Corso di Laurea in Informatica
Basi di Dati e Sistemi Informativi – 19 Luglio 1997

Esercizio 1

`YAWN` (Yet Another World-wide Network) è un'organizzazione che agisce come *broker* su Internet, gestendo un *meta-database* (database di database). Un utente di `YAWN` è un *provider* o un *client* (o entrambi) ed è noto al resto degli utenti tramite un acronimo univoco.

Ogni provider fornisce a `YAWN` lo schema – relazioni, attributi, chiavi (primarie, alternate e importate), domini degli attributi – di uno o più DB relazionali, e per ogni relazione la sua cardinalità. Tutti i DB di un provider sono accessibili in modo uniforme tramite un indirizzo formato dalla coppia (ProviderID,DBName), a cui `YAWN` fa corrispondere un indirizzo fisico (HostID,PortID). Un provider può fare uso di più host (non viceversa), per i quali fornisce le caratteristiche hardware, e più DB possono essere accessibili sulla stessa porta di uno stesso host (ma non viceversa).

Ogni client esegue ricerche sul meta-database di `YAWN`, allo scopo di trovare DB che, per il loro contenuto, possono interessargli. Per facilitare tale ricerca, `YAWN` assegna ad ogni DB alcune parole chiave appartenenti a un vocabolario controllato.

Quando un client trova un DB di interesse, `YAWN` attiva la fase di tariffazione, tenendo traccia della operazioni (query) del client. Un client paga un costo fisso di connessione (che dipende dal DB scelto) e un costo variabile che dipende dal numero di tuple cui il provider deve accedere per produrre il risultato di una query. In particolare, ogni relazione ha un costo base per singola tupla.

Si disegni uno schema concettuale E/R per il meta-database di `YAWN`.

`YAWN` non è responsabile dei criteri di ottimizzazione adottati dai singoli provider, che possono far crescere a dismisura i costi monetari pagati dai client per l'esecuzione delle loro richieste.

Esercizio 2

Date le dipendenze

$$\mathcal{F} = \{A \rightarrow BD, AC \rightarrow DE, D \rightarrow F\}$$

1. Si determini una copertura minima, $\mathcal{G}$, di $\mathcal{F}$.
2. Si disegni uno schema E/R che rappresenti $\mathcal{G}$.
3. Si modifichi infine lo schema E/R derivato in modo da rappresentare *direttamente* $\mathcal{F}$ e si commenti brevemente.

Esercizio 3

Si consideri il DB formato dagli schemi

`Afferenze(Ricercatore, Dipartimento)`
`Finanziamenti(Ricercatore, Progetto, Importo, Anno, Residuo)`

di cui la figura mostra un'istanza

Ricercatore	Dipartimento
Rossi	DM
Verdi	DM
Bianchi	CS
Neri	CS

Ricercatore	Progetto	Importo	Anno	Residuo
Rossi	P1	200	1996	100
Rossi	P2	150	1997	0
Verdi	P1	160	1996	100
Bianchi	P3	100	1995	20
Bianchi	P2	80	1996	10
Neri	P4	250	1997	200

Si esprimano in SQL le seguenti interrogazioni

1. Determinare i progetti che hanno finanziato ricercatori del dipartimento `'d'` nel 1996.
2. Determinare i ricercatori del dipartimento `'d'` che hanno speso almeno l'80% del totale dei finanziamenti ricevuti fino a tutto il 1996.
3. Determinare il totale dei finanziamenti ricevuti da ogni dipartimento, senza includere quelli erogati a ricercatori che hanno complessivamente ricevuto meno di `'tot_min'` lire.

Esercizio 4

Date due relazioni con schemi `R(A,B,C,...)`, `S(B,D,E,...)` e la query SQL

```
Select *
From   R
Where  R.B in ( Select S.B
                From   S
                Where  S.D > 10 );
```

si supponga che, tra le strategie considerate dall'ottimizzatore, vi sia quella caratterizzata dai seguenti passi di esecuzione:

1. Accedi a `S` e recupera le tuple che soddisfano il predicato `S.D > 10`.
2. Per ogni tupla ottenuta, estrai il valore di `S.B` e con questo accedi a `R` facendo uso di un indice unclustered costruito sull'attributo `R.B`.

Si valuti il costo stimato di esecuzione del passo 2, nell'ipotesi di conoscere, oltre alle informazioni di catalogo, anche $ET_{\mathtt{S}}$, numero di tuple residue di `S`, e considerando che i valori di `R.B` sono un sottoinsieme di quelli di `S.B`.