

## OTTIMIZZAZIONI:

NESTED LOOP JOIN:  $NP(R) + NR(R) \cdot NP(S)$

Ma Se Aggiungiamo un Indice B+Tree su S il Costo Cambia e diventa:  $NP(R) + NR \cdot (Profondità\ Indice + Selettività)$

Attributo A Nella Relazione S

$\leq 1$  Pagine di Indice  
Per Ogni livello

$$\frac{NR(R)}{VAL(A, S)}$$

Quante Righe Per Ogni  
Valore di A

## BLOCK NESTED LOOP JOIN:

Sfrutto i Dati che ho già in Memoria e Scricciare Pagine di S  
Solo Se ho finito tutti i Controlli:

Crescuna pagina di S (Tab. Interna) viene Letta Per Ogni  
Pagina di R invece che Per Ogni Riga di R

loop

$\forall t_r \in R \{$  caricata Interamente  
 $\forall t_s \in S \{$  s  
SE  $(t_s, t_r) \in PJ \{$   
aggiungi  
}  
}  
}

Block NESTED

$\forall$  PAGINA PR di R {  
 $\forall$  PAGINA PS di S {  
 $\forall$  RIGA  $t_r \in R \{$   
 $\forall$  RIGA  $t_s \in S \{$   
se  $(t_s, t_r) \in PJ \{$   
aggiungi  
}

Ottimizzo il Cacicamento delle Pagine di S visto che vengono Caciccate  $\forall$  PAGINA di R e non più per Ogni RIGA.

IL Costo DIVENTA  $\Rightarrow NP(R) + NP(R) \cdot NP(S)$

## MERGE-SCAN JOIN:

Si può Applicare Quando:

- Entrambe Tab hanno le TUPLE Ordinate rispetto agli Attributi di JOIN
- JOIN è un ugualanza stretta Tra Attributi di JOIN (EQU)

R	22	a
	32	s
	45	h
	87	b

S	22	z	8
	22	s	7
	35	h	4
	45	s	9
	45	c	3
	87	h	5
	87	g	6

$$R.A = S.A$$

Condizioni Soddisfatte

Sfrutto l'ORDINAMENTO per Evitare CONFRONTI INUTILI ed IL costo diventa:  $NP(R) + NP(S)$

R	22	a
	32	s
	45	h
	87	b

S	22	z	8
	22	s	7
	35	h	4
	45	s	9
	45	c	3
	87	h	5
	87	g	6

Da Qui in poi  
so che A è  
diverso

Se ho un Indice SU ENTRAMBE, devo fare una Stima per Eccesso  
e il costo è  $\Rightarrow \text{NR}(R) + \text{NR}(S)$  se INDICE Sta Nel BUFFER

Potrei NOTARE che Indice Qui non aiuta di molto, ma Potrebbe peggiorare se non Sta Nel BUFFER

### HASH JOIN:

Solo con EQUI-JOIN, ma non richiede presenza di INDICI Ne che le Tabelle Siano ORDINATE rispetto all' Attributo di JOIN.

Con Relazioni Molto Grandi è VANTAGGIOSO.

Si Basa sull' Avere a disposizione una fz. di HASH h che possiamo Applicare Agli Attributi di JOIN.

Se Parto da 2 Tuple:  $t_R \in R$  e  $t_S \in S$  e

$$t_R.J = t_S.J \Leftrightarrow h(t_R.J) = h(t_S.J)$$

Ma può essere che ci Siano dei falsi Positivi (COLLISIONI), che Sono dei CASI Limitati e Potrei usarli come una SCREMATURA

se  $t_R.J \neq t_S.J \Rightarrow$  sicuramente  $h(t_R.J) \neq h(t_S.J)$

IDEA Comune  $\Rightarrow$  Partizionare R ed S in Base ai Valori di h

Ricerca delle TUPLE che Soddisfano PJ avviene Solo Tra le Partizioni con lo Stesso Valore di h

**COSTO**  $\Rightarrow$  ① Costruzione **HASHMAP**  $\Rightarrow$   $NP(R) + NP(S)$   
② Ricerca  $\Rightarrow$  caso Pessimo =  $NP(R) \cdot NP(S)$   
caso Medio dipende da #Buffer ma  
ANCHE da distribuzione dei Valori

## OPTIMIZZATORE INTERROGAZIONI:

Genera **Tutti i possibili piani di Esecuzione** per poi Valutare  
con formule **Approssimate** (MODELLO DI COSTO) il Costo di Ogni  
Alternativa.

Si Sceglie l'ALBERO di Costo Minimo

Operazioni/Algoritmi Alternativi  
Usa o Meno di INDICI  
Determina l'ORDINE

Stima Cardinalità RELazione (#Tuples)

Stima Dimensione Ogni TUPLA

Stima Dimensione Attributi

Stima #Valori distinti  $\forall$  attributo in Tab.

Stima Valore min & max attributo

## STIMA COSTO INTERROGAZIONE:

INGREDIENTE (Codice, Nome, Calorie)

Composizione (Ricetta, Ingrediente, Quantità)

RICETTA (Codice, Nome, Regione, Tempo\_Preparazione)

M Trovare Gli Ingredienti Usati in Ricette della Regione Veneto  
Riportando il codice, Nome e le Calorie dell'Ingrediento

SELECT R.codice, I.Nome, I.calorie

FROM Ricetta R JOIN Composizione C ON R.codice = C.ricetta  
JOIN Ingrediento I ON C.ingrediente = I.codice

WHERE R.regione = 'Veneto'

M Calcola il costo (#Accessi Mem. Secondaria) con Queste ipotesi:

- ① Selezione Su RICETTA Vieni fatta con una Scansione Seq.
- ② Ordine Esecuzione del JOIN (R  $\bowtie$  C)  $\bowtie$  I
- ③ JOIN fatto con NESTED LOOP JOIN
- ④ RISULTATO del Primo JOIN Intermedio Rimane NEL BUFFER

M Ci Sono le Seguenti Statistiche Sulle TABELLE:

	NP	NR	VAL(Regione)	VAL(Ricetta)
RICETTA	12	260	20	
INGREDIENTE	40	1200		
COMPOSIZIONE	200	13000		260

Ci Sono 3 Operazioni

2 Join

1 Selezione

R.REGIONE = 'veneto'

Scan Seq

COSTO =  $NP(R) = 12$

Già State Selezionate

$R \bowtie C$



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

$$12 + \frac{NR(\text{rietta})}{VAL(\text{REGI, RICETTA})} \cdot 200$$

$$12 + \frac{260}{200} \cdot \frac{10}{200} = 26.12$$

Selezione + JOIN

\* Se UNICO Selezione con Scansione per il JOIN Vedo a Cambiare il Valore 'NR(R)' visto che NON Saranno TUTTE ma SONO Solo Quelle 'VENETO'

Risultato del Primo Join

$R' \bowtie S$

Mantenuto nel BUFFER

Costo =  $\emptyset$

$\emptyset$

$NP(R') + NR(R') \cdot NP(S)$

$$\frac{NR(C)}{VAL(R, C)} \cdot (\#Ric. \text{ Veneto})$$

$$\frac{50}{1300Q} \cdot \frac{13}{26Q}$$

COSTO  $\Rightarrow 650$

Costo Complessivo  $\Rightarrow$  Somma JOIN

$$\Rightarrow 26000 + 2612 = 28612$$

Cosa Cambia Se è DISPONIBILE un INDICE B+Tree su Attributo Codice di INGREDIENTE (con PROFONDITÀ 2)

Dove lo posso usare Questo Indice? Nel Secondo JOIN, il Primo Rimane Invaciato

$$\text{da } \overline{[NP(R) + NR(R) \cdot NP(S)]} \text{ a } \overline{[NP(R) + NR(R) \cdot (P_{cod, \text{Indice}} + \text{SELETTIVITÀ}_{cod, \text{Indice}})]}$$

$$\text{Costo DIVENTA } \Rightarrow \phi + 650 \cdot \left( 2 + \frac{NR(\text{ingrediente})}{val(cod, \text{Inged})} \right)$$

$\Rightarrow 1950$

→ Codice è chiave Primaria ed Allora  
è uguale a  $NR(\text{ingrediente})$

$$\text{Costo TOTALE } \Rightarrow 2612 + 1950 = 4562$$

## ESEMPIO:

PAZIENTE (codice\\_ssn, Nome, Cognome, ...)

VISITA (codice\\_ssn, codice\\_fiscale, Data, Ora\\_I, Ora\\_F, ...)

MEDICO (codice\\_fiscale, Cognome, Nome, Specialità)

① Se c'è Select Sequenziale è costo LETTURA + SCRITTURA

Si Azzerà se Viene Mantenuto Nel Buffer

② NR Vanno Stimate (Quelle Sopravvissute alla SELECT)

③ Se è PRIMARY KEY allora sono Pari a NR