

# Sistemi di gestione di basi di dati

7 febbraio 2011

1. (6 punti) Sono date le relazioni seguenti (le chiavi primarie sono sottolineate):

```
CANTANTE(CodC, NomeC, Città, DataNascita)
CASA_DISCOGRAFICA(CodCD, NomeCD, Indirizzo, Città)
DISCO(CodD, Titolo, CodC, CodCD, Tipo, Prezzo)
VENDITE(CodD, Data, NumeroCopieVendute)
```

Si ipotizzino le seguenti cardinalità per le tabelle:

- $\text{card}(\text{CANTANTE}) = 10^4$  tuple,  
 $\text{MIN}(\text{DataNascita}) = 1-1-1969$ ,  $\text{MAX}(\text{DataNascita}) = 31-12-1998$ ,
- $\text{card}(\text{CASA\_DISCOGRAFICA}) = 10^4$  tuple,  
numero di Città  $\simeq 100$ ,
- $\text{card}(\text{DISCO}) = 10^6$  tuple,  
 $\text{MIN}(\text{Prezzo}) = 10$ ,  $\text{MAX}(\text{Prezzo}) = 29$ ,
- $\text{card}(\text{VENDITE}) = 10^8$  tuple per l'anno 2010.

Inoltre si ipotizzi il seguente fattore di riduzione per la condizione di group by:

- $\text{having sum}(\text{NumeroCopieVendute}) \geq 10.000 \simeq \frac{1}{100}$ .

Si consideri la seguente query SQL:

```
select NomeCD, Città
from CASA_DISCOGRAFICA CD
where Città='Roma' and CodCD IN (select D.CodCD
                                from VENDITE V, DISCO D, CANTANTE C
                                where V.CodD=D.CodD and C.CodC=D.CodC
                                D.Prezzo ≤ 11 and C.DataNascita ≤ 31/12/1983
                                and V.Data ≤ 31/01/2010
                                group by V.CodD, D.CodCD
                                having sum(NumeroCopieVendute) ≥ 10.000)
```

Per l'interrogazione SQL

- Si scriva l'espressione algebrica corrispondente, indicando le operazioni svolte, la cardinalità e la selettività di ogni operazione. Dove necessario, si ipotizzi la distribuzione dei dati. Discutere la possibilità di anticipare l'operatore GROUP BY.
- Si scelgano le strutture fisiche accessorie per migliorare le prestazioni dell'interrogazione. Si motivi la scelta e si definisca il piano di esecuzione (ordine e tipo dei join, accesso alle tabelle e/o indici, etc.).

2. (7 Punti) Sono date le relazioni seguenti (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con \*).

EVENTO(CodE, NomeEvento, CategoriaEvento, CostoEvento, DurataEvento)

CALENDARIO\_EVENTI(CodE, Data, OraInizio, Luogo)

SOMMARIO\_CATEGORIA(CategoriaEvento, Data, NumeroTotaleEventi, CostoComplessivoEventi)

Si vuole gestire la pianificazione degli eventi nella città di Torino per l'anniversario dei 150 anni dell'unità d'Italia (Italia 150). Gli eventi appartengono a diverse categorie (attributo `CategoriaEvento`), quali mostre, dibattiti, proiezioni, e sono caratterizzati da un costo di realizzazione (attributo `CostoEvento`). Ciascun evento può essere ripetuto più volte in date diverse. La tabella `CALENDARIO_EVENTI` riporta la pianificazione degli eventi in diversi giorni e luoghi della città. Si scrivano i trigger per gestire le seguenti attività.

(1) *Aggiornamento della tabella SOMMARIO\_CATEGORIA.* La tabella `SOMMARIO_CATEGORIA` riporta, per ogni categoria di evento e per ogni data, il numero complessivo di eventi previsti e il costo complessivo per la loro realizzazione. Si scriva il trigger per propagare le modifiche alla tabella `SOMMARIO_CATEGORIA` quando viene inserito un nuovo evento a calendario (inserimento nella tabella `CALENDARIO_EVENTI`).

(2) *Vincolo di integrità sul costo massimo dell'evento.* Il costo di un evento della categoria proiezione cinematografica (attributo `CategoriaEvento`) non può essere superiore a 1500 euro. Se un valore di costo superiore a 1500 è inserito nella tabella `EVENTO`, all'attributo `CostoEvento` deve essere assegnato il valore 1500. Si scriva il trigger per la gestione del vincolo di integrità.

### 3. Progettazione Data Warehouse

La Regione Piemonte è interessata ad analizzare i ricoveri e l'utilizzo delle strutture ospedaliere per identificare le strutture che richiedono investimenti o riduzione del budget. Per analizzare queste caratteristiche, un opportuno data warehouse è stato commissionato.

Ogni ospedale è suddiviso in diversi reparti (es. medicina interna, pronto soccorso, cardiologia, ecc.). Per ognuno di questi reparti, è noto il numero di posti letto disponibili per ogni stanza. Ogni posto letto può essere equipaggiato con diverse attrezzature (es., ossigeno, materasso ortopedico, doghe reclinabili motorizzate, ecc.).

La Regione Piemonte è interessata nell'analizzare la percentuale di posti letto disponibili in funzione di:

- data, mese, trimestre, semestre e anno;
- giorno dell'anno (da 1 a 366), giorno della settimana (lunedì-domenica), giorno del mese (1-31);
- reparto, ospedale, distretto, città e provincia;
- configurazione di attrezzature del posto letto (e.g., ossigeno, materasso ortopedico, doghe reclinabili motorizzate, ecc.).

Sono riportate di seguito alcune delle interrogazioni frequenti di interesse per la regione:

- (a) Considerando solo l'anno 2010, per ogni ospedale e per ogni mese, calcolare la percentuale di posti letto disponibili. Assegnare un rank ai risultati (la posizione 1 corrisponde al valore percentuale più alto).
- (b) Calcolare il numero totale di letti disponibili in ciascun giorno della settimana e la percentuale di letti disponibili in ciascun giorno della settimana rispetto al totale (di tutti i giorni della settimana), separatamente per ogni configurazione di attrezzature. Assegnare un rank alle coppie (configurazione attrezzature, giorno della settimana) per numero di letti disponibili in ordine decrescente.
- (c) Considerando solo i posti letto che hanno l'ossigeno e il materasso ortopedico, calcolare il numero totale di posti disponibili per ogni ospedale e per ogni mese.

#### *Progettazione*

Nel data warehouse saranno contenuti i dati relativi agli anni 2006-2010. Sono inoltre note le seguenti statistiche (le informazioni ritenute necessarie ma non presenti in questa lista possono essere ipotizzate e stimate dal candidato):

- Reparti per ogni ospedale:  $\sim 20$
  - Ospedali:  $\sim 100$
  - Città:  $\sim 30$
  - Numero di attrezzature distinte per i posti letto: 10
- (a) (7 Punti) Progettare il data warehouse in modo da soddisfare le richieste descritte nelle specifiche del problema. Il data warehouse progettato deve inoltre permettere di rispondere in modo efficiente a tutte le interrogazioni frequenti indicate.
  - (b) (5 Punti) Esprimere l'interrogazione frequente (a) utilizzando il linguaggio SQL esteso.
  - (c) (*Opzionale*: 4 Punti) Esprimere l'interrogazione frequente (b) utilizzando il linguaggio SQL esteso.