Corso di Laurea in Informatica
Basi di Dati e Sistemi Informativi – 17 Luglio 1998

Esercizio 1

La TseTse (Tutto Subito E Tutto Sempre Esatto) è un'associazione (anonima!) che svolge progetti di Basi di Dati per gli studenti dei Corsi di Laurea (CdL) in Informatica, percependo un compenso fisso per ogni CdL più un ulteriore compenso integrativo che dipende dalla valutazione finale del progetto, rapportata ai criteri adottati dal docente titolare del corso nell'anno relativo (ogni anno in un CdL è attivo un solo corso di Basi di Dati).

Ogni progetto, il cui titolo deve essere univoco all'interno di un dato CdL, consiste di una relazione e del codice sorgente, in uno dei linguaggi stabiliti anno per anno nel corso (es: C e Java per il corso di Bologna del 1997-98), ed è sviluppato per un DBMS tra quelli previsti dal corso stesso (es: PostgreSQL e DB2 a Bologna nel 1997-98).

La relazione si compone di una serie di capitoli, ognuno relativo ad una fase della metodologia adottata (la stessa metodologia può essere adottata in più corsi e per più anni). Diverse metodologie possono avere fasi in comune, ma differire per come le numerano (es: la fase 1 della metodologia A corrisponde alla fase 2 della metodologia B) o le suddividono (es: la fase 1 della metodologia A corrisponde alle fasi 1 e 2 della metodologia B).

La TseTse, al fine di ridurre i tempi di consegna (il limite massimo viene specificato all'atto della commissione del progetto presso una delle sedi TseTse), sviluppa alcuni progetti apportando delle modifiche a partire da un progetto preesistente. Per questi "progetti delta" il compenso fisso viene ridotto del 50%.

Si disegni uno schema concettuale E/R per il database della TseTse.

La TseTse si scusa con i docenti, ma ritiene che i progetti siano troppo complessi per poter essere svolti dai soli studenti nei limiti di tempo imposti!

Esercizio 2

Dato lo schema

$$\langle R(ABCDEL), \mathcal{F} = \{AB \to CDE, BC \to D, C \to EL, E \to D\}\rangle$$

se ne fornisca una decomposizione in 3NF che preservi i dati e le dipendenze.

Sullo schema ottenuto si esprima in SQL l'interrogazione che determina "per ogni valore di A, non associato al valore b di B, la somma dei valori di D".

Esercizio 3

L'operatore di *ridenominazione*, ρ, dell'algebra relazionale permette di cambiare i nomi degli attributi, il che si può rendere necessario se si usa il join naturale in luogo del θ-join. Ad esempio, data una relazione con schema $R(ABC)$, per eseguire il *self-join* sul solo attributo C si può scrivere:

$$R \bowtie \rho_{A'B' \leftarrow AB}(R)$$

che produce una relazione con schema $ABCA'B'$.

Si consideri un database formato dalle relazioni $R_1(\underline{A}BC), R_2(\underline{B}D), R_3(\underline{C}D)$. Un valore di A "determina" in R_2 e R_3 due valori, distinti o meno, di D (nell'istanza in figura a_1 determina univocamente d_1, mentre a_2 determina d_1 in R_2 e d_2 in R_3).

Usando i soli operatori di proiezione, ridenominazione e join naturale, si scrivano le espressioni che calcolano:

1. I valori di A che determinano in R_2 e R_3 un solo valore di D (nell'esempio: $\{(a_1)\}$, che determina d_1).
2. Le coppie di valori di D (distinti o meno) determinate in R_2 e R_3 da uno stesso valore di A (nell'esempio: $\{(d_1, d_1), (d_1, d_2), (d_3, d_1)\}$, determinate rispettivamente da a_1, a_2 e a_3).
3. Le coppie di valori di A (distinti o meno) che determinano in R_2 e R_3 uno stesso valore di D (nell'esempio: $\{(a_1, a_1), (a_2, a_1), (a_1, a_3), (a_2, a_3)\}$, che determinano d_1).

A	B	C
a_1	b_1	c_1
a_2	b_1	c_2
a_3	b_2	c_1

B	D
b_1	d_1
b_2	d_3

C	D
c_1	d_1
c_2	d_2

Esercizio 4

Data una relazione con schema $R(\underline{A}BC)$, e un indice unclustered $IX(R.B)$ di tipo B^+-tree, si valutino i costi associati alla seguente strategia di esecuzione:

1. Accesso sequenziale a R e selezione delle sole tuple per cui $C = c$;
2. Esecuzione di un self-equi-join su B, usando l'indice e il metodo nested-loops.

Si scriva poi una query SQL da cui tale strategia è derivabile.