Corso di Laurea in Informatica
Basi di Dati e Sistemi Informativi – 16 Giugno 1997

Esercizio 1

La *Touristic Time Travels* (T^3) organizza viaggi nel tempo (passato) per singoli o gruppi (di non più di 5 persone). Se il viaggio è di gruppo, una persona del gruppo funge da responsabile. Ogni viaggiatore della T^3 deve mostrare di aver superato uno o più test di *time lag*. Un test autorizza viaggi non anteriori a un dato secolo (ad es. il test base TL1 permette di viaggiare solo nel XX secolo, ma già il test TL2 permette viaggi sino al XV secolo).

I programmi di viaggio specificano il secolo prescelto e il luogo, quest'ultimo espresso in coordinate spaziali (latitudine, longitudine). Se il viaggio è con ritorno, la T^3 garantisce il rientro con un ritardo massimo di 1 giorno.

Viaggi di sola andata, possibili solo per singoli, possono comunque essere interrotti, gratuitamente, se il viaggiatore riesce a trovare qualcuno con il biglietto di ritorno, comunicando il nome della persona con biglietto (nome del responsabile se si tratta di un gruppo) alla sede della T^3.

La tariffa di un viaggio è calcolata dalla T^3 a partire da un importo base, che dipende dal secolo prescelto e varia per singoli e gruppi.

Si disegni uno schema concettuale E/R per il database della T^3, che gestisca le informazioni sui soli viaggiatori attuali.

La T^3 declina ogni responsabilità sulle effettive condizioni di abitabilità del luogo prescelto, e ricorda ai suoi viaggiatori di portare con sé il T^3 *survival kit* per i casi di emergenza.

Esercizio 2

Si consideri lo schema

$$\langle \texttt{Sedi(Città, Settore, Responsabile)}, F = \{R \to C, CS \to R\}\rangle$$

(un `Responsabile` è responsabile di uno o più settori di una stessa città) di cui la figura mostra un'istanza legale.

Città	Settore	Responsabile
Bologna	Vendite	Rossi
Bologna	Amministrazione	Rossi
Bologna	Produzione	Verdi
Roma	Vendite	Bianchi

Data la decomposizione formata dagli schemi

`Resp_Citta(Responsabile, Città)`
`Resp_Settori(Responsabile, Settore)`

si determini, facendo uso di statement SQL, se è possibile inserire nella relazione `Resp_Settori` una tupla `('r','s')` senza violare i vincoli definiti dalle dipendenze in F.

Esercizio 3

Dati i seguenti schemi di relazioni

$$R_1(\underline{K_1}, X_1), R_2(\underline{K_2}, X_2), R_3(\underline{K_3}, X_3), R_4(\underline{K_1, K_2}, K_3, X_4)$$

si disegni uno schema E/R da cui tali relazioni siano derivabili, e si commenti brevemente.

Esercizio 4

L'ottimizzatore di un DBMS adotta un modello di costo semplificato in cui il costo di un piano di accesso è determinato dalla *somma delle cardinalità dei risultati parziali* che il piano stesso genera (la cardinalità del risultato finale non è influente perché indipendente dal piano specifico).

Ad esempio, data la query $R_1 \bowtie R_2$, se si considera il piano che accede prima a R_1 e poi a R_2, il costo è stimato pari a:

$$C(R_1, R_2) = NT_{R_1}$$

Viceversa, se si considera R_2 come relazione esterna, il costo si esprime come

$$C(R_2, R_1) = NT_{R_2}$$

Data la query $R_1 \bowtie R_2 \bowtie R_3$, in cui:

- Il join tra R_1 e R_2, è del tipo primary key - foreign key (primary key in R_1).
- Il join tra R_2 e R_3, è del tipo primary key - foreign key (primary key in R_3).
- R_1 e R_3 non hanno attributi in comune (il join è un prodotto Cartesiano)

si enumerino i costi dei possibili piani di accesso, e si dica sotto quali condizioni un piano che esegue il prodotto Cartesiano tra R_1 e R_3 può risultare ottimale.

Esercizio 5 (facoltativo)

Con riferimento alla decomposizione dell'esercizio 2, si esprima in SQL l'interrogazione che restituisce tutte le sedi (`Città`) in cui non ci sono responsabili di più settori (ad es.: data l'istanza in figura, il risultato include Roma, ma non Bologna).