

OTTIMIZZAZIONE FISICA

Creazione di indici e elaborazione del workload per rendere la base di dati più ottimizzata.

WORKLOAD:

Le tre interrogazioni più frequenti (e quindi facenti parte del workload) sono:

1.

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

2.

Determinare l'identificatore delle sfide relative a un gioco A di vostra scelta (specificare direttamente l'identificatore nella richiesta) che, in alternativa:

o 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

3.

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

PRIMA INTERROGAZIONE

La prima interrogazione può essere interpretata in SQL come:

```
SELECT id_g  
FROM GIOCO  
WHERE n_squadre <= 4 AND n_dadi > 2;
```

Per questa interrogazione è utile creare 2 indici: uno su `n_squadre` e uno su `n_dadi` così da rendere più rapida e meno costosa la ricerca.

Utilizzando gli indici è possibile ridurre di molto gli accessi a tuple inutili che sono molto costosi.

I due indici saranno di tipo ordinato perché entrambe le condizioni del WHERE richiedono una ricerca per intervallo e il tipo di indice hash come sappiamo non la implementa.

```
CREATE INDEX SquadreInGioco ON GIOCO(n_squadre);
```

```
CREATE INDEX DadiInGioco ON GIOCO(n_dadi);
```

Per decidere quale dei due indici clusterizzare la scelta è indifferente anche se per ora è meglio non clusterizzare nessuno dei due.

E' infatti importante ricordare che solo un attributo può essere clusterizzato in una tabella e magari avrà più senso clusterizzare un indice più utile in una interrogazione successiva del workload.

SECONDA INTERROGAZIONE

```
SELECT cod_Sfida
FROM SFIDA NATURAL JOIN GIOCO
WHERE SFIDA.id_g = 'ab0'
AND ((Sfida.data BETWEEN '01/01/2021' AND '31/01/2021'
AND SFIDA.durata_max > '02:00:00')
OR
(SFIDA.data BETWEEN '01/03/2021' AND '31/03/2021'
AND SFIDA.durata_max = '00:30:00'));
```

In questo caso gli attributi chiaramente utilizzati maggiormente sono data e durata_max di sfida.

Per questo sarà necessario creare due indici su questi attributi:

```
CREATE INDEX DataInSfida on SFIDA(data);
CREATE INDEX DurataInSfida on SFIDA(durata_max);
```

Anche in questo caso ha senso utilizzare indici ordinati visto la ricerca per intervallo.

In più per facilitare l'operazione di join potrebbe essere utile creare due indici sugli attributi id_g di GIOCO e SFIDA.

In questo modo il sistema è facilitato perché può utilizzare il metodo di index nested loop.

```
CREATE INDEX IDgioco on SFIDA(id_g);
CREATE INDEX IDSfida on SFIDA(id_g);
```

Una buona idea è quella di facilitare e velocizzare ulteriormente l'operazione di join clusterizzando questi due indici che sono chiaramente molto importanti.

```
CLUSTER GIOCO USING IDgioco;
CLUSTER SFIDA USING IDSfida;
```

TERZA INTERROGAZIONE

```
SELECT id_g , cod_Sfida
FROM GIOCO NATURAL JOIN SFIDA
WHERE GIOCO.n_dadi > 2 AND SFIDA.durata_max > '02:00:00';
```

Questa interrogazione prende parti della prima e della seconda.

Per questo tutti gli indici necessari sono già stati creati precedentemente nelle prime due interrogazioni.

Gli indici utili in questa interrogazione sono quelli ordinati su num_dadi, durata_max e poi quelli clusterizzati su id_g di gioco e sfida per il join.

Tutti e quattro sono già stati creati precedentemente.