

# **Esercizi d'esame**

# Esercizio

---

- Dato il seguente log
  1. B(T1)
  2. U(T1,O1,B1,A1)
  3. B(T2)
  4. I(T1,O2,A2)
  5. B(T3)
  6. D(T3,O3,B2)
  7. U(T2,O4,B3,A3)
  8. CK(T1,T2,T3)
  9. I(T3,O5,A4)
  10. B(T4)
  11. U(T4,O6,B4,A5)
  12. I(T4,O7,A6)
  13. U(T4,O2,B5,A7)
  14. C(T3)
  15. I(T2,O8,A9)
  16. A(T1)
  17. U(T4,O3,B7,A10)
  18. guasto
- si mostrino le operazioni di recovery da effettuare supponendo che il guasto avvenga subito dopo l'ultimo record del log.

# Esercizio: soluzione

---

- |                     |                                      |
|---------------------|--------------------------------------|
| 1. B(T1)            | 8 UNDO={T1,T2,T3} REDO={}            |
| 2. U(T1,O1,B1,A1)   | 10 B(T4)->UNDO={T1,T2,T3,T4} REDO={} |
| 3. B(T2)            | 14 C(T3)->UNDO={T1,T2,T4} REDO={T3}  |
| 4. I(T1,O2,A2)      | UNDO                                 |
| 5. B(T3)            | 17 O3=B7                             |
| 6. D(T3,O3,B2)      | 15 D(O8)                             |
| 7. U(T2,O4,B3,A3)   | 13 O2=B5                             |
| 8. CK(T1,T2,T3)     | 12 D(O7)                             |
| 9. I(T3,O5,A4)      | 11 O6=B4                             |
| 10. B(T4)           | 7 O4=B3                              |
| 11. U(T4,O6,B4,A5)  | 4 D(O2)                              |
| 12. I(T4,O7,A6)     | 2 O1=B1                              |
| 13. U(T4,O2,B5,A7)  | REDO                                 |
| 14. C(T3)           | 6 D(O3)                              |
| 15. I(T2,O8,A9)     | 9 I(O5,A4)                           |
| 16. A(T1)           |                                      |
| 17. U(T4,O3,B7,A10) |                                      |

# Esercizio

---

- Dire se il seguente schedule

$S1=r1(a), r2(a), r3(a), w1(b), w2(b), w3(b)$

- e' view-equivalente al seguente schedule seriale

$S2=r2(a), w2(b), r1(a), w1(b), r3(a), w3(b)$

Soluzione

- Si', i due schedule sono view equivalenti.
- In  $S1$  la relazione legge-da e' vuota e le scritture finali comprendono solo la scrittura di  $b$  da parte di  $T3$ .  $S2$  ha la stessa relazione legge-da e la stessa scrittura finale su  $b$ .

# Esercizio

---

- Dire se il seguente schedule

$S1 = r1(a), r3(d), w1(b), r2(b), w3(b), r4(b), w2(c), r5(c), w4(e), r5(e), w5(b)$

- E' view-equivalente allo schedule seriale:

$S2 = r1(a), w1(b), r2(b), w2(c), r3(d), w3(b), r4(b), w4(e), r5(c), r5(e), w5(b)$

Soluzione

- La relazione legge-da per  $S1$  ha le seguenti coppie:  $\{(r2(b), w1(b)), (r4(b), w3(b)), (r5(c), w2(c)), (r5(e), w4(e))\}$
- e le seguenti scritture finali:  $\{w5(b), w2(c), w4(e)\}$
- $S2$  ha la stessa relazione legge-da e le stesse scritture finali

# Esercizio

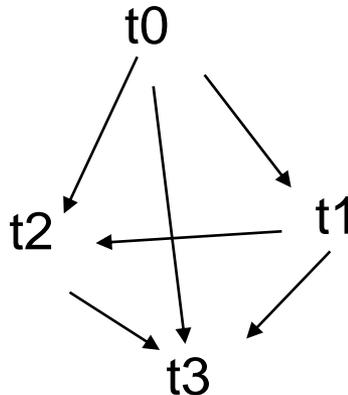
---

Dato il seguente schedule:

$w_0(x)$   $r_1(x)$   $r_0(y)$   $w_1(y)$   $w_1(x)$   $r_2(x)$   $w_3(x)$   $r_2(y)$

si costruisca il grafo dei conflitti e si dica se e' conflict serializzabile.

Soluzione



Lo schedule e' conflict serializzabile

# Esercizio

---

- Dire se il seguente schedule e' in 2PL

S1 = r1(x) r2(y) w1(z) r2(x) r3(x) w2(z) w2(y) r1(y)

- Nel caso rispetti il 2PL, si indichi una sequenza di acquisizione e rilascio di lock che rispetti sia lo schedule che il two-phase locking. Nel caso non lo rispetti, si indichi almeno uno dei conflitti.

# Soluzione

r1(x) r2(y) w1(z) r2(x) r3(x) w2(z) w2(y) r1(y)

Transazione 1	Transazione 2	Transazione 3
r_lock(x)		
	r_lock(y)	
w_lock(z)		
unlock(x)		
	r_lock(x)	
		r_lock(x)
unlock(z)		
	w_lock(z)	
	w_lock(y)	
	unlock(y)	
	unlock(z)	
r_lock(y) -> <b>NO</b>		

Lo schedule non rispetta il two-phase locking. Infatti poichè sia possibile w2(z), la transazione 1 deve rilasciare il lock esclusivo su z, ma deve effettuare r1(y) alla fine dello schedule e deve acquisire il lock in lettura su y dopo w2(y)

## Esercizio (13/6/07)

---

- Dato il seguente schedule:
- $r_2(x) w_3(y) w_1(x) r_5(p) w_5(x) w_2(z) r_4(w) r_3(z) r_5(y) w_4(p)$
- si indichi se rispetta il two-phase locking. Nel caso lo rispetti, si indichi una sequenza di acquisizione e rilascio di lock compatibile con lo schedule e che rispetti il two-phase locking. Nel caso non lo rispetti, si indichi almeno uno dei conflitti.

# Soluzione

r2(x) w3(y) w1(x) r5(p) w5(x) w2(z) r4(w) r3(z) r5(y) w4(p)

Transazione 1	Transazione 2	Transazione 3	Transazione 4	Transazione 5
	r_lock(x)			
		w_lock(y)		
	w_lock(z)			
	unlock(x)			
w_lock(x)				
unlock(x)				
				r_lock(p)
				w_lock(x)
	unlock(z)			
			r_lock(w)	

# Soluzione

r2(x) w3(y) w1(x) r5(p) w5(x) w2(z) r4(w) r3(z) r5(y) w4(p)

Transazione 1	Transazione 2	Transazione 3	Transazione 4	Transazione 5
		r_lock(z)		
		unlock(y)		
		unlock(z)		
				r_lock(y)
				unlock(p)
				unlock(x)
				unlock(y)
			w_lock(p)	
			unlock(p)	
			unlock(w)	

# Esercizio

---

- Dire se il seguente schedule e' in 2PL

S1=r1(a), r3(d), w1(b), r2(b), w3(b), r4(b), w2(c), r5(c),  
w4(e), r5(e), w5(b)

- Nel caso lo sia, si indichi una sequenza di acquisizione e rilascio di lock compatibile con lo schedule e che rispetti il two-phase locking. Nel caso non lo rispetti, si indichi almeno uno dei conflitti.

# Soluzione

r1(a), r3(d), w1(b), r2(b), w3(b), r4(b), w2(c), r5(c), w4(e), r5(e), w5(b)  
S1 e' in 2PL perche' ha la seguente sequenza di acquisizione e rilascio di lock

Transazione 1	Transazione 2	Transazione 3	Transazione 4	Transazione 5
r_lock(a)				
		r_lock(d)		
w_lock(b)				
unlock(b)				
unlock(a)				
	r_lock(b)			
	w_lock(c)			
	unlock(b)			
		w_lock(b)		

# Soluzione

r1(a), r3(d), w1(b), r2(b), w3(b), r4(b), w2(c), r5(c), w4(e), r5(e), w5(b)

Transazione 1	Transazione 2	Transazione 3	Transazione 4	Transazione 5
		unlock(b)		
		unlock(d)		
			r_lock(b)	
	unlock(c)			
				r_lock(c)
			w_lock(e)	
			unlock(b)	
			unlock(e)	
				r_lock(e)
				w_lock(b)
				unlock(e)
				unlock(b)

e' in 2PL

## Esercizio 9/9/2009

---

- Dato lo schedule
- $r1(x) r2(x) r4(y) w3(z) w1(x) r2(z) w4(y) r3(y)$
- si indichi se rispetta il two-phase locking. Nel caso lo rispetti, si indichi una sequenza di acquisizione e rilascio di lock compatibile con lo schedule e che rispetti il two-phase locking. Nel caso non lo rispetti, si indichi almeno uno dei conflitti.

# Soluzione

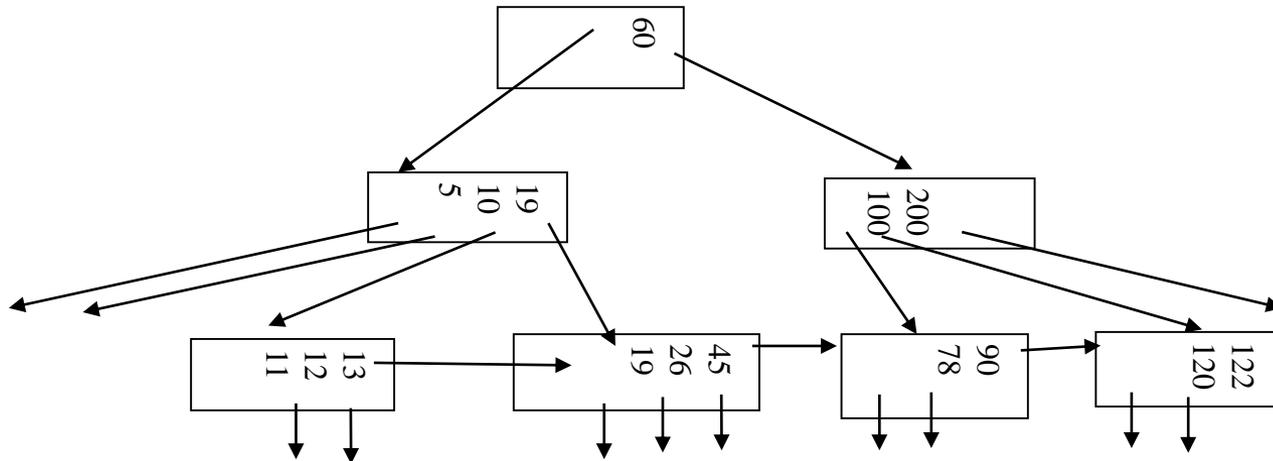
- r1(x) r2(x) r4(y) w3(z) w1(x) r2(z) w4(y) r3(y)

Transazione 1	Transazione 2	Transazione 3	Transazione 4
r_lock(x)			
	r_lock(x)		
			r_lock(y)
		w_lock(z)	
		unlock(z)	
	r_lock(z)		
	unlock(x)		
w_lock(x)			
unlock(x)			
	unlock(z)		
			w_lock(y)
			unlock(y)
		r_lock(y) -> NO	

Lo schedule non è in 2-phase locking perché t3 deve rilasciare il lock su z affinché t2 possa proseguire. t3 deve però acquisire un lock su y dopo w4(y).

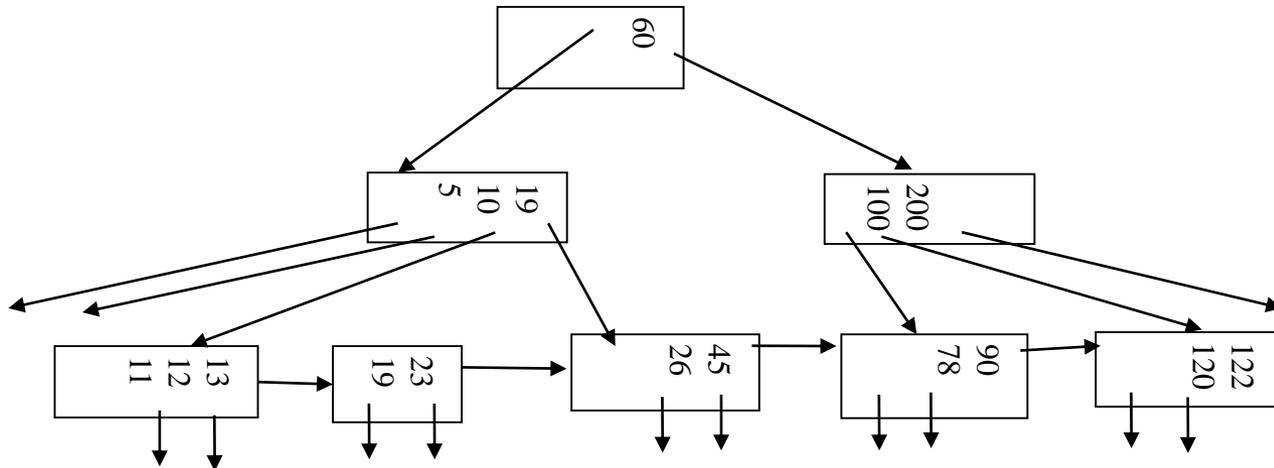
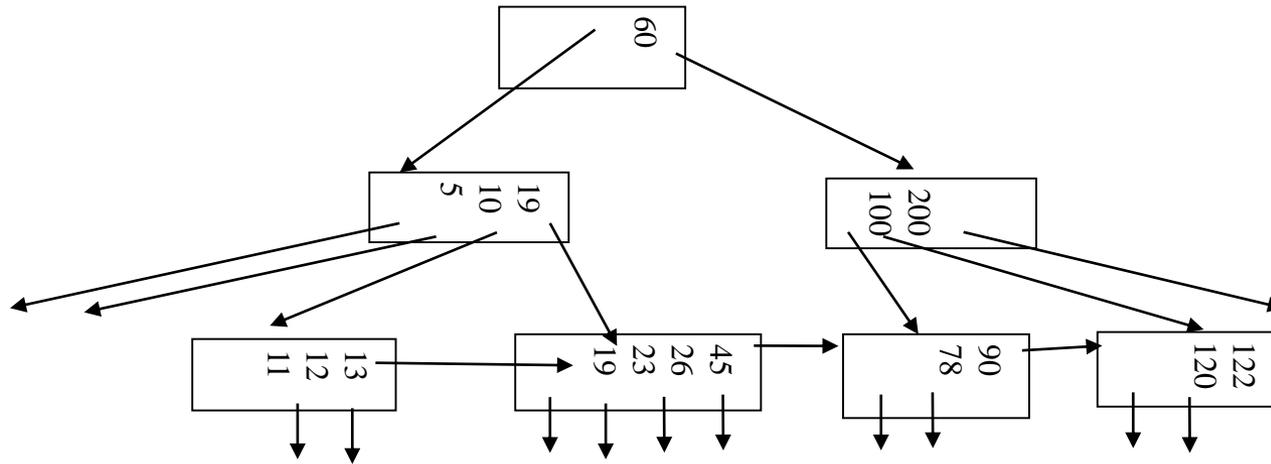
# Esercizio 14/7/2011

- Sia dato il seguente B+ tree di ordine 3.

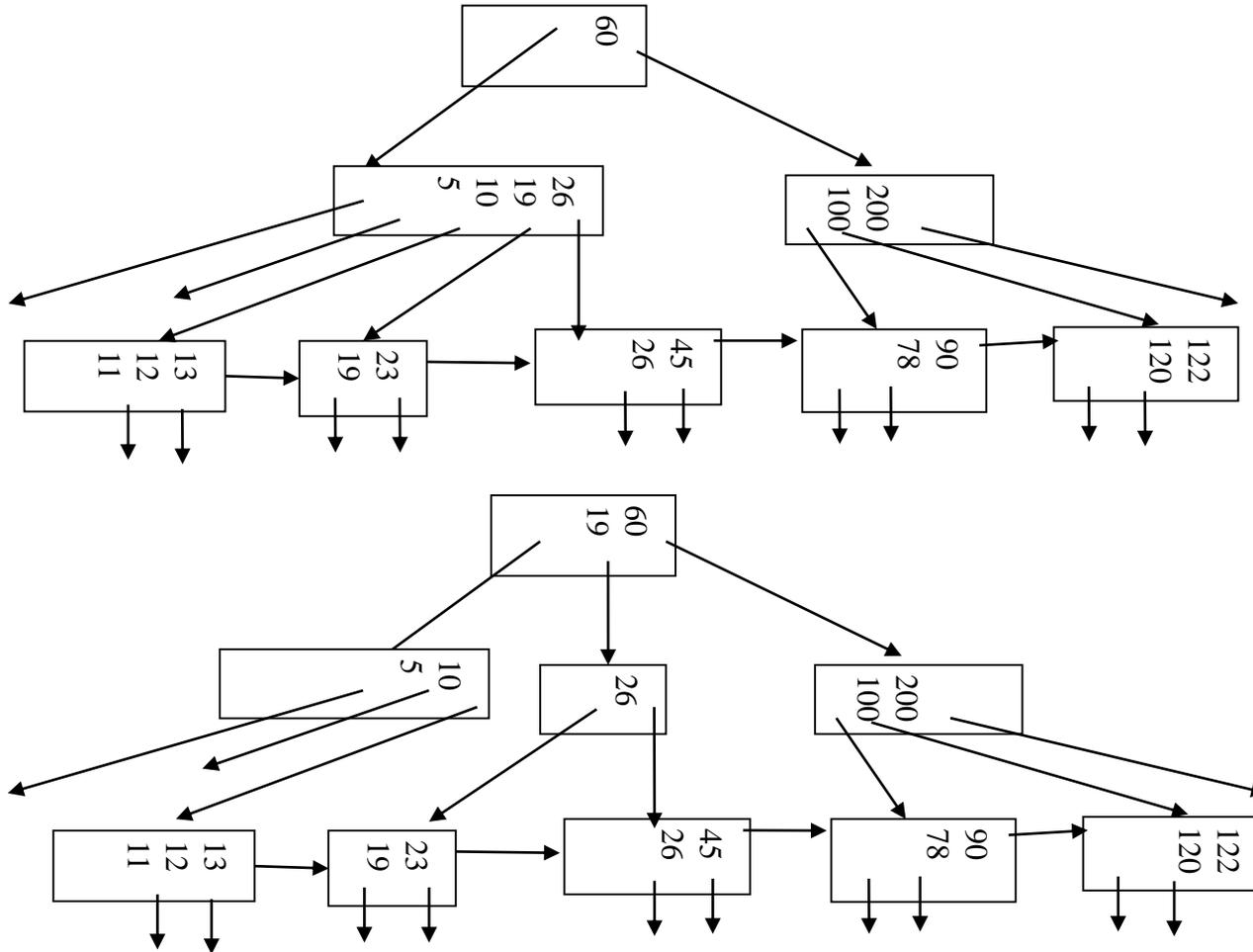


- Si mostri l'albero risultante dall'inserimento della chiave 23

# Soluzione

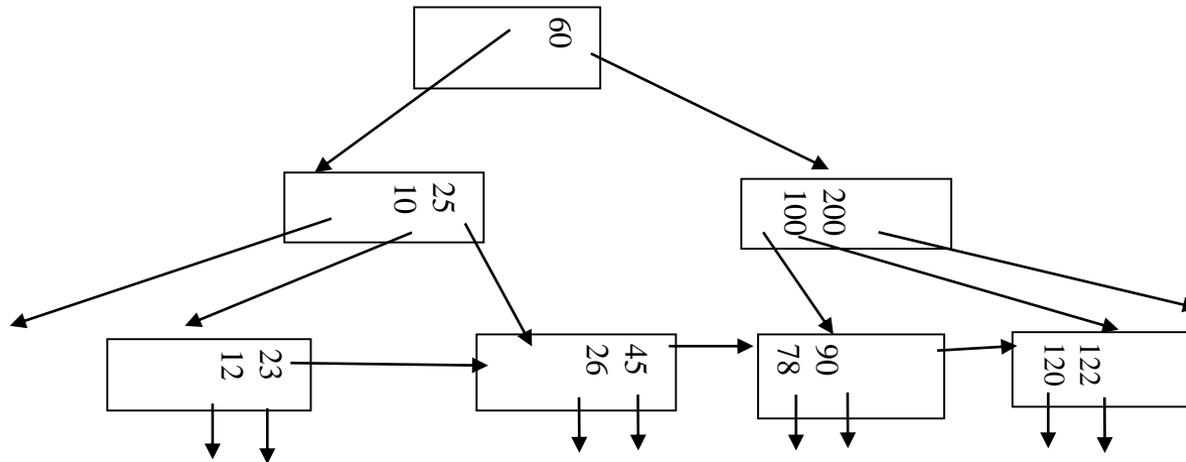


# Soluzione



# Esercizio 29/6/2011

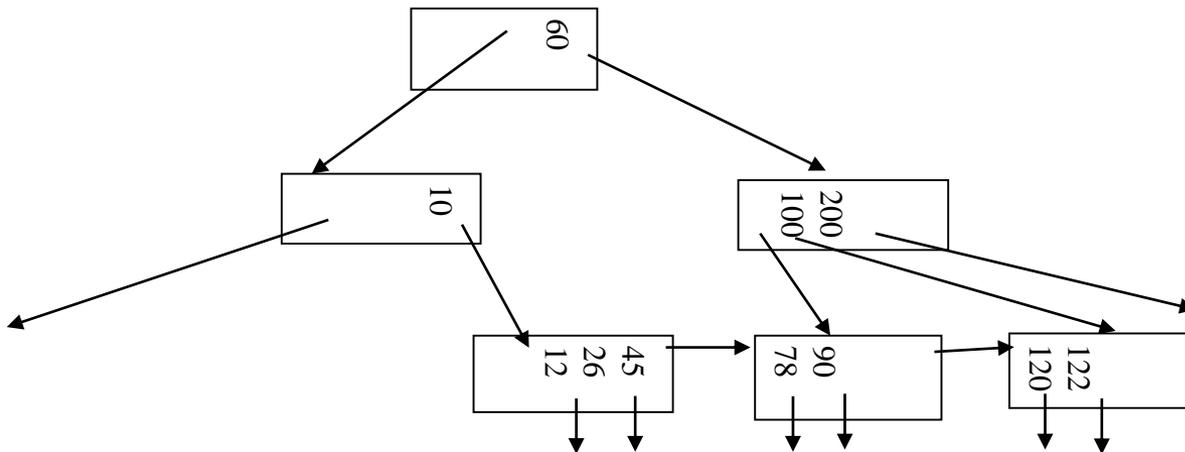
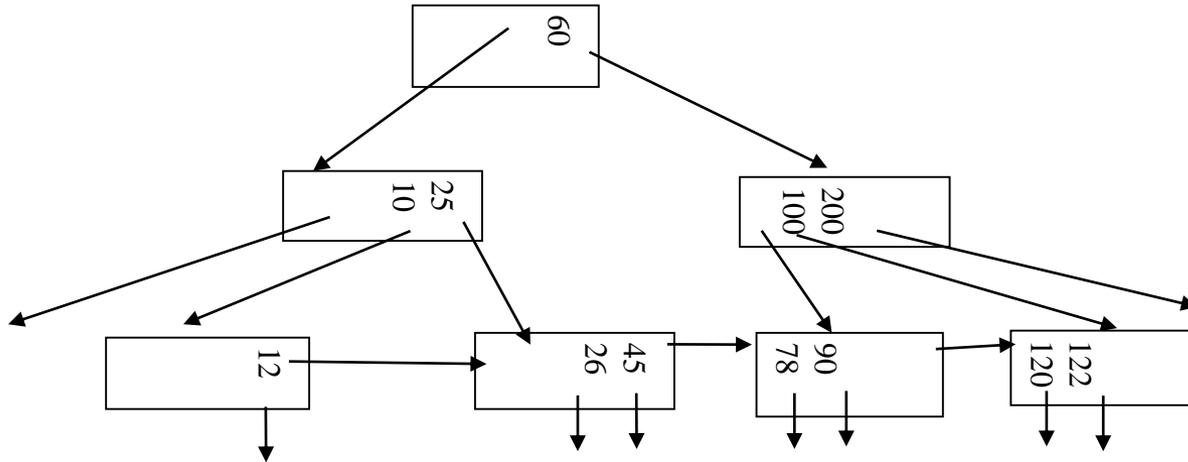
- Sia dato il seguente B+ tree di ordine 4.



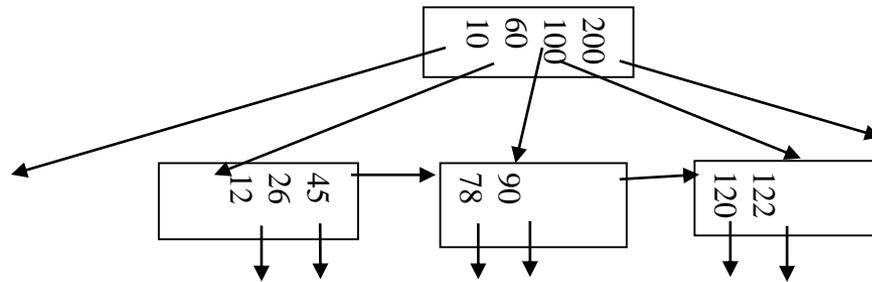
- Si mostri l'albero risultante dalla cancellazione della chiave 23. Si mostri l'albero risultante dalla cancellazione della chiave 78 nell'albero ottenuto in precedenza

# Soluzione

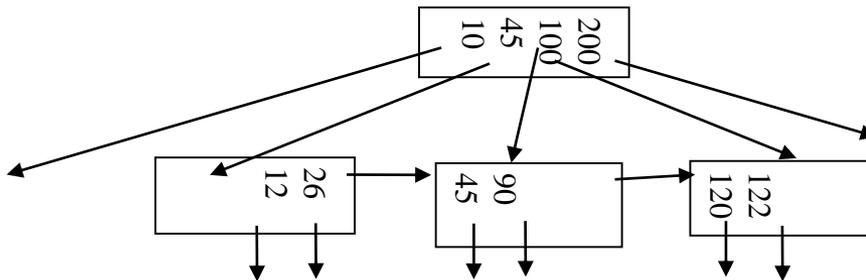
- Cancellazione della chiave 23



# Soluzione

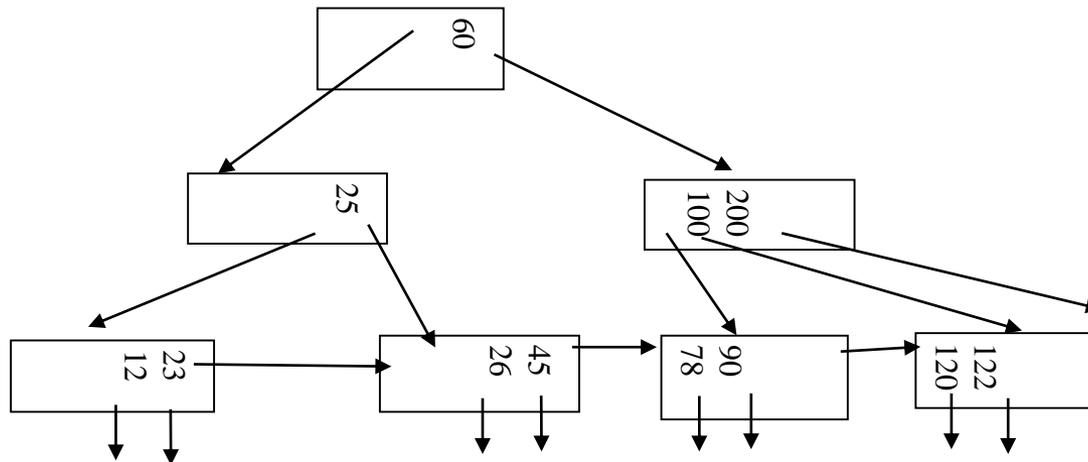


- Cancellazione della chiave 78



# Esercizio 13/7/2010

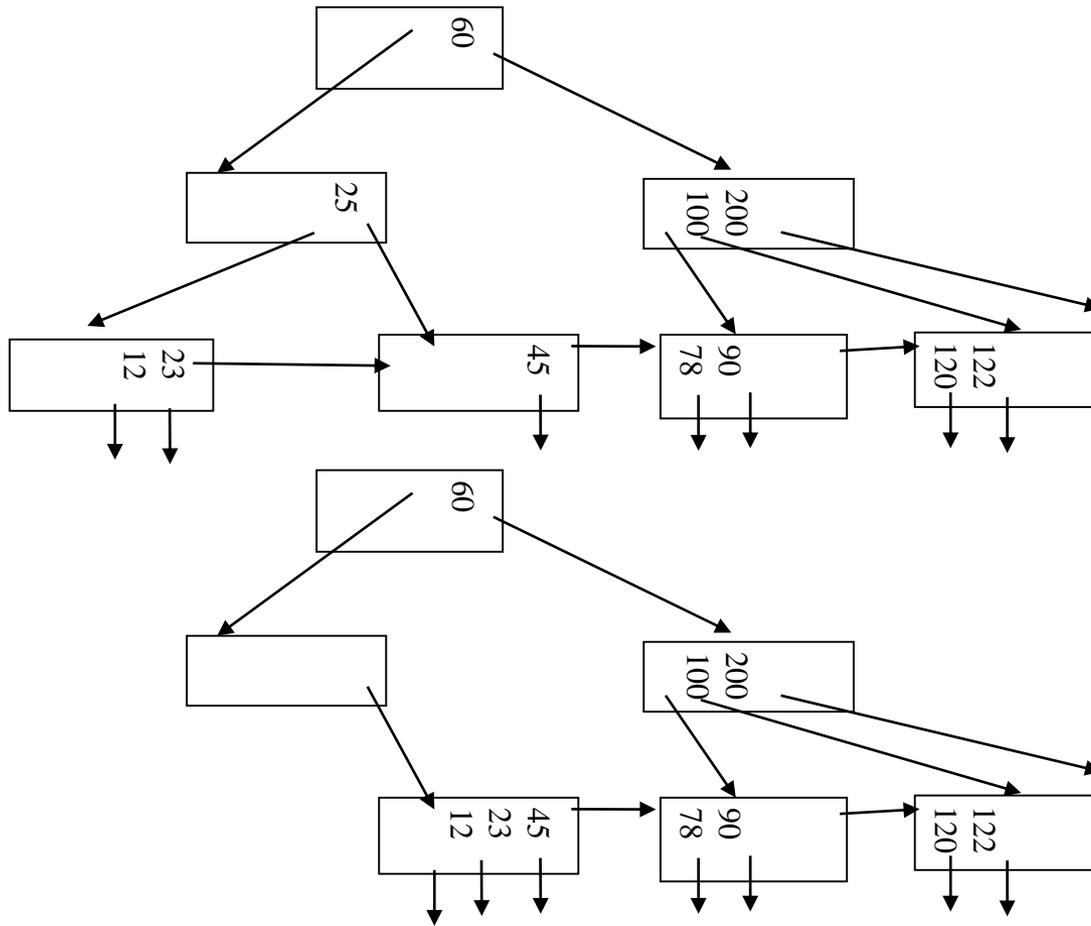
- Sia dato il seguente B+ tree di ordine 3.



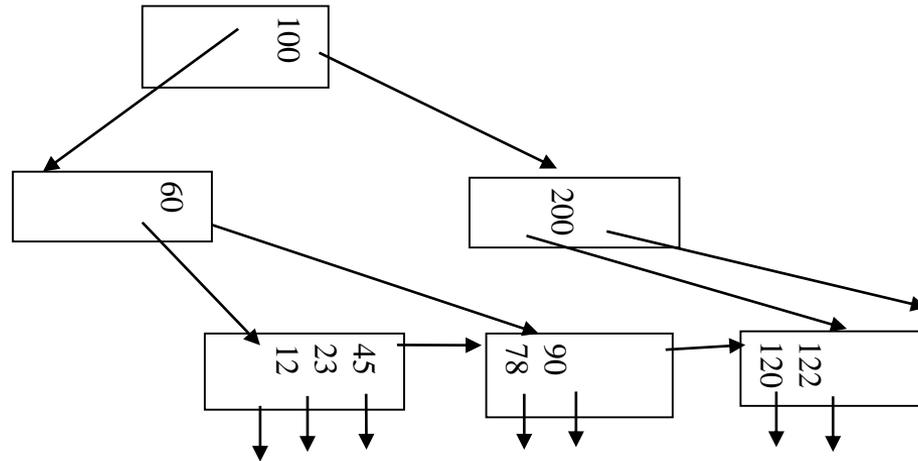
- Si mostri l'albero risultante dalla cancellazione della chiave 26. Si mostri l'albero risultante dalla cancellazione della chiave 78 nell'albero ottenuto in precedenza.

# Soluzione

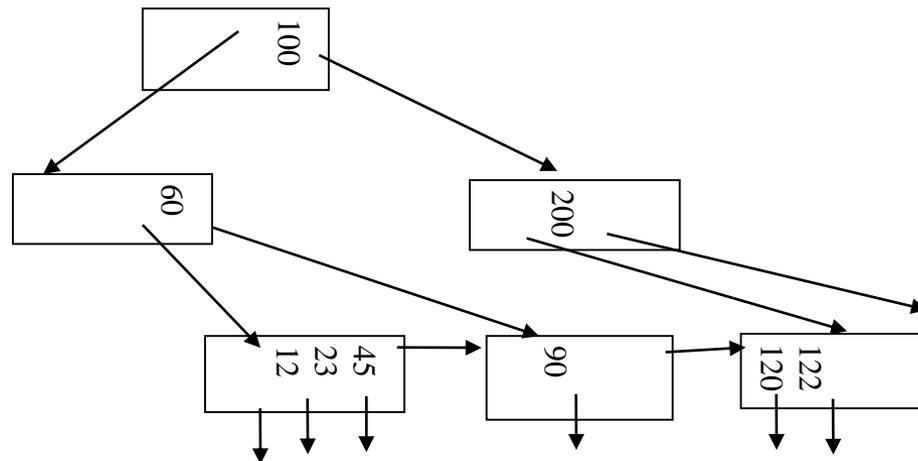
- Cancellazione di 26



# Soluzione

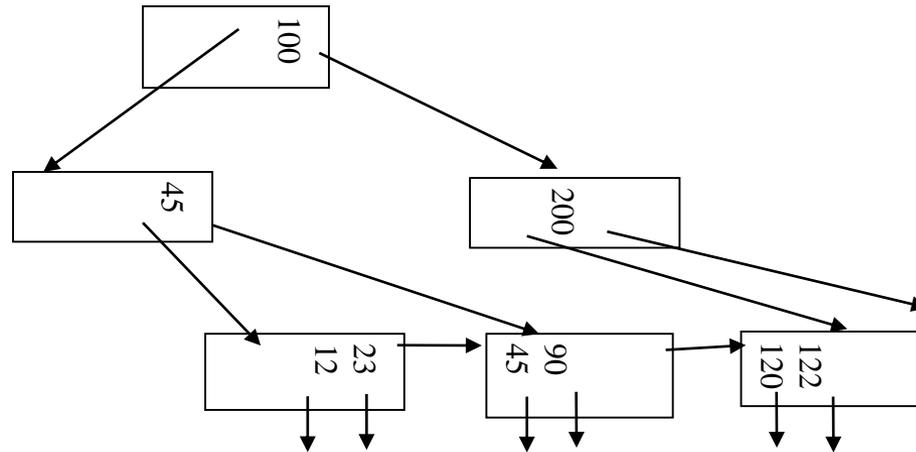


- Cancellazione di 78



# Soluzione

---



# Esercizio

---

- Date le relazioni

Impiegato(CodImp, CodDip, Nome, Cognome, Data\_di\_nascita, Stipendio)

Dipartimento(CodDip, Nome, Indirizzo, N\_afferenti)

- Si calcoli il costo di esecuzione della query

```
SELECT I.Nome, I.Cognome, D.Nome
```

```
FROM Impiegato AS I, Dipartimento AS D
```

```
WHERE I.CodDip=D.CodDip
```

- supponendo che vengano utilizzati gli algoritmi di Block-Based Nested-Loops Join, Sort Join, Hash Join Ibrido e Join con Indice.
- Si supponga di avere  $M=100$  buffer di memoria centrale disponibili, che  $B(\text{Impiegato})=1500$ ,  $T(\text{Impiegato})=15.000$ ,  $B(\text{Dipartimento})=200$ ,  $T(\text{Dipartimento})=2000$  e si supponga di avere un indice primario su Dipartimento.CodDip.

# Esercizio: soluzione

---

- Block-Based Nested-Loops Join:
  - $\text{Costo} = B(\text{Dipartimento}) + B(\text{Dipartimento})B(\text{Impiegato}) / (M-1)$   
 $= 200 + 200 * 1500 / 100 = 3200$
- Sort Join:
  - puo' essere fatto se  $B(\text{Dipartimento}) + B(\text{Impiegato}) \leq M^2$   
 $B(\text{Dipartimento}) + B(\text{Impiegato}) \leq 10.000$   $1700 \leq 10.000$  ok
  - $\text{Costo} = 3(B(\text{Dipartimento}) + B(\text{Impiegato})) = 3(1500 + 200) = 5100$
- Hash Join Ibrido:
  - puo' essere fatto se  $B(\text{Dipartimento}) \leq M^2$   
 $200 \leq 10.000$  ok
  - $\text{Costo} = (3 - 2M/B(\text{Dipartimento}))(B(\text{Impiegato}) + B(\text{Dipartimento})) =$   
 $(3 - 2 * 100/200)(1500 + 200) = (3 - 1)(1700) = 3400$
- Index Join:
  - $\text{Costo} = B(\text{Impiegato}) + T(\text{Impiegato}) * 1 = 1.500 + 15.000 = 16.500$

# Esercizio (29/11/05)

---

- Siano date le relazioni

Cliente(CodCli, Nome, Cognome, DataDiNascita, Città)

Venditore(CodVend, Nome, Cognome, Città, Auto)

Supervisore(CodSup, Nome, Cognome, Auto)

- la query

```
SELECT C.*, V.*, S.*
```

```
FROM Cliente AS C, Venditore AS V, Supervisore AS S
```

```
WHERE C.Città=V.Città AND V.Auto=S.Auto
```

- e i parametri:

- buffer di memoria centrale disponibili:  $M=100$
- dimensione del buffer:  $B=500$  bytes
- numero di tuple:  $T(\text{Cliente})=20.000$ ,  $T(\text{Venditore})=1.000$ ,  
 $T(\text{Supervisore})=500$
- dimensione delle tuple:  $S(\text{Cliente})=200$  bytes,  $S(\text{Venditore})=125$ ,  
 $S(\text{Supervisore})=125$ ,  $S(\text{Città})=25$ ,  $S(\text{Auto})=30$
- numero di valori:  $V(\text{Cliente}, \text{Città})=1.000$ ,  $V(\text{Venditore}, \text{Città})=500$ ,  
 $V(\text{Venditore}, \text{Auto})=100$ ,  $V(\text{Supervisore}, \text{Auto})=80$

# Esercizio

---

- Si stabilisca qual'è l'ordine migliore con cui eseguire i singoli join supponendo di utilizzare sempre l'Hash Join Ibrido. In particolare, occorrerà calcolare il costo delle due sequenze:  
(Cliente▷◁Venditore)▷◁Supervisore e  
(Venditore▷◁Supervisore)▷◁Cliente.
- Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record. Quindi, il numero di blocchi di una relazione e' dato da  $B(R)=T(R)*S(R)/B$ .

# Esercizio: soluzione

---

Calcoliamo innanzitutto il numero di blocchi occupati da ciascuna relazione:

$$B(\text{Cliente}) = T(\text{Cliente}) * S(\text{Cliente}) / B \\ = 20.000 * 200 / 500 = 8.000$$

$$B(\text{Venditore}) = 1.000 * 125 / 500 = 250$$

$$B(\text{Supervisore}) = 500 * 125 / 500 = 125$$

# Esercizio: soluzione

---

Prima sequenza: (Cliente▷◁Venditore)▷◁Supervisore

$$\text{Costo}(\text{Cliente} \triangleright \triangleleft \text{Venditore}) = (3 - 2 * 100 / 250) * (8.000 + 250) = 18.150$$

Sia  $X = \text{Cliente} \triangleright \triangleleft \text{Venditore}$

$$S(X) = S(\text{Cliente}) + S(\text{Venditore}) - S(\text{Città}) = 200 + 125 - 25 = 300$$

$$T(X) = T(\text{Cliente}) * T(\text{Venditore}) / \max(V(\text{Cliente}, \text{Città}), V(\text{Venditore}, \text{Città})) = 20.000 * 1.000 / 1.000 = 20.000$$

$$B(X) = 20.000 * 300 / 500 = 12.000$$

$$\text{Costo}(X \triangleright \triangleleft \text{Supervisore}) = (3 - 2 * 100 / 125) * (125 + 12.000) = 16.975$$

$$\text{Costo}((\text{Cliente} \triangleright \triangleleft \text{Venditore}) \triangleright \triangleleft \text{Supervisore}) = 18.150 + 16.975 = 35.125$$

## Esercizio: soluzione

---

Seconda sequenza: (Venditore  $\triangleright$   $\triangleleft$  Supervisore)  $\triangleright$   $\triangleleft$  Cliente

$$\text{Costo}(\text{Venditore} \triangleright \triangleleft \text{Supervisore}) = (3 - 2 * 100 / 125) * (250 + 125) = 525$$

Sia  $Z = \text{Venditore} \triangleright \triangleleft \text{Supervisore}$

$$S(Z) = 125 + 125 - 30 = 220$$

$$T(Z) = 1.000 * 500 / 100 = 5.000$$

$$B(Z) = 5000 * 220 / 500 = 2.200$$

$$\text{Costo}(Z \triangleright \triangleleft \text{Cliente}) = (3 - 2 * 100 / 2200) * (2200 + 8000) = 29.673$$

$$\text{Costo}((\text{Venditore} \triangleright \triangleleft \text{Supervisore}) \triangleright \triangleleft \text{Cliente}) = 525 + 29.673 = 30.198$$

La sequenza migliore per realizzare il join è la seconda sequenza.

# Esercizio (9/1/06)

---

- Siano date le relazioni

Impiegato(CodImp, Nome, Cognome, DataDiNascita, Dipartimento, Stipendio)

Consulente(CodCons, Nome, Cognome, Dipartimento, Stipendio, Indirizzo)

- la query

```
SELECT I.*, C.*
```

```
FROM Impiegato AS I, Consulente AS C
```

```
WHERE I.Dipartimento=C.Dipartimento AND I.Nome="Francesco"
```

- e i parametri:

- buffer di memoria centrale disponibili:  $M=100$
- dimensione del buffer:  $B=500$  bytes
- numero di tuple:  $T(\text{Impiegato})=250.000$ ,  $T(\text{Consulente})=800$ ,
- dimensione delle tuple:  $S(\text{Impiegato})=160$  bytes,  
 $S(\text{Consulente})=250$ ,  $S(\text{Dipartimento})=50$
- numero di valori:  $V(\text{Impiegato}, \text{Dipartimento})=2.000$ ,  
 $V(\text{Consulente}, \text{Dipartimento})=500$ ,  $V(\text{Impiegato}, \text{Nome})=400$

# Esercizio

---

- Si stabilisca qual'è l'ordine migliore con cui eseguire le operazioni supponendo di utilizzare l'Hash Join Ibrido. In particolare, occorrerà calcolare il costo delle due sequenze:  
 $(\sigma_{\text{Impiegato.Nome}='Francesco'} \text{Impiegato}) \triangleright \triangleleft \text{Consulente}$  e  
 $\sigma_{\text{Impiegato.Nome}='Francesco'} (\text{Impiegato} \triangleright \triangleleft \text{Consulente})$ .
- Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record. Si supponga inoltre che ci sia un indice secondario su Impiegato.Nome. Si supponga che non vengano costruiti indici su Impiegato  $\triangleright \triangleleft$  Consulente.

# Esercizio: soluzione

---

Calcoliamo innanzitutto il numero di blocchi occupati da ciascuna relazione:

$$B(\text{Impiegato}) = T(\text{Impiegato}) * S(\text{Impiegato}) / B \\ = 250.000 * 160 / 500 = 80.000$$

$$B(\text{Consulente}) = 800 * 250 / 500 = 400$$

Prima sequenza:  $(\sigma_{\text{Impiegato.Nome}='Francesco'} \text{Impiegato}) \triangleright \triangleleft \text{Consulente}$

Sia  $X = \sigma_{\text{Impiegato.Nome}='Francesco'} \text{Impiegato}$

$$\text{Costo}(X) = 3 + T(\text{Impiegato}) / V(\text{Impiegato}, \text{Nome}) = 3 + 250.000 / 400 \\ = 3 + 625 = 628.$$

$$T(X) = 250.000 / 400 = 625$$

$$B(X) = 625 * 160 / 500 = 200$$

$$\text{Costo}(X \triangleright \triangleleft \text{Consulente}) = (3 - 2 * 100 / 200) * (200 + 400) = 1.200$$

$$\text{Costo}((\sigma_{\text{Impiegato.Nome}='Francesco'} \text{Impiegato}) \triangleright \triangleleft \text{Consulente}) = 628 + 1.200 \\ = 1.828$$

# Esercizio: soluzione

---

Seconda sequenza:

$$\sigma_{\text{Impiegato.Nome='Francesco'}}(\text{Impiegato} \triangleright \triangleleft \text{Consulente})$$

Sia  $Y = \text{Impiegato} \triangleright \triangleleft \text{Consulente}$

$$\text{Costo}(Y) = (3 - 2 * 100 / 400) * (80.000 + 400) = 201.000$$

$$S(Y) = S(\text{Impiegato}) + S(\text{Consulente}) - S(\text{Dipartimento}) = 160 + 250 - 50 = 360$$

$$T(Y) = T(\text{Impiegato}) * T(\text{Consulente}) / \max(V(\text{Impiegato}, \text{Dipartimento}), V(\text{Consulente}, \text{Dipartimento})) = 250.000 * 800 / 2.000 = 100.000$$

$$B(Y) = 100.000 * 360 / 500 = 72.000$$

$$\text{Costo}(\sigma_{\text{Impiegato.Nome='Francesco'}} Y) = B(Y) = 72.000$$

$$\text{Costo}(\sigma_{\text{Impiegato.Nome='Francesco'}}(\text{Impiegato} \triangleright \triangleleft \text{Consulente})) = 72.000 + 201.000 = 273.000$$

La sequenza migliore per realizzare l'operazione è la prima.

# Esercizio (12/7/07)

---

- Siano date le relazioni

CD(CodiceCD, CodiceMus, Titolo, Anno, Durata)

Musicista(CodiceMus, CasaDiscografica, Nazionalità, Età)

- la query

```
SELECT C.*, M.*
```

```
FROM CD AS C, Musicista AS M
```

```
WHERE C.CodiceMus=M.CodiceMus AND C.Durata>30 AND  
C.Durata<70
```

- e i parametri:

- buffer di memoria centrale disponibili:  $M=20.000$
- dimensione del buffer:  $B=4.000$  bytes
- numero di tuple:  $T(\text{CD})= 83102$ ,  $T(\text{Musicista})= 40000$ ,
- dimensione delle tuple:  $S(\text{CD})=2000$  bytes,  $S(\text{Musicista})=3000$ ,
- $S(\text{CodMusicista})=200$
- numero di valori:  $V(\text{CD}, \text{CodiceMus})=40000$ ,
- $\text{Min}(\text{CD}, \text{Durata})=25$   $\text{Max}(\text{CD}, \text{Durata})=80$

# Esercizio

---

- Si calcoli il costo minimo di ciascuna delle seguenti sequenze:

$(\sigma_{\text{Durata}>30 \text{ AND Durata}<70} \text{ CD}) \triangleright \triangleleft \text{Musicista}$

$\sigma_{\text{Durata}>30 \text{ AND Durata}<70} (\text{CD} \triangleright \triangleleft \text{Musicista})$

- Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record. Si supponga che CD e Musicista siano ordinate sulla base del campo CodiceMus. Se ne tenga conto nel calcolo del costo del join.
- Si supponga di avere un indice secondario su CD.Durata. Per il calcolo del costo e della cardinalità della selezione su Durata si utilizzi la stima dei valori nel range. Si noti che CodiceMus è chiave primaria di Musicista.

## Esercizio: soluzione

---

$$B(\text{CD}) = T(\text{CD}) * S(\text{CD}) / B = 83102 * 2000 / 4000 = 41551$$

$$B(\text{Musicista}) = 40000 * 3000 / 4000 = 30000$$

Prima sequenza:  $(\sigma_{\text{Durata} > 30 \text{ AND Durata} < 70} \text{ CD}) \triangleright \triangleleft \text{Musicista}$

Sia  $X = \sigma_{\text{Durata} > 30 \text{ AND Durata} < 70} \text{ CD}$

Utilizzando l'indice secondario su Durata

$$f = (70 - 30 - 1) / (80 - 25 + 1) = 0.696$$

$$\text{Costo}(X) = 3 + f * T(\text{CD}) = 3 + 0.696 * 83102 = 57842$$

$$T(X) = f * T(\text{CD}) = 0.696 * 83102 = 57839$$

$$B(X) = 57839 * 2000 / 4000 = 28919$$

## Esercizio: soluzione

---

Né X né Musicista stanno in memoria centrale, quindi si prova con il join nested-loop basato sui blocchi e con l'hash join ibrido. Inoltre X non è ordinato su CodiceMus

$$\text{CostoJNL}(X \triangleright \triangleleft \text{Musicista}) = B(X) + B(X) * B(\text{Musicista}) / M = 28919 + 28919 * 30000 / 20000 = 72297$$

$$\begin{aligned} \text{CostoHJI}(X \triangleright \triangleleft \text{Musicista}) &= (3 - 2M/B(X))(B(X) + B(\text{Musicista})) = \\ &= (3 - 2 * 20000 / 28919) * (28919 + 30000) = 95262 \end{aligned}$$

$$\text{Costo totale} = 57842 + 72297 = 130139$$

## Esercizio: soluzione

---

Seconda sequenza:  $\sigma_{\text{Durata}>30 \text{ AND } \text{Durata}<70}$  (CD  $\triangleright\triangleleft$  Musicista)

Sia  $Z = \text{CD} \triangleright\triangleleft \text{Musicista}$

CD e Musicista sono ordinati quindi posso usare il sort-based join senza il passo preliminare di ordinamento

$$\text{CostoSJ}(Z) = B(\text{CD}) + B(\text{Musicista}) = 41551 + 30000 = 71551$$

$$T(Z) = T(\text{CD}) * T(\text{Musicista}) / \max\{V(\text{CD}, \text{CodiceMus}), V(\text{Musicista}, \text{CodiceMus})\} =$$

$$83102 * 40000 / \max\{40000, 40000\} = 83102$$

## Esercizio: soluzione

---

$$S(Z) = S(\text{CD}) + S(\text{Musicista}) - S(\text{CodiceMus}) = 2000 + 3000 - 200 = 4800$$

$$B(Z) = 83102 * 4800 / 4000 = 99722$$

Dato che non ci sono indici su Z:

$$\text{Costo}(\sigma_{\text{Durata} > 30 \text{ AND } \text{Durata} < 70} (\text{CD} \triangleright \triangleleft \text{Musicista})) = B(Z) = 99722$$

$$\text{Costo totale} = 71551 + 99722 = 171273$$

## Esercizio (13/2/08)

---

- Siano date le relazioni

Paziente(CodicePaz, Nome, Cognome, Età, CodiceMed)

Medico(CodiceMed, Nome, Cognome, Indirizzo)

- la query

```
SELECT P.*, C.*
```

```
FROM Paziente AS P, Medico AS M
```

```
WHERE P.CodiceMed=M.CodiceMed AND P.Età>=10  
AND P.Età<=20
```

# Esercizio

---

- e i parametri:
  - buffer di memoria centrale disponibili:  $M=1000$
  - dimensione del buffer:  $B=1.000$  bytes
  - numero di tuple:  $T(\text{Paziente})= 80000$ ,  $T(\text{Medico})= 10000$ ,
  - dimensione delle tuple:  $S(\text{Paziente})=500$  bytes,  $S(\text{Medico})=400$ ,
  - $S(\text{CodiceMed})=50$
- Si calcoli il costo minimo di ciascuna delle seguenti sequenze:
  - $\sigma_{\text{Età} \geq 10 \text{ AND } \text{Età} \leq 20} (\text{Paziente} \triangleright \triangleleft \text{Medico})$
  - $(\sigma_{\text{Età} \geq 10 \text{ AND } \text{Età} \leq 20} \text{Paziente}) \triangleright \triangleleft \text{Medico}$

# Esercizio

---

- Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record.
- Si supponga di avere un indice primario su Medico.CodiceMed e un indice secondario su Paziente.Età. Per il calcolo del costo e della cardinalità della selezione su Età si utilizzi la stima dei valori nel range, sapendo che il range di Età è [1,105].

## Esercizio: soluzione

---

$$B(\text{Paziente}) = T(\text{Paziente}) * S(\text{Paziente}) / B$$
$$= 80000 * 500 / 1000 = 40000$$

$$B(\text{Medico}) = 10000 * 400 / 1000 = 4000$$

Prima sequenza:  $(\sigma_{\text{Età} \geq 10 \text{ AND } \text{Età} \leq 20} \text{ Paziente}) \triangleright \triangleleft$   
Medico

Sia  $X = \sigma_{\text{Età} \geq 10 \text{ AND } \text{Età} \leq 20} \text{ Paziente}$

C'è un indice secondario su Paziente.Età

$$F = (20 - 10 + 1) / (105 - 1 + 1) = 11 / 105 = 0.105$$

$$\text{Costo}(X) = 3 + f * T(\text{Paziente}) = 3 + 0.105 * 80000 = 8403$$

$$T(X) = 0.105 * 80000 = 8400$$

$$B(X) = 0.105 * 40000 = 4200$$

# Esercizio: soluzione

---

Né X né Medico stanno in memoria centrale, quindi si prova con il join nested-loop basato sui blocchi, con l'hash join ibrido e con il join con indice visto che c'è un indice su CodiceMed.

CostoJNL( $X \bowtie \triangleleft$  Medico)=

$$B(\text{Medico}) + B(X) * B(\text{Medico}) / M = 4000 + 4200 * 4000 / 1000 = 20200$$

CostoHJI( $X \bowtie \triangleleft$  Medico)=  $(3 - 2M/B(\text{Medico}))(B(X) + B(\text{Medico})) = (3 - 2 * 1000/4000) * (4000 + 4200) = 20500$

CostoIJ( $X \bowtie \triangleleft$  Medico)=  $B(X) + T(X) *$

$$\lceil B(\text{Medico}) / V(\text{Medico}, \text{CodiceMed}) \rceil = 4200 + 8400 * 1 = 12600$$

Costo totale=  $8403 + 12600 = 21.003$

# Esercizio: soluzione

---

Seconda sequenza:  $\sigma_{\text{Età} \geq 10 \text{ AND } \text{Età} \leq 20}$  (Paziente  $\triangleright \triangleleft$  Medico)

Sia  $Z = \text{Paziente} \triangleright \triangleleft \text{Medico}$

Né Paziente né Medico stanno in memoria centrale, quindi si prova con il join nested-loop basato sui blocchi, con l'hash join ibrido e con il join con indice visto che c'è un indice su CodiceMed.

$$\text{CostoJNL}(Z) = B(\text{Medico}) + B(\text{Paziente}) * B(\text{Medico}) / M = \\ 4000 + 4000 * 40000 / 1000 = 164.000$$

$$\text{CostoHJI}(Z) = (3 - 2M / B(\text{Medico})) (B(\text{Medico}) + B(\text{Paziente})) = \\ (3 - 2 * 1000 / 4000) * (4000 + 40000) = 110.000$$

$$\text{CostoIJ}(Z) = B(\text{Paziente}) + T(\text{Paziente}) \\ \lceil B(\text{Medico}) / V(\text{Medico}, \text{CodiceMed}) \rceil = 40000 + 80000 * 1 = 120.000$$

# Esercizio: soluzione

---

$$\begin{aligned} T(Z) &= T(\text{Medico}) * T(\text{Paziente}) / \\ &\quad \max\{V(\text{Medico}, \text{CodiceMed}), V(\text{Paziente}, \text{CodiceMed})\} \\ &= 80000 * 10000 / 10000 = 80000 \end{aligned}$$

$$\begin{aligned} S(Z) &= S(\text{Medico}) + S(\text{Paziente}) - S(\text{CodiceMed}) = \\ &\quad 400 + 500 - 200 = 700 \end{aligned}$$

$$B(Z) = 80000 * 700 / 1000 = 56000$$

Dato che non ci sono indici su Z:

$$\text{Costo: } (\sigma_{\text{Età} \geq 10 \text{ AND } \text{Età} \leq 20} Z) = B(Z) = 56000$$

$$\text{Costo totale} = 56000 + 110000 = 166.000$$

## Esercizio (21/6/2012)

---

- Siano date la seguenti relazioni

Agenzia(CodAg, Nome, Via, Città, CAP, Telefono, Fax)

Cliente(CodiceCli, Nome, Cognome, Via, Città, CAP, Telefono, CodAg, Età)

- la query

```
SELECT A.*, C.*
```

```
FROM Agenzia AS A, Cliente AS C
```

```
WHERE A.Città = C.Città AND C.Età >=30
```

## Esercizio (21/6/2012)

---

e i parametri:

- buffer di memoria centrale disponibili:  $M=10000$
- dimensione del buffer:  $B=4000$  bytes
- numero di tuple:  $T(\text{Cliente})=200000$ ,  $T(\text{Agenzia})=50000$ ,
- dimensione delle tuple:  $S(\text{Cliente})=2000$ ,  
 $S(\text{Agenzia})=2000$ ,  $S(\text{Città})=200$
- numero di valori  $V(\text{Cliente}, \text{Città})=20000$ ,  $V(\text{Agenzia}, \text{Città})=1000$
- $\text{Min}(\text{Cliente}, \text{Età})=20$   $\text{Max}(\text{Cliente}, \text{Età})=60$

## Esercizio (21/6/2012)

---

- Si calcoli il costo minimo di ciascuna delle seguenti sequenze:

$(\sigma_{\text{Età} \geq 30} \text{ Cliente}) \triangleright \triangleleft \text{ Agenzia}$

$\sigma_{\text{Età} \geq 30} (\text{Cliente} \triangleright \triangleleft \text{ Agenzia})$

- Si indichi quale delle sequenze ha costo inferiore.
- Si supponga che i record siano impaccati nei buffer e che tutto lo spazio nei buffer sia occupato dai record.
- Si supponga di avere un indice secondario su Cliente.Età.

# Esercizio: soluzione

---

- Calcoliamo innanzitutto il numero di blocchi occupati da ciascuna relazione:
- $B(\text{Cliente}) = T(\text{Cliente}) * S(\text{Cliente}) / B$   
 $= 200000 * 2000 / 4000 = 100000$
- $B(\text{Agenzia}) = 50000 * 2000 / 4000 = 25000$

# Esercizio: soluzione

---

- Prima sequenza:  $(\sigma_{\text{Età} \geq 30} \text{ Cliente}) \triangleright \triangleleft \text{ Agenzia}$
- Sia  $X = (\sigma_{\text{Età} \geq 30} \text{ Cliente})$
- C'e' un induce secondario su Cliente.Età
- $f = (60 - 30 + 1) / (60 - 20 + 1) = 0.756$
- $\text{Costo}(X) = 3 + f * T(\text{Cliente}) = 3 + 0.756 * 200000 = 151203$
- $B(X) = 0.756 * 100000 = 75600$

## Esercizio: soluzione

---

- Né Agenzia né X stanno in memoria centrale, quindi si prova con il join nested-loop basato sui blocchi e con l'hash join ibrido.
- Sia  $Z = X \bowtie \triangleleft$  Agenzia
- $\text{CostoJNL}(Z) = B(\text{Agenzia}) + B(X) * B(\text{Agenzia}) / M$   
 $= 25000 + 75600 * 25000 / 10000 = 214000$
- $\text{CostoHJI}(Z) = (3 - 2M / B(\text{Agenzia})) (B(X) + B(\text{Agenzia})) =$   
 $(3 - 2 * 10000 / 25000) * (75600 + 25000) = 221320$
- Costo totale =  $151203 + 214000 = 365203$

# Esercizio: soluzione

---

- Seconda sequenza:  $\sigma_{Età \geq 30} (\text{Cliente} \triangleright \triangleleft \text{Agenzia})$
- Sia  $Z = \text{Cliente} \triangleright \triangleleft \text{Agenzia}$
- Né Agenzia né Cliente stanno in memoria centrale, quindi si prova con il join nested-loop basato sui blocchi e con l'hash join ibrido.
- $\text{CostoJNL}(Z) = B(\text{Agenzia}) + B(\text{Cliente}) * B(\text{Agenzia}) / M$   
 $= 25000 + 100000 * 25000 / 10000 = 275000$
- $\text{CostoHJI}(Z) = (3 - 2M / B(\text{Agenzia})) * (B(\text{Cliente}) + B(\text{Agenzia}))$   
 $= (3 - 2 * 10000 / 25000) * (100000 + 25000) = 275000$

# Esercizio: soluzione

---

- $T(Z) = T(\text{Agenzia}) * T(\text{Cliente}) / \max\{V(\text{Cliente}, \text{Città}), V(\text{Agenzia}, \text{Città})\} = 50000 * 200000 / 20000 = 500000$
- $S(Z) = S(\text{Cliente}) + S(\text{Agenzia}) - S(\text{Città}) = 2000 + 2000 - 200 = 3800$
- $B(Z) = 500000 * 3800 / 4000 = 475000$
- Dato che non ci sono indici su Z:
- Costo  $\sigma_{\text{Età} \geq 30}(Z) = B(Z) = 475000$
- Costo totale =  $275000 + 475000 = 750000$
- La sequenza di costo minore è la prima