

GESTORE AFFIDABILITÀ:

DUMP: B(T1), B(T2), I(T1, O1, A1), O(T2, O2, B2), B(T3)
U(T3, O3, B3, A3) U(T1, O4, B4, A4) CK(T1, T2, T3)
B(T4) C(T1) U(T4, O5, B5, A5), B(T5) C(T3)
U(T2, O6, B6, A6) U(T4, O3, B7, A7) B(T6) A(T2) GUASTO

① Risalire fino all'ultimo CheckPoint $\Rightarrow CK(T1, T2, T3)$

② Inizializzo 2 Insiemi:

$\text{UNDO} = \{T1, T2, T3\}$

$\text{REDO} = \emptyset$

③ Scorro il file di Log Per Giungere Allo Stato finale dei 2

Insiemi:

$\text{UNDO} = \{T2, T5, T6, T4\}$

$\text{REDO} = \{T1, T3\}$

④ Disfo Op. insieme UNDO ripercorrendo Log al Contrario:

$O3 := B7$

$O6 := B6$

$O5 := B5$

$O2 := B2$

⑤ Scorro log dell'inizio Rifare Op. del REDO

$O1 = A1$

$O3 = A3$

$O4 = A4$

Ricordarsi il Verso delle OPERAZIONI:



RIPRESA A FREDDO:

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

ESECUZIONE CONCORRENTE:

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

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

$T_0 \rightarrow w_0(x)$

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

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

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

$\begin{matrix} \text{/// } T_1 < T_3 \\ \text{/// } T_3 < T_0 \\ \text{/// } T_3 < T_1 \end{matrix} \left. \vphantom{\begin{matrix} T_1 < T_3 \\ T_3 < T_0 \\ T_3 < T_1 \end{matrix}} \right\} \text{NOTE}$

$\begin{matrix} \text{/// } T_2 < T_3 \\ \text{/// } T_3 < T_1 \\ \text{/// } T_2 < T_1 \end{matrix} \left. \vphantom{\begin{matrix} T_2 < T_3 \\ T_3 < T_1 \\ T_2 < T_1 \end{matrix}} \right\} \text{NOTE}$

Ho un ASSURDO,
Contraddizione e so
che S è VSR,
Non Potrò MAI avere

Una Schedule Seriale View-EQUIVALENTE

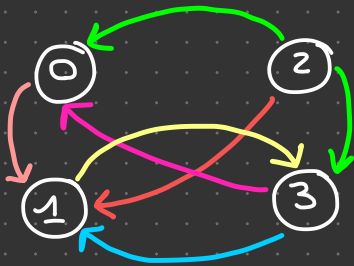
Qui mi Sono Bastate le CONDIZIONI Nate/Immedie dei 2 Insiemi ma avrei potuto/Dovuto Analizzare Anche Quello che non ci Sta dentro.

PARTO DA QUELLO CHE C'È E POI AMUZZO

So già che $S \notin VSR \Rightarrow \notin CSR \Rightarrow \notin 2PL$

CSR?

$$\text{Conflicti} = \left\{ \left(\underline{w_3(z)}, \underline{w_1(z)} \right), \left(\underline{r_2(x)}, \underline{w_3(x)} \right), \left(\underline{r_2(x)}, \underline{w_0(x)} \right), \right. \\ \left(\underline{w_2(y)}, \underline{r_3(y)} \right), \left(\underline{w_2(y)}, \underline{r_1(y)} \right), \\ \left(\underline{w_1(t)}, \underline{w_3(t)} \right), \\ \left(\underline{w_3(x)}, \underline{r_1(x)} \right), \left(\underline{w_3(x)}, \underline{w_0(x)} \right), \\ \left. \left(\underline{r_1(x)}, \underline{w_0(x)} \right) \right\}$$



Ho Confermato che:
 $S \notin CSR$

Tutto Quello che Si SCRIVE Viene Corretto

Ricordare Sempre di MOTIVARE Tutto! e Anche VSR CSR 2PL

2 PL? Ogni T_i 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

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

Perché 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, CFMedico, data, Ora I, ...)

Medico (CFMedico, 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 Seq. 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

È SEQ. Quindi Sarà
NP(Medico)

\Rightarrow NP(Medico)

$\rightarrow \emptyset$ 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

\emptyset
 \uparrow
 $NP(R) + NR(R) \cdot NP(S) = 312'000$

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

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

Attributo di JOIN

Cambierà il Costo

2

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

$$\frac{380'000}{1'900} = 200$$

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

Se INDICE è su Qualcosa su cui NON Centra 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) + Scritture

diventa

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

Scansione Sequenziale

Indice

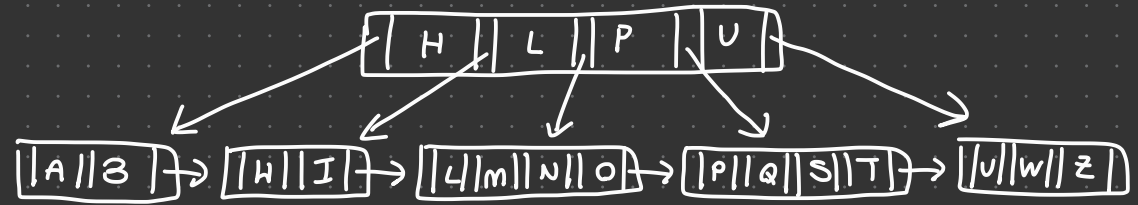
NB a NON fare ERRORI:

STARE SEMPRE ATTENTI AL COSTO CHE VA PUÒ ESSERE:

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

ESEMPIO B+TREE:

fan-Out = 5



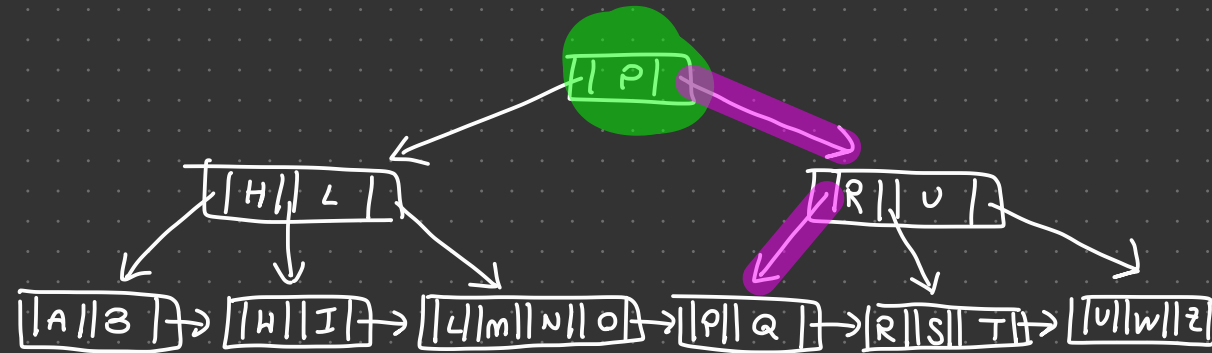
Riesco a costruire una radice unica

① AGGIUNGO VALORE CHIAVE 'R'

Nodo foglia è

	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:

