

# Tecnologie delle Basi di Dati M

Appello del 27/9/2010

## Esercizio 1 (2 punti)

Date le relazioni con schema:

```
Personale(matricola, nome, stipendio, progetto, dipartimento)
Progetti(codice, nome, luogo, budget)
Dipartimenti(codice, nome, direttore)
```

si supponga di dover ottimizzare la seguente interrogazione SQL:

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

Si indichino i piani di accesso che verrebbero generati da un ottimizzatore che utilizzi unicamente piani *left-deep*. In ogni piano si indichi anche quale sia l'attributo coinvolto nell'operazione di join. (Suggerimento: piani di accesso che presentino lo stesso ordine di accesso alle relazioni ma attributi di join diversi sono da considerarsi come piani diversi).

## Esercizio 2 (5 punti)

Date le relazioni con schema:

```
Personale(matricola, nome, stipendio, progetto, caporeparto)
Progetti(codice, nome, luogo, budget)
```

si ottimizzi l'esecuzione della seguente interrogazione SQL (si considerino solamente i piani PR-P-C e C-P-PR):

```
SELECT P.nome, PR.nome, C.nome
FROM Personale P, Personale C, Progetti PR
WHERE P.caporeparto=C.matricola
      AND P.progetto=PR.codice
      AND PR.luogo = "Bologna"
      AND C.stipendio < 60000
```

tenendo conto che dai cataloghi della base di dati risulta:

- Numero di tuple Personale = 100K
- Numero di pagine Personale = 1000
- Numero di tuple Progetti = 1000
- Numero di pagine Progetti = 100
- Numero di chiavi per l'attributo caporeparto = 1000
- Indice unclustered (TID ordinate) su luogo: numero foglie = 20, numero chiavi = 50
- Indice clustered su progetto: numero foglie = 400
- Indice unclustered (TID disordinate) su stipendio: numero foglie = 2000, valore minimo = 10000, valore massimo = 210000
- Indice hash su matricola e codice

Si disegni infine l'albero corrispondente al piano di accesso di costo minimo e stimi il numero di risultati dell'interrogazione (supponendo non esistano omonimie tra il personale e tra i progetti).

Suggerimento: per la formula di Cardenas si utilizzino i seguenti valori, validi per P = 100:

R	$\Phi(R, P)$
10	9.561792
20	18.20931
30	26.02996
40	33.10282
50	39.49939
60	45.28434
70	50.51613
80	55.24768
90	59.5268
100	63.39677

R	$\Phi(R, P)$
110	66.89669
120	70.06196
130	72.92457
140	75.51347
150	77.85482
160	79.9723
170	81.8873
180	83.6192
190	85.1855
200	86.60203

R	$\Phi(R, P)$
210	87.88312
220	89.04171
230	90.08952
240	91.03714
250	91.89415
260	92.66921
270	93.37017
280	94.0041
290	94.57741
300	95.09591

## Esercizio 3 (5 punti)

Si illustrino le varie possibilità di gestione degli overflow per le strutture di static hashing.

Facoltativo (1 punto): si discuta sull'opportunità di aumentare la capacità delle pagine per diminuire la percentuale di overflow, rispetto al diminuire il fattore di caricamento delle pagine stesse.

## Esercizio 4 (3 punti)

Discutere sull'opportunità di mantenere statistiche "incrociate" su più attributi di una o più relazioni ai fini di ottenere una più precisa stima dei costi in fase di ottimizzazione delle interrogazioni. In particolare, si discuta sulla possibilità di mantenere tali statistiche in forma compatta utilizzando strutture multi-dimensionali (per semplicità, è possibile considerare la combinazione di soli due attributi).

### Soluzione Esercizio 1

I piani che vengono generati sono i seguenti:

1.  $(P \bowtie_{(P.dipartimento=D.codice)D}) \bowtie_{(P.progetto=R.codice)R}$
2.  $(P \bowtie_{(P.matricola=D.direttore)D}) \bowtie_{(P.progetto=R.codice)R}$
3.  $(D \bowtie_{(P.dipartimento=D.codice)P}) \bowtie_{(P.progetto=R.codice)R}$
4.  $(D \bowtie_{(P.matricola=D.direttore)P}) \bowtie_{(P.progetto=R.codice)R}$
5.  $(P \bowtie_{(P.progetto=R.codice)R}) \bowtie_{(P.dipartimento=D.codice)D}$
6.  $(P \bowtie_{(P.progetto=R.codice)R}) \bowtie_{(P.matricola=D.direttore)D}$
7.  $(R \bowtie_{(P.progetto=R.codice)P}) \bowtie_{(P.dipartimento=D.codice)D}$
8.  $(R \bowtie_{(P.progetto=R.codice)P}) \bowtie_{(P.matricola=D.direttore)D}$

### Soluzione Esercizio 2

#### Selettività dei predicati:

Predicato su R. luogo =  $1/50 = 0.02$

Predicato su P. caporeparto =  $1000/100K = 0.01$

Predicato su C. stipendio =  $(60000 - 10000)/(210000 - 10000) = 0.25$

#### Accesso a PR:

Costo scan sequenziale = **100**

Costo indice su PR. luogo:  $NL \times 0.02 + \Phi(NT \times 0.02, NP) = 30 \times 0.02 + \Phi(1000 \times 0.02, 100) = 1 + 19 = 20$

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

Numero tuple residue =  $NT \times 0.02 = 20$

#### Accesso a P:

Costo scan sequenziale = **1000**

Costo indice su P. progetto:  $NL/1K + NP/1K = 400/1000 + 1000/1000 = 1 + 1 = 2$

Numero tuple residue PR esterna:  $NT \times 0.02 = 2000$

Numero tuple residue C esterna:  $NT \times 0.25 = 25000$

#### Accesso a C:

Costo scan sequenziale = **1000**

Costo indice su C. matricola:  $1 + 1 = 2$

Costo indice su C. stipendio:  $NL \times 0.25 + NT \times 0.25 = 2000 \times 0.25 + 100K \times 0.25 = 500 + 25000 = 25500$

Numero tuple residue:  $NT \times 0.25 = 25000$

#### Costi di join:

PR-P-C: costo = costo indice su PR. luogo +  $20 \times$  costo indice P. progetto +  $2000 \times$  costo indice C. matricola =  $20 + 20 \times 2 + 2000 \times 2 = 4060$

C-P-PR: costo = costo scan sequenziale +  $5000 \times$  costo sequenziale +  $25000 \times$  costo indice

PR. codice =  $1000 + 25000 \times 1000 + 25000 \times 2 = 25051000$

Il numero di risultati dell'interrogazione è  $100K \times 0.02 \times 0.25 = 500$