

Compito di Complementi di Basi di Dati

1 settembre 2009

Esercizio 1:

Si supponga che per memorizzare i dati relativi ai teatri di una data città sia stata usata un'unica tabella *TEATRI* con la seguente struttura:

TEATRI(*NomeTeatro*, *Capienza*, *Indirizzo*, *Proprietario*, *Direttore*, *Dipendente*, *GiornoSettimana*, *CompensoGiornaliero*)

Sia dato il seguente insieme di requisiti. Ogni teatro sia identificato univocamente dal nome e possieda un proprietario, un direttore e un insieme di persone che vi lavorano (dipendenti). Di ogni teatro vengano, inoltre, memorizzati l'indirizzo e la capienza. Non vi siano teatri diversi con lo stesso indirizzo. Teatri diversi possano avere la medesima capienza. Proprietari, direttori e dipendenti siano identificati univocamente dai loro codici fiscali. Ogni proprietario possieda un unico teatro. Un direttore possa dirigere uno o più teatri. Ogni dipendente lavori per un unico teatro. Il compenso giornaliero di ogni dipendente possa variare da un giorno all'altro della settimana.

Con riferimento allo schema dato, si esegua quanto segue.

- (a) Determinare le dipendenze funzionali della relazione *TEATRI*, indicando, per ciascuna di esse, il requisito codificato.
- (b) Determinare le chiavi candidate e gli attributi primi e non primi di *TEATRI*.
- (c) Stabilire se *TEATRI* è o meno in 3NF (in caso di risposta negativa, con riferimento all'insieme determinato al punto (a), si determini il sottoinsieme delle dipendenze che violano la 3NF).
- (d) Stabilire se *TEATRI* è o meno in 2NF (in caso di risposta negativa, si indichino gli attributi che violano la 2NF e se ne illustri il perché).
- (e) Stabilire se *TEATRI* è o meno in BCNF (in caso di risposta negativa, con riferimento all'insieme determinato al punto (a), si determini il sottoinsieme delle dipendenze che violano la BCNF).
- (f) Nel caso in cui *TEATRI* non sia in 3NF, fornire una scomposizione lossless join in 3NF di *TEATRI* che conservi le dipendenze.

Esercizio 2:

Si determini se, e in caso affermativo come, il metodo 2PL previene le anomalie di perdita di aggiornamento, di lettura sporca e di aggiornamento fantasma.

Successivamente, si stabilisca se i seguenti schedule appartengono o meno a 2PL, 2PL stretto e TS.

1. $r_2(x), r_1(x), w_1(x), w_2(x);$
2. $r_2(x), r_1(y), w_2(x), w_1(x);$
3. $r_2(y), r_2(z), r_1(x), r_1(z), w_1(z), w_1(x), r_2(x);$
4. $w_1(y), r_2(x), w_1(x), r_3(y), r_0(x), w_0(x), w_2(y).$

Esercizio 3:

Si illustrino le informazioni/strutture dati utilizzate dal buffer manager per la gestione del buffer. Successivamente, si descrivano brevemente le primitive per il caricamento e lo scaricamento di pagine messe a disposizione dal buffer manager.

Esercizio 4:

Si consideri un disco con dimensione del blocco $B = 1024$ byte. Si consideri un B -albero che abbia come campo di ricerca un campo chiave non ordinante di dimensione $V = 10$ byte, dimensione dei puntatori ai record di dati $P_r = 6$ byte e dimensione dei puntatori ai blocchi $P = 5$ byte.

[4a] Si calcoli l'ordine p del B -albero che massimizza l'occupazione dei nodi.

[4b] Assumendo che ogni nodo del B -albero sia pieno al 70%, determinare il numero medio di entry (livello per livello e totale) di un B -albero a due livelli (più radice) e di un B -albero a tre livelli (più radice).

Si immagini ora di sostituire il B -albero con un B^+ -albero.

[4c] Si calcolino gli ordini p e p_{leaf} dei nodi interni e dei nodi foglia, rispettivamente, del B^+ -albero che massimizzano l'occupazione dei nodi.

[4d] Assumendo che ogni nodo del B^+ -albero sia pieno al 70%, determinare il numero medio di entry (livello per livello e totale) di un B^+ -albero a due livelli (più radice) e di un B^+ -albero a tre livelli (più radice).