



UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA  
DIPARTIMENTO DI SCIENZE TEORICHE E APPLICATE

### Basi di Dati II

04 Giugno 2024 -- Pag 1/2

(Tempo a disposizione totale: 2 ore) TEMA A

- Lo studente deve rispondere in modo esauriente alla seguente lista di domande.
- Lo studente è tenuto a rispondere ad ogni domanda con una calligrafia comprensibile.
- Ogni foglio che verrà consegnato deve riportare la data dell'esame, nome, cognome e numero di matricola dello studente.
- **Tutte le risposte devono essere completate con adeguata motivazione**

SONO RIPORTATE SINTETICAMENTE LE RISPOSTE ALLE DOMANDE.

**SI RICORDA CHE ALL'ESAME LO STUDENTE E' TENUTO A MOTIVARE LE RISPOSTE.**

#### Quiz 1 (pt 1.5)

Si consideri il seguente file di log

B(T1)  
U(T1, O1, B1, A1)  
B(T2)  
U(T2, O2, B2, A2)  
B(T3)  
U(T3, O1, B3, A3)  
U(T2, O3, B4, A4)  
C(T1)  
U(T3, O2, B5, A5)  
B(T4)  
U(T4, O1, B6, A6)  
U(T3, O3, B7, A7)  
C(T2)  
CK(T3, T4)  
U(T4, O2, B8, A8)  
C(T3)  
B(T5)  
U(T5, O1, B9, A9)  
U(T4, O3, B10, A10)  
C(T4)  
B(T6)  
U(T6, O2, B11, A11)  
U(T5, O2, B12, A12)  
C(T5)

Ipotizzando un failure, dopo una ripresa a caldo che valori hanno gli oggetti O1 e O2?

[2.a] O1= A9, O2= B11

**[2.b] O1= A9, O2= A12**

[2.c] O1= B9, O2= B12

[2.d] O1= A6, O2= B5

[2.e] nessuna delle risposte precedenti

#### Quiz 2 (pt 1.5)

Si supponga di avere una relazione R(A,...) contenente 200.000 record. Quanti livelli (compreso la root) e sono necessari per indicizzare tutti i valori di A con un indice B-tree di ordine 150? Quanti nodi sono necessari per memorizzare l'indice?

[1.a] 2 livelli, 22651 nodi

[1.b] 2 livelli, 22500 nodi

[1.c] 3 livelli, 22500 nodi

**[1.d] 3 livelli, 22651 nodi**

[1. e] Nessuna delle risposte precedenti

### ESERCIZIO 1 (pt. 7)

Si consideri il seguente schema relazionale:

Biblioteche(bid, nomeB, città, cap)  
Libri(isbn, titolo, autore, annoPubblicazione, biblioteca<sup>Biblioteche</sup>)  
Prestiti(cliente<sup>Clienti</sup>, libro<sup>Libri</sup>, dataInizio, dataFine)  
Clienti(CF, nome, cognome, Città, annoNascita)

1. Scrivere un'espressione algebrica che restituisca il CF dei clienti nati il 2004 che hanno in prestito libri (i.e., non li hanno ancora restituiti, dataFine=null) solo da biblioteche di Varese

L'espressione algebrica che restituisce il CF dei clienti del 2004 che hanno in prestito (ossia, 'dataFine=NULL') libri presi da biblioteche di Varese è la seguente

$$\pi_{CF}(\sigma_{AnnoNascita = 2004}(Clienti) \bowtie \sigma_{CF=cliente}(\sigma_{dataFine = NULL}(Prestiti) \bowtie_{Libro=ISBN} (Libri \bowtie \sigma_{biblioteca=bid}(\sigma_{città='Varese'}(Biblioteche)))))$$

Questa espressione non garantisce che tutti i prestiti siano stati fatti solo in biblioteche di Varese. Questo lo possiamo garantire togliendo ( - ) dall'espressione precedente i CF dei clienti del 2004 con prestiti da biblioteche non di Varese.

L'espressione finale è la seguente:

$$\pi_{CF}(\sigma_{AnnoNascita = 2004}(Clienti) \bowtie \sigma_{CF=cliente}(\sigma_{dataFine = NULL}(Prestiti) \bowtie_{Libro=ISBN} (Libri \bowtie \sigma_{biblioteca=bid}(\sigma_{città='Varese'}(Biblioteche)))))$$
$$\pi_{CF}(\sigma_{AnnoNascita = 2004}(Clienti) \bowtie \sigma_{CF=cliente}(\sigma_{dataFine = NULL}(Prestiti) \bowtie_{Libro=ISBN} (Libri \bowtie \sigma_{biblioteca=bid}(\sigma_{città \neq 'Varese'}(Biblioteche)))))$$

2. Per l'espressione ottenuta al punto 1, disegnare il query tree (albero dell'interrogazione) ottimizzato

3. Si consideri due relazioni Prestiti e Clienti con la seguente configurazione

- Prestiti occupa 8 blocchi identificati con i seguenti indirizzi: P1, P2, P3, .....P7, P8
- Clienti occupa 5 blocchi identificati con i seguenti indirizzi: C1, C2, C3, C4, C5

Considerare l'esecuzione ottimizzata di un join tra Prestiti e Clienti con nested loop. Si assuma che il buffer metta a disposizione in totale 4 blocchi per implementare il join. Indicare nell'ordine di esecuzione le operazioni di carico dei blocchi nel buffer (i.e., fix), specificando l'indirizzo del blocco (e.g., fix(R8)) e le operazioni di scarico (e.g., unfix(R8)).

L'esecuzione ottimizzata del nested loop join tra Prestiti e Clienti, prevede la relazione più piccola come outer. I 4 blocchi sono usati così: 1 per output, 1 per Prestiti e 2 per Clienti.

L'ordine di esecuzione è

Fix(C1), Fix(C2), Fix(P1)  
Unfix(P1), Fix(P2)  
Unfix(P2), Fix(P3)  
Unfix(P3), Fix(P4)  
Unfix(P4), Fix(P5)  
Unfix(P5), Fix(P6)  
Unfix(P6), Fix(P7)  
Unfix(P7), Fix(P8)  
Unfix(C1), Unfix(C2),

Fix(C3), Fix(C4),  
 Unfix(P8), Fix(P1)  
 Unfix(P1), Fix(P2)  
 .....  
 Unfix(P7), Fix(P8)  
 Unfix(C3), Unfix(C4),  
 Fix(C5)  
 Unfix(P8), Fix(P1)  
 Unfix(P1), Fix(P2)  
 .....  
 Unfix(P7), Fix(P8)  
 Unfix(C5), Unfix(P8)

## ESERCIZIO 2 (pt. 6.5)

Si ipotizzi la tabella CLIENTI con 100.000 record di lunghezza fissa costituiti dai seguenti campi:

- CF (16 byte), campo unique
- NOME (30 byte),
- COGNOME (30 byte),
- CITTA (9 byte),
- ANNONASCITA(8 Byte)
- Un ulteriore byte utilizzato come indicatore di cancellazione record.

Si supponga la tabella sia memorizzata in un file non ordinato su un disco con dimensione di blocco  $B = 512$  byte, dove un puntatore al blocco occupa 6 byte e un puntatore al record occupa 7 byte

Rispondere alle seguenti domande:

2.1) Assumendo che su CF è costruito un indice B-tree con fattore di riempimento al 80%, calcolare il numero di accessi necessari per cercare e reperire un record dal file, dato il valore di CF

Si calcola l'ordine dei nodi foglia e interno

$P_{\text{interno}} = 18 \rightarrow$  si riduce all'80%  $P_{\text{interno}} = 14$

$P_{\text{foglia}} = 23 \rightarrow$  si riduce all'80%  $P_{\text{interno}} = 18$

E' necessario un albero di 5 livelli

Il costo di ricerca e accesso è  $5+1 = 6$  I/O

2.2) Assumendo che su CITTA è costruito un indice B+-tree con fattore di riempimento al 80%, calcolare il numero di accessi necessari per cercare e reperire un record dal file, dato il valore di CITTA

Si calcola l'ordine dei nodi foglia e interno

$P_{\text{interno}} = 34 \rightarrow$  si riduce all'80%  $P_{\text{interno}} = 27$

$P_{\text{foglia}} = 32 \rightarrow$  si riduce all'80%  $P_{\text{interno}} = 25$

E' necessario un albero di 4 livelli

Il costo di ricerca e accesso è  $4+1 = 5$  I/O

2.3) Assumendo che su CITTA è costruito un indice secondario, calcolare

il numero di accessi necessari per cercare e reperire un record dal file, dato il valore di CITTA

Il fattore di blocco dell'indice secondario è

Voce indice  $9 + 6 = 15$

Fattore blocco  $512/15 = 34$

I blocchi dell'indice sono quindi:  $100.000/34 = 2942$

Ricerca binaria su indice secondario:  $\log_2 2942 = 12$

Il costo di ricerca e accesso è  $12 + 1 = 13$  I/O

### ESERCIZIO 3 (pt. 6.5)

---

Considerare un file che occupa 550.000 blocchi nella memoria secondaria. Si ipotizzi l'esecuzione di un algoritmo di merge sort esterno generico con 10 blocchi di buffer disponibili. Rispondere alle seguenti domande:

- 2.1) Quanti run sono creati al passo iniziale? 55.000
- 2.2) Quanti passaggi saranno necessari per ordinare completamente il file? 6 Passaggi
- 2.2) Assumendo che un'operazione I/O richiede 8ms, quanti secondi impiega l'ordinamento del file? 52800 sec.

### ESERCIZIO 4 (pt. 7)

Indicare quale dei seguenti schedule è CSR, VSR o non serializzabili

- 4.1 r2(a) r1(a) w1(c) r3(c) w1(b) r4(b) w3(b) r4(c) w2(d) r2(b) w4(a) w4(b) no csr, no vsr
  - 4.2 r1(a) r2(a) w1(a) r3(b) w2(a) r4(b) w3(b) w4(b) r2(b) w2(b) no csr, no vsr
  - 4.3 r1(a) w3(a) r1(b) w1(a) w2(b) r1(d) w2(a) r3(a) no csr, no vsr
-



UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA  
DIPARTIMENTO DI SCIENZE TEORICHE E APPLICATE

### Basi di Dati II

04 Giugno 2024 -- Pag 1/2

(Tempo a disposizione totale: 2 ore) TEMA B

- Lo studente deve rispondere in modo esauriente alla seguente lista di domande.
- Lo studente è tenuto a rispondere ad ogni domanda con una calligrafia comprensibile.
- Ogni foglio che verrà consegnato deve riportare la data dell'esame, nome, cognome e numero di matricola dello studente.
- **Tutte le risposte devono essere completate con adeguata motivazione**

SONO RIPORTATE SINTETICAMENTE LE RISPOSTE ALLE DOMANDE.

**SI RICORDA CHE ALL'ESAME LO STUDENTE E' TENUTO A MOTIVARE LE RISPOSTE.**

#### Quiz 1 (pt 1.5)

Si supponga di avere una relazione  $R(A, \dots)$  contenente 250.000 record. Quanti livelli (compreso la root) e sono necessari per indicizzare tutti i valori di A con un indice B-tree di ordine 200? Quanti nodi sono necessari per memorizzare l'indice?

[1.a] 2 livelli, 22650 nodi

[1.b] 3 livelli, 40201 nodi

[1.c] 2 livelli, 40201 nodi

[1.d] 3 livelli, 40000 modi

[1. e] Nessuna delle risposte precedenti

.ò

#### Quiz 2 (pt 1.5)

Si consideri il seguente file di log

B(T1)

U(T1, O1, B1, A1)

B(T2)

U(T2, O2, B2, A2)

B(T3)

U(T3, O1, B3, A3)

U(T2, O3, B4, A4)

C(T1)

U(T3, O2, B5, A5)

B(T4)

U(T4, O1, B6, A6)

U(T3, O3, B7, A7)

C(T2)

CK(T3, T4)

U(T4, O2, B8, A8)

B(T5)

U(T5, O1, B9, A9)

U(T4, O3, B10, A10)

C(T4)

B(T6)

\O2, B11, A11)

U(T5, O2, B12, A12)

Ipotizzando un failure, dopo una ripresa a caldo che valori hanno gli oggetti O1 e O2?

[2.a] O1= A9, O2= B11

[2.b] O1= A6, O2= B5

[2.c] O1= A9, O2= B5

[2.d] O1= A6, O2= B12

[2.e] nessuna delle risposte precedenti

### ESERCIZIO 1 (pt. 6.5)

Considerare un file che occupa 500.000 blocchi nella memoria secondaria. Si ipotizzi l'esecuzione di un algoritmo di merge sort esterno generico con 12 blocchi di buffer disponibili. Rispondere alle seguenti domande:

- 2.1) Quanto sono grandi (# blocchi) i run creati al passo iniziale? Ogni run contiene 12 blocchi
- 2.2) Quanti passaggi saranno necessari per ordinare completamente il file? 6 Passaggi
- 2.3) Di quanti blocchi di buffer si ha bisogno per ordinare completamente il file in due soli passaggi? 708 blocchi

### ESERCIZIO 2 (pt. 7)

Si consideri il seguente schema relazionale:

Concessionaria(cid, nomeC, città, cap)

Auto(TargaA, modello, annoImmatricolazione, concessionaria<sup>Concessionarie</sup>)

Noleggi(cliente<sup>Clients</sup>, Targa<sup>Auto</sup>, dataInizio, dataFine)

Clients(CF, nome, cognome, Città, annoNascita)

1. Scrivere un'espressione algebrica che restituisca il CF dei client che hanno in noleggio auto immatricolate nel 2022 (i.e., non hanno ancora restituite, dataFine=null) solo da concessionarie di Milano.

Per scrivere un'espressione algebrica che restituisca il CF dei clienti che hanno noleggio auto immatricolate nel 2022 (ossia, non ancora restituite, `dataFine=NULL`) \*\*solo\*\* da concessionarie di Milano, dobbiamo assicurarci che i clienti con auto del 2022 noleggiati da concessionarie di Milano non abbiano anche noleggi attivi da altre concessionarie. Questo lo possiamo fare con espressione algebrica che computa il CF dei clienti con noleggi attivi di auto del 2022 di concessionarie di Milano, al cui risultato togliamo (-) i CF dei clienti con noleggi attivi di auto del 2022 di concessionarie non di Milano. L'espressione finale è la seguente:

$$\pi_{\text{cliente}}((\sigma_{\text{dataFine}=\text{NULL}}(\text{Noleggi}) \bowtie_{\text{Targa}=\text{TargaA}} (\sigma_{\text{annoImmatricolazione}=2022}(\text{Auto}) \bowtie \sigma_{\text{concessionaria}=\text{cid}}(\sigma_{\text{città}='Milano'}(\text{Concessionaria})))) \\ - \\ \pi_{\text{cliente}}((\sigma_{\text{dataFine}=\text{NULL}}(\text{Noleggi}) \bowtie_{\text{Targa}=\text{TargaA}} (\sigma_{\text{annoImmatricolazione}=2022}(\text{Auto}) \bowtie \sigma_{\text{concessionaria}=\text{cid}}(\sigma_{\text{città} \neq 'Milano'}(\text{Concessionaria}))))))$$

2. Per l'espressione ottenuta al punto 1, disegnare il query tree (albero dell'interrogazione) ottimizzato
3. Si consideri due relazioni Noleggi e Auto con la seguente configurazione

- Noleggi occupa 10 blocchi identificati con i seguenti indirizzi: N1, N2, ....., N9, N10
- Clienti occupa 6 blocchi identificati con i seguenti indirizzi: C1, C2, ....., C5, C6

Considerare l'esecuzione ottimizzata di un join tra Noleggi e Clienti con nested loop. Si assuma che il buffer metta a disposizione in totale 5 blocchi per implementare il join. Indicare nell'ordine di esecuzione le operazioni di carico dei blocchi nel buffer (i.e., fix), specificando l'indirizzo del blocco (e.g., fix(R8)) e le operazioni di scarico (e.g., unfix(R8)).

L'esecuzione ottimizzata del nested loop join tra Noleggi e Clienti, prevede la relazione più piccola come outer. I 5 blocchi sono usati così: 1 per output, 1 per Noleggi e 3 per Clienti.

L'ordine di esecuzione è

Fix(C1), Fix(C2), Fix(C3), Fix(N1)

Unfix(N1), Fix(N2)

Unfix(N2), Fix(N3)

Unfix(N3), Fix(N4)

Unfix(N4), Fix(N5)

Unfix(N5), Fix(N6)

Unfix(N6), Fix(N7)

Unfix(N7), Fix(N8)

Unfix(C1), Unfix(C2), Unfix(C3)

Fix(C4), Fix(C5), Fix(C6)

Unfix(N1), Fix(N2)

Unfix(N2), Fix(N3)

....

Unfix(N7), Fix(N8)

Unfix(C4), Unfix(C5), Unfix(C6)

Unfix(N8)

### ESERCIZIO 3 (pt. 6.5)

Si ipotizzi la tabella CLIENTI con 100.000 record di lunghezza fissa costituiti dai seguenti campi:

- CF (16 byte), campo unique
- NOME (30 byte),
- COGNOME (30 byte),
- CITTA (9 byte),
- ANNONASCITA(8 Byte)
- Un ulteriore byte utilizzato come indicatore di cancellazione record.

Si supponga la tabella sia memorizzata in un file non ordinato su un disco con dimensione di blocco  $B = 512$  byte, dove un puntatore al blocco occupa 6 byte e un puntatore al record occupa 7 byte

Rispondere alle seguenti domande:

3.1 Calcolare il fattore di blocco e il numero di blocchi del file supponendo un'organizzazione unspanned **5, 20000**

3.2 Calcolare il numero di accessi necessari per cercare e reperire un record dal file, dato il valore di CF, usando un indice secondario su CF

Fattore di blocco per file indice=  $512/22 = 23$ . NB. si può considerare sia il puntatore al blocco sia quello al record.

Blocchi per file indice  $100000/23 = 4348$

Ricerca binaria su file indice:  $\log_2 4348 = 13$

Il costo di ricerca e accesso è  $13 + 1 = 14$  I/O

3.3 Calcolare il numero di accessi necessari per cercare e reperire un record dal file, dato il valore di CF, usando l'indice multilivello su CF

L'indice secondario rappresenta il primo livello

Sono necessari 4 livelli (inclusa la root)

Il costo di ricerca e accesso è  $4 + 1 = 5$  I/O

### ESERCIZIO 4 (pt. 7)

Indicare quale dei seguenti schedule è CSR, VSR o non serializzabili

4.1) r1(d) r3(a) w2(b) r2(c) w4(c) r1(b) r2(a) w3(a) w2(d) r4(d) w3(c) r4(b) w1(a) r3(d) **no csr, no vsr**

4.2) r1(a) r4(b) w1(c) r4(c) w2(b) r3(b) w1(a) w2(a) w3(c) w4(a) w3(a) **no csr, si vsr**

4.3) w3(a) r1(b) w2(b) r1(d) w2(a) r3(a) w1(a) r1(a) **no csr, no vsr**