

Basi di dati
Prof. Stefano Paraboschi
Prova (b) del 18-12-2002

A. Si deve costruire una base di dati che consenta di gestire un sistema di autorizzazioni interno a un sistema informatico. Ogni autorizzazione è caratterizzata dal soggetto che beneficia dell'autorizzazione, dall'oggetto su cui si applica l'autorizzazione, dall'azione (lettura, scrittura, etc.) e dal segno (positivo se l'autorizzazione consente al soggetto di svolgere l'azione sull'oggetto; negativo se invece l'autorizzazione impedisce di svolgere quella particolare azione). I soggetti possono essere utenti, gruppi, o ruoli. Gli utenti corrispondono normalmente a persone fisiche, caratterizzate dai dati anagrafici e dall'incarico all'interno dell'organizzazione; ogni utente è caratterizzato da un nome di login e da una password, che viene conservata in formato cifrato. Un gruppo rappresenta un insieme di utenti o di gruppi. Un gruppo/utente può far parte di diversi gruppi. Un ruolo rappresenta una responsabilità che un utente o un membro di un gruppo può assumere in un certo istante, normalmente per svolgere uno specifico compito. Utenti e gruppi devono essere autorizzati a ricoprire i ruoli che assumono.

Vi sono quindi due tipologie di autorizzazioni: autorizzazioni che consentono a un utente o a un gruppo di assumere un ruolo, e autorizzazioni che consentono a un utente, gruppo o ruolo di svolgere un'azione su un oggetto. Bisogna anche tener conto di chi ha concesso un'autorizzazione. Il privilegio di assumere un ruolo può essere concesso solo dal database administrator. I privilegi di accesso alle risorse possono invece essere attribuiti sia dal proprietario della risorsa che dall'amministratore della base di dati; questi privilegi possono essere concessi con la grant option. Il sistema deve automaticamente eliminare un privilegio derivato, quando il privilegio in base al quale era stato concesso viene revocato, per mantenere la consistenza dell'insieme di autorizzazioni.

Per quanto riguarda le risorse su cui si possono concedere i privilegi, queste possono essere organizzate in gerarchie (ad esempio, un direttorio che contiene file e altri direttori, per un numero arbitrario di livelli). Le autorizzazioni possono propagarsi da una risorsa a tutte le risorse in essa contenute.

1. Svolgere il progetto concettuale; si ricorda di specificare un identificatore per ogni entità e cardinalità minima e massima di ogni relazione. (6 punti)
2. Svolgere il progetto logico, descrivendo le chiavi di ogni tabella e i cammini di join. (3 punti)
3. Realizzare, a partire dai dati presenti nello schema sopra descritto, un modello multidimensionale per effettuare analisi sulle autorizzazioni. (3 punti)

B. Si ha il seguente schema relazionale per una società di trasporti pubblici:

AUTOBUS(Targa,Modello,Deposito)
AUTISTA(Mat~~r~~,Nome,Residenza,Deposito)
CORSA(MatAutista,Data,Ora,TargaMezzo,Linea,Verso)
LINEA(Linea,Verso,Tipo,ServizioNotturmo)
PERCORSO(Linea,Verso,NroProgr,Via,NumCivico)

1. Scrivere il comando SQL che assegna all'attributo ServizioNotturmo di LINEA il valore true se sulla linea, indipendentemente dal valore di Verso, si svolgono tutti i giorni almeno 6 corse con inizio compreso tra le ore 0.30 e le ore 6.00 (si tenga conto che non si può assumere che per tutte le linee ci sia tutti i giorni almeno una corsa nell'intervallo 0.30-6.00). (5 punti)
2. Formulare in algebra relazionale ottimizzata o nel calcolo relazionale delle tuple una query che estrae i nomi degli autisti che hanno svolto corse per due anni consecutivi nel giorno di Natale (le funzioni scalari year, month e day applicate a un valore del dominio date restituiscono rispettivamente l'anno, il mese e il giorno). (3 punti)
3. Supponendo che lo schema di AUTISTA sia esteso con un attributo NumTotCorse che rappresenta il numero totale di corse effettuate dall'autista, creare una regola attiva che mantiene aggiornato l'attributo in seguito a inserimenti in CORSA. (2 punti; 1 punto in più per una soluzione efficiente)

C. Data la gerarchia di risorse $A \rightarrow \{B, C, D\}$ (in cui A è la radice della gerarchia e B, C e D sono le foglie, considerare la sequenza di richieste di lock:

$isl_1(A) \text{ } ixl_2(A) \text{ } sl_3(B) \text{ } sl_1(D) \text{ } xl_1(C) \text{ } xl_2(B) \text{ } xl_2(D) \text{ } sl_2(C) \text{ } ixl_5(A) \text{ } xl_5(C) \text{ } sl_5(B) \text{ } xl_2(C)$

- Eliminare le richieste di lock che non rispettano le regole della tecnica di lock gerarchico (se una richiesta è corretta, ma non può essere soddisfatta immediatamente, si ponga in attesa la transazione, eventualmente rimuovendo dallo schedule tutte le richieste successive prodotte dalla transazione).
- Analizzare se il sistema al termine si trova in una situazione di deadlock. (4 punti per le due domande)

D. Nel protocollo 2PC, lo stesso nodo può svolgere il ruolo di TM e di RM, per transazioni diverse. Si ha quindi il seguente log, in cui si prevedono i record aggiuntivi p (prepare) e r (ready).

$Kcpt(), b(t_1), b(t_2), b(t_3), d(t_1, o1, b1), d(t_2, o2, b2), i(t_3, o3, a3), p(t_1), r(t_3), \text{guasto}$

Discutere l'uso delle tecniche di recovery in questa situazione. (4 punti)