Università degli Studi di Bologna
Corso di Laurea in Informatica
Basi di Dati e Sistemi Informativi - 20 Settembre 1999

Esercizio 1

La Biblioteca Universale di Alexandria (BUA) ha bandito una gara per il progetto del proprio database.

Per ogni copia di un testo deve essere riportata la collocazione e la disponibilità attuale. Le copie di pregio sono disponibili alla sola consultazione in loco e previa autorizzazione (la richiesta di autorizzazione va presentata con sufficiente anticipo, data la numerosità delle richieste stesse).

Una copia non di pregio può essere presa in prestito o consultata in loco, presso una delle sale di lettura della BUA. In entrambi i casi bisogna registrarsi (se non già fatto in altre occasioni) presentandosi con due documenti di identità e con una persona già registrata da almeno un anno che funga da garante.

Una copia può essere presa in prestito per un numero di giorni non superiore al limite stabilito per la copia stessa. Se presa in consultazione, una copia può essere tenuta tutta il giorno. Per le copie di pregio questo limite può essere esteso fino a tre giorni, a condizione che sia stato specificato all'atto della richiesta.

Per ogni testo, oltre al codice assegnato dalla BUA, deve essere riportato il titolo, l'anno di edizione (se noto), l'editore (se noto) e l'elenco degli autori. Si noti che testi con lo stesso titolo ma con editori diversi sono considerati distinti a tutti gli effetti.

Si disegni uno schema concettuale E/R per il database della BUA, in cui si mantenga traccia di tutti i prestiti e le consultazioni, passate e presenti.

Esercizio 2

Si dimostri che per produrre una copertura minima di un insieme di dipendenze funzionali $\mathcal{F}$ un algoritmo che elimini *prima* le dipendenze ridondanti e *poi* gli attributi estranei non è corretto.

Suggerimento: si consideri che una FD può non essere inizialmente ridondante, ma diventarlo dopo che gli attributi estranei sono stati rimossi.

Esercizio 3

Dato il DB formato dagli schemi

`Movies(CodM, Title, Director, Year)`
`Actors(CodA, Name, Gender,BirthYear)`
`Casts(CodM, CodA, Role)`

si esprimano in SQL le interrogazioni che determinano:

1. I nomi degli attori che hanno interpretato più ruoli in uno stesso film;
2. Per ogni regista (`Director`) l'attrice (`Gender = 'F'`) preferita, ovvero l'attrice che ha interpretato più film con tale regista (senza considerare gli eventuali ruoli molteplici interpretati in uno stesso film);
3. Gli attori che hanno interpretato il loro primo film a meno di 20 anni.

Esercizio 4

Si consideri la seguente query in SQL "esteso", in cui la clausola `Stop After` serve a produrre in output solo un numero limitato di tuple (50, nell'esempio), rispettando il criterio di ordinamento (in pratica la query restituisce informazioni sui 50 dipendenti più pagati):

```
Select     Emp.*, D.NameD
From       Emp E, Dept D
Where      E.WorksIn = D.CodD
Order by   E.Salary DESC
Stop After 50 ;
```

Sapendo che `E.WorksIn` è una *foreign key*, ovvero che un dipendente lavora in uno e un solo dipartimento, e che il DBMS mette a disposizione, tra gli altri, i seguenti operatori:

SeqScan(F): implementa uno scan sequenziale applicando la condizione F

Filter(X;F): applica alla tuple in input la condizione F e le proietta sui campi X

NestedLoop(F): esegue il join secondo il metodo nested-loop applicando la condizione F

Sort(X,N): ordina le tuple in input sulla base dei valori di X e restituisce solo le prime N

1. si disegni un piano di accesso "ottimizzato" per la query, e si commenti sulla sua convenienza;
2. se la clausola `Where` fosse

   ```
   Where E.WorksIn = D.CodD
   And   D.Type = 'research'
   ```

   lo stesso piano di accesso sarebbe ancora applicabile? Si giustifichi la risposta.