Università degli Studi di Bologna

Corso di Laurea in Informatica

Basi di Dati e Sistemi Informativi - 17 Giugno 1999

Esercizio 1

Il sistema elettorale e legislativo dell'Isola di Random è estremamente semplice. Non esistono partiti, e ogni elettore può anche essere votato (ed eventualmente eletto) a condizione che sia stato preventivamente sorteggiato (non sono ammessi rifiuti).

Ad ogni elezione si stabilisce, in maniera casuale, il numero di abitanti che saranno eletti e ad ogni elettore viene assegnato, sempre in maniera casuale, un numero massimo di preferenze che può esprimere. All'atto della votazione l'elettore riceve una scheda che permette di esprimere un numero massimo di preferenze pari a quelle assegnategli. In fase di scrutinio delle schede si totalizzano i voti ricevuti da ciascun candidato e, se il numero di voti ottenuti è primo, il totale si incrementa di una percentuale casuale, comunque non superiore al 10%.

I candidati eletti vengono successivamente aggregati, casualmente, in gruppi, il cui numero è pari a 5 (qualche costituzionalista sostiene da tempo che tale numero dovrebbe essere scelto casualmente!). Solo i gruppi possono presentare proposte di legge, mentre l'attività legislativa dei singoli eletti è limitata alla formulazione di emendamenti alle proposte di legge (del proprio gruppo o di altri). Ogni proposta (eventualmente emendata) viene votata e può essere approvata (se ottiene la maggioranza dei voti) o respinta.

Una proposta approvata deve essere poi sottoposta a un costituzionalista (scelto casualmente) per la ratifica. In caso di esito positivo la legge diventa operativa, altrimenti deve essere emendata e poi votata nella nuova forma.

Si disegni uno schema concettuale E/R per il database delle elezioni dell'Isola di Random, che garantisca l'anonimato delle votazioni.

Esercizio 2

Dato l'insieme di dipendenze funzionali

$$\mathcal{F} = \{A \to B, B \to C, BD \to A, CD \to A\}$$

si produca una copertura minima di $\mathcal{F}$. Si inidividuino inoltre le chiavi dello schema $R(ABCD)$ e si specifichi il livello di normalizzazione, motivando la risposta.

Esercizio 3

Dato il DB formato dagli schemi

`Computer(CodC, Marca, Modello)`
`PacchettiSW(CodC, CodSW, Installatore, DataInstall)`
`Installatori(Marca, Installatore)`

si esprimano in SQL le interrogazioni che determinano:

1. Il numero di pacchetti software installati da `'Bowman'` su modelli `'9000'` nel 2001;
2. Per ogni modello della marca `'HAL'`, il pacchetto con più installazioni;
3. Gli installatori della marca `'HAL'` che non hanno mai eseguito alcuna installazione su computer di tale marca e modello `'9000'`.

Esercizio 4

Si consideri la seguente variante dell'algoritmo di join *nested-loops*, in cui R è la relazione esterna e S quella interna:

> Vengono riservate B pagine di buffer per S e 1 sola pagina per R. Le B pagine di buffer di S vengono gestite con una strategia LRU (Least Recently Used), che prevede, a seguito di un *page fault*, di rimpiazzare la pagina usata meno di recente. Per ogni tupla residua di R viene eseguito uno scan di S. Il primo scan è di tipo usuale (*forward*), e pertanto procede "in avanti", mentre il secondo (*backward*) procede "a ritroso". In generale, gli scan dispari sono forward e quelli pari sono backward.

Supponendo che sia l'accesso a R che a S sia basato su scan sequenziale, e che su R non siano definiti predicati locali, si stimi il costo (numero di operazioni di I/O) di esecuzione del join secondo la variante descritta.