

Compito di Complementi di Basi di Dati

1 luglio 2008

Esercizio 1:

Si supponga che per memorizzare i dati relativi alla gestione dei medici di base operanti sul territorio italiano sia stata usata un'unica tabella *MediciDiBase* con la seguente struttura:

MediciDiBase(*NomeMedico*, *CittàAmbulatorio*, *AnnoNascitaMedico*, *NomePaziente*,
CittàResidenza, *AnnoNascitaPaziente*, *AnnoAssegnamento*)

Sia dato il seguente insieme di requisiti. Ogni medico sia identificato univocamente dal nome e sia caratterizzato da un anno di nascita. Ogni medico possieda uno o più ambulatori (nel caso possieda più ambulatori, questi si trovino in città diverse). Si assuma che medici diversi possano ovviamente possedere ambulatori nella stessa città. Ogni paziente sia identificato univocamente dal nome e sia caratterizzato da una città di residenza e da un anno di nascita. Ogni paziente sia assegnato ad uno e un solo medico. Si tenga traccia dell'anno in cui un paziente viene assegnato ad un medico (si assuma, inoltre, che qualora un paziente venga assegnato ad un nuovo medico, l'assegnamento precedente decada automaticamente e venga, per tale ragione, rimosso).

- Determinare le dipendenze funzionali della relazione *MediciDiBase*, indicando, per ciascuna di esse, il requisito codificato.
- Determinare le chiavi candidate e gli attributi primi e non primi di *MediciDiBase*.
- Stabilire se *MediciDiBase* è o meno in BCNF.
- Stabilire se *MediciDiBase* è o meno in 2NF.
- Stabilire se *MediciDiBase* è o meno in 3NF.
- Nel caso in cui *MediciDiBase* non sia in 3NF, fornire una scomposizione lossless join in 3NF di *MediciDiBase* che conservi le dipendenze.

Esercizio 2:

Si stabilisca se i seguenti schedule appartengono o meno a VSR, CSR, TS, 2PL e 2PL stretto:

(s₁) $r_2(z), r_2(y), w_2(y), r_3(z), r_1(x), w_1(x), r_3(y), w_3(y), r_2(x), w_3(z), r_1(y), w_1(y), w_2(x)$;

(s₂) $r_3(z), r_1(x), w_1(x), r_3(y), w_3(z), r_2(z), w_3(y), r_1(y), w_1(y), r_2(y), w_2(y), r_2(x), w_2(x)$.

Esercizio 3:

Si descriva la ripresa a freddo quando, nella situazione determinata dal seguente input:

DUMP, *B*(*T1*), *I*(*T1*, *O1*, *A1*), *B*(*T2*), *B*(*T3*), *D*(*T2*, *O2*, *B2*), *C*(*T2*), *B*(*T4*), *U*(*T1*, *O4*, *B4*, *A4*),
U(*T4*, *O3*, *B3*, *A3*), *CK*(*T1*, *T3*, *T4*), *B*(*T5*), *U*(*T5*, *O5*, *B5*, *A5*), *A*(*T3*), *B*(*T6*), *CK*(*T1*, *T4*, *T5*, *T6*),
B(*T7*), *U*(*T7*, *O6*, *B6*, *A6*), *A*(*T4*), *U*(*T6*, *O3*, *B7*, *A7*), *B*(*T8*), *A*(*T7*),

si verifica un guasto generale di dispositivo.

Esercizio 4

Si consideri un file contenente 150000 record di dimensione prefissata pari a 100 byte, memorizzati in blocchi di dimensione pari a 1024 byte in modo unspanned.

Si determini la dimensione di un indice multilivello ottenuto a partire da un indice secondario denso costruito su un campo chiave non ordinante V di 9 byte, con dimensione del puntatore a blocco pari a 6 byte.

Successivamente, si determini la dimensione di un B-albero, con campo di ricerca il campo chiave V , puntatore ai dati di dimensione pari a 7 byte e puntatore ausiliario di dimensione pari a 6 byte, assumendo che ciascun nodo del B-albero sia pieno al 70