Università degli Studi di Bologna

Corso di Laurea in Informatica

Basi di Dati e Sistemi Informativi - 22 Settembre 2000

Esercizio 1

La DoReMi (Documenting Relational Model Instances) fornisce una documentazione esauriente dei vostri database relazionali in forma di schema E/R. Più precisamente, per ogni db che la DoReMi tratta, vengono mantenute informazioni sul creatore, sulla data di creazione, sul sistema (HW e SW) su cui è installato (il SW comprende sia il DBMS che il sistema operativo, l'HW include informazioni sull'host, quali indirizzo IP, memoria RAM e dischi usati) e sul nome del db. Un DBMS è caratterizzato da modello e versione (es.: UDB 7.0) e ha un produttore univoco (es.: IBM). Si noti che su un host possono essere installati più DBMS e anche più copie (dette *istanze*) dello stesso DBMS. Il nome di un db è univoco solo considerando i db gestiti da un'istanza di DBMS su un dato host.

Per ogni db la DoReMi mantiene anche informazioni (nome, creatore, data di creazione, attributi, tipo degli attributi) per le relazioni esistenti e per gli indici costruiti sui vari attributi. Si ricorda che in generale un indice può esser costruito su più attributi, e che in tal caso l'ordine è rilevante (es.: un indice su `(Cognome, Città)` è diverso da un indice su `(Città, Cognome)`). Tutti gli indici costruiti su un db hanno nomi distinti.

Infine, la DoReMi mantiene statistiche aggiornate sui volumi di dati gestiti (n. tuple e pagine delle relazioni, n. valori distinti per gli attributi, n. livelli degli indici e, solo nel caso di indici multi-attributo, n. di valori di chiave distinti indicizzati).

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

Esercizio 2

Dato lo schema $R(ABCDEG)$ e l'insieme di dipendenze funzionali:

$$\mathcal{F} = \{A \rightarrow BDE, BD \rightarrow DG, E \rightarrow BDG\}$$

1. si produca una copertura minima di $\mathcal{F}$;
2. si operi una decomposizione di R in 3NF che preservi i dati e le dipendenze.

Esercizio 3

Dato il DB formato dagli schemi

`Voli(IdVolo, Partenza, OraP, Arrivo, OraA, Compagnia)`
`Prenotazione(IdPren, IdVolo, Data, Nome, Cognome, Classe)`
`Aeroporti(IdAP, Citta, Nazione)`

in cui i campi `Partenza` e `Arrivo` sono codici di aeroporti, `Data` è la data di partenza del volo prenotato, e considerando che una città può avere più aeroporti, si esprimano in SQL le interrogazioni che determinano:

1. La compagnia che con un volo diretto impiega meno tempo da `'FCO'` (Roma Fiumicino) a `'JFK'` (J.F. Kennedy a New York);
2. Il numero dei passeggeri in classe `'economica'` per ogni volo in partenza da Roma il 15 Agosto 2000 dopo le 20:00 per un aeroporto degli USA;
3. I collegamenti (coppie di voli) da Bologna (`'BLQ'`) a Sydney (`'SYD'`) basati su 2 voli in cui tra l'arrivo del primo e la partenza del secondo ci siano almeno 2 e non più di 5 ore.

Per semplicità si assuma che i tempi di percorrenza/attesa siano calcolabili mediante semplici differenze (ad es.: `OraA - OraP`).

Esercizio 4

Si esprimano i costi di accesso a indice e dati per l'esecuzione di un Merging-Scans join quando la relazione esterna R è ordinata sul campo di join $R.J$ (chiave primaria di R) e alla relazione interna S si accede mediante un indice secondario unclustered costruito sul campo $S.J$.