

## GESTORE AFFIDABILITÀ:

DUMP; B(T1), B(T2), I(T1, 01, A1), D(T2, 02, B2), B(T3)  
U(T3, 03, B3, A3) U(T1, 04, B4, A4) CK(T1, T2, T3)  
B(T4) C(T1) U(T4, 05, B5, A5), B(T5) C(T3)  
U(T2, 06, B6, A6) U(T4, 03, B7, A7) B(T6) A(T2) GUASTO

① Risalire fino all' ultimo CheckPoint  $\Rightarrow CK(T_1, T_2, T_3)$

② Inizializzo 2 Insiemi :

$$\begin{aligned} \text{# UNDO} &= \{T_1, T_2, T_3\} \\ \text{# REDO} &= \emptyset \end{aligned}$$

③ Scorro il file di Log per Giungere Allo Stato finale dei 2 Insiemi:

$$\begin{aligned} \text{# UNDO} &= \{T_2, T_5, T_6, T_4\} \\ \text{# REDO} &= \{T_1, T_3\} \end{aligned}$$

④ Distro Op. insieme UNDO ripercorrendo Log al Contrario:

$$03 := B7$$

$$06 := B6$$

$$05 := B5$$

$$02 := B2$$

⑤ Scorro log dell' inizio Rifare op. del REDO

$$01 = A1$$

$$03 = A3$$

$$04 = A4$$

Ricordarsi il Verso delle OPERAZIONI:



### RIPRESA A FREDDO:

Checkpoint prima del Dump mi Serve e devo Ripristinare la Memoria.

### ESECUZIONE CON CORRENTE:

Categorizzare lo Schedule che viene dato

$r_2(y), w_3(z), r_2(x), w_1(z), w_2(y), w_1(t), r_3(y), w_3(x), r_1(x), r_1(y), w_0(x), w_3(t)$

■ VSR?

LeggeDa =  $\{(r_3(y), w_2(y)), (r_1(x), w_3(x)), (r_1(y), w_2(y))\}$

Scritturaefinali =  $\{w_3(t), w_0(x), w_2(y), w_1(z)\}$

$T_0 \rightarrow (w_0(x))$

$T_1 \rightarrow (w_1(z), w_1(t), r_1(x), r_1(y),$

$T_2 \rightarrow r_2(y), r_2(x), w_2(y),$

$T_3 \rightarrow (w_3(z), r_3(y), w_3(x), w_3(t))$

$\begin{cases} T_2 < T_3 \\ T_3 < T_1 \\ T_2 < T_1 \end{cases}$

$\begin{cases} T_1 < T_3 \\ T_3 < T_0 \\ T_3 < T_1 \end{cases}$

Ho un ASSURDO,  
Contraddizione e so  
che S  $\neq$  VSR,  
Non Posso MAI avere

Uno Schedule Seriale VIEW-EQUIVALENTE

Qui mi Sono Bastate le condizioni Note/Immediate dei 2 Insiemi ma avrei potuto/Dovuto Analizzare Anche Quello Che non ci Sta dentro.

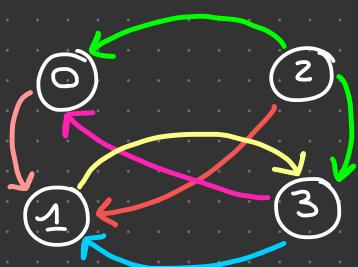
PARTO DA Quello che C'È E Poi Analizzo

So già che  $S \not\models \text{CSR} \Rightarrow \not\models \text{CSR} \Rightarrow \not\models \text{2PL}$

---

Ma CSR?

$$\text{Conflitti} = \left\{ \left( \underline{W_3(z)}, \underline{W_1(z)} \right), \left( \underline{r_2(x)}, \underline{W_3(x)} \right), \left( \underline{r_2(x)}, \underline{W_0(x)} \right), \right. \\ \left( \underline{W_2(y)}, \underline{r_3(y)} \right), \left( \underline{W_2(y)}, \underline{r_1(y)} \right), \\ \left( \underline{W_1(t)}, \underline{W_3(t)} \right), \\ \left( \underline{W_3(x)}, \underline{r_1(x)} \right), \left( \underline{W_3(x)}, \underline{W_0(x)} \right), \\ \left. \left( \underline{r_1(x)}, \underline{W_0(x)} \right) \right\}$$



Ho Confermato che:  
 $S \not\models \text{CSR}$

---

Tutto Quello che Si SCRIVE Viene Corretto

Ricordare Sempre di MOTIVARE Tutto! e Anche



2 PL? Ogni Ti ha 3 fasi:

- Acquisisce
- Sfrutta
- Rilascia



$T_0 \rightarrow W_0(x)$

$T_1 \rightarrow (W_1(z), W_1(t), r_1(x), r_1(y),$

$T_2 \rightarrow r_2(y), r_2(x), W_2(y),$

$T_3 \rightarrow W_3(z), r_3(y), W_3(x), W_3(t)$

$T_3$  Acquisisce un NUOVO LOCK

$r_2(y), W_3(z), r_2(x), W_1(z), W_2(y), W_1(t), r_3(y), W_3(x), r_1(x), r_1(y), W_0(x), W_3(t)$

Mi Serve LOCK che ha Acquisito  $W_3$

$T_3$  ha Rilasciato il LOCK ed ha Iniziato la  
fase discendente

---

$S_{2PL}$ :  $r_1(x), W_2(y), r_1(z), W_1(t), r_2(t), W_2(x)$

Perche  $T_2$  possa Eseguire  $r_2(t)$  Vuol dire che è Necessario che  $T_1$   
Abbia Rilasciato il LOCK Sulla Risorsa T

## OTTIMIZZAZIONE :

Paziente (Codice SSN, ...)

Visita (Codice SSN, CF Medico, data, Ora I, ...)

Medico (CF Medico, Nome, Cognome, Specialità)

SELECT M.nome, M.cognome, V.codSSN

FROM Medico M JOIN Visita V ON M.CFMedico = V.CFMedico

WHERE SPEC = 'Ortopedia'

① SELEZIONE Con Seg. Scan Su Medico e Risultato Mantenuto  
Nel BUFFER

② Join  $\Rightarrow$  Medico  $\bowtie$  Visita (Nested Loop)

	NP	NR	Val (Spec, Med)	Val (CFMedico, Visita)
Medico	12	1200	12	
Visita	3100	380.000		1900
Paziente	160	16.000		

Costo SELEZIONE  $\Rightarrow$  Leggere + Scrivere

$\rightarrow$  È SEQ. Quindi Sarà  
NP (Medico)

$\Rightarrow$  NP (Medico)

$\rightarrow$  Perché Sta  
Nel BUFFER

**COSTO DEL JOIN  $\Rightarrow$   $NP(R) + NR(R) \cdot NP(S)$  [Nested Loop Join]**

R Sarà MEDICO

S Sarà Visita

NB! Non tutti i Medici, Ma  
Solo Quelli filtrati sulla Specialità

Essendo che i DATI Sono Già in MEM. CENTRALE

$$NP(R) + NR(R) \cdot NP(S) = 312.000$$

$$\frac{NR(\text{Medico})}{\text{Val}(\text{Spec, Medico})} = \frac{1200}{12} = 100 \text{ che Sono i Medici con Specialità = Octopedia}$$

Indice B+Tree definito su CF Medico di Visita con Prof. = 2

Attributo di JOIN

Cambierà il  
Costo

$$\text{COSTO JOIN} \Rightarrow NP(R) + NR(R) \left( \text{Prof. I} + \frac{NR(S)}{\text{VAL}(A, S)} \right)$$
$$\frac{380.000}{1.900} = 200$$

$$\Rightarrow \emptyset + \frac{1200}{12} \cdot \frac{202}{1} = \text{Se si ha Calcolatrice, Altrimenti ok così.}$$

Se INDICE è su Qualcosa su cui NON C'è Nulla Allora devo Accorgermene

COLLEGIO (

VOTO (

CANDIDATO (

Collegio  $\bowtie$  Voto

Costo Select  $\Rightarrow$  Selezione su COLLEGIO.REGIONE + VOTO. #VOTI > 180

Calcolo Come prima Cardinalità dei 2 Argomenti del JOIN  
Cosa

Se c'è INDICE B+Tree prof. 2 Costruito su COLLEGIO.REGIONE  
devo Cambiare il Costo della Selezione e non nel JOIN.

NP(Collegio) + Scrivere diventa

$$P_{\text{prof. Indice}} + \frac{NR(\text{Collegio})}{\text{VAR}(\text{REGIONE}, \text{COLLEGIO})}$$

Scansione Sequenziale

Indice

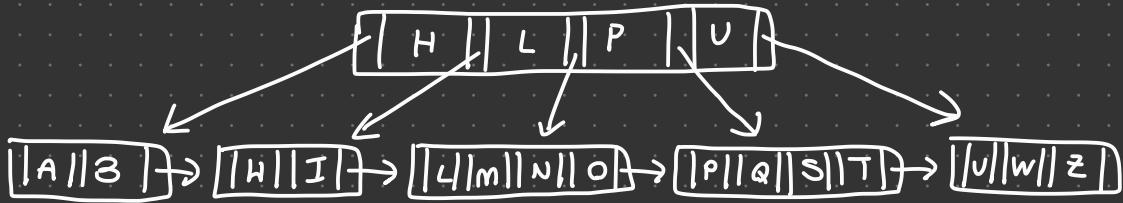
NB a NON fare ERRORI:

STARE SEMPRE ATTENTI AL COSTO CHE VA PUÒ ESSERE:

- Su Attributo di JOIN
- Su Attributo di SELEZIONE
- Su Attributo INUTILE

## ESEMPIO B+TREE:

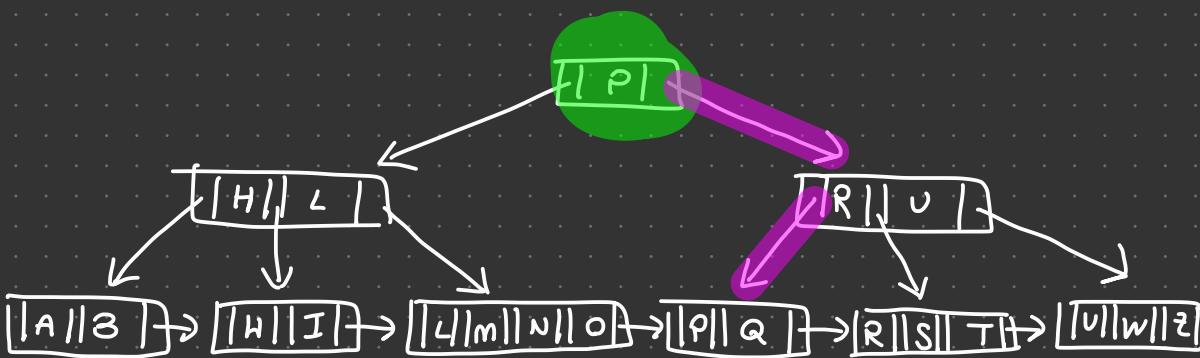
$f_{in} - f_{out} = 5$



Riesco a COSTRUIRE UNA RADICE UNICA

② AGGIUNGO VALORE CHIAVE 'R'

Nodo foglie è  $\boxed{P||Q||R||S||T}$  ma viola vincoli di riempimento  
e devo fare lo SPLIT e propagarlo anche alla RADICE



IL VALORE P lo Trovo trovando il Minimo del Sotto-Albero Destro

② Rimuovo H, devo fare il MERGE

viola ed ho solo 5 figli  $\Rightarrow$  Gestisco con un solo lvl



### ALBERO RISULTANTE:

