



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

Basi di Dati II

28 Giugno 2023 -- Pag 1/2
(Tempo a disposizione totale: 2 ore)

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

Si consideri il seguente schema relazionale:

Studenti(matricola, nomeS, cittàS, annoNascita)

Esami(codiceE, nomeE, CorsoLaurea)

Voti(esame^{esame}, studente^{studenti}, voto, data)

Indicare quale di queste espressioni restituisce le informazioni sugli esami del corso di laurea di informatica sostenuti con voto superiore a 25 da almeno uno studente nato dopo il 2000

[1.a] $(\sigma_{\text{CorsoLaurea}=\text{"Informatica"}}(\text{Esami})) \cap ((\sigma_{\text{voto}>25}(\text{Voti})) \bowtie_{\text{studente}=\text{matricola}} (\sigma_{\text{annoNascita}>2000}(\text{Studenti})))$

[1.b] $\pi_{\text{nomeE}, \text{codiceE}}((\sigma_{\text{CorsoLaurea}=\text{"Informatica"}}(\text{Esami})) \bowtie_{\text{CodiceE}=\text{Esame}} ((\sigma_{\text{voto}>25}(\text{Voti})) \bowtie_{\text{studente}=\text{matricola}} (\sigma_{\text{annoNascita}>2000}(\text{Studenti}))))$

[1.c] $\pi_{\text{nomeE}}(\text{Voti} \bowtie_{\text{studente}=\text{matricola}} (\sigma_{\text{annoNascita}>2000}(\text{Studenti}))) \cap \pi_{\text{nomeE}}(((\sigma_{\text{voto}>25}(\text{Voti})) \bowtie_{\text{esame}=\text{codiceE}} (\sigma_{\text{corsoLaurea}=\text{"Informatica"}}(\text{Esami}))))$

[1.d] $\pi_{\text{matricola}}(\sigma_{\text{annoNascita}>2000}(\text{Studenti})) \cap \pi_{\text{studente}}((\sigma_{\text{voto}>25}(\text{Voti})) \bowtie_{\text{esame}=\text{codiceE}} (\sigma_{\text{corsoLaurea}=\text{"Informatica"}}(\text{Esami})))$

Quiz 2 (pt 1.5) (in giallo è evidenziata la risposta corretta)

Si consideri il seguente file di log (visualizzato in due colonne per ridurre lo spazio)

- | | |
|----------------------|-----------------------|
| 1. B(T5) | 10. U(T8, O3, B5, A5) |
| 2. B(T6) | 11. C(T8) |
| 3. U(T6, O1, B1, A1) | 12. CK(T6,T7) |
| 4. I(T5, O2, A2) | 13. U(T7, O5, B6, A6) |
| 5. B(T7) | 14. U(T6, O5, B7, A7) |
| 6. C(T5) | 15. A(T7) |
| 7. B(T8) | 16. B(T9) |
| 8. U(T6, O2, B3, A3) | 17. U(T9, O2, B8, A8) |
| 9. U(T7, O2, B4, A4) | 18. C(T6) |

Ipotizzando un failure all'istante 19, dopo una ripresa a caldo che valori hanno gli oggetti O2 e O5

[2.a] O2= B8, O5= B6

[2.b] O2= B4, O5= B6

[2.c] O2= A3, O5= A7

[2.d] O2= B3, O5= B7

Quiz 3 (pt 1.5) (in giallo è evidenziata la risposta corretta)

Si supponga di costruire un B⁺-tree su un campo di ricerca V che occupa 10 byte, usando blocchi di dimensione 1000 byte, con puntatore al blocco che occupa 5 byte e puntatori ai record che occupano 4 byte. Qual è l'ordine massimo di un nodo interno del B⁺-tree?

[3.a] 66

[3.b] 45

[3.c] 71

[3.d] 70

[3.e] nessuno dei valori indicati

ESERCIZIO 1 (pt. 7)

Si consideri il seguente schema di base di dati per la gestione di conferenze

CONFERENZE(titolo, Anno, Città, Nazione)

PERSONE(CF, Nome, Cognome, email, AnnoNascita)

ORGANIZZA(titoloConf^{conferenze}, persona^{persone}, funzione)

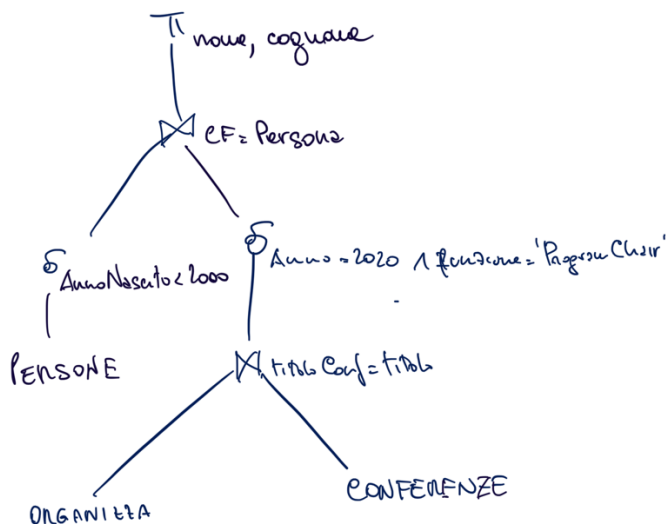
- 1.1 Scrivere un'espressione algebrica che restituisce il nome e cognome delle persone nate prima del 2000 e che hanno partecipato all'organizzazione di conferenze nel 2020 con la funzione di "Program chair"
- 1.2 Per l'espressione ottenuta al punto 1, disegnare il query tree (albero dell'interrogazione) ottimizzato
- 1.3 Indicare quante operazioni I/O richiede l'esecuzione ottimizzata di un join tra ORGANIZZA e PERSONE con l'algoritmo merge sort ipotizzando che:
 - i due file non sono ordinati fisicamente
 - Il buffer mette a disposizione 10 blocchi
 - la tabella PERSONE occupa 1000 blocchi
 - la tabella ORGANIZZA occupa 100 blocchi

SOLUZIONE ESERCIZIO 1

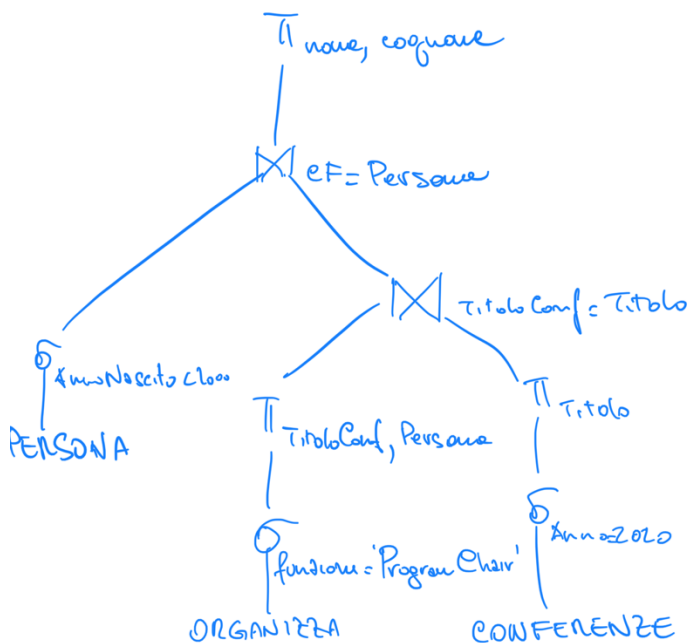
1.1 Una possibile espressione algebrica è la seguente:

$\pi_{\text{nome, cognome}}(\sigma_{\text{AnnoNascita} < 2000}(\text{Persone}) \bowtie_{\text{CF} = \text{persona}}(\sigma_{\text{Anno} = 2020 \text{ AND } \text{funzione} = \text{'Program Chair'}}(\text{ORGANIZZA} \bowtie_{\text{titoloConf} = \text{titolo}} \text{CONFERENZE})))$

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



La 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 I due file non sono ordinati, per poter applicare il join merge sort è necessario prima ordinarli entrambi.

Il costo dell'ordinamento con l'algoritmo di ordinamento esterno merge sort $2N * (1 + \lceil \log_{B-1} \lceil N/B \rceil \rceil)$

Nel caso dell'ordinamento del file che contiene la tabella PERSONA questo è

$$2 * 1000 * (1 + \lceil \log_9 \lceil 1000/10 \rceil \rceil) = 2000 * (1 + \lceil \log_9 100 \rceil) = 2000 * (1 + 3) = 8000$$

Nel caso dell'ordinamento del file che contiene la tabella ORGANIZZA questo è

$$2 * 100 * (1 + \lceil \log_9 \lceil 100/10 \rceil \rceil) = 200 * (1 + \lceil \log_9 10 \rceil) = 200 * (1 + 2) = 600$$

Il costo del merge sort esterno è dato dal costo di ordinamento dei file , più i costi di lettura di tutti i blocchi dei due file, quindi:

$$8000 + 600 + 1000 + 1000 = 9700 \text{ I/O}$$

Esercizio 2 (pt 6.5)

Considerare un file che occupa 2.500.000 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) Quanti passaggi saranno necessari per ordinare completamente il file?
- 2.2) Qual è il costo totale di I/O dell'ordinamento del file?
- 2.3) Di quanti blocchi di buffer si ha bisogno per ordinare completamente il file in due soli passaggi?

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 $\lceil 2.500.000/12 \rceil = 208.334$ run.

Seguono un numero di passi di merge sui run ottenuti al passo 0. Il numero di merge con 12 buffer è pari a $\lceil \log_{11} [208.334] \rceil = 6$.

Il numero di passi totale è quindi $1+6=7$

2.2

Il costo totale è $2N \cdot \text{numero passi}$, dove N è il numero di blocchi da ordinare. Quindi $2.500.000 \cdot 7 = 17.500.000$

2.3

L'ordinamento in solo due passi implica che viene eseguita un solo passo di merge. Sia X il numero di blocchi di buffer, per poter eseguire un solo merge su 2.500.000

$$\log_{x-1} (2.500.000/x) = 1$$

$$(x-1)^1 = (2.500.000/x)$$

$$\sim x^2 = 2.500.000 \quad x = 1582$$

Esercizio 3 (pt. 7)

Si consideri la relazione $R(A,B,C,...)$ con la seguente configurazione

- $T(R) = 100.000.000$ tuple, ognuna di 40 byte
- A occupa 4 byte
- C occupa 8 byte ed è unique
- $B=4000$, il blocco B ha dimensione 4000 byte
- P , puntatore al blocco, occupa 6 byte
- Pr , puntatore al record, occupa 7 byte

3.1) Assumendo che su A è costruito un indice B^+ -tree con fattore di riempimento al 70%, indicare la stima di costo della seguente operazione: $SELECT * FROM R WHERE A = 1000$

3.2) Assumendo che su C è costruito un indice secondario, indicare la stima di costo della seguente operazione: $SELECT * FROM R WHERE C = 1000$

3.3) Assumendo che su C è costruito un indice multilivello, indicare la stima di costo della seguente operazione: $SELECT * FROM R WHERE C = 1000$

SOLUZIONE ESERCIZIO 3

3.1

La query $SELECT * FROM R WHERE A = 1000$ richiede di attraversare il B^+ -tree costruito su A per trovare il puntatore al record avente $A=1000$, più un accesso al blocco che contiene il record.

Per avere una stima sul costo di ricerca su B^+ -tree devo calcolare il numero di livelli. Calcolo prima l'ordine massimo per i nodi interni e nodi foglia.

Ordine massimo nei nodi interni

$$Pp + A(p-1) \leq 4000$$

$$6p + 4(p-1) \leq 4000$$

$$6p + 4p - 4 \leq 4000$$

$$10p \leq 4001$$

$$p_{\text{interno}} = 400$$

Ordine massimo nei nodi foglia

$$p \cdot A + p_{Pr} + P \leq 4000$$

$$4p + 7p + 6 \leq 4000$$

$$11p + 6 \leq 4000$$

$$11p \leq 3994 \quad p_{foglia} = 363$$

Con fattore di riempimento al 70%, l'ordine massimo diventa

$$p_{interno} = 400 \cdot 0.70 = 280$$

$$p_{foglia} = 363 \cdot 0.70 = 254$$

Il numero di livelli per ospitare 100.000.000 tuple è calcolato

Livello	#nodi	#valori di chiavi	#puntatori record
Root	1	(280-1)=279	
1 liv.	280	$280 \times 279 = 78.120$	
2 liv.	280^2	$280^2 \times 279 = 21.873.600$	
3 liv./foglie	280^3	$280^3 \times 254 = 5.575.808.000$	5.575.808.000

Quindi ricerca un record in questo B+-tree richiede 4 accessi I/O.
Il costo della query è quindi $4+1=5$ I/O

3.2

La query *SELECT * FROM R WHERE C = 1000* richiede di attraversare l'indice secondario costruito su C per trovare il puntatore al record avente $C=1000$, più un accesso al blocco che contiene il record.

Per avere una stima sul costo di ricerca sull'indice secondario, devo calcolare il numero di blocchi contenenti le voci dell'indice su cui fare la ricerca binaria.

Il file non è ordinato su C, quindi l'indice è denso. E' necessaria una voce per ogni record, quindi 100.000.000 voci.

Ogni voce richiede 14 byte, 8 per il campo C e 6 per il puntatore al record (N.b. si poteva anche usare il puntatore al record)

In un blocco da 4000 byte si possono memorizzare $\lceil 4000/14 \rceil = 285$ voci

L'indice secondario occupa, quindi, $\lceil 100.000.000/285 \rceil = 350.878$ blocchi

La ricerca sull'indice secondario è una ricerca binaria su 350.878 blocchi, richiede $\log_2(350.878) = 19$ operazioni I/O

Il costo della query è quindi $19 + 1 = 20$ operazioni I/O.

3.3

Per stimare il costo della query devo calcolare di quanti livelli è costituito l'indice multilivello su C. 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. 350.878 voci richiedono un numero di blocchi pari a $\lceil 350.878/285 \rceil = 1232$

Livello 3 - come per il precedente. 1232 voci richiedono un numero di blocchi pari a $\lceil 1232/285 \rceil = 5$

Livello 4 - 5 voci richiede un solo blocco.

Per ricerca il record con $C=1000$ nell'indice multilivello è necessario leggere 4 blocchi (uno per livello). Il costo della query 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) $r_3(z)$ $w_1(y)$ $w_3(z)$ $r_2(z)$ $r_2(y)$ $w_3(y)$ $w_2(y)$

4.2) $r_1(a)$ $w_2(a)$ $r_3(a)$ $r_1(b)$ $r_4(c)$ $w_3(d)$ $r_4(d)$ $w_4(b)$ $w_5(b)$ $w_5(c)$ $w_2(b)$ $r_1(d)$

4.3) $r_1(x)$ $w_2(x)$ $r_3(x)$ $r_1(y)$ $w_3(v)$ $w_2(y)$ $r_1(v)$

SOLUZIONE ESERCIZIO 4.1

$S_1 = r_3(z)$ $w_1(y)$ $w_3(z)$ $r_2(z)$ $r_2(y)$ $w_3(y)$ $w_2(y)$

È CSR?

① CONSIDERO I CONFLITTI SULLE RISORSE

$z = r_3, w_3, r_2$

$y = w_1, r_2, w_3, w_2$

CONFLITTI:

w_3, r_2

w_1, r_2

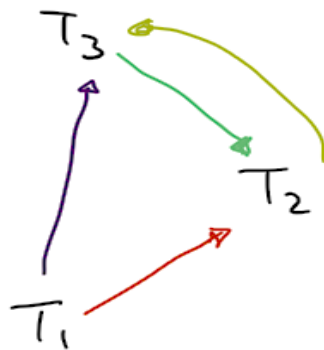
w_1, w_3

w_1, w_2

r_2, w_3

w_3, w_2

② COSTRUISCO IL GRAFO DEI CONFLITTI



È ciclico, quindi S_1 NON È CSR

E' USR?

① DEFINISCO LE RELAZIONI LEGGI-DA DI S_1

$$\begin{aligned} & (w_3(z), r_2(z)) \\ & (w_1(y), r_2(y)) \end{aligned}$$

② DETERMINO LE ULTIME SCRITTURE IN S_1

Per y : $w_2(y)$

Per z : $w_3(z)$

③ CONSIDERO I POSSIBILI SCHEDULE SERIALI

$$T_1 = w_1(y)$$

$$T_2 = r_2(z) r_2(y) w_2(y)$$

$$T_3 = r_3(z) w_3(z) w_3(y)$$

PER ESSERE USR CON S_1 , LO SCHEDULE DEVE AVERE LE STESSA RELAZIONI LEGGI-DA E SCRITTURE FINALI

LE SCRITTURE FINALI DI S_1 ESCLUDONO GLI SCHEDULE SERIALI DOVE T_1 E' ESEGUITA PER ULTIMA, PERCHE' POTREBBE AD UNA SCRITTURA FINALE DI y , $w_1(y)$, DIVERSA DA QUELLA DI S_1

LO STESSO RAGIONAMENTO ESCLUDE T_3 COME ULTIMA TRANSIZIONE NEL SCHEDULE SERIALE, PERCHE' AVREMMO ULTIMA SCRITTURA SU y DATA DA $w_3(z)$, MENTRE IN S_1 E' DI $T_2 (w_2(z))$

GLI SCHEDULE SERIALI CHE POTREBBERO ESSERE USR EQUIVALENTI A S_1 DEVONO AVERE T_2 FINALE

$$\Rightarrow S_{11} = T_1 T_3 T_2 \quad \left. \begin{array}{l} S_{12} = T_3 T_1 T_2 \end{array} \right\} \begin{array}{l} \text{ENTRAMBI HANNO LE} \\ \text{STESSE ULTIME SCRITTURE} \\ \text{DI } S_1 \end{array}$$

CONTROLLA LEGGI DA

$$1. \omega_1(4) \quad r_3(z) \quad \omega_3(z) \quad \omega_3(4)$$

$$(\omega_3(z), r_2(z))$$

$$(\omega_3(4), r_2(4))$$

E' DIVERSO

S_{11} NON E' USR
CON S_1 .

$$S_{12} =$$

$$r_3(z) \quad \omega_3(z) \quad \omega_3(4) \quad \omega_1(4) \quad r_2(z) \quad r_2(4) \quad \omega_2(4)$$

$$(\omega_3(z), r_2(z))$$

$$(\omega_2(4), r_2(4))$$

HA LE STESSI
RELATIONI LEGGI-DA

S_{12} E' USR
EQUIVALENTE A S_2

S_1 E USR

SOLUZIONE ESERCIZIO 4.2

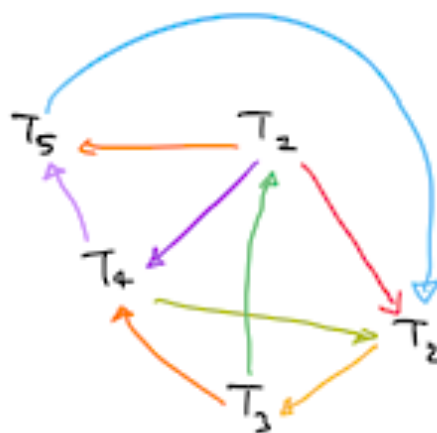
$$S_2 = r_1(a) w_2(a) r_3(a) r_2(b) r_4(c) w_3(d) r_4(d) \\ w_4(b) w_5(b) w_5(c) w_2(b) r_2(d)$$

E' CSR?

$$\begin{aligned} a &= r_2 w_2 r_3 \\ b &= r_2 w_4 w_5 w_2 \\ c &= r_4 w_5 \\ d &= w_3 r_4 r_2 \end{aligned}$$

CONFLITTI

$r_2 w_2$
 $w_2 r_3$
 $r_2 w_4$
 $r_2 w_5$
 $r_2 w_2$
 $w_4 w_5$
 $w_4 w_2$
 $w_5 w_2$
 $r_4 w_5$
 $w_3 r_4$
 $w_3 r_2$



IL GRAFO HA CICLI

S_2 NON E' CSR.

E' CSR?

① RELAZIONI LEGGI-DA

$$\begin{aligned} (w_2(a), r_3(a)) \\ (w_3(d), r_4(d)) \\ (w_3(d), r_2(d)) \end{aligned}$$

② ULTIME SOSTITUIRE

Per a: $w_2(a)$
 Per b: $w_2(b)$
 Per c: $w_5(c)$
 Per d: $w_3(d)$

② PER TROVARE UNO SCHEDULE SERIALE
 VSRL EQUIVALENTE A S_2 (4.2) DEVO
 CONSIDERARE GLI SCHEDULE SERIALI
 CHE SI POSSONO COMPORRE DA QUESTE
 5 TRANSACTION:

$$T_1 = r_1(a) \ r_1(b) \ r_1(d)$$

$$T_2 = w_2(a) \ w_2(b)$$

$$T_3 = r_3(a) \ w_3(d)$$

$$T_4 = r_4(c) \ r_4(d) \ w_4(b)$$

$$T_5 = w_5(b) \ w_5(c)$$

PRIMA DI CONSIDERARE TUTTI I CASI
 POSSIBILI POSSO ESCLUDERE DELLE
 COMBINAZIONI CHE DANNO SCHEDULE
 SERIALI NON VIEW EQUIVALENTI,
 OVVERO CON RELAZIONI LEGGI-DA
 DIVERSE E/O SCRITTURE FINALI DIVERSE.

OSSERVAZIONE 1

CONSIDERO LA SCRITTURA FINALE SUL
 L'OGGETTO b IN S_2 , OVVERO $w_2(b)$.

POSSO DEDURRE CHE T_2 DEVE ESSERE
 ESEGUITO DOPO T_4 E T_5 . INFATTI ENTRAMBE
 QUESTE DUE TRANSACTION SCRIVONO SUL
 L'OGGETTO b .

SE ANCHE SOLO UNA DI LORO VENISSE
 ESEGUITA DOPO T_2 AVREMMO UNO SCHEDULE
 SERIALE CON UNA SCRITTURA DIVERSA -
 SU b , QUINDI NON VIEW EQUIVALENTE
 A S_2 .

QUINDI DEVO CONSIDERARE SCHEDULE
 SERIALI DOVE

$$T_4 \text{ E } T_5 \text{ PRECEDONO } T_2$$

OSSERVAZIONE 2

CONSIDERO QUESTA RELAZIONE LEGGI-DA

$$(w_2(a), r_3(a))$$

POSSO DEDURRE CHE LO SCHEDULE SERVA
PER AVERE LA STESSA RELAZIONE DEVE
AVERE T_2 PRIMA DI T_3 .

QUINDI

T_4 e T_5 PRECEDONO T_2 , PRECEDE T_3 .

OSSERVAZIONE 3

DA QUESTA RELAZIONE LEGGI-DA

$$(w_3(d), r_4(d))$$

DEDUCO CHE PER AVERE LA STESSA RELAZIONE
 T_3 DEVE ESSERE PRIMA T_4

QUINDI ABBIAMO CHE

OSSERVAZIONI 1 & 2 IMPLICANO

T_4 e T_5 PRECEDONO T_2 , PRECEDE T_3 .

OSSERVAZIONE 3 IMPLICA

T_3 PRECEDE T_4

⇒ NON PUÒ ESISTERE UNA SCHEDULE
CHE SODDISFA LA CONDIZIONE DATA
DALE OSSERVAZIONI 1, 2 E CONTEMPORA-
NEAMENTE LA CONDIZIONE IMPLICATA
DALL'OSSERVAZIONE 3

QUINDI POSSIAMO CONCLUDERE CHE
NON ESISTE UNA SCHEDULE SERIALE
CHE HA LE STESSA RELAZIONI LEGGI-DA
DI S_2 E LE STESSA SORTITE
FINALI.

S_2 NON È USR.

SOLUZIONE ESERCIZIO 4.3

$$S_3 = r_1(x) w_2(x) r_3(x) r_1(y) w_3(y) w_2(y) r_3(y)$$

E' CSR?

① CONSIDERO I CONFLITTI SULLE RISONSE

$$x = r_1 w_2 r_3$$

$$y = r_1 w_2$$

$$v = w_3 r_1$$

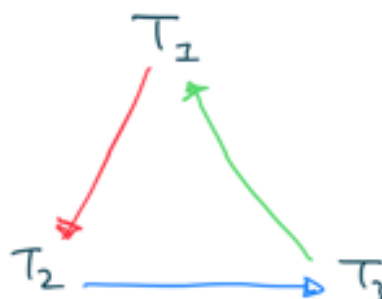
CONFLITTI

$$r_1 w_2$$

$$w_2 r_3$$

$$r_1 w_2$$

$$w_3 r_1$$



IL GRAFO HA CICLI, S_3 NON E' CSR.

E' CSR?

① RELAZIONI LEGGI-DA DI S_3

$$(w_3(v), r_1(v))$$

$$(w_2(x), r_3(x))$$

② ULTIME SCRITTURE DI S_3

$$\text{Per } x: w_2(x)$$

$$\text{Per } y: w_2(y)$$

$$\text{Per } v: w_3(v)$$

③ PER TROVARE UNA SCHEDULE SERIALE
 UPR EQUIVALENTE A S_3 CONSIDERO
 LE POSSIBILI COMBINAZIONI DI QUESTE
 TRANSAZIONI:

$$T_1 = r_2(x) \ r_1(y) \ r_1(v)$$

$$T_2 = w_2(x) \ w_2(y)$$

$$T_3 = r_3(x) \ w_3(v)$$

• $T_1 \ T_2 \ T_3$

$$= r_2(x) \ r_1(y) \ r_1(v) \ w_2(x) \ w_2(y) \ r_3(x) \ w_3(v)$$

ULTIME SCRITTURE

Per x : $w_2(x)$

Per y : $w_2(y)$

Per v : $w_3(v)$

= ALLE ULTIME
 SCRITTURE DI S_3

LEGGI-DA

$$w_2(x) \ r_3(x)$$

\neq DUE RELAZIONI
 LEGGI DA DI S_3 .

• $T_1 T_3 T_2$

$$= r_2(x) \ r_1(y) \ r_1(v) \ r_3(x) \ w_3(v) \ w_2(x) \ w_2(y)$$

ULTIME SCRITTURE

Per x : $w_2(x)$

Per y : $w_2(y)$

Per v : $w_3(v)$

= ALLE ULTIME
SCRITTURE DI S_3

LEGGI-DA

NESSUNA

\neq DUE RELAZIONI
LEGGI DA DI S_3 .

• $T_3 T_1 T_2$

$$= r_3(x) \ w_3(v) \ r_2(x) \ r_1(y) \ r_1(v) \ w_2(x) \ w_2(y)$$

ULTIME SCRITTURE

Per x : $w_2(x)$

Per y : $w_2(y)$

Per v : $w_3(v)$

= ALLE ULTIME
SCRITTURE DI S_3

LEGGI-DA

$(w_3(v), r_1(v)) \neq$ DUE RELAZIONI
LEGGI DA DI S_3 .

• $T_3 T_2 T_1$

$$\bullet T_3 T_2 T_1$$

$$= r_3(x) w_3(y) w_2(x) w_2(y) r_2(x) r_1(y) r_1(y)$$

ULTIME SCRITTURE

$$\text{Per } x: w_2(x)$$

$$\text{Per } y = w_2(y) = \text{ALLE ULTIME}$$

$$\text{Per } v = w_3(y)$$

SCRITTURE DI S_3

LEGGI DA :

$$(w_3(y), r_2(y))$$

$$(w_2(y), r_1(y))$$

$$(w_2(x), r_2(x))$$

\neq DUE RELAZIONI
LEGGI DA DI S_3

$$\bullet T_2 T_1 T_3$$

$$= w_2(x) w_2(y) r_2(x) r_1(y) r_1(y) r_3(x) w_3(y)$$

ULTIME SCRITTURE

$$\text{Per } x: w_2(x)$$

$$\text{Per } y = w_2(y) = \text{ALLE ULTIME}$$

$$\text{Per } v = w_3(y)$$

SCRITTURE DI S_3

LEGGI-DA :

$$\left\{ \begin{array}{l} (w_2(x), r_3(x)) \\ (w_2(y), r_2(y)) \\ (w_2(x), r_1(x)) \end{array} \right\} \neq \text{DUE RELAZIONI LEGGI DA DI } S_3$$

● $T_2 T_3 T_1$

$$= w_2(x) w_2(y) r_3(x) w_3(y) r_2(x) r_1(y) r_1(y)$$

ULTIME SCRITTURE

Per x : $w_2(x)$

Per y : $w_2(y)$

Per v : $w_3(y)$

= ALLE ULTIME SCRITTURE DI S_3

LEGGI-DA :

$$\left\{ \begin{array}{l} (w_3(y), r_1(y)) \\ (w_2(y), r_1(y)) \\ (w_2(x), r_1(x)) \\ (w_2(x), r_3(x)) \end{array} \right\} \neq \text{DUE RELAZIONI LEGGI DA DI } S_3$$

S_3 NON È USR, PERCHÉ NON ESISTONO SCHEDULE SERIALI VIEW EQUIVALENTI A S_3 .

