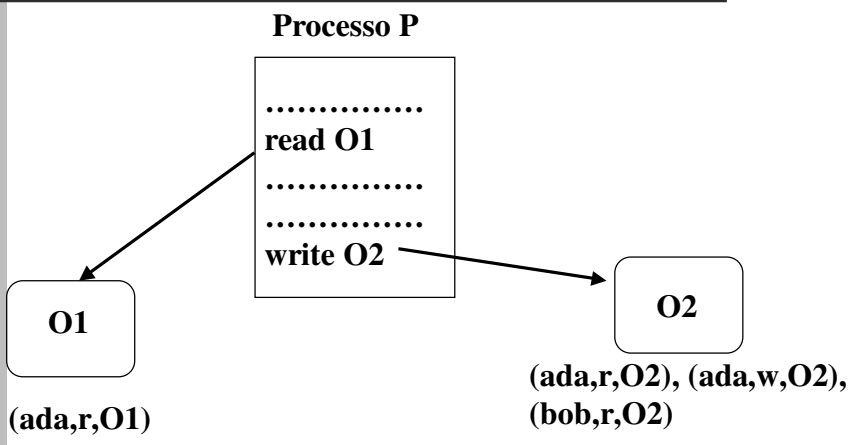
Politiche discrezionali

- Richiedono che vengano stabiliti, mediante apposite regole di autorizzazione, i diritti che ogni soggetto possiede sugli oggetti del sistema
- il meccanismo di controllo esamina le richieste di accesso accordando solo quelle che sono autorizzate da una regola
- permettono ai soggetti di concedere o revocare dei diritti di accesso sugli oggetti, a loro discrezione

Politiche discrezionali

- vantaggio:
  - sono estremamente flessibili e adatte a numerosi contesti applicativi
- svantaggio:
  - non forniscono alcun controllo sul flusso di informazioni nel sistema, non impongono restrizioni sull'uso che un utente fa dell'informazione, una volta acceduta

Cavallo di Troia



Politiche mandatorie

- Gli accessi consentiti nel sistema sono stabiliti classificando oggetti e soggetti in livelli di sicurezza
- questo tipo di sicurezza si chiama anche ***sicurezza multilivello***
- i DBMS che adottano una politica mandatoria si chiamano multilevel secure database management systems (MLS/DBMSs)

Politiche mandatorie

- Il controllo dell'accesso avviene sulla base di un insieme di *assiomi di sicurezza*
- gli assiomi di sicurezza stabiliscono le relazioni che devono intercorrere tra il livello di sicurezza di un soggetto *s* ed un oggetto *o* affinchè *s* possa accedere ad *o*
- le relazioni dipendono dal modo di accesso considerato

Modelli di controllo dell'accesso

- Mandatorio: Modello di Bell e La Padula
- Discrezionale: Modello del System R

Modello di Bell e La Padula

- Modello di riferimento per sistemi che adottano una politica di tipo mandatorio
  - **oggetti**: entità passive che contengono informazioni da proteggere
  - **soggetti**: entità attive che richiedono accesso agli oggetti (utenti e processi)
  - **modi di accesso**: tipi di accesso di un soggetto su un oggetto:
    - *read*: leggere ma non modificare
    - *append*: modificare ma non leggere
    - *write*: sia modificare che leggere
    - *execute*: eseguire ma non leggere né modificare direttamente (modalità usata per programmi applicativi, utility, ecc.).

Modello di Bell e La Padula

- I soggetti e gli oggetti del sistema vengono classificati mediante l'assegnamento di una **classe di accesso**
- una classe di accesso è costituita da due componenti: un livello di sicurezza ed un insieme di categorie
- il **livello di sicurezza** è un elemento di un insieme totalmente ordinato

  ad esempio: Top Secret (TS), Secret (S), Confidential (C) e Unclassified (U), dove: TS > S > C > U
- l'**insieme di categorie** è un insieme non ordinato di elementi, che dipendono dal tipo di ambiente e dall'area applicativa

  ad esempio: Army, Navy, Air Force, Nuclear

Modello di Bell e La Padula
Relazione di dominanza

- Una classe di accesso $c_1 = (L_1, SC_1)$ **domina** una classe di accesso $c_2 = (L_2, SC_2)$, indicato con $c_1 \geq c_2$, se entrambe le seguenti condizioni sono verificate:
  - il livello di sicurezza di c_1 è maggiore o uguale al livello di sicurezza di c_2 (cioè $L_1 \geq L_2$)
  - l'insieme di categorie di c_1 include l'insieme di categorie di c_2 (cioè $SC_1 \supseteq SC_2$)
- c_1 e c_2 si dicono incomparabili ($c_1 <> c_2$) se né $c_1 \geq c_2$ né $c_2 \geq c_1$ valgono
- Indichiamo con c(s) e c(o) la classe di accesso di un soggetto s e di un oggetto o, rispettivamente

Modello di Bell e La Padula
Relazione di dominanza - Esempio

- Classi di accesso:
  - c1 = (TS,{Nuclear,Army})
  - c2 = (TS,{Nuclear})
  - c3 = (C,{Army})
- c1 $\geq$ c2 (TS $\geq$ TS e {Nuclear} $\subset$ {Nuclear,Army})
- c1 $\geq$ c3 (TS > C e {Army} $\subset$ {Nuclear,Army})
- c 2 <> c 3 (c2 $\not\geq$ c3 perché {Nuclear} $\not\supseteq$ {Army} e c3 $\not\geq$ c2 perché TS > C e {Army} $\supseteq$ {Nuclear})

Modello di Bell e La Padula Assiomi

- **Proprietà di sicurezza semplice** *(no read up)*
  - Un soggetto s può effettuare l'azione m sull'oggetto o se una delle seguenti condizioni è verificata:
    - m= execute oppure m = append
    - (m = read oppure m = write) e c(s) ≥ c(o)
  - un soggetto con classe di accesso (C,{Army}) non può leggere dati con classi di accesso (C,{Navy,Air Force}) o (U,{Air Force})
  - evita che soggetti leggano informazioni con classe di accesso maggiore o incomparabile, e assicura che i soggetti abbiano accesso diretto solo alle informazioni per cui hanno la necessaria classificazione

Modello di Bell e La Padula Assiomi

- **Star (*) proprietà** *(no write down)*
  - Un soggetto s può effettuare l'azione m sull'oggetto o se una delle seguenti condizioni è verificata:
    - m= execute oppure m = read
    - (m = append oppure m = write) e c(s) ≤ c(o)
  - un soggetto con classe di accesso (C,{Army,Nuclear}) non può scrivere informazioni in oggetti con classe di accesso (U, {Army,Nuclear})
  - previene il flusso di informazioni verso classi di accesso minori o non comparabili
- Poiché entrambi gli assiomi devono essere verificati, s può effettuare write su o solo se c(s) = c(o)

Covert Channel

- Un **covert channel** permette un trasferimento di informazione che viola la politica multilivello
- Timing covert channel:
  - il passaggio dell'informazione avviene tracciando il tempo necessario ad eseguire determinati processi o la durata di un evento
- Il modello di Bell e La Padula non protegge contro i covert channel

Covert Channel - Esempio

- Un covert channel molto noto si basa sull'utilizzo nel protocollo 2PL per il controllo della concorrenza
- Si considerino due transazioni T_l e T_h con classe di accesso low and high rispettivamente
- si consideri un dato d_1 con classe di accesso low
- si assuma che le uniche transazioni in esecuzione siano T_l e T_h
- si supponga che T_h richieda un read lock su d_1
  - il lock è concesso perché nessun altra transazione è in esecuzione

Covert Channel - Esempio

- Si supponga adesso che T_l voglia scrivere lo stesso dato
  - richiede un write lock su d_1
- poiché T_h possiede un read lock to d_1, T_l deve aspettare finché T_h rilascia il lock su d_1
- l'attesa di T_l può essere modulata da T_h poiché T_h ha accesso totale a dati classificati high, T_h può utilizzare questa attesa per passare informazione high a T_l
- sono stati proposti algoritmi di controllo della concorrenza particolari per evitare varie tipologie di covert channel

Modello di autorizzazione del System R

- Gli oggetti del modello sono relazioni (sia di base che viste)
- i privilegi previsti sono:
  - *alter*: aggiungere una nuova colonna ad una relazione
  - *index*: creare un indice su una relazione
  - *delete*: cancellare tuple da una relazione
  - *insert*: inserire tuple in una relazione
  - *select*: selezionare tuple da una relazione
  - *update*: modificare valori di attributi in tuple di una relazione
  - non esiste il privilegio di drop

Modello di autorizzazione del System R

- 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

Modello di autorizzazione del System R - Delega dei privilegi

- 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 soggetto può concedere un privilegio su una determinata relazione solo se è il proprietario della relazione, o ha ricevuto tale privilegio con grant option

Modello del System R
L'operazione di GRANT

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

- si possono concedere privilegi sia su relazioni che su viste
- i privilegi alter e index si applicano solo a relazioni
- i privilegi si applicano ad intere relazioni (o viste)

Modello del System R
L'operazione di GRANT

- per il privilegio di update è necessario specificare le colonne a cui si applica
- le parole chiave ALL o ALL PRIVILEGES (equivalenti) consentono di concedere con un solo comando tutti i privilegi su una determinata relazione
- non possono essere utilizzate su viste

Modello del System R
L'operazione di GRANT

- Con un unico comando di GRANT si possono concedere più privilegi su una stessa relazione e concedere privilegi sulla stessa relazione a più utenti (in entrambi i casi l'ordine è irrilevante)
- un comando di GRANT con soggetto PUBLIC si applica a tutti gli utenti del sistema
- se la clausola WITH GRANT OPTION non è specificata l'utente che riceve i privilegi non può concederli ad altri utenti
- i privilegi di ogni utente sono quindi suddivisi in:
  - privilegi delegabili (concessi con grant option)
  - privilegi non delegabili (senza grant option)

Modello del System R
L'operazione di GRANT - Esempio

GRANT update(Stipendio,Premio P) ON Impiegati TO Rossi;

GRANT select,insert ON Impiegati TO Verdi,Gialli;

GRANT ALL PRIVILEGES ON Impiegati TO Neri WITH GRANT OPTION;

- Rossi può modificare gli attributi Stipendio e Premio P delle tuple della relazione Impiegati
- Verdi e Gialli possono selezionare ed inserire tuple nella relazione Impiegati
- Neri ha tutti i privilegi sulla relazione Impiegati e può delegare ad altri tali privilegi

Modello del System R
L'operazione di GRANT - Esempio

Bianchi: GRANT select,insert ON Impiegati
TO Verdi WITH GRANT OPTION;
Bianchi: GRANT select ON Impiegati
TO Rossi WITH GRANT OPTION;
Verdi: GRANT select,insert ON Impiegati
TO Rossi;

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

Modello del System R
Cataloghi 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*: identificatore 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')

Modello del System R
Cataloghi Sysauth e Syscolauth

- *alter* in {Y,N}: indica se il soggetto ha (alter=`Y') o meno (alter=`N') il privilegio di aggiungere una nuova colonna alla relazione
- Sysauth contiene un analogo attributo per i privilegi *index, delete, insert e select;*
- *update* in {ALL,SOME,N}: indica se il soggetto ha il privilegio di update su tutte (update=`ALL'), alcune (update=`SOME'), o nessuna (update=`N') colonna della relazione;
- *grantopt* in {Y,N}: indica se i privilegi sono delegabili (grantopt=`Y') o meno (grantopt=`N')

Modello del System R
Cataloghi Sysauth e Syscolauth
Esempio

id_ut.	nome	creat.	t	a	i	d	ins	s	u	go
Bianchi	Impiegati	Bianchi	R	Y	Y	Y	Y	Y	ALL	Y
Verdi	Impiegati	Bianchi	R	N	N	N	Y	Y	N	Y
Rossi	Impiegati	Bianchi	R	N	N	N	N	Y	N	Y
Rossi	Impiegati	Bianchi	R	N	N	N	Y	Y	N	N

- si suppone che la relazione Impiegati sia stata creata da Bianchi
- per ogni relazione (o vista) su cui un utente ha privilegi, sono presenti al più due tuple nel catalogo Sysauth:
  - una rappresentante i privilegi delegabili (grantop=`Y')
  - una rappresentante i privilegi non delegabili (grantop=`N')

Modello del System R
Cataloghi Sysauth e Syscolauth

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

id_utente	nome	colonna	grantopt
Bianchi	Impiegati	Imp#	Y
Bianchi	Impiegati	Nome	Y
Bianchi	Impiegati	Mansione	Y
Bianchi	Impiegati	Data_A	Y
Bianchi	Impiegati	Stipendio	Y
Bianchi	Impiegati	Premio_P	Y
Bianchi	Impiegati	Dip#	Y

Modello del System R
L'operazione di GRANT

- Quando un utente u esegue un comando di GRANT, il reference monitor esegue una query su Sysauth e Syscolauth per determinare se u ha il diritto di concedere 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

Modello del System R
L'operazione di GRANT

- tre possibili casi
  - se l'intersezione è vuota, il comando non viene eseguito
  - se l'intersezione coincide con i privilegi specificati nel comando, il comando viene eseguito, e tutti i privilegi specificati vengono concessi
  - altrimenti il comando viene eseguito parzialmente, cioè solo i privilegi contenuti nell'intersezione vengono accordati

Modello del System R
L'operazione di GRANT - Esempio

Bianchi: GRANT select,insert ON Impiegati
TO Gialli WITH GRANT OPTION;

Verdi: GRANT update, ON Impiegati
TO Gialli WITH GRANT OPTION;

Rossi: GRANT select,insert ON Impiegati TO Neri;

- il primo comando di GRANT viene eseguito (Bianchi è il proprietario della relazione Impiegati)
- il secondo non viene eseguito (Verdi non possiede il privilegio di update sulla relazione Impiegati)
- il terzo viene parzialmente eseguito (Rossi ha i privilegi di select ed insert sulla relazione Impiegati ma non ha la grant option per insert) a Neri viene concesso solo privilegio di select

Modello del System R
L'operazione di 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 unico comando di REVOKE, ed un unico comando di REVOKE può essere utilizzato per revocare gli stessi privilegi sulla stessa relazione ad utenti diversi

Modello del System R
L'operazione di REVOKE

- la parola chiave PUBLIC indica che la revoca si applica a tutti gli utenti, mentre ALL (o ALL PRIVILEGES) indica che tutti i privilegi previsti dal modello sono revocati
- se la clausola ON non è specificata, la revoca si applica a tutte le relazioni della base di dati

Modello del System R
L'operazione di REVOKE

- Il comando di REVOKE non consente di revocare in modo selettivo il diritto di update
- non è possibile revocare solo la grant option: occorre revocare tutto il privilegio concesso con grant option e successivamente rieseguire il comando di GRANT senza grant option
- quando si esegue una operazione di revoca, l'utente a cui i privilegi sono stati revocati perde tali privilegi, a meno che essi non gli provengano anche da altre sorgenti indipendenti da quella che effettua la revoca

Modello del System R
L'operazione di REVOKE - Esempio

REVOKE select, insert ON Impiegati FROM Verdi,Gialli;
REVOKE update ON Impiegati FROM Rossi;
REVOKE ALL ON Impiegati FROM Neri;

- vengono revocati a Verdi ed a Gialli i diritti di selezionare ed inserire tuple nella relazione Impiegati
- revoca a Rossi il diritto di modificare tuple della relazione Impiegati
- revoca a Neri tutti i diritti che possedeva sulla relazione Impiegati

Modello del System R
L'operazione di REVOKE - Esempio

Bianchi:	GRANT select ON Impiegati TO Verdi WITH GRANT OPTION;
Bianchi:	GRANT select ON Impiegati TO Gialli WITH GRANT OPTION;
Verdi:	GRANT select ON Impiegati TO Rossi;
Gialli:	GRANT select ON Impiegati TO Rossi;
Verdi:	REVOKE select ON Impiegati FROM Rossi;

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

Modello del System R
Revoca e grant option

Bianchi:	GRANT select ON Impiegati TO Verdi WITH GRANT OPTION;
Verdi:	GRANT select ON Impiegati TO Rossi;
Bianchi:	REVOKE select ON Impiegati FROM Verdi;

Cosa succede dell'autorizzazione di Rossi?

Modello del System R
Revoca ricorsiva

- Un'operazione di revoca del modo di accesso **m** sulla relazione **rel** all'utente **u1** da parte dell'utente **u2** ha l'effetto non solo di far perdere **m** ad **u1** su **rel**, se **u1** non ha ottenuto tale privilegio da fonti indipendenti, ma anche di modificare il sistema portandolo in uno stato equivalente a quello in cui si sarebbe trovato se **u2** non avesse mai concesso ad **u1 m** su **rel**
- il sistema deve ricorsivamente revocare tutti i privilegi che non avrebbero potuto essere concessi se **u1** non avesse ricevuto il privilegio revocato
- è utilizzata dalla maggior parte dei DBMS relazionali oggi in commercio, quali per esempio Oracle, Informix e DB2

Modello del System 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,…,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
- utilizzando la revoca ricorsiva lo stato del sistema dopo l'esecuzione della sequenza:

  $G_1,\ldots,G_n, R_i$

  è identico allo stato che si sarebbe ottenuto dopo l'esecuzione della sequenza:

  $G_1,\ldots,G_{i-1}, G_{i+1},\ldots,G_n$

Modello del System R
Revoca ricorsiva - Esempio

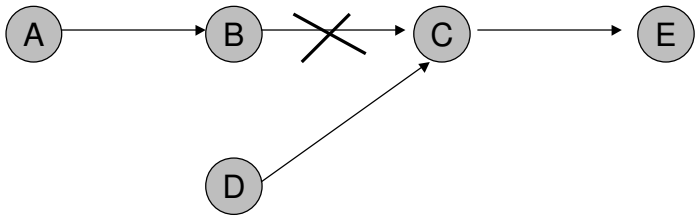
REVOKE select ON Impiegati FROM Verdi;

- sia Verdi che Rossi perdono il privilegio di select sulla relazione Impiegati: Verdi a seguito della richiesta contenuta nel comando, Rossi perché non avrebbe potuto ricevere il privilegio di select da Verdi se quest'ultimo non lo avesse ricevuto, con grant option, da Bianchi

Modello del System R
Grafo delle autorizzazioni

- Lo stato delle autorizzazioni rispetto ad un dato modo di accesso **m** su una certa relazione **T** può essere rappresentato mediante un grafo orientato in cui:
  - i nodi rappresentano i soggetti
  - esiste un arco dal nodo u1 al nodo u2 se u1 ha concesso ad u2 il modo di accesso m su T

Modello del System R
Grafo delle autorizzazioni

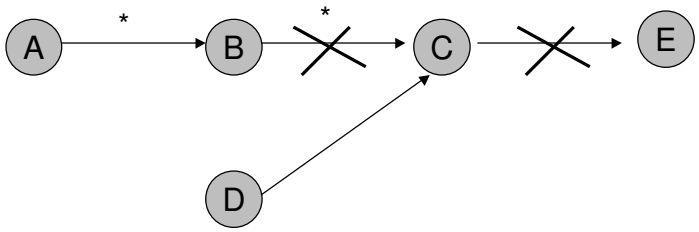


C avrebbe potuto concedere a E il privilegio?

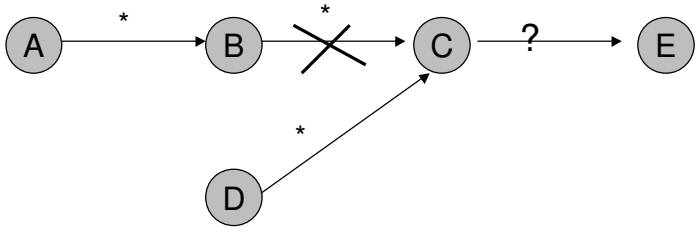
Modello del System R
Grafo delle autorizzazioni

- Si deve innanzitutto sapere se D ha concesso a C il privilegio con grant option:
  - modifica del grafo delle autorizzazioni
  - un arco da u1 ad u2 viene etichettato con * se u1 ha concesso il privilegio ad u2 con grant option

Modello del System R
Grafo delle autorizzazioni

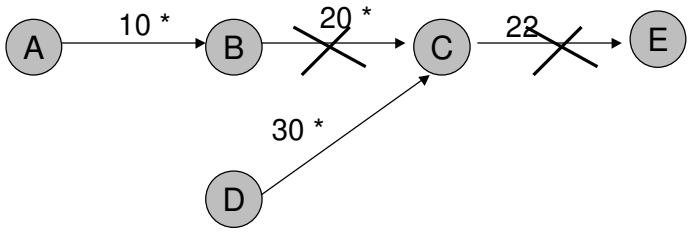


Modello del System R
Grafo delle autorizzazioni



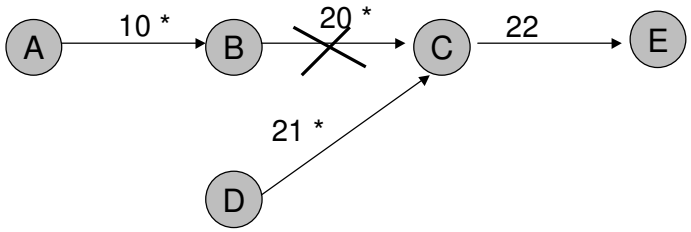
Le informazioni non sono sufficienti

Modello del System R
Grafo delle autorizzazioni con timestamp



C non avrebbe potuto concedere l'autorizzazione a E al tempo 22 se non l'avesse ricevuta al tempo 20 da B

Modello del System R
Grafo delle autorizzazioni con timestamp



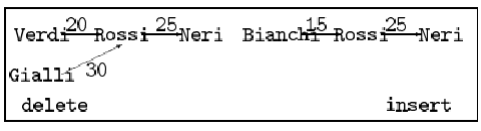
C avrebbe potuto concedere l'autorizzazione a E al tempo 22 se non l'avesse ricevuta al tempo 20 da B

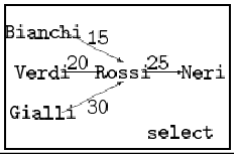
Modello del System R
Revoca ricorsiva

- Necessità di determinare se un privilegio proviene da sorgenti indipendenti rispetto a quella specificata nel comando di revoca
  - Sysauth e Syscolauth sono modificati per mantenere, per ogni privilegio, anche l'utente che ha concesso il privilegio (grantor)
- 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, mentre un valore t diverso da 0 indica che il privilegio è stato concesso al tempo t
- privilegi concessi con lo stesso comando di GRANT hanno lo stesso timestamp

Modello del System R
Revoca ricorsiva - Esempio

id_utente	grantor	nome	t	s	i	d	go
Rossi	Bianchi	Impiegati	R	15	15	0	Y
Rossi	Verdi	Impiegati	R	20	0	20	Y
Neri	Rossi	Impiegati	R	25	25	25	Y
Rossi	Gialli	Impiegati	R	30	0	30	Y





Tutti gli archi hanno l'etichetta *

Modello del System R
Revoca ricorsiva - Esempio

- Al tempo 35 Verdi esegue il comando:

  REVOKE ALL ON Impiegati FROM Rossi;

  - si elimina la tupla (Rossi,Verdi,Impiegati,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
  - per ogni utente i cui privilegi sono stati modificati in seguito all'operazione di revoca:

Modello del System R
Revoca ricorsiva – Esempio

1. si forma la lista dei timestamp dei privilegi delegabili rimanenti a Rossi, dopo che i privilegi concessi da Verdi sono stati eliminati:

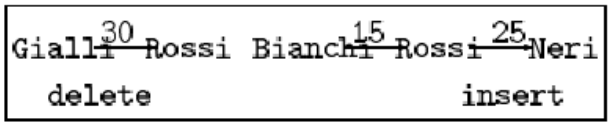
   delete={30}, insert={15}, select={15,30}
2. si forma la lista dei timestamp dei privilegi concessi da Rossi ad altri utenti (nell'es. solo a Neri):

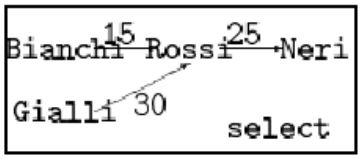
   delete={25}, insert={25}, select={25}
3. un privilegio garantito da Rossi è revocato se Rossi non ha più il privilegio, oppure se ha ancora il privilegio ma con un timestamp maggiore

Modello del System R
Revoca ricorsiva – Esempio

- si revoca il privilegio di delete a Neri (il timestamp associato al privilegio di delete per Rossi -30- è maggiore di 25, cioè del timestamp del privilegio di delete garantito da Rossi a Neri)
- Neri mantiene sia il privilegio di insert sia quello di select (il privilegio di insert concesso da Rossi a Neri era stato concesso a Rossi da Bianchi e Rossi avrebbe potuto concedere al tempo 25 il privilegio di select a Neri anche se non avesse ricevuto tale privilegio da Verdi al tempo 10, grazie all'autorizzazione ricevuta al tempo 15 da Bianchi)

Modello del System R
Revoca ricorsiva – Esempio





Autorizzazione su viste

- Le viste permettono di supportare il controllo dell'accesso in base al contenuto
  - esempio: per autorizzare un utente a selezionare solo le tuple della relazione Impiegati relative ad impiegati che non guadagnano più di due milioni di lire, si definisce una vista che seleziona dalla relazione Impiegati le tuple che soddisfano la condizione e si concede all'utente il privilegio di select sulla vista
- permettono di delegare privilegi su singole colonne di relazioni:
  - basta definire una vista come proiezione sulle colonne su cui si vogliono concedere i privilegi
- permettono di delegare privilegi statistici (media, somma, ecc.)

Autorizzazione su viste

- Esempio: supponiamo di voler autorizzare Ann ad accedere solo alle informazioni su impiegati che guadagnano meno di 2000 euro – passi necessari:
  - CREATE VIEW L_imp AS
    SELECT * FROM Impiegati
    WHERE Stipendio < 2000;
  - GRANT Select ON L_imp TO Ann;

Autorizzazione su viste

- Le query sulle viste sono trasformate in query sulle corrispondenti tabelle di base mediante il meccanismo di composizione delle viste
- esempio:

```
Ann>  SELECT * FROM L_imp
      WHERE Mansione= 'tecnico';
                ⇓
SELECT * FROM Impiegati
WHERE Stipendio < 2000 AND Mansione = 'tecnico';
```

- la verifica delle autorizzazioni avviene prima della composizione delle viste ⇒ avviene rispetto alle viste e non alle tabelle di base su cui sono definite

Autorizzazione 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 delle relazioni o viste componenti
  - le autorizzazioni che l'utente possiede sulle relazioni o viste componenti
- i privilegi alter e index non si applicano alle viste
- non si ottengono privilegi relativi ad operazioni non consentite sulla vista (ad esempio, se la vista è definita come un join non è possibile cancellare tuple)

Autorizzazione su viste - Esempio

```
Bob> CREATE VIEW V1 (Imp#, Stip_tot)
       AS SELECT Imp#, Stipendio + Bonus
            FROM Impiegati
            WHERE Mansione ='ingegnere';
```

non si possono effettuare update sulla colonna Stip_tot

⇓

Bob non riceverà il privilegio di update sulla colonna anche se lo aveva sulla relazione Impiegati

Autorizzazione su viste

- Per determinare quali sono i privilegi che chi crea una vista può esercitare sulla vista stessa, si deve fare l'intersezione tra i privilegi che l'utente che crea la vista ha sulle tabelle di base e quelli esercitabili sulla vista, in base alla sua definizione

Autorizzazione su viste - Esempio

- Si supponga che Bob abbia creato la relazione Impiegati e si consideri la seguente sequenza di comandi:
  - Bob> GRANT Select, Insert, Update ON Impiegati to Tim;
  - Tim> CREATE VIEW V1 AS
    SELECT Imp#, Stipendio FROM Impiegati;
  - Tim> CREATE VIEW V2 (Imp#, Stipendio_Annuale) AS
    SELECT Imp#, Stipendio*12 FROM Impiegati;
- Tim può esercitare su V1 tutti i privilegi che ha su Impiegati, cioè: Select, Insert, Update
- Tim può esercitare su V2 solo il privilegio di Select e quello di Update limitatamente alla colonna Imp#

Autorizzazione su viste

- E' possibile concedere privilegi su viste: un utente può concedere su una vista tutti i privilegi che ha con grant option sulle tabelle di base e che si possono applicare sulla vista in base alla sua definizione, oppure quelli che gli sono stati concessi sulla vista con grant option
- nell'esempio: Tim non può concedere alcun privilegio su V1 e V2, perché non ha privilegi con grant option sulla tabella Impiegati

Autorizzazione su viste - Esempio

- Bob> GRANT Select ON Impiegati TO Tim
  WITH GRANT OPTION;
- Bob> GRANT Update, Insert ON Impiegati TO Tim;
- Tim> CREATE VIEW V4 AS
  SELECT Imp#, Stipendio FROM Impiegati;

Autorizzazioni di Tim su V4:

- Select con Grant Option;
- Update, Insert senza Grant Option;

Autorizzazione su viste

- Le operazioni di revoca dei privilegi su una vista sono più complicate di quelle sulle relazioni
- è necessario stabilire cosa succede alla vista se un privilegio di select sulle relazioni di base è revocato
- anche in questo caso si adotta la revoca ricorsiva: la vista viene cancellata se il privilegio revocato era l'unico che consentiva la sua definizione
- Nell'esempio:
  - Bob> REVOKE Select ON Impiegati TO Tim;
  - la vista V4 creata da Tim viene ricorsivamente cancellata

Modelli basati sui ruoli (RBAC) Motivazioni

- RBAC (Role-based Access Control) si basa sulla considerazione che nella maggior parte delle organizzazioni gli utenti finali non sono i proprietari dell'informazione che gestiscono, che invece è proprietà dell'organizzazione stessa
- gli accessi che gli utenti possono esercitare sugli oggetti del sistema dipendono prevalentemente dalla funzione che essi ricoprono all'interno dell'organizzazione piuttosto che dalla loro identità

RBAC: Concetti Base

- I ruoli rappresentano specifiche funzioni all'interno dell'azienda/organizzazione
- Le autorizzazioni sono concesse ai ruoli invece che ai singoli utenti
- Gli utenti sono autorizzati a ricoprire uno o più ruoli a seconda della loro posizione all'interno dell'organizzazione
- Quanto un utente riceve l'autorizzazione a ricoprire un ruolo, automaticamente eredita tutte le autorizzazioni di quel ruolo

RBAC: Vantaggi

- Le politiche di sicurezza che si possono specificare sono strettamente correlate alle dinamiche aziendali
- La concessione e revoca dei privilegi agli utenti è notevolmente semplificata
- I modelli RBAC possono implementare qualsiasi politica di controllo dell'accesso

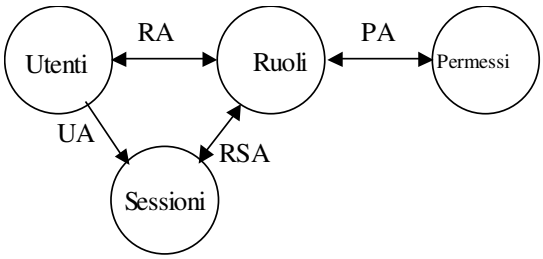
RBAC

- I produttori di DBMS hanno riconosciuto l'importanza del concetto di ruolo, e oggi molti DBMS commerciali supportano RBAC
- Non esiste però una standardizzazione
- Il modello NIST [Sandhu,Ferraiolo,Kuhn 00] rappresenta un primo tentativo di standardizzazione

RBAC- Componenti Base

- *Utente* – un essere umano, una macchina, un processo, un agente attivo nel sistema
- *Ruolo* – una funzione all'interno di un contesto organizzativo, con associati un insieme di privilegi
- *Permesso* – modo di accesso che può essere esercitato sugli oggetti del sistema, sia gli oggetti che i modi di accesso sono dipendenti dal dominio
- *Sessione* – una sessione rappresenta una connessione al sistema da parte di un utente
  - durante una sessione un utente può attivare un sottoinsieme dei ruoli che è autorizzato a ricoprire
  - un utente può attivare più sessioni contemporaneamente

RBAC Flat



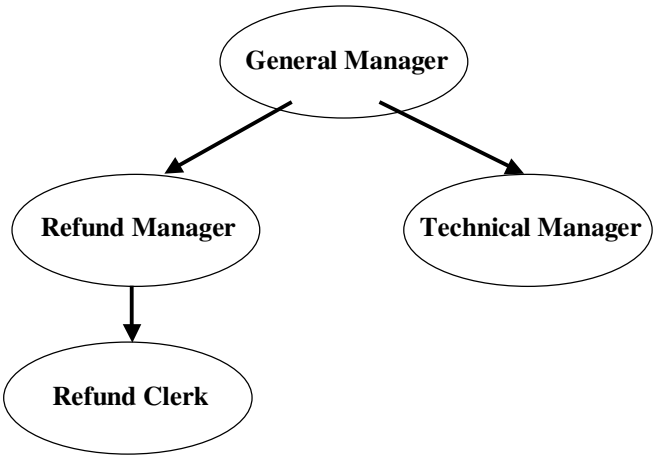
RBAC Gerarchico

- Le gerarchie sui ruoli sono un mezzo naturale per rappresentare una relazione di autorità che di solito si instaura tra ruoli in una certa organizzazione
- Tale gerarchia ha impatto sulle autorizzazioni che ogni ruolo possiede

RBAC Gerarchico

- Dati due ruoli ri ed rj, tali che ri precede rj nella gerarchia:
  - ri è detto senior rispetto a rj
  - rj è un ruolo junior rispetto ad ri
- Le autorizzazioni concesse ad un ruolo sono automaticamente ereditate da tutti i ruoli senior

RBAC Gerarchico



RBAC – Comandi SQL

- CREATE ROLE nome-ruolo IDENTIFIED BY passw |NOT IDENTIFIED;

  esempio:

  CREATE ROLE contabile IDENTIFIED BY cnt;

- DROP ROLE nome-ruolo;

Comandi SQL

- GRANT role TO user | role | PUBLIC [WITH ADMIN OPTION];

  un utente può effettuare il grant di un ruolo solo se è autorizzato a ricoprire quel ruolo con ADMIN option, oppure ha il privilegio di sistema GRANT ANY ROLE

  l'ADMIN option consente anche di modificare o fare il drop di un ruolo
- Esempio:

  GRANT contabile TO Bob;

Comandi SQL

- Il comando di grant del System R viene esteso con la possibilità di avere un ruolo come soggetto:

  GRANT select ON Impiegato TO contabile;

Comandi SQL

- SET ROLE nome-ruolo IDENTIFIED BY passwd;
  per abilitare/disabilitare un ruolo durante una sessione

  SET ROLE contabile IDENTIFIED by cnt;
- SET ROLE ALL [EXCEPT nome-ruolo]
  SET ROLE ALL;
  SET ROLE ALL EXCEPT banker;
- SET ROLE NONE;
  disabilita tutti i ruoli per la sessione corrente