

# Tecnologie delle Basi di Dati M

Appello del 19/7/2010

## Esercizio 1 (2 punti)

Data la relazione con schema:

Docenti(matricola, nome, luogonascita, datanascita)

si effettui una stima del numero di pagine necessarie per memorizzare la relazione e del numero di livelli (e di nodi) di un B<sup>+</sup>-tree costruito sull'attributo (luogonascita). Si supponga di avere pagine di dimensione 4KB, di cui 96 B riservati per il page header, e si considerino i seguenti valori:

- Numero di tuple = 800K
- Numero di chiavi (luogonascita) = 200K
- Dimensione matricola = 4 byte
- Dimensione nome = 24 byte
- Dimensione datanascita = 6 byte
- Dimensione luogonascita = 16 byte
- Dimensione RID = 4 byte
- Dimensione PID = 4 byte
- Percentuale di riempimento foglie = 80%

## Esercizio 2 (5 punti)

Date le relazioni con schema:

Personale(matricola, nome, stipendio, progetto, dipartimento)

Progetti(codice, nome, luogo, budget)

Dipartimenti(codice, nome, direttore)

si ottimizzi l'esecuzione della seguente interrogazione SQL (si considerino solamente i piani Progetti-Personale-Dipartimenti e Dipartimenti-Personale-Progetti):

```
SELECT P.matricola, D.direttore
FROM Personale P, Dipartimenti D, Progetti R
WHERE P.dipartimento=D.codice
      AND P.progetto=R.codice
      AND R.luogo = "Bologna"
      AND D.nome = "design"
```

tenendo conto che dai cataloghi della base di dati risulta:

- Numero di tuple Personale = 100K
- Numero di pagine Personale = 1K
- Numero di tuple Dipartimenti = 100
- Numero di pagine Dipartimenti = 5
- Numero di tuple Progetti = 10K
- Numero di pagine Progetti = 100
- Indice unclustered (TID disordinate) su R.luogo: numero foglie = 30, numero chiavi = 100
- Indice unclustered (TID ordinate) su D.nome: numero foglie = 3, numero chiavi = 20
- Indice clustered su P.progetto: numero foglie = 400
- Indice unclustered (TID ordinate) su P.dipartimento: numero foglie = 200
- Indici hash su D.codice e R.codice

Si disegni infine l'albero corrispondente al piano di accesso di costo minimo e stimi il numero di risultati dell'interrogazione.

Suggerimento: per la formula di Cardenas si utilizzino i seguenti valori, validi per P = 1000:

R	$\Phi(R, P)$
100	95.20785
200	181.3512
300	259.293
400	329.8141
500	393.6211
600	451.3531
700	503.5886
800	550.8509
900	593.6134
1000	632.3046

R	$\Phi(R, P)$
1100	667.3121
1200	698.9866
1300	727.6454
1400	753.5757
1500	777.0372
1600	798.265
1700	817.4718
1800	834.8499
1900	850.5735
2000	864.8001

R	$\Phi(R, P)$
2100	877.6722
2200	889.3187
2300	899.8565
2400	909.3909
2500	918.0176
2600	925.823
2700	932.8852
2800	939.2751
2900	945.0566
3000	950.2876

## Esercizio 3 (5 punti)

Si illustri il funzionamento dell'indice B<sup>+</sup>-tree in fase di inserimento di oggetti, mostrandone le prestazioni in termini di costo medio per oggetto inserito.

Facoltativo (1 punto): si discutano le implicazioni dell'aumentare/diminuire la dimensione della pagina (nodo) sulle prestazioni dell'indice in fase di ricerca e di aggiornamento del B<sup>+</sup>-tree.

## Esercizio 4 (3 punti)

Discutere sull'opportunità di adottare politiche "force" e "no steal" all'interno del buffer manager, considerando l'impatto che tali politiche hanno sul crash recovery. In particolare, si considerino sistemi aventi differenti frequenze di aggiornamento dei dati e di accessi concorrenti.

### *Soluzione Esercizio 1*

#### **Dimensionamento relazione:**

Dimensione di ogni tupla =  $4 + 24 + 6 + 16 = 50B$

Numero di tuple per pagina =  $(4096 - 96)/50 = 4000/50 = 80$

Numero di pagine della relazione =  $NT/80 = 800000/80 = 10000$

#### **Dimensionamento indice (luogonascita):**

Numero di chiavi = 200K, quindi mediamente ci sono  $800K/200K = 4$  tuple per ogni valore di chiave.

Dimensione di ogni record (foglia) =  $16 + 4 \times 4 = 32B$

Dimensione "reale" foglia =  $(4096 - 96) \times 0.80 = 3200B$

Numero di foglie =  $200K \times 32/3200 = 2000$

Dimensione di ogni record (nodo interno) =  $16 + 4 = 20B$

Numero nodi livello 1 =  $2000 \times 20/4000 = 10$

Numero nodi livello 2 =  $10 \times 20/4000 = 1$

Il B<sup>+</sup>-tree corrispondente si compone quindi di 3 livelli per un totale di 11 nodi interni e 2000 foglie.

### *Soluzione Esercizio 2*

#### **Selettività dei predicati:**

Predicato su R.luogo =  $1/100 = 0.01$

Predicato su D.nome =  $1/20 = 0.05$

#### **Accesso a R:**

Costo scan sequenziale = **100**

Costo indice su R.luogo:  $NL \times 0.01 + NT \times 0.01 = 30 \times 0.01 + 10K \times 0.01 = 1 + 100 = 101$

Costo indice su R.codice:  $1 + 1 = 2$

Numero tuple residue =  $NT \times 0.01 = 100$

#### **Accesso a P:**

Costo scan sequenziale = **1000**

Costo indice su P.progetto:  $NL \times 0.0001 + NP \times 0.0001 = 400 \times 0.0001 + 1000 \times 0.0001 = 1 + 1 = 2$

Costo indice su P.dipartimento:  $NL \times 0.01 + \Phi(NT \times 0.01, NP) = 200 \times 0.01 + \Phi(100K \times 0.01, 1K) = 2 + 633 = 635$

Numero tuple residue R esterna:  $NT \times 0.01 = 1000$

Numero tuple residue D esterna:  $NT \times 0.05 = 5000$

#### **Accesso a D:**

Costo scan sequenziale = **5**

Costo indice su D.nome:  $NL \times 0.05 + \Phi(NT \times 0.05, NP) = 3 \times 0.05 + \Phi(100 \times 0.05, 5) = 1 + 4 = 5$

Costo indice su D.codice:  $1 + 1 = 2$

Numero tuple residue =  $NT \times 0.05 = 5$

#### **Costi di join:**

R-P-D: costo = costo sequenziale +  $100 \times$  costo indice P.progetto +  $1000 \times$  costo indice

D.codice =  $100 + 100 \times 2 + 1000 \times 2 = 2300$

D-P-R: costo = costo sequenziale +  $5 \times$  costo indice P.dipartimento +  $5000 \times$  costo indice

R.codice =  $5 + 5 \times 635 + 5000 \times 2 = 13180$

Il numero di risultati dell'interrogazione è  $100K \times 0.01 \times 0.05 = 50$