**Parte III**

**11. Descrizione dei Piani di Esecuzione**

Descriviamo i piani di esecuzione delle tre interrogazioni del carico di lavoro, prima e dopo il tuning fisico effettuato al fine di ottimizzarle.

Andiamo quindi ad analizzarle una alla volta;

1. Determinare l’identificatore dei giochi che coinvolgono al più quattro squadre e richiedono l’uso di due dadi.

EXPLAIN ANALYZE SELECT CodGioco

FROM Gioco

WHERE maxSquadre = 4 AND dadiRichiesti = 2;

Osserviamo il piano di esecuzione **senza l'indice:**

"**Seq Scan** on gioco (cost=0.00..232.47 rows=6758 width=4) (actual time=0.025..3.719 rows=6761 loops=1)"

" Filter: ((maxsquadre <= '4'::numeric) AND (dadirichiesti = '2'::numeric))"

" Rows Removed by Filter: 4"

"Planning time: 0.188 ms"

"Execution time: **4.111** **ms**"

Viene eseguita una semplice scansione sequenziale, filtrando rispetto alla condizione presente nel WHERE.

Creiamo quindi un indice multiattributo ordinato su maxSquadre e dadiRichiesti:

CREATE INDEX gioco\_multi

ON Gioco(maxSquadre, dadiRichiesti);

Osserviamo il piano di esecuzione con l'indice:

"**Index Scan using gioco\_multi on gioco** (cost=0.28..8.30 rows=1 width=4) (actual time=0.029..0.029 rows=1 loops=1)"

" Index Cond: ((maxsquadre = '4'::numeric) AND (dadirichiesti = '2'::numeric))"

"Planning time: 0.155 ms"

"Execution time: **0.041 ms**"

Viene eseguita dal sistema una scansione mediante l’indice creato, che risulta molto efficiente; il tempo di esecuzione passa infatti da 4.111 ms a 0.041 ms, riducendosi del **99**%.

1. Determinare l’identificatore delle sfide relative a un gioco A di vostra scelta = OCA che, in alternativa:

* hanno avuto luogo a gennaio 2021 e durata massima superiore a 2 ore, o
* hanno avuto luogo a marzo 2021 e durata massima pari a 30 minuti.

SELECT CodSfida

FROM Sfida

WHERE codGioco = 'OCA' AND

EXTRACT(YEAR FROM Sfida.dataOra) = 2021 AND

((EXTRACT(MONTH FROM Sfida.dataOra) = 1 AND Sfida.durataMax > 120)

OR (EXTRACT(MONTH FROM Sfida.dataOra) = 3 AND Sfida.durataMax = 30));

Osserviamo il piano di esecuzione senza l'indice:

"Seq Scan on sfida (cost=0.00..398.94 rows=1 width=5) (actual time=0.041..9.092 rows=3972 loops=1)"

" Filter: (((codgioco)::text = 'OCA'::text) AND (date\_part('year'::text, dataora) = '2021'::double precision) AND (((date\_part('month'::text, dataora) = '1'::double precision) AND (duratamax > 120)) OR ((date\_part('month'::text, dataora) = '3'::double precision) AND (duratamax = 30))))"

" Rows Removed by Filter: 6026"

"Planning time: 0.739 ms"

"Execution time: 9.305 ms"

Creiamo degli indici su codGioco, dataOra e durataMax:

CREATE INDEX sfida\_codGioco

ON Sfida(codGioco);

CLUSTER Sfida USING sfida\_codGioco;

CREATE INDEX sfida\_dataOra

ON Sfida(dataOra);

CREATE INDEX sfida\_durataMax

ON Sfida(durataMax);

Osserviamo il piano di esecuzione con gli indici:

"Seq Scan on sfida (cost=0.00..398.94 rows=1 width=5) (actual time=0.044..9.440 rows=3972 loops=1)"

" Filter: (((codgioco)::text = 'OCA'::text) AND (date\_part('year'::text, dataora) = '2021'::double precision) AND (((date\_part('month'::text, dataora) = '1'::double precision) AND (duratamax > 120)) OR ((date\_part('month'::text, dataora) = '3'::double precision) AND (duratamax = 30))))"

" Rows Removed by Filter: 6026"

"Planning time: 1.603 ms"

"Execution time: 8.015 ms"

Purtroppo, in questo caso il sistema non ha considerato altri piani oltre la scansione sequenziale e quindi l’aggiunta di indici non ha portato a nessun miglioramento a livello prestazionale.

Se fossimo ancora in fase di progettazione, potremmo riconsiderare questa interrogazione per provare qualche altro piano di esecuzione che la renda più efficiente; se nuovamente questi tentativi non portassero a nessun risultato potremmo quindi considerare l’eliminazione di questi indici, in quanto essi occupano memoria e rendono più costoso l’aggiornamento.

1. Determinare le sfide, di durata massima superiore a 2 ore, dei giochi che richiedono almeno due dadi. Restituire sia l’identificatore della sfida sia l’identificatore del gioco.

SELECT Sfida.CodSfida, Gioco.CodGioco

FROM Dado NATURAL JOIN Gioco JOIN Sfida ON Gioco.codGioco = Sfida.codGioco

WHERE Sfida.durataMax > 120

GROUP BY Sfida.CodSfida, Gioco.CodGioco

HAVING COUNT(Gioco.CodGioco) >= 2;

Creiamo due indici ordinati su Dado.codGioco e Gioco.codGioco per il **merge join**; per la selezione abbiamo già l’indice ad albero su durataMax creato nel punto precedente.

CREATE INDEX gioco\_codGioco

ON Gioco(codGioco);

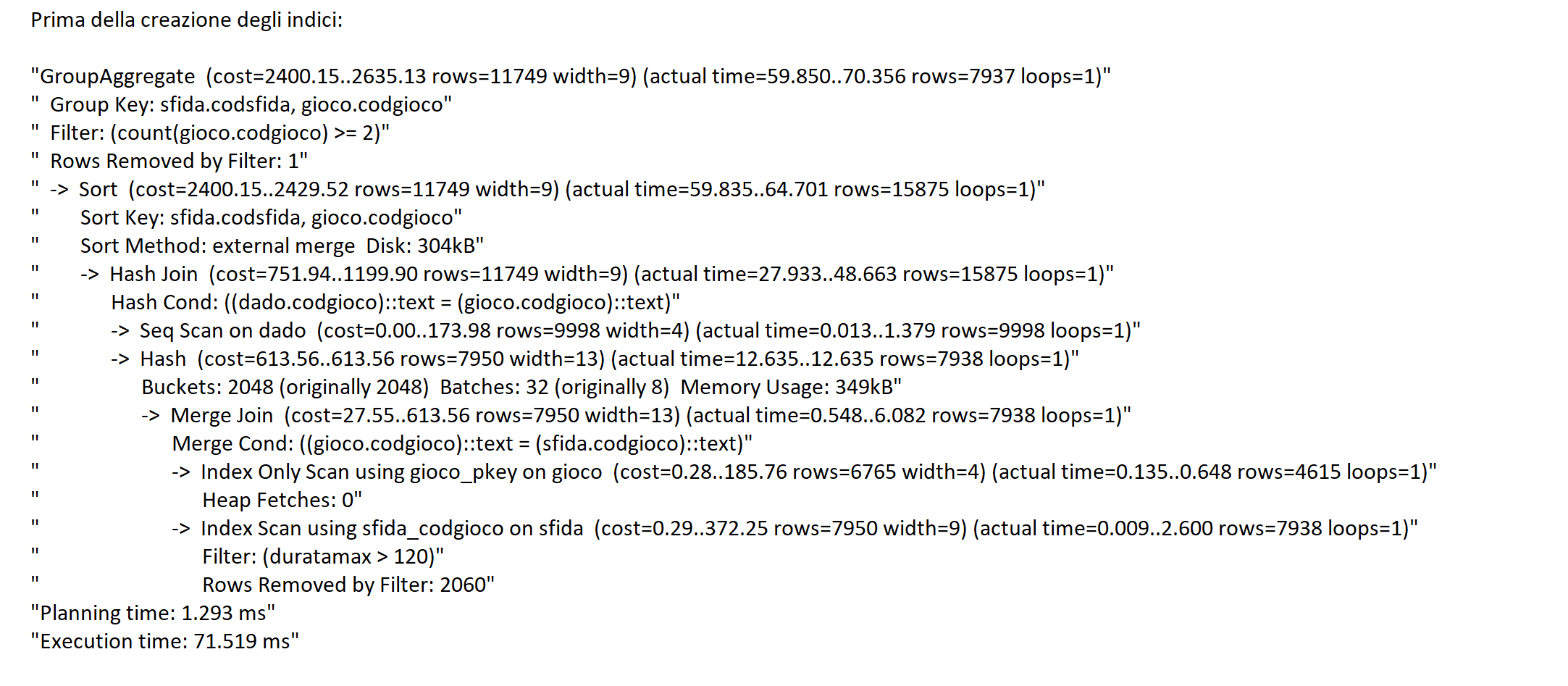
CLUSTER Gioco USING gioco\_codGioco;

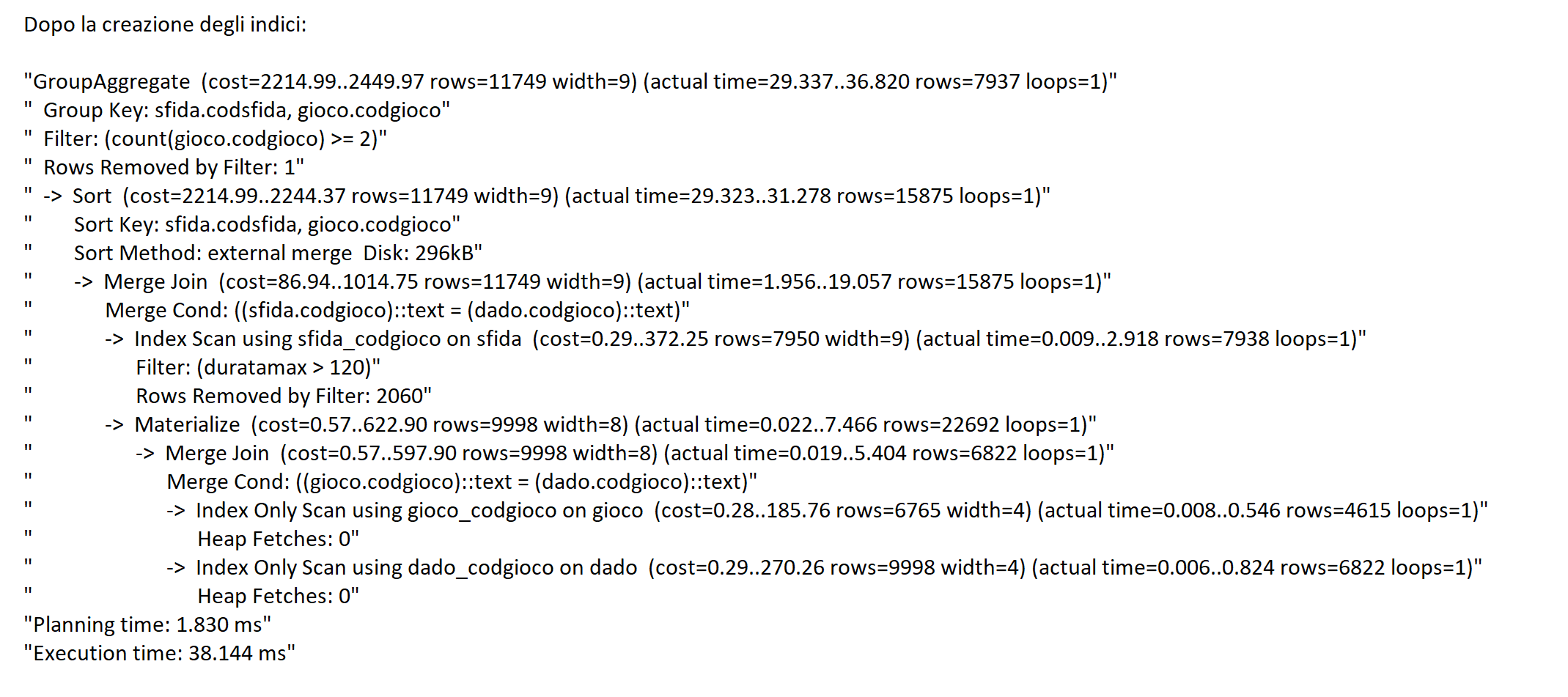
CREATE INDEX dado\_codGioco

ON Dado(codGioco);

CLUSTER Dado USING dado\_codGioco;

Confrontiamo i piani di esecuzione con e senza indice:





Notiamo come nel primo caso venga eseguito un **Hash Join** mentre nel secondo esso venga sostituito da un **Merge Join**.

Questo rende l’interrogazione più efficiente, infatti il tempo di esecuzione passa da **71.519 ms** a **38.144 ms**, con un miglioramento del **46.67**%.