

# Tecnologia delle basi di dati (ex Basi di dati, primo modulo)

## 14 giugno 2006 — Compito A

**Tempo a disposizione:** 2 ore e 15 minuti. **Nota:** è richiesta una “bella copia” comprensibile e ordinata.

**Domanda 1** (20%) Sintetizzare in meno di una pagina (circa 300 parole) i risultati ottenuti in una delle parti del progetto (scegliendo liberamente fra le tre svolte). Nota bene: la risposta a questa domanda sarà valutata anche sulla base della chiarezza espositiva ed è quindi indispensabile redigerla con cura.

**Domanda 2** (20%) Alcuni DBMS prevedono la possibilità di includere in un indice i valori di altri attributi delle ennuple, oltre a quelli degli attributi su cui l'indice è realizzato. Ad esempio, una istruzione del tipo

```
CREATE INDEX contoCorrenteIX ON contoCorrente (numero) INCLUDE (saldo)
```

crea un indice sulla relazione `contoCorrente` includendo nelle foglie dell'indice, oltre ai valori di `numero` (su cui l'indice è realizzato), anche quelli di `saldo`. Confrontare (con riferimento ad esempio a specifiche operazioni di ricerca e di aggiornamento che potrebbero essere avvantaggiate o penalizzate) questa soluzione con le due seguenti

```
CREATE INDEX contoCorrenteIX ON contoCorrente (numero)
```

```
CREATE INDEX contoCorrenteIX ON contoCorrente (numero, saldo)
```

**Domanda 3** (20%) Dimostrare che due schedule conflict-equivalenti hanno lo stesso grafo dei conflitti, mentre il viceversa non è necessariamente vero.

**Domanda 4** (20%) Si considerino un sistema con blocchi di dimensione  $B = 1000$  byte e puntatori ai blocchi di  $p = 5$  byte e una base di dati sulle seguenti relazioni, ognuna delle quali (i) ha una struttura heap (ii) ha un indice secondario sulla chiave (iii) non ammette valori nulli

- $R_1(\underline{ABC})$ , contenente  $S_1 = 1.000.000$  ennuple di  $l_1 = 40$  byte, di cui  $l_A = 5$  per il campo chiave  $A$
- $R_2(\underline{DEF})$ , contenente  $S_2 = 400.000$  ennuple di  $l_2 = 100$  byte, di cui  $l_D = 4$  per il campo chiave  $D$
- $R_3(\underline{GHL})$ , contenente  $S_3 = 500.000$  ennuple di  $l_3 = 25$  byte, di cui  $l_G = 5$  per il campo chiave  $G$

e con una vista definita come segue:

- `CREATE VIEW V AS SELECT * FROM (R1 LEFT JOIN R2 ON B=D) JOIN R3 ON C=G`

In tale contesto,

- mostrare un possibile piano di esecuzione (in termini di operatori dell'algebra relazionale e loro realizzazioni) per ciascuna delle seguenti interrogazioni
  1. `SELECT A, L FROM V`
  2. `SELECT A FROM V`
  3. `SELECT A, E FROM V`
- stimare il costo, in termini di numero di accessi a memoria secondaria (ignorando la presenza di eventuali buffer) per l'operazione 1.

**Domanda 5** (20%) Si consideri la seguente base di dati relazionale, relativa alle vendite presso una catena di supermercati:

- Vendite(NumeroScontrino, CodNegozio, CFCliente, Data, Ora)
- ElementiVendita(NumeroScontrino, NumeroLinea, CodArticolo, Quantità)
- Articoli(CodArticolo, Descrizione, CodMarca, Prezzo)
- Clienti(CFCliente, Cognome, Nome, DataNascita, CodCategoria, CittàResidenza)
- Negozi(CodNegozio, Nome, Indirizzo, Città)
- Città(Nome, Provincia)
- Province(Sigla, Regione)
- Marche(CodMarca, Nome, CodNazione)
- Nazioni(CodNazione, Nazione)
- Categorie(CodCategoria, Descrizione)

Si noti che, ovviamente, il prezzo di un prodotto cambia molto rapidamente nel tempo, mentre la residenza di un cliente può cambiare, ma molto di rado.

Costruire, in tale contesto, uno schema dimensionale che permetta di analizzare le vendite giornaliere (quantità e incassi) rispetto a

- data (dimensione standard aziendale, i cui dettagli possono essere omessi);
- prodotti, con tutte le loro proprietà;
- negozi, con tutte le loro proprietà;
- clienti, con le relative fasce d'età (ad esempio 18-24 anni, 25-35 anni, etc.), categorie e città, provincia e regione di residenza.

Indicare sinteticamente come si ottengono le relazioni dello schema dimensionale da quelle della base di dati.