UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA

DIPARTIMENTO DI SCIENZE TEORICHE E APPLICATE

Basi di Dati II

09 Giugno 2023

(Tempo a disposizione totale: 2 ore)

ISTRUZIONI

- 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 rispose devono essere completate con adeguata motivazione

Quiz 1 (pt 1) (in giallo è evidenziata la risposta corretta)

Si consideri il seguente file di log

B(T5) B(T6) U(T6, O1, B1, A1) I(T5, O2, A2) B(T7) C(T5) B(T8) U(T7, O2, B3, A3) U(T8, O3, B4, A4) C(T8) CK(T6,T7) U(T7, O5, B5, A5) A(T7) B(T9) C(T6)

Ipotizzando un failure e assumendo una ripresa a caldo, quali sono le transazioni da annullare (UNDO_SET) e le transazioni da rieseguire (REDO_SET)

[1.a] UNDO_SET={T9}, REDO_SET={T6, T7}
[1.b] UNDO_SET={T7, T9}, REDO_SET={T6}
[1.c] UNDO_SET={T9}, REDO_SET={T6}
[1.d] UNDO_SET={T7, T9}, REDO_SET={}

Quiz 2 (pt 1)

Si supponga di avere una relazione R(A,….) contenente 100.000 record. Quanti livelli (compreso la root) sono necessari per indicizzare tutti i valori di A con un indice B-tree di ordine 100?

[1.a] 2 livelli
[1.b] 4 livelli
[1.c] 3 livelli
[1.d] 5 livelli

Quiz 3 (pt 1)

Indicare cosa restituisce la seguente espressione algebrica

$\rho_{\text{NomeDip,CittàDip,Manager} \neg \text{ NomeDipC,CittàDipC,ManagerC}}(\text{DIPARTIMENTO})$

[1.a] La relazione DIPARTIMENTO dove gli attributi che memorizzano il nome, la città e il manager del dipartimento hanno nome *NomeDipC*, *CittàDipC*, *ManagerC*

[1.b] Le tuple di DIPARTIMENTO con attributi che soddisfano queste condizioni: Nome=*NomeDipC*, città=*CittàDipC*, e Manager=*ManagerC*

[1.c] La relazione DIPARTIMENTO dove gli attributi che memorizzano il nome, la città e il manager del dipartimento hanno nome *NomeDip*, *CittàDip*, *Manager*

[1.d] Solo gli attributi *NomeDip*, *CittàDip*, *Manager* delle tuple della relazione DIPARTIMENTO

ESERCIZIO 1 (pt. 7)

Si consideri il seguente schema di base di dati per gestione di spettacoli teatrali

PERSONE(CF, nome, cognome, nazionalità, AnnoNascita)
SPETTACOLI(Titolo, regista$^{\text{Persone}}$, autore$^{\text{Persone}}$, genere)
CAST(TitoloC$^{\text{Spettacoli}}$, registaC$^{\text{Spettacoli}}$, attore$^{\text{Persone}}$, ruolo)

1.1) Scrivere un'espressione algebrica che restituisca il nome e cognome dell'attore nato nel 2000 di nazionalità Italiana che è stato attore con ruolo da protagonista in uno spettacolo di genere Musical

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

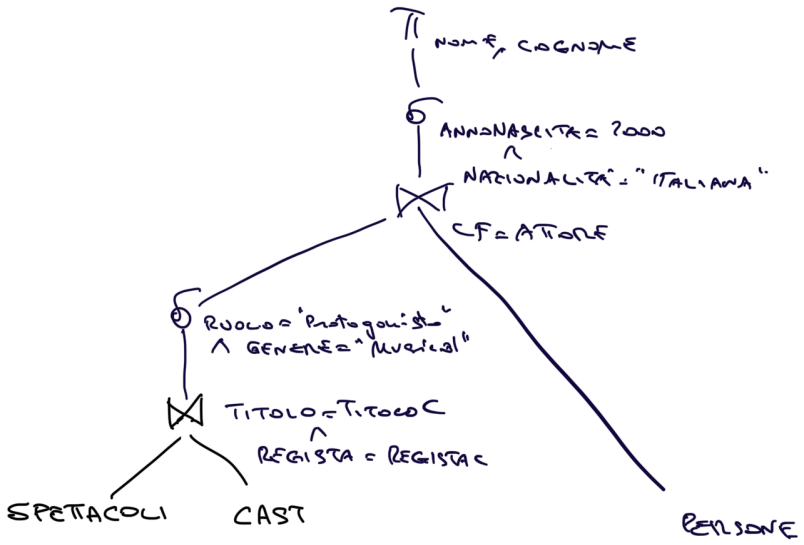
1.3) Indicare quante operazioni I/O richiede l'esecuzione ottimizzata di un join tra PERSONE e CAST con l'algoritmo nested loop e ipotizzando la seguente configurazione:
Il buffer mette a disposizione per l'esecuzione 10 blocchi
la tabella PERSONE occupa 1000 blocchi
la tabella CAST occupa 100 blocchi

SOLUZIONE ESERCIZIO 1

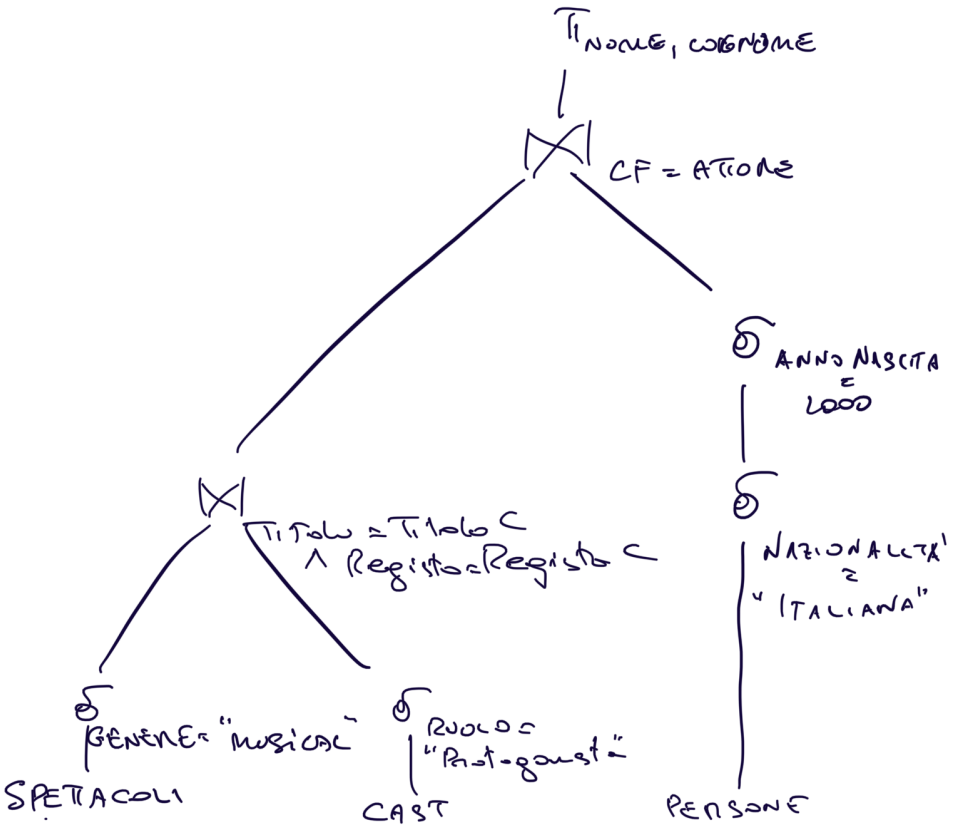
1.1 Una possibile espressione algebrica è la seguente:

$\pi_{\text{nome,cognome}}(\sigma_{\text{AnnoNascita=2000 AND Nazionalità="italiana"}}(\text{PERSONE} \bowtie_{\text{CF=Attore}} (\sigma_{\text{Ruolo="Protagonista" AND Genere="Musical"}}(\text{SPETTACOLI} \bowtie_{\text{titolo=titoloC AND Regista=RegistaC}} \text{CAST}))))$

1.2 Il query tree associato all'espressione definita nel punto 1.1 è il seguente



Per definire la versione ottimizzata si sfrutta la proprietà commutativa del σ, si sposta ogni operatore σ quanto più possibile verso il basso del query tree. Sfruttando la proprietà commutativa del π, si sposta l'operatore π quanto più possibile verso il basso dell'albero di interrogazione, mantenendo solo gli attributi necessari per eseguire le operazioni successive. Il query ottimizzato è il seguente:



1.3 Sia:

B(PERSONA)= 2000 il numero di blocchi per la tabella PERSONA
B(CAST)=100 il numero di blocchi per la tabella CAST
B=10 il numero di blocchi del buffer a disposizione

Dei 10 blocchi di buffer per l'operazione di Join, 9 vengono destinati alla memorizzazione dei blocchi di input ed 1 per la memorizzazione dell'output del join. Tra i 9 blocchi per l'input, 8 sono destinati alla relazione OUTER e 1 alla relazione INNER. Per ottimizzare il nested loop, si tiene come relazione outer, la relazione con minor numero di blocchi, ovvero CAST. Si esegue quindi il nested loop CAST ⋈ PERSONA.

Il numero di operazioni I/O è dato da
B(CAST) +(⌈ B(CAST)/**B** ⌉ * B(PERSONE))
=100 + (⌈ 100/8 ⌉ * 1000)= 100+(13*1000)= 13100

Esercizio 2 (pt 6.5)

Considerare un file che occupa 3.000.000 blocchi nella memoria secondaria. Si ipotizzi l'esecuzione di un algoritmo di merge sort esterno generico con 15 blocchi di buffer disponibili. Rispondere alle seguenti domande:

2.1) Quanti run verranno prodotti nel primo passaggio?
2.2) Quanti passaggi saranno necessari per ordinare completamente il file?
2.3) Qual è il costo totale di I/O dell'ordinamento del file?

SOLUZIONE ESERCIZIO 2

2.1) L'algoritmo di merge sort esterno richiede un passo iniziale (Passo 0) per il sort iniziale. In questo passo si ottengono ⌈ N/B ⌉, dove N è il numero di blocchi dati da riordinare e B il numero di blocchi di buffer disponibili. Il numero di run è quindi ⌈3.000.000/15 ⌉= 200.000.

2.2) Dopo il passo 0, seguono un numero di passi di merge sui run ottenuti al passo 0. Il numero di merge con 15 buffer su 200.000 run è pari a $\lceil \log_{14} 200.000 \rceil = 5$. Il numero di passaggi totale è 1+5=6.

2.3) Il costo totale è 2N* numero passi, dove N è il numero di blocchi da ordinare.
Quindi 2*(3.000.000) *6= 36.000.000

Esercizio 3 (pt. 7)

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

- CF (16 byte), campo unique
- NOME (30 byte),
- COGNOME (30 byte),
- NAZIONALITA' (9 byte),
- ANNONASCITA (4 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.
Rispondere alle seguenti domande:

3.1) Calcolare il fattore di blocco e il numero di blocchi del file supponendo un'organizzazione unspanned
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
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

SOLUZIONE ESERCIZIO 3

3.1) Il record ha una dimensione totale di 90 byte. Il fattore di blocco del file dati è B/90= 5
Il numero di blocchi del file è quindi dato da ⌈100.000/5 ⌉ =20.000

3.2) Per avere una stima del numero di accessi, devo calcolare il numero di blocchi contenenti le voci dell'indice su cui fare la ricerca binaria.
Il file non è ordinato su CF, quindi l'indice è denso. E' necessaria una voce per ogni record, quindi 100.000 voci. Ogni voce richiede 22 byte, 16 per il campo CF e 6 per il puntatore al blocco.
In un blocco da 512 si possono memorizzare ⌈512/22⌉=23 voci. L'indice secondario occupa, quindi, ⌈100.000/23⌉= 4.348 blocchi.
La ricerca sull'indice secondario è una ricerca binaria su 4.348 blocchi e richiede $\log_2(4.348)$= 13 operazioni I/O
Il costo per recuperare il record è quindi il costo della ricerca (13 operazioni I/O) più il costo della lettura del blocco dati, ovvero 13+1=14

3.3) Per stimare il costo della query devo calcolare di quanti livelli è costituito l'indice multilivello su CF. Il primo livello è l'indice secondario calcolato nel punto 3.2.
Livello 2 - Il secondo livello è un indice sparso sull'indice primo livello, con una voce per blocco dell'indice del primo livello. 4.348 voci richiedono un numero di blocchi pari a ⌈4.348/23⌉= 190.
Livello 3 – come per il precedente. 190 voci richiedono un numero di blocchi pari a ⌈190/23⌉= 9
Livello 4 – 9 voci richiede un solo blocco.
L'indice multilivello si compone quindi di 4 livelli.
Per ricerca il record nell'indice multilivello è necessario leggere 4 blocchi (uno per livello). Il costo per reperire un record sull'indice è quindi 4 (ricerca sull'indice) + 1 (lettura blocco dati), ovvero 5 operazioni I/O.

Esercizio 4 (pt. 6.5)

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

4.1) r2(c) w2(a) r1(b) w3(c) r5(b) w5(d) w3(a) w2(d) w3(b) w1(d) r4(c)
4.2) w1(d)r1(a)w2(a)r1(d)r2(d)w1(a)
4.3) r2(a)r3(c)w3(a)w2(d)r2(b)w2(c)

SOLUZIONE ESERCIZIO 4

4.1)

$S_1 = r_2(c)\, w_2(a)\, r_1(b)\, w_3(c)\, r_5(b)\, w_5(d)\, w_3(a)$
$w_2(d)\, w_3(b)\, w_1(d)\, r_4(c)$

E' CSR?

① CONSIDERO I CONFLITTI SULLE RISORSE

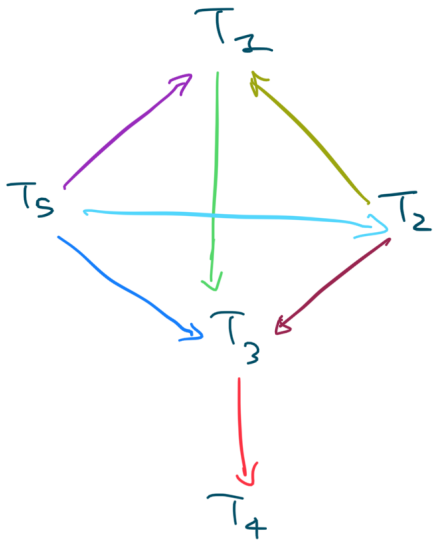
a = w_2 w_3
b = r_1 r_5 w_3
c = r_2 w_3 r_4
d = w_5 w_2 w_1

CONFLITTI

w_2 w_3
r_1 w_3
r_5 w_3
r_2 w_3
w_3 r_4
w_5 w_1
w_5 w_2
w_5 w_1
w_2 w_1

② COSTRUISCO IL GRAFO DEI CONFLITTI

IL GRAFO NON HA CICLI, QUINDI S_1 E' CSR E DI CONSEGUENZA ANCHE VSR.



4.2)

$S_2 = w_1(d)\, r_1(a)\, w_2(a)\, r_1(d)\, r_2(d)\, w_1(a)$

E' CSR?

① CONSIDERO I CONFLITTI SULLE RISORSE

$a_2\ r_1\ w_2\ w_1$

$d_2\ w_1\ r_1\ r_2$

CONFLITTI

$r_1\ w_2$

$w_2\ w_1$

$w_1\ r_2$

② COSTRUISCO IL GRAFO DEI CONFLITTI

IL GRAFO HA UN CICLO, QUINDI S_2 NON E' CSR.

E' USR?

① DEFINISCO LE RELAZIONI LEGGI-DA DI S_2

$w_1(d)\ r_2(d)$

② DETERMINO LE ULTIME SCRITTURE IN S_2

Per a : w_2

Per d : w_2

③ PER TROVARE UNO SCHEDULE SERIALE USR EQUIVALENTE A S_2 CONSIDERO LE POSSIBILI COMBINAZIONI DELLE SUE TRANSAZIONI

$T_1 = w_2(d)\, r_1(e)\, r_1(d)\, w_2(a)$

$T_2 = w_2(a)\, r_2(d)$

- $T_1 T_2$

$= w_2(d)\, r_1(e)\, r_1(d)\, w_2(a)\, w_2(a)\, r_2(d)$

ULTIME SCRITTURE

Per a : w_2

Per d : w_1 $\neq$ DA ULTIME SCRITTURE DI S_2.

- $T_2 T_1$

$= w_2(a)\, r_2(d)\, w_2(d)\, r_1(e)\, r_1(d)\, w_2(a)$

ULTIME SCRITTURE

Per a : w_1

Per d : w_1 $=$ ALLE ULTIME SCRITTURE DI S_2.

RELAZIONI LEGGI-DA

$w_2(a),\ r_1(a)$ $\neq$ ALLE RELAZIONI LEGGI-DA DI S_2

$\Rightarrow$ S_2 NON E' USR, PERCHE' NON ESISTE UNO SCHEDULE SERIALE USR EQUIVALENTE A S_2

4.3)

$S_3 = r_2(a)\ r_3(c)\ w_3(a)\ w_2(d)\ r_2(b)\ w_2(c)$

E' CSR?

① CONSIDERO I CONFLITTI SULLE RISORSE

$a = r_2\ w_3$
$b = r_2$
$c = r_3\ w_2$
$d = w_2$

CONFLITTI
$r_2\ w_3$
$r_3\ w_2$

② COSTRUISCO IL GRAFO DEI CONFLITTI

IL GRAFO HA UN CICLO QUINDI S_3 NON E' CSR.

E' VSR?

① DEFINISCO LE RELAZIONI LEGGI-DA DI S_3

NESSUNA

② DETERMINO LE ULTIME SCRITTURE

Per a: w_3
Per b: /
Per c: w_2
Per d: w_2

③ PER TROVARE UNO SCHEDULE SERIALE VSR EQUIVALENTE A S_3, CONSIDERO LE POSSIBILI COMBINAZIONI DELLE SUE TRANSAZIONI

$T_2 = r_2(a)\ w_2(d)\ r_2(b)\ w_2(c)$
$T_3 = r_3(c)\ w_3(a)$

- $T_2 T_3$

$= r_2(a)\ w_2(d)\ r_2(b)\ w_2(c)\ r_3(c)\ w_3(a)$

ULTIME SCRITTURE

Per a: w_3
Per b: /
Per c: w_2
Per d: w_2

= ALLE ULTIME SCRITTURE DI S_3

RELAZIONI LEGGI-DA
$w_2(c), r_3(c)$ ≠ DALLE RELAZIONI DI S_3

- $T_3 T_2$

$= r_3(c)\ w_3(a)\ r_2(a)\ w_2(d)\ r_2(b)\ w_2(c)$

ULTIME SCRITTURE

Per a: w_3
Per b: /
Per c: w_2
Per d: w_2

= ALLE ULTIME SCRITTURE DI S_3

RELAZIONI LEGGI-DA
$w_3(a), r_2(a)$ ≠ DALLE RELAZIONI DI S_3

⇒ S_3 NON E' VSR, PERCHÉ NON ESISTE NESSUN SCHEDULE SERIALE DI T_1, T_2 CHE SIA VIEW EQUIVALENTE A S_3.