CALCOLO DEL COSTO DI ACCESSO AI DATI

Nelle lezioni precedenti

- **Avete visto:**
  - **i le basi di dati relazionali ed il linguaggio SQL**
  - **la struttura dei files**
  - **gli indici B^+ tree**
- **questa è dedicata al calcolo del costo di accesso ai dati (tuple)**

scopo della lezione

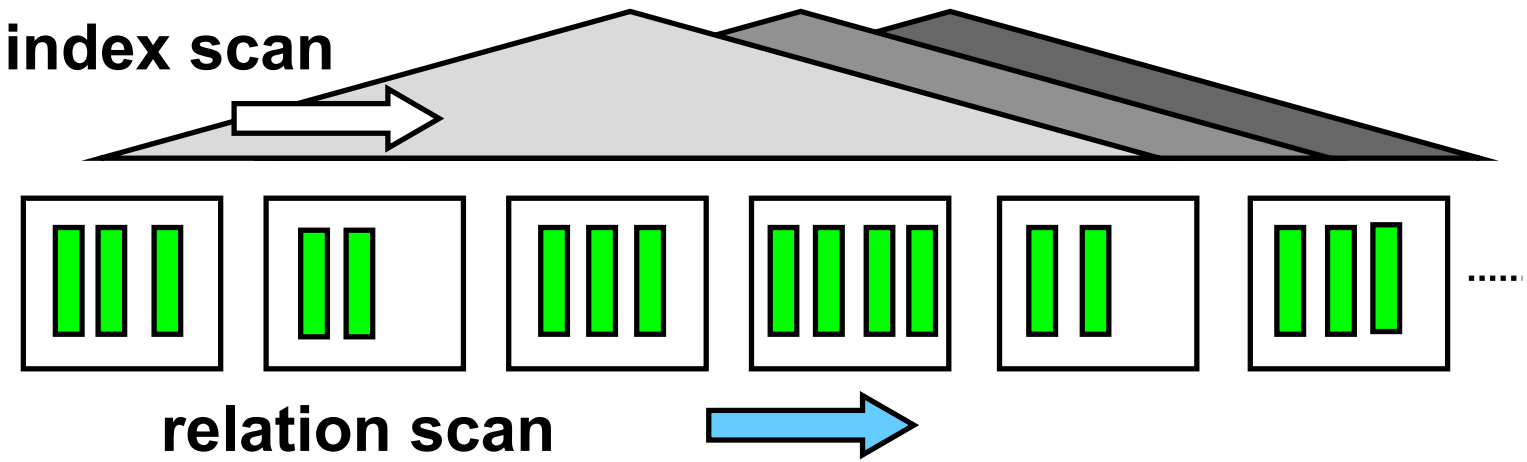
- **scopo:**
  - **valutare quale sia la migliore strategia di accesso per interrogazioni SQL sia su singola relazione sia nel caso di join**
  - **i criteri di valutazione servono anche a prendere decisioni sull'ordinamento delle relazioni e quali indici costruire**

ottimizzatori

- I criteri che vedremo sono in linea con i metodi e le scelte utilizzati dai query-optimizer dei DBMS relazionali
- lo scopo dei query-optimizer è infatti valutare quale sia la migliore strategia di accesso (access path) per le interrogazioni SQL degli utenti
- gli ottimizzatori non prendono decisioni sull'ordinamento delle relazioni e su quali indici costruire
- queste decisioni sono lasciate al DBA che deve valutare sulla base del carico di lavoro

decisioni

quali indici?



Relazione di NT tuple in NB blocchi (pagine)

utilizzo degli indici

- **Un indice può essere utilizzato per eseguire una interrogazione SQL se l'attributo su cui è costruito:**
- **compare nella clausola WHERE**
- **è contenuto in un FATTORE BOOLEANO**
- **il fattore booleano è ARGOMENTO DI RICERCA attraverso indice**
- **compare in un ORDER BY o GROUP BY**

utilizzo degli indici

- **Esempi: per la relazione**

IMPIEGATI (matr, cognome, nome, lavoro,
qualifica, salario, straordinario, dno)

1) la query: SELECT cognome, salario
FROM impiegati
WHERE dno = 51
AND salario > 2000
AND (lavoro = ‘fattorino’
OR lavoro = ’guardiano’)

oppure ….

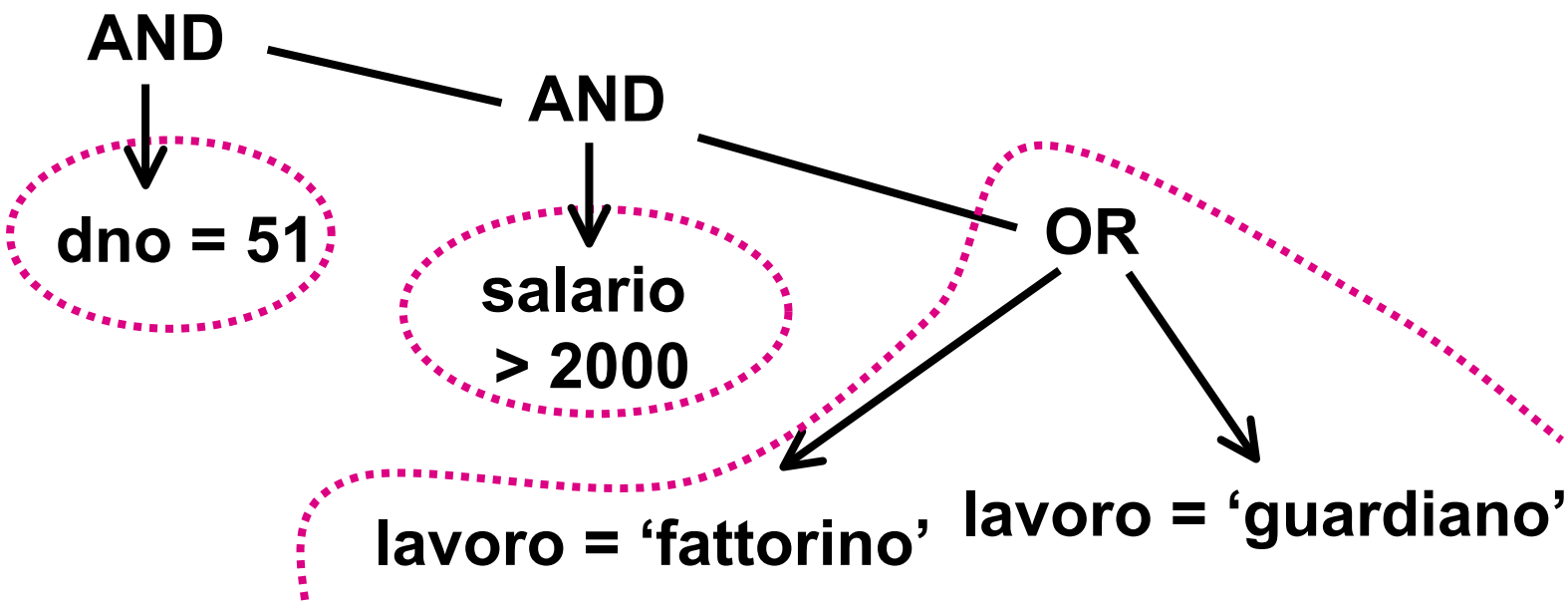
utilizzo degli indici

2) la query: SELECT cognome, salario
FROM impiegati
WHERE dno = 51
AND salario + straordinario > 3000
AND (lavoro = 'fattorino'
OR qualifica = 7)

utilizzo degli indici

Separazione della condizione WHERE in fattori booleani: un predicato è un fattore boleano se è collegato alla radice del WHERE-tree da AND

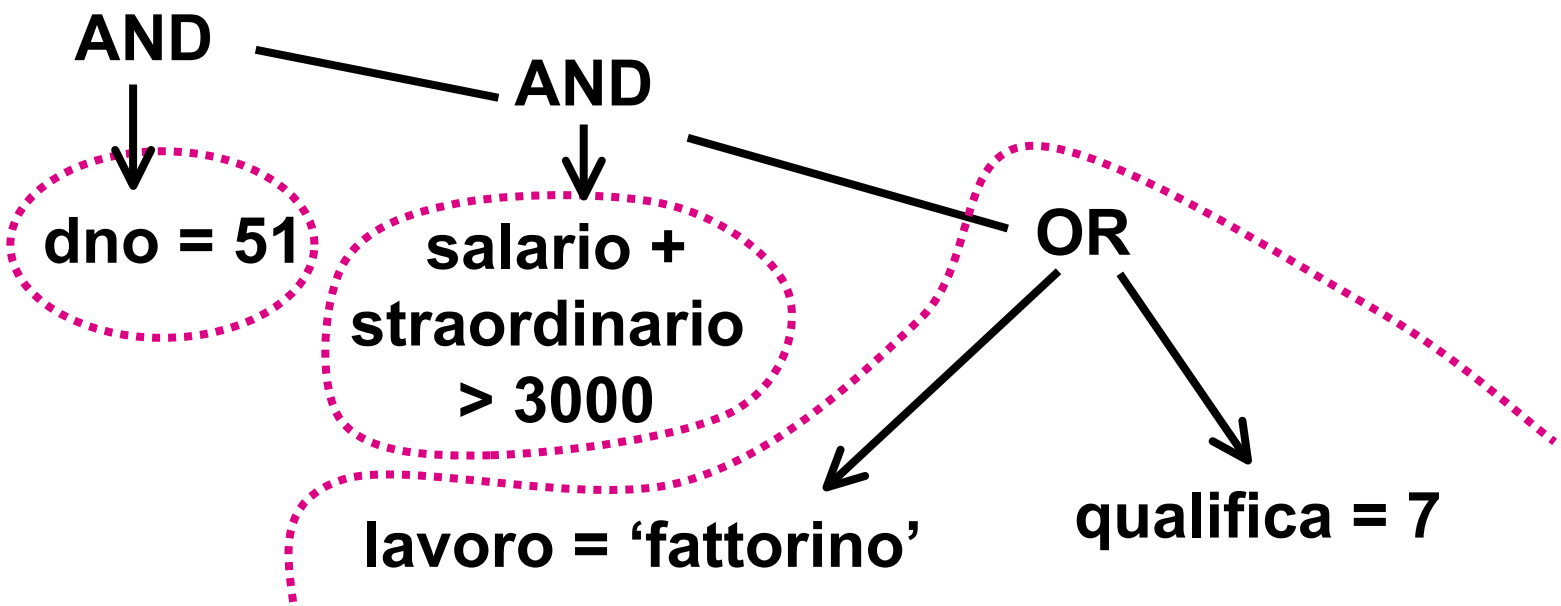
(query 1)



utilizzo degli indici

ovvero, un predicato è un fattore boleano se col valore falso determina un risultato vuoto per la query

(query 2)

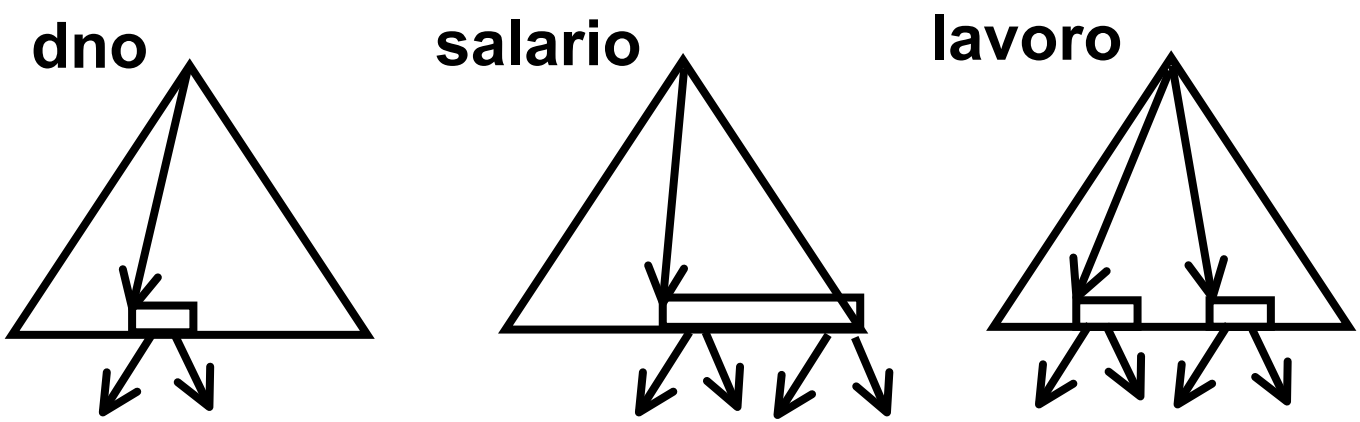


utilizzo degli indici

1) per la query 1 sono fatt. bool. argomenti di ricerca:

   dno = 51 , salario > 2000

   (lavoro = ‘fattorino’ OR lavoro = ’guardiano’)

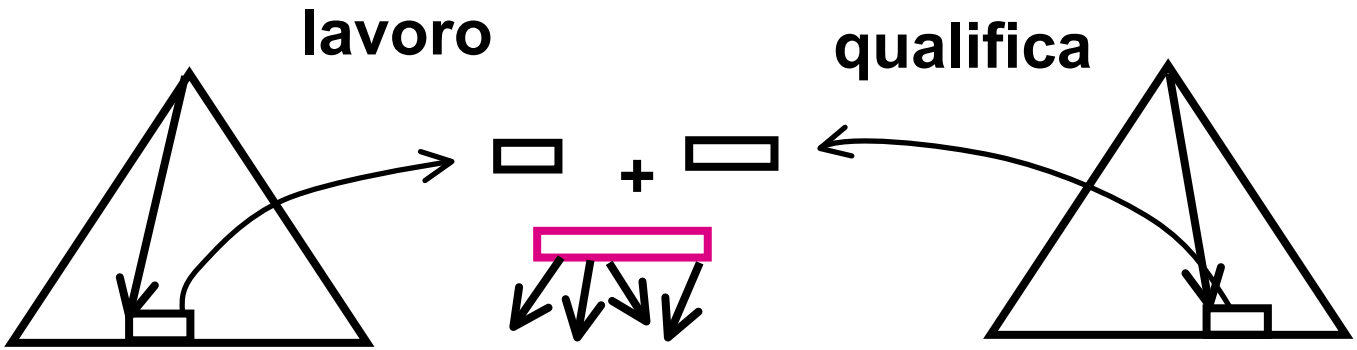


utilizzo degli indici

2) per la query 2 è fatt. bool. argomento di ricerca

solo: dno = 51

per (lavoro = 'fattorino' OR qualifica = 7) se il DBMS può usare più indici per una query:

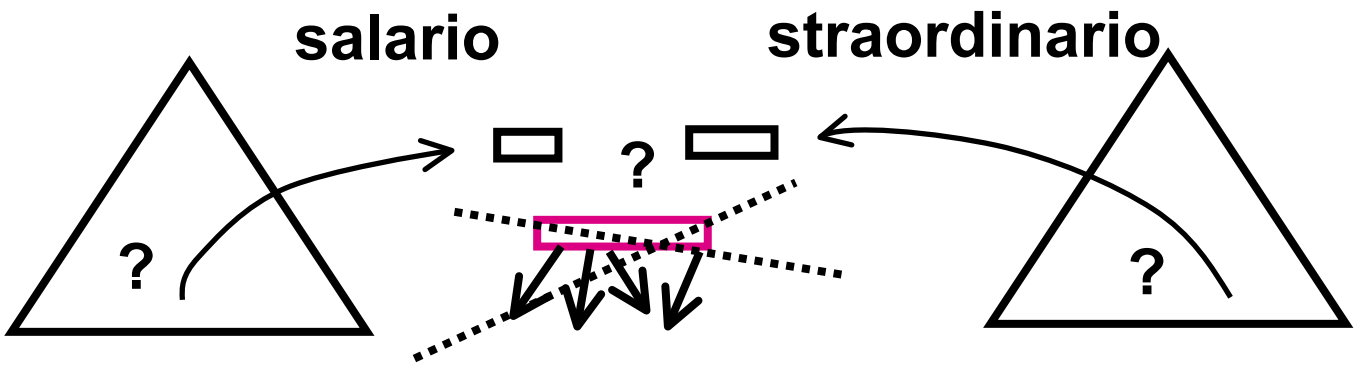


si può effettuare l'unione delle liste di TID

utilizzo degli indici

2) per la query 2 è non è fatt. bool. argomento di ricerca: **salario + straordinario > 3000**

sia che il DBMS possa usare più indici per una query che uno solo :



non si può effettuare l'unione delle liste di TID

utilizzo degli indici

per una query sono argomento di ricerca i predicati del tipo: attributo . comparatore. valore,

ad es.:

dno = 47 (<, < = ,> = , > , between)	**SI**
dno = $D (variabile di programma)	**SI**
dno = 47 OR dno = 32	**SI**
(IN corrisponde ad OR ma non sempre è SI)	
(dno = 47) OR (qualifica = 3)	**SI/NO**
salario + straordinario > 3000	**NO**
salario = straordinario (stessa relazione)	**NO**

modello di costo

- Un indice è utile per una query solo se il costo di accesso con l'indice è < costo dell'accesso sequenziale cioè < NB (NB/2 se attributo unique)
- il modello comunemente utilizzato (ce ne sono di molto più sofisticati e precisi) serve per previsioni di massima e si basa su:

– per la relazione : NT, NB
– per ogni indice :NL (numero di foglie)
– per ogni attributo :NK (cardinalità), max, min
– tutti valori desumibili dai cataloghi dei DBMS
– le grandezze sono uniformemente distribuite

selettività

- **Un predicato è selettivo se ci si aspetta che non tutte le tuple lo soddisfino**
- **fattore di selettività (filtro) F di un predicato : frazione di tuple che soddisfano il predicato**

$$F = ET / NT = EK / NK$$

(ET = tuple selezionate, EK = valori selezionati)

- **A = valore :** $F_A = 1 / NK_A$

  dno = 24 : $F_{dno} = 1/ NK_{dno}$
- **A IN (val1, val2, val3) :** $F_A = 3 / NK_A$

  dno IN (24, 36) : $F_{dno} = 2/ NK_{dno}$

  analogamente per dno = 24 OR dno = 36

selettività

- A > valore : $F_A = EK / NK_A$

  voto > 27 : $F_{voto} = 4/ 15$

  se i voti vanno da 17 a 31 e supponendo che tutti siano stati assegnati almeno una volta

  per salario > 2000 (range la cui cardinalità non è controllabile) si può prendere:

  $F_{salario} = (max(sal) - 2000) / (max(sal) - min(sal))$

  analogamente per BETWEEN 2000 AND 2300
  $F_{salario} = (2300 - 2000) / (max(sal) - min(sal))$

selettività

- **Predicati su attributi diversi**

– **pred1 OR pred2 :**

$$F = F_{pred1} + F_{pred2} - F_{pred1} F_{pred2}$$

– **attributo A = attributo B** *(predicato di join)*

$$F = 1 / \max(NK_A, NK_B)$$

- **Numero di tuple del risultato:**

– $E = NT \times F_{pred1}$ **e**

–**nell'ipotesi di assenza di correlazione tra i valori degli attributi, per più predicati:**

$$E = NT \times \Pi_i F_{pred\ i}$$

costo di accesso

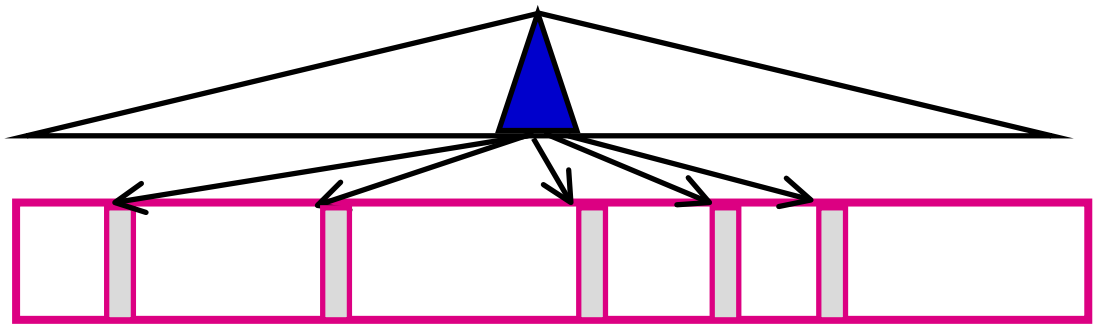
I casi esaminati si differenziano a seconda che:

- **indice clustered / unclustered**
- **attributo unique / con ripetizione dei valori**
- **predicato di uguaglianza / di range / OR**
- **uso di un solo indice / più indici**

costo di accesso

Il costo C è dato dalla somma :

$$C_{indice} + C_{relazione}$$

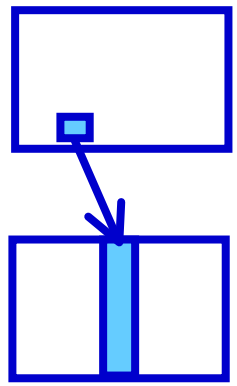


costo di accesso

- **indice clustered / unclustered su attributo unique con predicato di uguaglianza:**

  esempio, matr = 236 (sulla relazione impiegati)

foglia indice



blocco dati

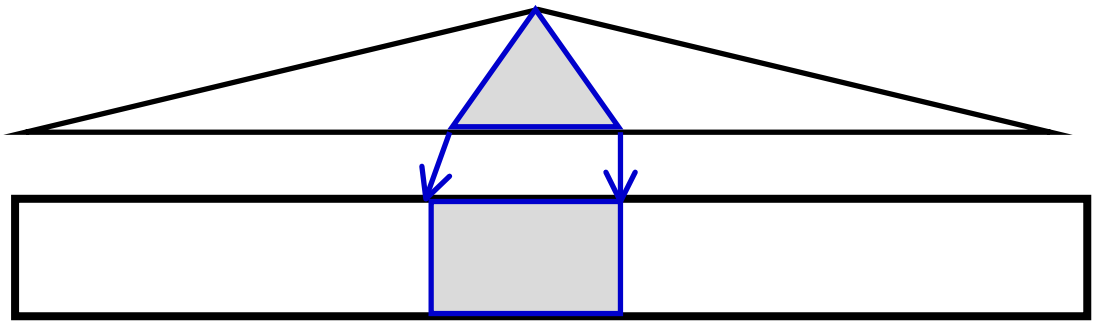
Costo = 1 foglia + 1 blocco = 2

(se l'indice è clustered o unclustered è lo stesso)

costo di accesso

- **indice clustered con ET > 1:**

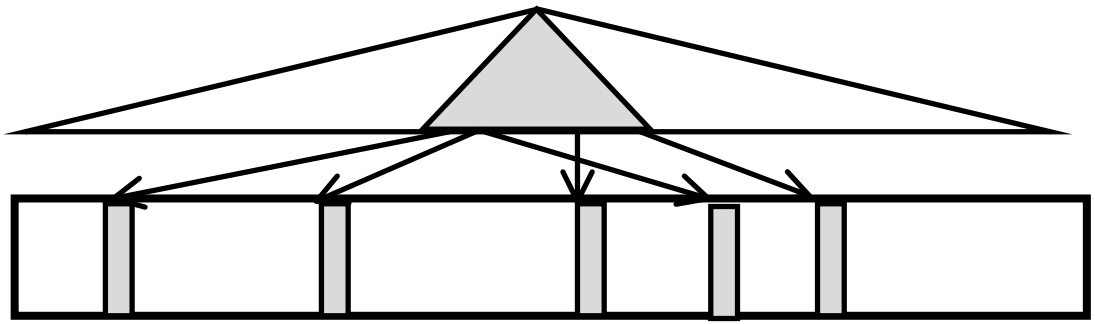
  matr between 236 and 312, lavoro = ‘guardiano’



Costo = $\lceil F \times NL \rceil + \lceil F \times NB \rceil$

costo di accesso

- **indice unclustered:**
  lavoro = guardiano



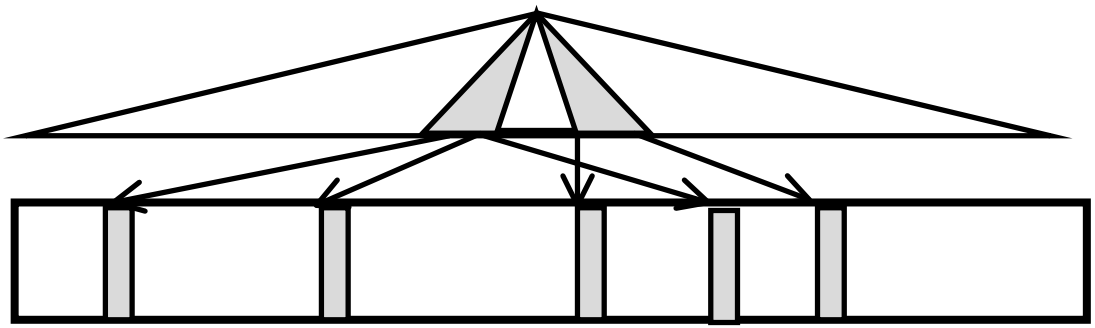
$$Costo = (\lceil F \times NL \rceil + \lceil F \times NT \rceil) \quad \text{con } F = F_{valore}$$

analogamente per il caso clustered:

$$Costo = (\lceil F \times NL \rceil + \lceil F \times NB \rceil)$$

costo di accesso

- **indice unclustered / clustered (predicato OR):**
  lavoro = guardiano OR portiere



Costo = 2 × (⌈F × NL⌉ + ⌈F × NT⌉) **con F = F_{valore}**

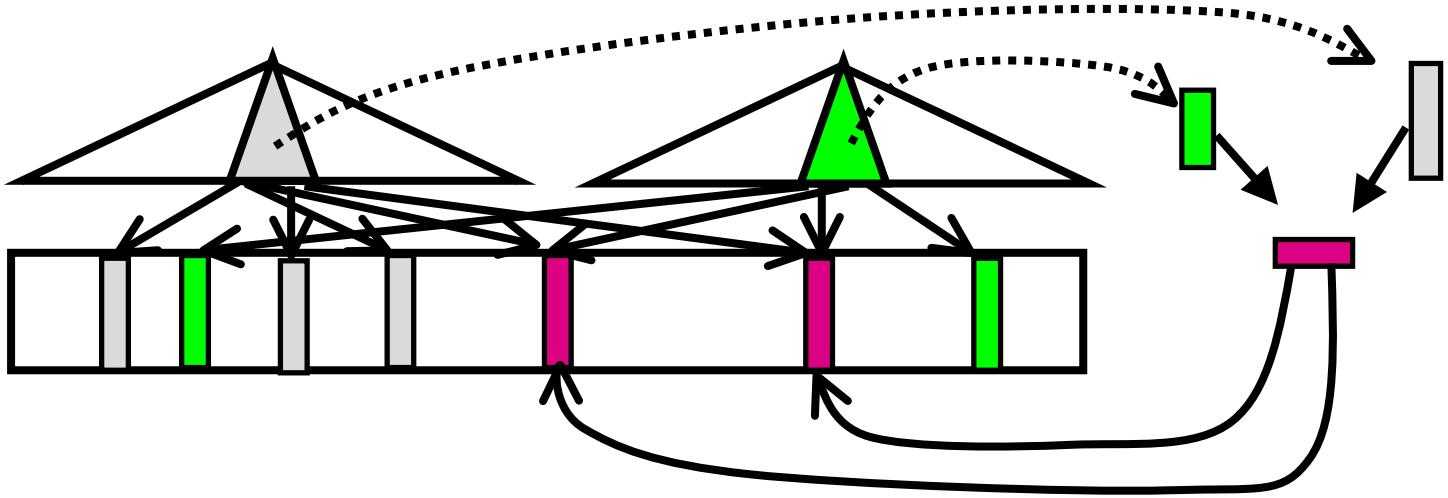
sono 2 accessi distinti,
analogamente per il caso clustered

Costo = 2 × (⌈F × NL⌉ + ⌈F × NB⌉)

costo di accesso

- **Uso di più indici unclustered con ET > 1: intersezione /unione dei TID estratti dagli indici**

  matr between 236 and 312, lavoro = guardiano



$$\text{Costo} = \Sigma_k \lceil F_k \times NL_k \rceil + \lceil \Pi_k F_k \times NT \rceil$$

costo di accesso

Esempio:

SELECT * FROM IMPIEGATI
WHERE lavoro = 'fattorino' AND salario < 1500

con $NT = 10000$, $NB = 1000$,
$NK_{sal} = 100$, $NK_{lav} = 50$

indici: **clustered** su salario con $NL = 160$
unclustered su lavoro con $NL = 100$

supponiamo che la selettività sia:

$F_{lav} = 1 / 50 = 0.02$
$F_{sal} = (1500 - min) / (max- min) = 0.1$
[es. min=1400, max=2400]

costo di accesso

costo delle scansione sequenziale:

$C_{seq} = 1000$

costo dell'indice su lavoro (unclustered):

$C_{lav} = \lceil F_{lav} \times NL_{lav} \rceil + \lceil F_{lav} \times NT \rceil =$

$0.02 \times 100 + 0.02 \times 10000 = 2 + 200 = 202$

costo dell'indice su salario (clustered):

$C_{sal} = \lceil F_{sal} \times NL_{sal} \rceil + \lceil F_{sal} \times NB \rceil =$

$0.1 \times 160 + 0.1 \times 1000 = 16 + 100 = 116$

$C_{seq} > C_{lav} > C_{sal}$

costo di accesso

scambiamo adesso l'ordinamento per gli indici:

costo dell'indice su lavoro (clustered):

$$C_{lav} = \lceil F_{lav} \times NL_{lav} \rceil + \lceil F_{lav} \times NB \rceil =$$

$$0.02 \times 100 + 0.02 \times 1000 = 2 + 20 = 22$$

costo dell'indice su salario (unclustered):

$$C_{sal} = \lceil F_{sal} \times NL_{sal} \rceil + \lceil F_{sal} \times NT \rceil =$$

$$0.1 \times 160 + 0.1 \times 10000 = 16 + 1000 = 1016$$

$$C_{sal} > C_{seq} > C_{lav}$$

costo di accesso

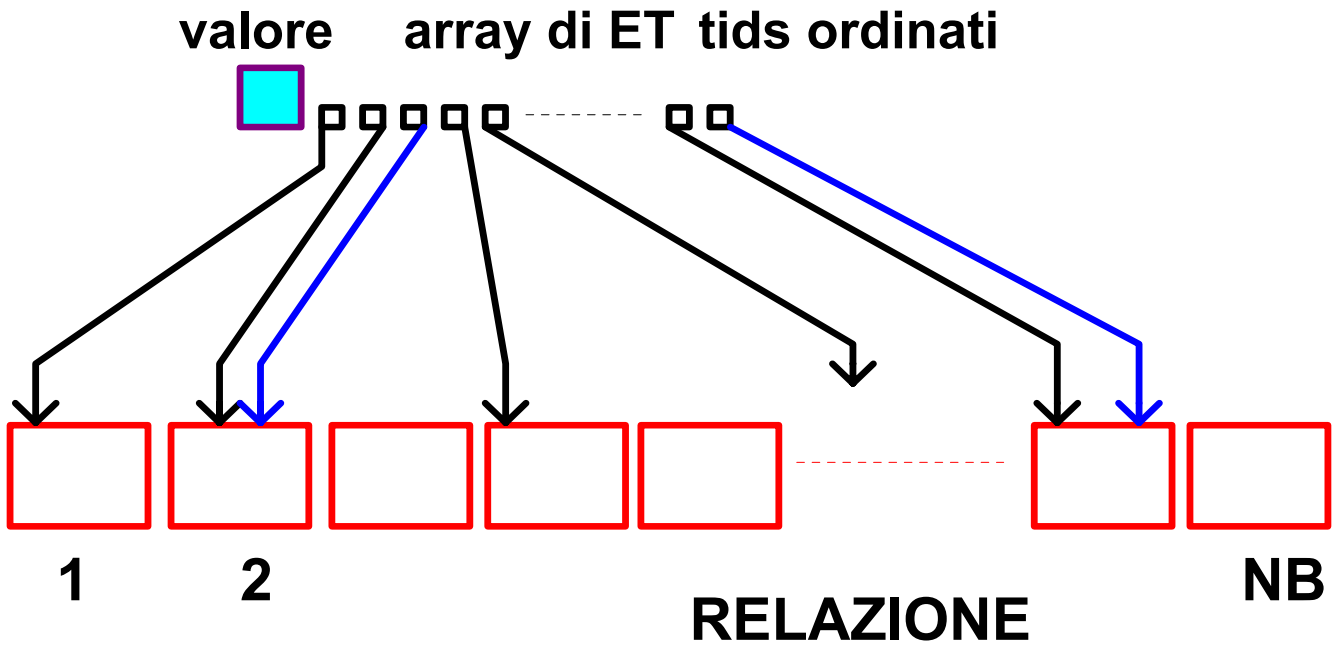
tuple del risultato:

$$E = F_{lav} \times F_{sal} \times NT = 20$$

- la **scelta migliore** per la query è avere un ordinamento su lavoro e un indice su lavoro, mentre l'indice su salario non deve essere costruito
- il **miglioramento** che si ottiene rispetto all'assenza di indici e ordinamenti ò di **1 a 45**
- nel caso in cui l'ordinamento su salario si di utilità per altre query l'indice **unclustered** su lavoro porta ad un miglioramento di **1 a 9**

uso degli indici

Miglioramento della formula di costo nel caso di ordinamento dei TIDs



Quindi in generale si ha $C_{relazione} \leq min(ET, NB)$.

formula di Cardenas

$C_{relazione}$ puo' essere calcolato meglio usando la formula di CARDENAS (Comm. ACM 1975):

$$\Phi(k,n) = \lceil n \times (1-(1-1/n)^{k}) \rceil$$

$$C_{relazione} = \Phi(ET,NB) = \lceil NB \times (1-(1-1/NB)^{ET}) \rceil$$

La formula e' valida sotto le seguenti ipotesi:

a) tutti i record sono (e rimangono) equiprobabili,

b) tutti i blocchi contengono lo stesso numero di tuple,

c) come conseguenza di a) e b) tutti i blocchi sono equiprobabili.

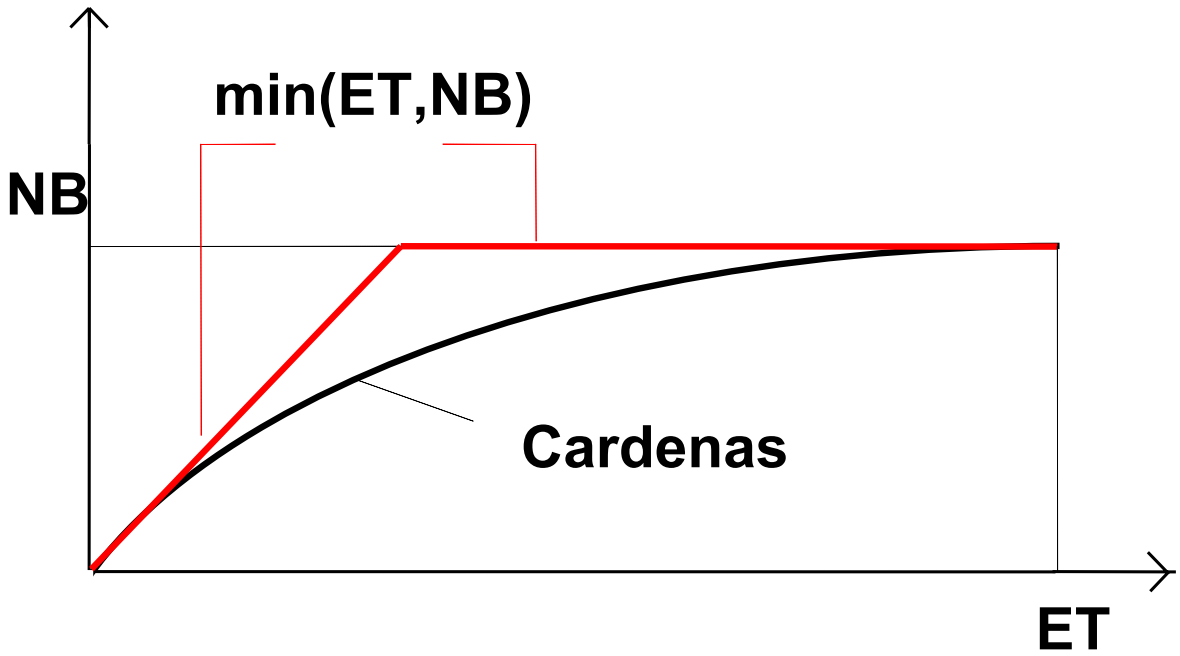
Cardenas

Φ(ET,NB) può essere ricavata come segue:

- 1/NB è la probabilità che un blocco contenga una data tupla estratta dalle ET,
- 1-1/NB è la probabilità che un blocco non contenga una data tupla estratta dalle ET
- $(1-1/NB)^{ET}$ è la probabilità che un blocco non contenga nessuna delle ET tuple
- $1-(1-1/NB)^{ET}$ è la probabilità che un blocco contenga almeno una delle ET tuple e quindi venga visitato,
- $NB \times (1-(1-1/NB)^{ET})$ è il numero di blocchi che ci si aspetta contengano almeno una delle ET tuple

andamento di Φ

L'andamento generale della Φ è riportato in figura:



andamento di Φ

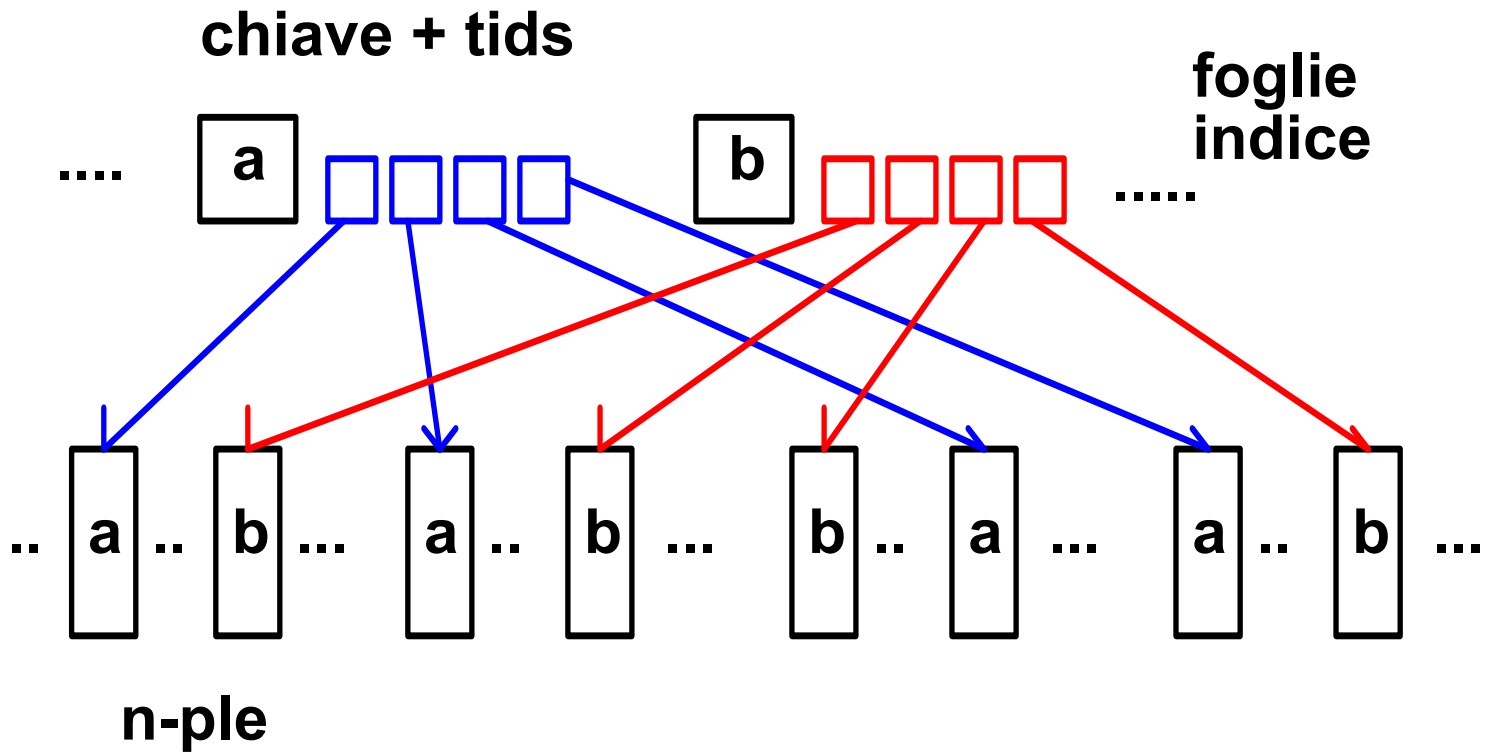
Tabella con NB = 100

- **La formula di Cardenas calcola in ogni caso il numero Φ di blocchi visitati (con qualsiasi successione).**
- **Il numero di accessi $C_{relazione}$ è calcolabile con la formula di Cardenas solo se i TIDs sono in ordine.**

ET	⌈Φ⌉
10	10
20	18
30	26
40	33
50	39
60	45
70	51
80	55
90	60
100	63
150	78
200	87
250	95
300	95
500	99

utilizzo degli indici

Caso di indice unclustered "localmente ordinato" : con i TIDs in ordine crescente per ogni valore della chiave



utilizzo degli indici

- **Predicato :** **Col = valore**

$$C = \lceil F_{col} \times NL \rceil + \lceil \Phi(ET,NB) \rceil$$

dove $ET = F_{col} \times NT$

- **Predicato di tipo :** **valore 1< Col < valore2**

$$C = \lceil F_{col} \times NL \rceil + \lceil EK \times \Phi(DK,NB) \rceil$$

dove $EK = F_{col} \times NK$, $DK = NT/NK$

utilizzo degli indici

- **Rivediamo i calcoli con Φ per l'esercizio precedente con indici unclustered:**

  predicato: lavoro = 'fattorino'

  $$C = \lceil F_{lav} \times NL_{lav} \rceil + \lceil \Phi(ET,NB) \rceil$$

  dove $ET = F_{lav} \times NT = 200$

  $$C = \lceil F_{lav} \times NL_{lav} \rceil + \lceil \Phi(200,1000) \rceil$$

  $= 2 + 182 = 184$ **(era 202)**

utilizzo degli indici

- **Predicato : salario < 1500**

$$C = \lceil F_{sal} \times NL_{sal} \rceil + \lceil EK \times \Phi(DK,NB) \rceil$$

dove $EK = F_{sal} \times NK_{sal}$, $DK = NT/NK$:

$$EK = F_{sal} \times NK = 0.1 \times 100$$

$$DK = NT/NK = 10000/100 = 100$$

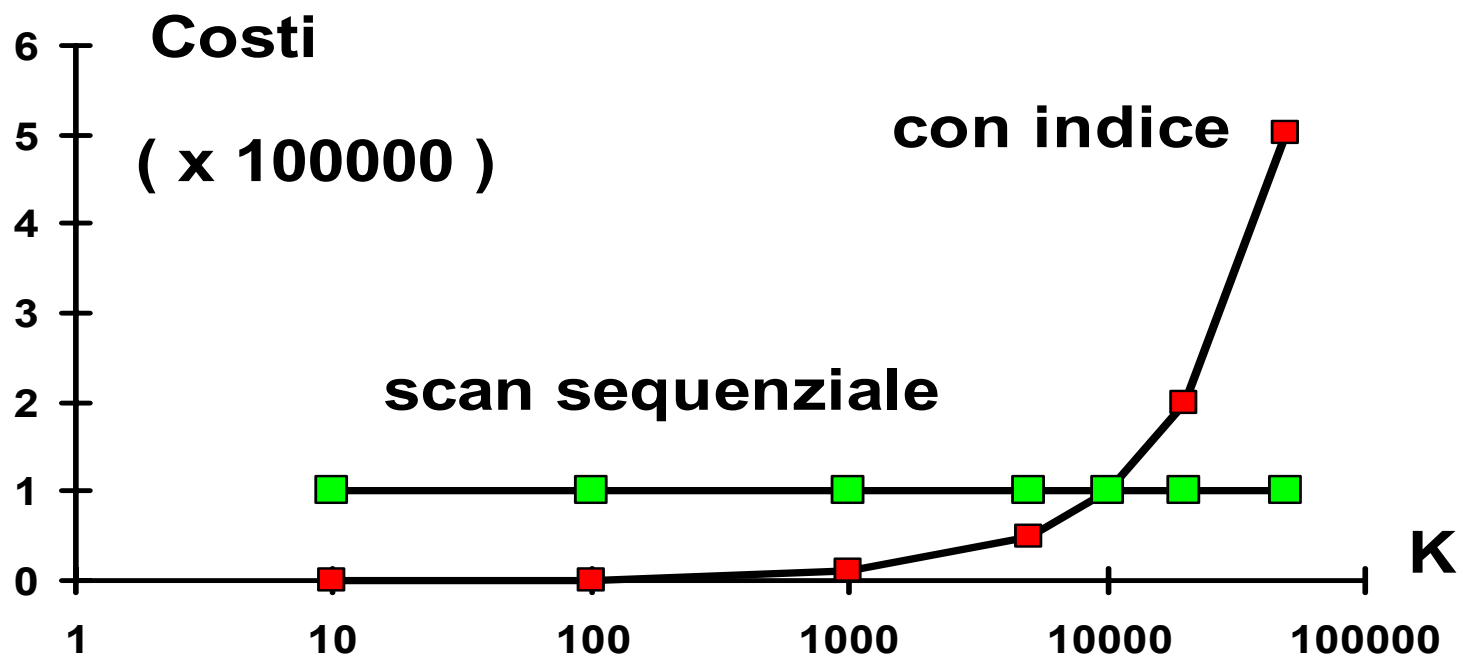
$$C = 16 + \lceil 10 \times \Phi(100,1000) \rceil =$$

$$= 16 + 952 = 968$$ (era 1016)

utilizzo degli indici

Esempio: variazione con K

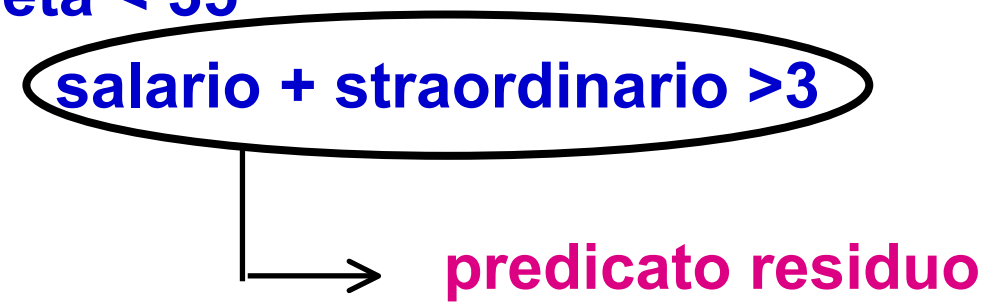
NL = 1·800, NR = 1·000·000, NK = 100·000, NB = 100·000



utilizzo degli indici

- **Le tuple del risultato vengono poi controllate con i predicati residui:**

```
SELECT * FROM IMPIEGATI
WHERE lavoro = 'fattorino'
AND età < 35
AND salario + straordinario >3
```

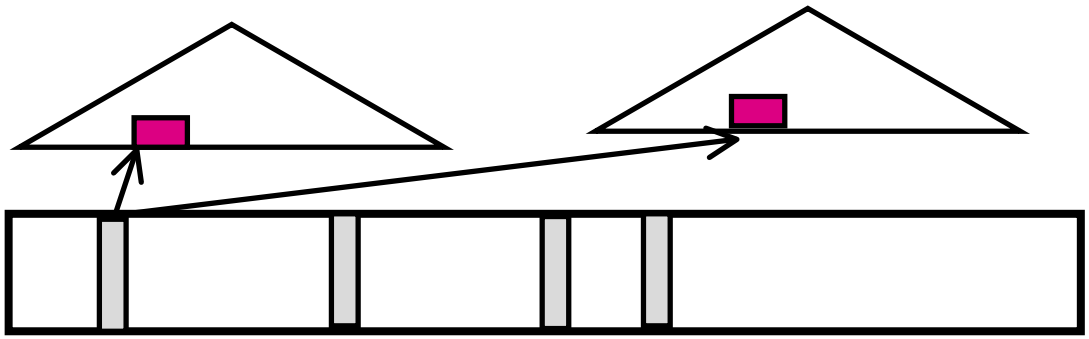


La selettività dei predicati residui è difficile da calcolare

uso degli indici

- **Perché non mettere indici su tutti gli attributi?**
  Il query optimizer potrebbe poi scegliere:

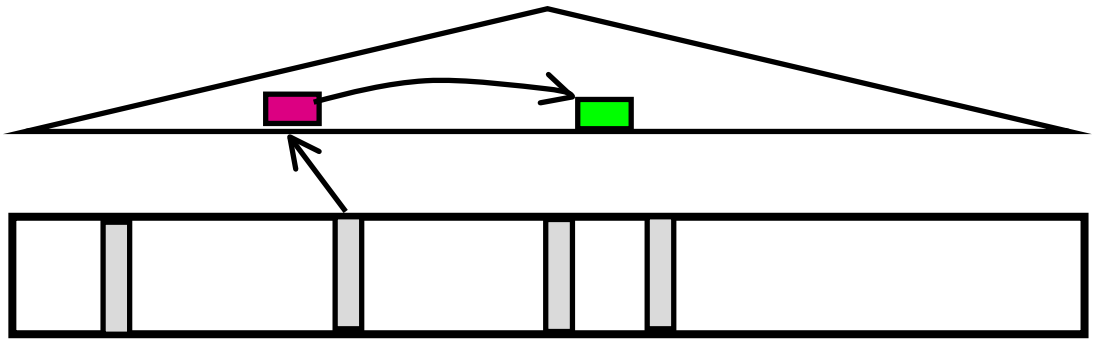
caso DELETE: eliminazione di TID



Costo = $N_{ind} \times E'$ **(N_{ind} : numero indici**
$E' = E \times 2$)

uso degli indici

caso UPDATE: spostamento di TID



Costo = $N_{ind} \times 2 \times E'$ **(N_{ind}: numero indici**
$E' = E \times 2$)

Un numero indici troppo elevato comporta un eccessivo costo di modifica della relazione

proiezione

- **Le tuple risultato della proiezione sono distinte:**

```
SELECT DISTINCT B, C
FROM REL
```

Un calcolo approssimato del numero di tuple del risultato è il seguente:

se NT sono le tuple di REL, NK_B e NK_C le cardinalità di B e C, $NK_B \times NK_C$ sono le possibili coppie (b,c) di valori distinti, quindi

$E_{B,C}$ è calcolabile con la formula di Cardenas:

$$E_{B,C} = \Phi(NT, NK_B \times NK_C)$$

proiezione

- **Nel caso di query con selezione e proiezione:**

```
SELECT DISTINCT C
FROM REL
WHERE B = 10
```

NT/ NK_B sono le tuple di REL che soddisfano il predicato, ciascuna di queste può assumere uno dei NK_C valori di C, quindi E_C è calcolabile con la formula di Cardenas:

$$E_C = \Phi(NT/NK_B , NK_C)$$