

# HOMEWORK 4

SELECT SELECT OTTO, SUM(COSTO), SUM(Numero Bre) 450

FILTER  $G_{sum(COSTO) \geq 2000} 4500 \cdot \frac{1}{10} = 450$   
GB NO HASH GB EID 4500

~~$9 \cdot 10^4 \cdot \frac{1}{20} = 4500$~~  NO

SID  
NESTED  
LOOP  
JOIN

$10^8 \cdot \frac{9}{10000} = 9 \cdot 10^4$

TABLE ACCESS  
BY ROW ID

$10^9 \cdot \frac{1}{10} = 10^8$   
DATA = 1/2/2023 AM  
DATA = 3/2/2023

FILTER  $G 5 \cdot 10^4 \cdot \frac{9}{10} = 45 \cdot 10^3$   
TABLE ACCESS  
BY ROW ID

$G 5 \cdot 10^3 \cdot \frac{1}{1000} = 5 \cdot 10^4$   
CITTA = Torino

SP  $10^9$  PT-TRG  
FAST FULL INDEX SCAN  
ON SP(DATA)

FATTORIO DI  
RIDUZIONE  $\frac{9}{10000}$   
IPOTIZZANDO che  
anche OTTO segue  
la stessa distribuzione  
di CITTA e TRONCINO

FAST FULL  
INDEX SCAN  
ON E(CITTA)

TABLE ACCESS BY ROWID  
FATTORIO DI  
RIDUZIONE  $\frac{2}{10}$   
PROBARE: 'PIEMONTE' OR  
PROBARE: 'LIGURIA' segue la stessa  
distribuzione di  
REGIONE

BITMAP INDEX ACCESS ON IP(REGIONE)  
FATTORIO DI RIDUZIONE  $\frac{2}{10}$

IPOTIZZANDO che anche  
SID segue la stessa distribuzione  
di CATEGORIA

FATTORIO DI  
RIDUZIONE  $\frac{1}{20}$

FILTER  $G_{COUNT(*) > 1} 10 \cdot \frac{1}{2} = 5$   
GB HASH  
SOSID

PID  
HASH  
JOIN

NESTED  
LOOP  
JOIN

$2 \cdot 10^5$  SO  
FTA

FILTER  
CATEGORIA  
'INTERNI'

100 S  
FTA



# STRUTTURE FISICHE ACCESSORIE

osservo che il DBMS abbia già indicizzato le chiavi primarie

IP (P.d)

~~SPC (Dat)~~

SPC (Dat)

DB+IPOT

S(Siol) | E(Eid)

non indicizzo S perché è di dimensione ridotta.  
In più volentieri di creare un indice su IP (REGIONE), anche se la tabella non è troppo grande e la selettività è solo  $\frac{1}{10}$ .  
In questi casi sarebbe più appropriato un indice hash, perché  
dovrebbe prestare di efficienza, ~~spesso~~ clustered, per non dover  
accedere tramite B+ 10. È alternativa, avendo 2 divisioni in  
OR, anche un indice bitmap unclustered con accessi molto  
meno migliorabili le prestazioni.

Un indice più conveniente da creare sarebbe quello su  
E(CITTA'), perché consente di accedere a molte voci  
bloccando rispetto a una full table scan, avendo una tabella  
grande e un predicato la selettività molto alta ( $\frac{1}{2000}$ ).  
Anche in questo caso conviene un indice hash.

E(TIPODIPICCO) invece ha una selettività troppo bassa per essere  
contenuto da solo, al massimo si potrebbe creare un indice  
composito su <sup>(EID, CITTA', TIPODIPICCO)</sup> ~~per avere un~~ per avere un  
indice grande ~~per avere un~~ ma anche in  
questo caso i vantaggi non superano gli svantaggi dovuti  
probabilmente ~~prestanzi~~  
all'intensità dell'indice.

## ANTICIPO della GROUP BY

Non lo schema in figura nessuna delle 2 GB è possibile  
anticipare, perché si richiederebbe l'attributo necessario  
per il join.  
Considerando l'ordine dei join (prima SID e poi SID, prima SID e poi EID),  
ovviamente non è possibile anticipare il GB, perché vorrebbe  
dove anticiparlo prima del join su SID (query stessa) e prima di quella su  
EID (query stessa), se non si aspetta il risultato della join su SID sotto dei filtri, ma sotto  
questo caso il join è molto più complesso mettere la join dopo il join, rendendo l'operazione  
impossibile.