

# Compito di Complementi di Basi di Dati

28 marzo 2006

## Esercizio 1:

Si consideri il seguente insieme di requisiti che caratterizza un sistema per la gestione degli esami degli studenti di una facoltà universitaria.

Ogni studente presenta un piano di studi. Tale piano di studi può essere approvato così come proposto dallo studente o può essere restituito allo studente con l'indicazione di un insieme di modifiche richieste. Una volta apportate le modifiche, lo studente ripresenta il piano di studi. I piani di studio approvati vengono memorizzati nell'archivio *PianiDiStudio*. Successivamente, lo studente può chiedere di apportare delle modifiche al suo piano di studi; tali modifiche possono essere approvate o rifiutate. Nel caso in cui vengano approvate, il sistema provvede ad aggiornare il piano di studi.

Ogni docente immette nel sistema i risultati degli esami relativi ai corsi da lui tenuti. Il sistema controlla i dati comunicati dal docente e, in base all'esito di tale controllo, decide se accettare o rifiutare l'inserimento. In dettaglio, il sistema verifica la consistenza dei dati relativi allo studente (nome e cognome, numero di matricola) consultando l'archivio *Studenti*; inoltre, si accerta che l'esame sostenuto dallo studente faccia parte del suo piano di studi e che il corso faccia parte dell'insieme dei corsi tenuti dal docente (la corrispondenza tra corsi e docenti è mantenuta nell'archivio *DocentiCorsi*). I dati degli inserimenti accettati vengono registrati nell'archivio *Esami*.

Gli studenti possono richiedere dei certificati che attestino gli esami da loro sostenuti. Il sistema produce i certificati richiesti, in cui vengono riportati gli estremi degli studenti (nome e cognome, matricola) e il dettaglio degli esami sostenuti (corso, voto, data), e li consegna agli studenti.

Si costruisca il diagramma di contesto del sistema di gestione degli esami e, successivamente, lo si trasformi in un diagramma di flusso. In particolare, si mettano in evidenza i diversi processi di sistema, i depositi dei dati, i flussi di dati fra le interfacce e il sistema (con l'indicazione esplicita del loro contenuto) e i flussi fra i depositi dei dati e i processi di sistema.

## Esercizio 2:

Si supponga che per memorizzare i dati relativi alla gestione dei corsi offerti da un ente di formazione professionale sia stata usata un'unica tabella *ENTE* con la seguente struttura:

*ENTE(Corso, TipologiaCorso, Docente, Allievo, IndirizzoDocente, TelefonoStudente, QuotaVersata)*

Sia dato il seguente insieme di requisiti. Ogni corso abbia un codice, che lo identifica univocamente, un docente e alcuni allievi. Di ogni corso venga, inoltre, specificata la tipologia (corso di base, corso avanzato, corso di laboratorio, ..). Un docente possa tenere più corsi, anche di tipologie diverse. Un allievo possa essere iscritto a più corsi. Ogni docente sia caratterizzato da un nome, che lo identifica univocamente, e da un indirizzo. Ogni allievo sia caratterizzato da un nome, che lo identifica univocamente, e da un numero di telefono. Di ogni allievo si voglia sapere, per ogni corso che frequenta, quanto ha già versato.

- Determinare le dipendenze funzionali della relazione *ENTE*.
- Determinare le chiavi candidate e gli attributi primi e non primi di *ENTE*.

- (c) Stabilire se *ENTE* è o meno in 2NF.
- (d) Stabilire se *ENTE* è o meno in 3NF.
- (e) Stabilire se *ENTE* è o meno in BCNF.
- (f) Nel caso in cui *ENTE* non sia in 3NF, fornire una scomposizione lossless join in 3NF di *ENTE* che conservi le dipendenze.
- (g) (facoltativo) Stabilire se la scomposizione di cui al punto precedente è o meno in BCNF?

### Esercizio 3:

Si stabilisca se i seguenti schedule appartengono o meno a VSR, CSR, 2PL, 2PL stretto e TS. Nel caso di schedule appartenenti a VSR, si forniscano tutti gli schedule seriali ad essi equivalenti; lo stesso nel caso di schedule appartenenti a CSR.

1.  $s_1 : r_1(y), w_1(x), r_3(x), w_4(y), w_1(z), w_5(y), r_3(t), r_2(t), w_2(t), w_5(t), r_5(z), r_4(z);$
2.  $s_2 : r_1(y), w_2(y), r_2(y), r_3(y), r_4(x), w_1(y), r_3(z), r_3(y), w_1(z), w_5(y), w_1(x), r_5(x), r_5(z);$
3.  $s_3 : r_3(t), r_1(z), w_2(t), w_1(x), r_4(t), r_1(y), r_3(y), r_4(y), w_4(y), w_3(x), w_4(x), w_2(z).$

### Esercizio 4:

Si descriva l'algoritmo di inserimento in un *B*-albero. Successivamente, dato il seguente insieme di chiavi:

23, 20, 14, 22, 5, 4, 7, 18, 16, 9, 17, 3, 12, 30, 2, 27, 1

mostrare il *B*-albero di ordine  $p = 4$  ottenuto inserendo un elemento dopo l'altro nell'ordine dato.