QUADERNO_3

lunedì 12 dicembre 2022 17:16

Sono date le relazioni seguenti (le chiavi primarie sono sottolineate):

```
IMPRESA-PULIZIE(Pid, Nome, Indirizzo, Città, Regione)
SERVIZI-OFFERTI(Pid, Sid)
SERVIZIO(Sid, NomeServizio, Categoria)
EDIFICIO(Eid, NomeEdificio, TipoEdificio, Indirizzo, Città , Regione)
SERVIZI-PULIZIA(Pid, Eid, Data, Sid, Costo, NumeroOre)
```

Si ipotizzino le seguenti cardinalità:

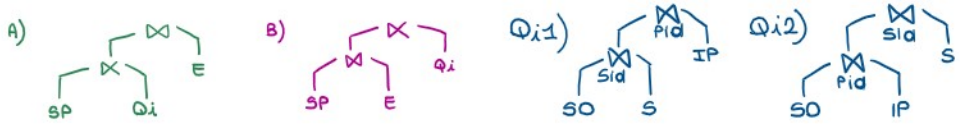
- card(IMPRESA-PULIZIE) = 10^4 tuple,
  valori distinti di Regione = 20
- card(SERVIZI-OFFERTI)= $2 \cdot 10^5$ tuple,
- card(SERVIZIO)= 100 tuple,
  valori distinti di Categoria = 10
- card(EDIFICIO)= $5 \cdot 10^7$ tuple,
  valori distinti di Città = 1000
  valori distinti di TipoEdificio = 10
- card(SERVIZI-PULIZIA)= 10^9 tuple,
  MIN(Data) = 1/1/2013, MAX(Data) = 31/12/2022

Inoltre si ipotizzi il seguente fattore di riduzione per le condizioni di group by:

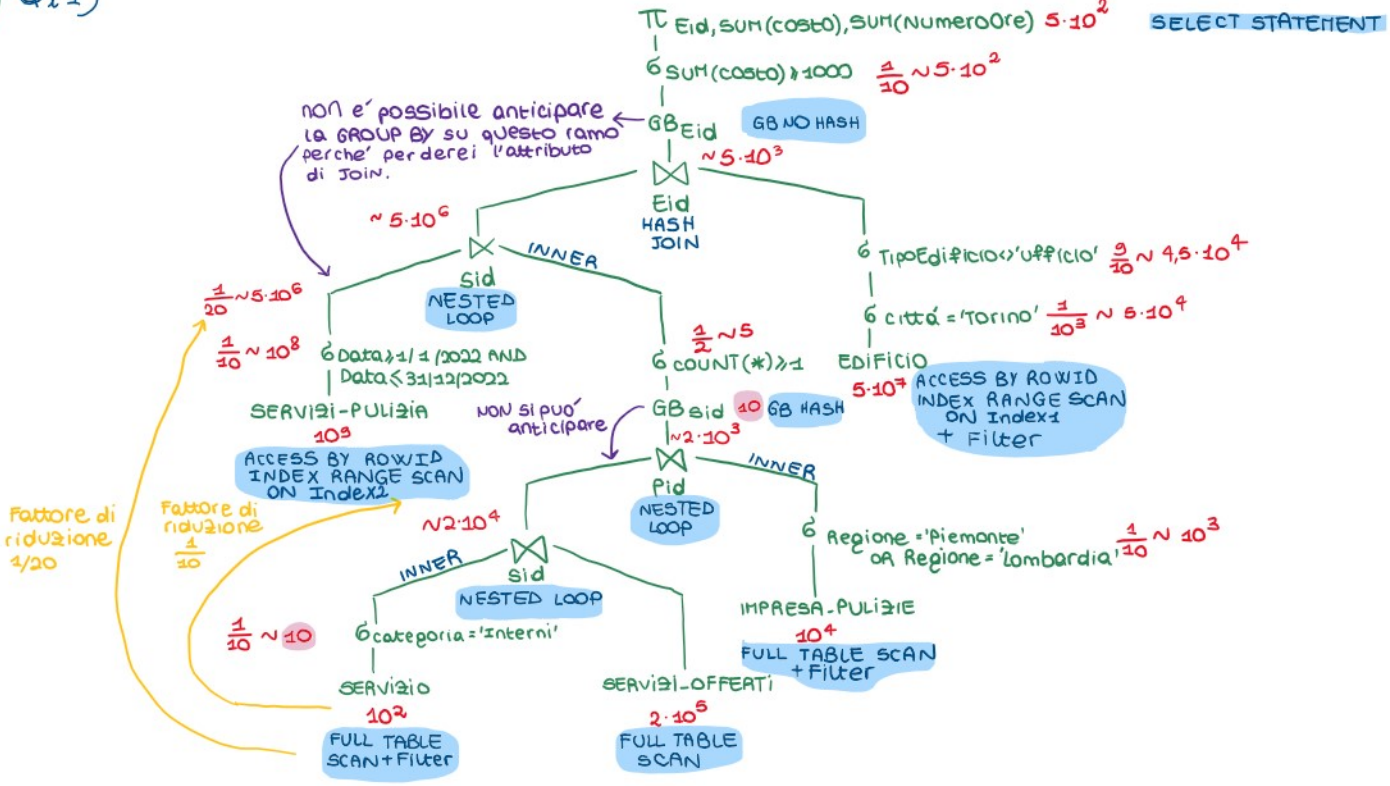
- having COUNT(*)≥1 $\simeq \frac{1}{2}$.
- having SUM(Costo)≥1000 $\simeq \frac{1}{10}$.

Si consideri la seguente query SQL:

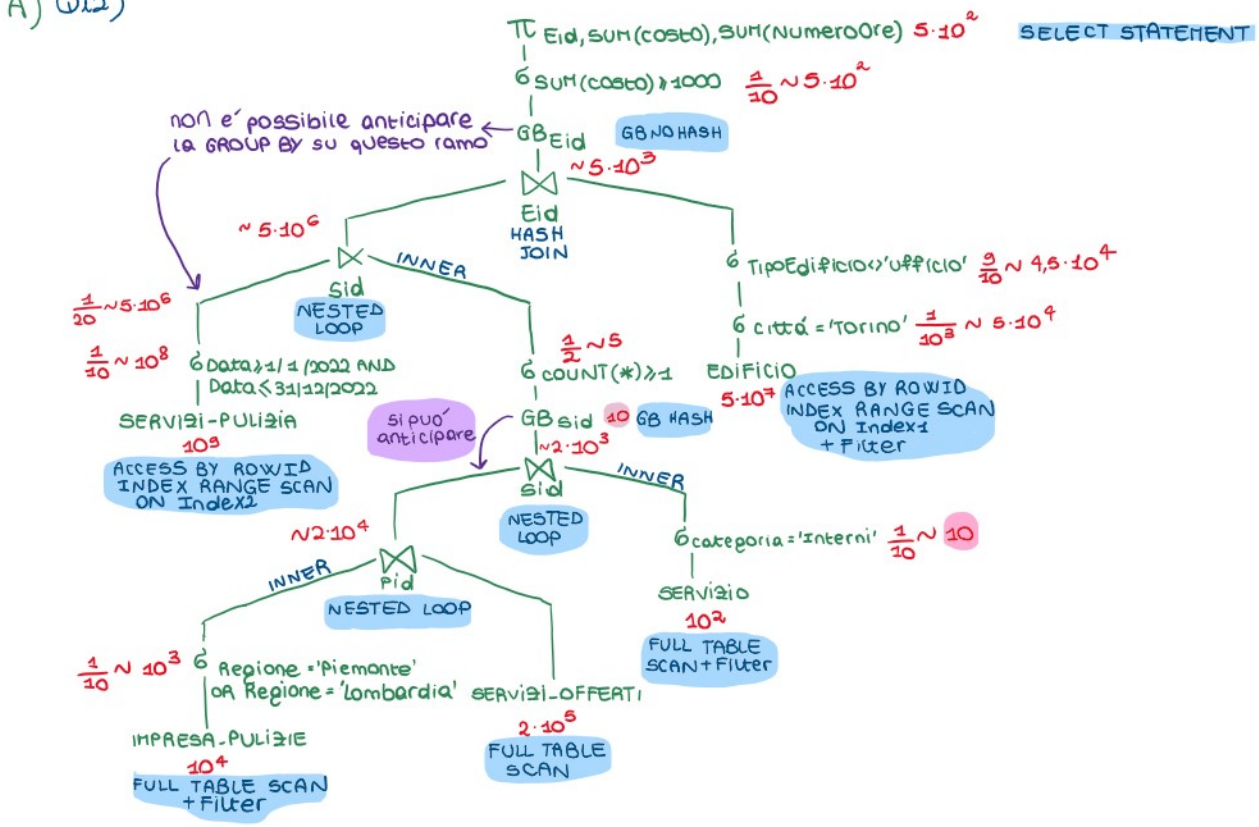
```
select Eid, SUM(Costo) as TotCost, SUM(NumeroOre) as TotOre
from SERVIZI-PULIZIA SP, EDIFICIO E
where SP.Data>=1/1/2022 and SP.Data<=31/12/2022
  and E.TipoEdificio <> 'Ufficio'
  and E.Città='Torino'
  and SP.Eid=E.Eid
  and SP.Sid IN ( select SO.Sid
                  from IMPRESA-PULIZIE IP, SERVIZIO S, SERVIZI-OFFERTI SO
                  where SO.Sid=S.Sid and SO.Pid=IP.Pid
                    and (Regione='Piemonte' or Regione='Liguria')
                    and Categoria='Interni'
                  group by SO.Sid
                  having COUNT(*)>=1)
group by SP.Eid
having SUM(Costo)>=1000
```



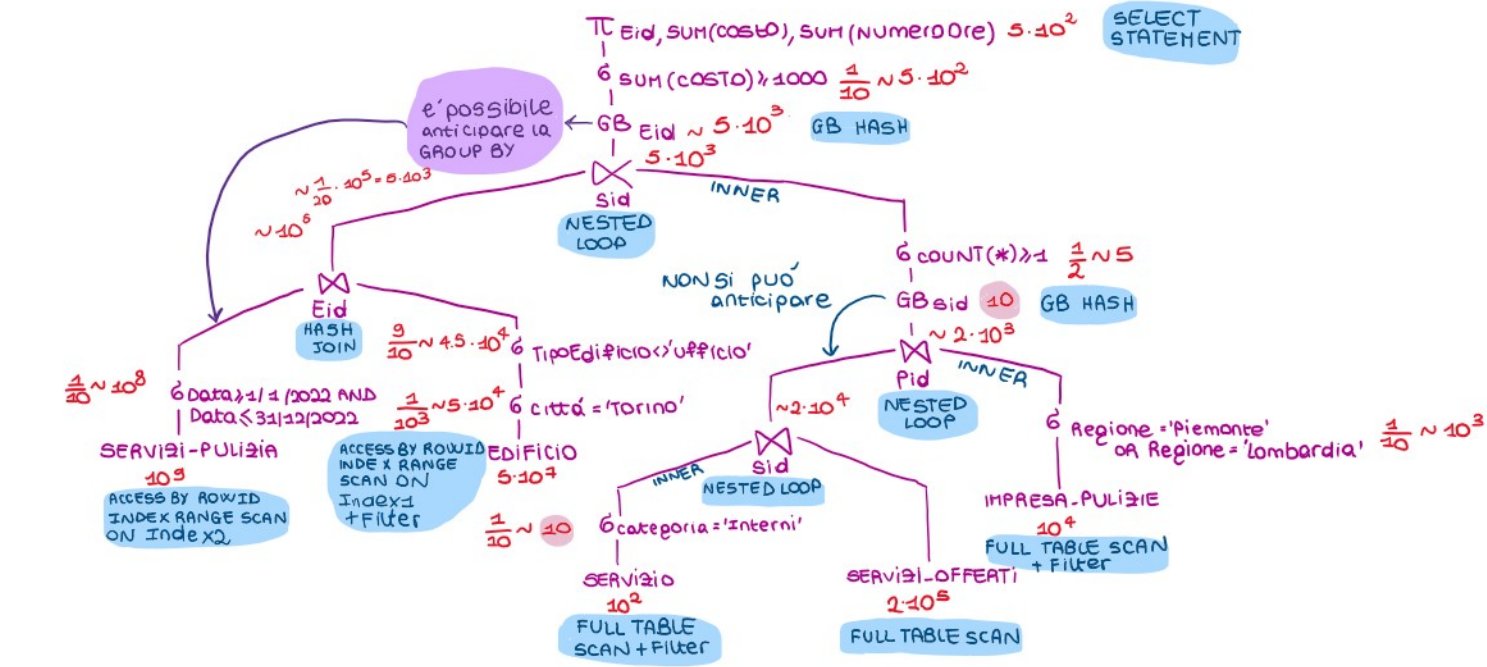
A) Qi1)

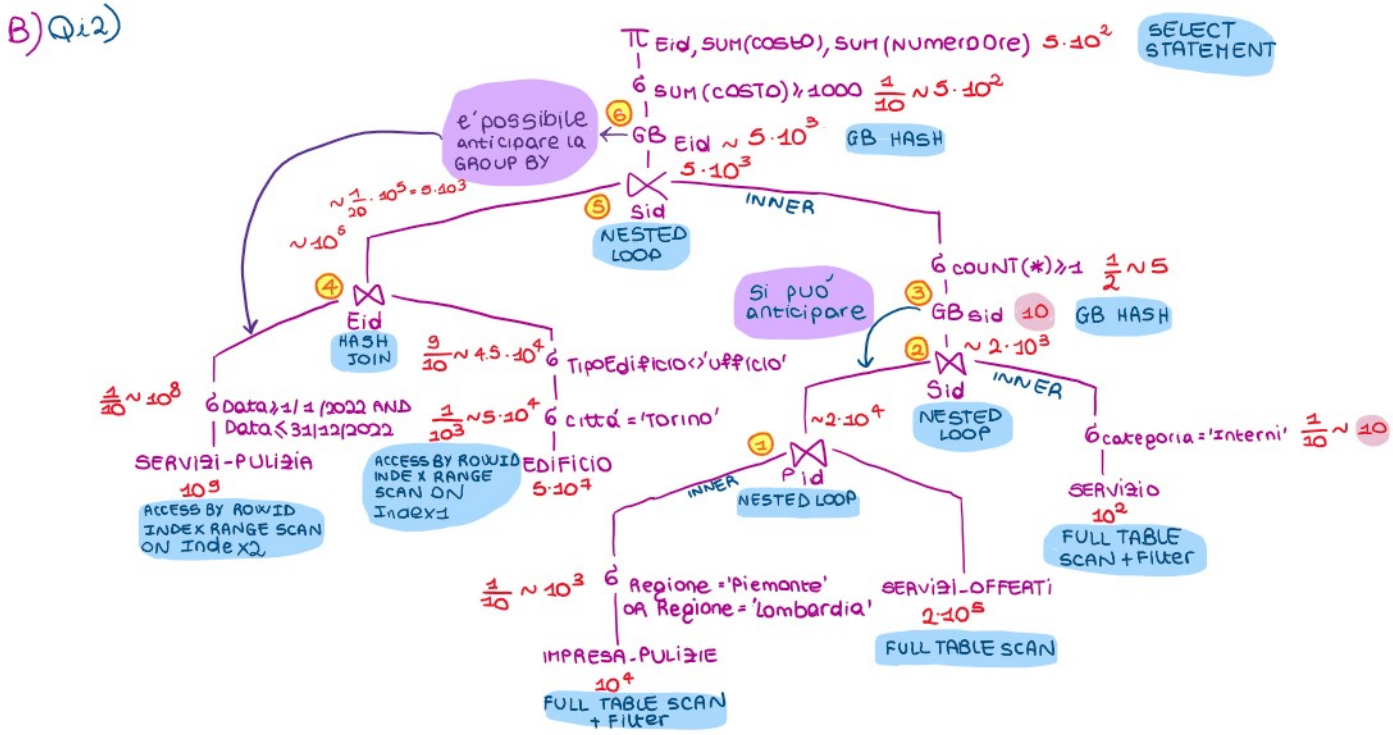


A) Qi2)


π Eid, SUM(costo), SUM(NumeroOre) 5·10^2
SELECT STATEMENT
σ SUM(costo)>1000 1/10 ~ 5·10^2
non è possibile anticipare la GROUP BY su questo ramo
GB Eid
GB NO HASH
~5·10^3
Eid
HASH JOIN
~ 5·10^6
Sid
NESTED LOOP
INNER
σ TipoEdificio<>'ufficio' 9/10 ~ 4,5·10^4
1/20 ~5·10^6
1/10 ~ 10^8
σ Data>1/1/2022 AND Data<31/12/2022
SERVIZI-PULIZIA
10^9
ACCESS BY ROWID INDEX RANGE SCAN ON Index2
si può anticipare
σ COUNT(*)>1 1/2 ~5
GB Sid 10
GB HASH
~2·10^3
σ città='Torino' 1/10^3 ~ 5·10^4
EDIFICIO
5·10^7
ACCESS BY ROWID INDEX RANGE SCAN ON Index1 + Filter
Sid
NESTED LOOP
INNER
~2·10^4
Pid
NESTED LOOP
INNER
σ categoria='Interni' 1/10 ~ 10
SERVIZIO
10^2
FULL TABLE SCAN+Filter
1/10 ~ 10^3
σ Regione='Piemonte' OR Regione='Lombardia'
SERVIZI-OFFERTI
2·10^5
FULL TABLE SCAN
IMPRESA-PULIZIE
10^4
FULL TABLE SCAN + Filter

B) Qi1)


π Eid, SUM(costo), SUM(NumeroOre) 5·10^2
SELECT STATEMENT
σ SUM(costo)>1000 1/10 ~ 5·10^2
è possibile anticipare la GROUP BY
GB Eid ~ 5·10^3
GB HASH
5·10^3
~1/20 · 10^5 = 5·10^3
~10^5
Sid
NESTED LOOP
INNER
σ COUNT(*)>1 1/2 ~5
NON si può anticipare
GB Sid 10
GB HASH
~2·10^3
Eid
HASH JOIN
9/10 ~ 4,5·10^4
σ TipoEdificio<>'ufficio'
1/10 ~ 10^8
σ Data>1/1/2022 AND Data<31/12/2022
1/10^3 ~5·10^4
σ città='Torino'
SERVIZI-PULIZIE
10^9
ACCESS BY ROWID INDEX RANGE SCAN ON Index2
ACCESS BY ROWID INDEX RANGE SCAN ON Index1 + Filter
EDIFICIO
5·10^7
Pid
NESTED LOOP
INNER
~2·10^4
Sid
NESTED LOOP
INNER
σ Regione='Piemonte' OR Regione='Lombardia' 1/10 ~ 10^3
1/10 ~ 10
σ categoria='Interni'
SERVIZIO
10^2
FULL TABLE SCAN + Filter
IMPRESA-PULIZIE
10^4
FULL TABLE SCAN + Filter
SERVIZI-OFFERTI
2·10^5
FULL TABLE SCAN



Come è possibile vedere dagli alberi delle operazioni di algebra relazionale, l'ordine delle operazioni nella query interna influisce sulla possibilità di anticipare la GROUP BY. Allo stesso modo anche l'ordine con cui viene effettuato il join tra la query esterna e quella interna influisce sull'operazione di anticipo del GROUP BY. Pertanto l'ultimo albero raffigurato risulta essere una soluzione ottimale in quanto permette l'anticipo di entrambi gli operatori di GROUP BY semplificando la complessità computazionale delle operazioni di JOIN.

TIPOLOGIE DI JOIN E GROUP BY

1. NESTED LOOP perche' IMPRESA PULIZIE (INNER TABLE) e' piccola (card = 10^3)
2. NESTED LOOP perche' SERVIZIO (INNER TABLE) e' piccola (card = 10)
3. GROUP BY HASH
4. HASH JOIN perche' le tabelle sono grandi
5. NESTED LOOP < INNER TABLE = ③ card = 5; OUTER TABLE = ④ card = $5 \cdot 10^3$
6. GROUP BY HASH

STRUTTURE FISICHE ACCESSORIE

E' possibile valutare la creazione dei seguenti indici in quanto l'attributo 'Città' ha una selettività inferiore a 1/10 e l'attributo 'Data' ha una selettività di 1/10 su una tabella molto grande (cardinalità di 10^9 tuple).

CREATE INDEX Index1 ON Edificio(Città)
CREATE INDEX Index2 ON SERVIZI-PULIZIA(Data)

Pertanto il metodo di accesso alle tabelle sarebbe il seguente:

IMPRESA-PULIZIE --> FULL TABLE SCAN +FILTER
SERVIZI-OFFERTI --> FULL TABLE SCAN
SERVIZIO --> FULL TABLE SCAN + FILTER
EDIFICIO --> ACCESS BY ROWID + INDEX RANGE SCAN ON Index1
SERVIZI-PULIZIA --> ACCESS BY ROWID + INDEX RANGE SCAN ON Index2