

PIANI DI ESECUZIONE WORKLOAD

Il workload è definito tramite le tre seguenti interrogazioni considerate frequenti

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: 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.

Andremo ad analizzare come il DBSM esegue le query sopra descritte prima e dopo la creazione degli indici di ottimizzazione.

1. La prima query è formulata in questo modo:

```
SELECT IDgioco FROM Gioco WHERE Gioco.MaxSquadre <= 4 AND NumDadi=2;
```

Questa query risulta molto semplice, difatti, per come ho strutturato la base di dati, necessita l'accesso ad un'unica tabella. Ciò non vuol dire che non possano essere fatte ottimizzazioni; infatti, con l'aggiunta di due indici si ha un miglioramento nell'efficienza, soprattutto per quanto riguarda la condizione con range :

```
CREATE INDEX Giochi ON Gioco (MaxSquadre);
```

```
CREATE INDEX NumeroDadi ON Gioco (NumDadi);
```

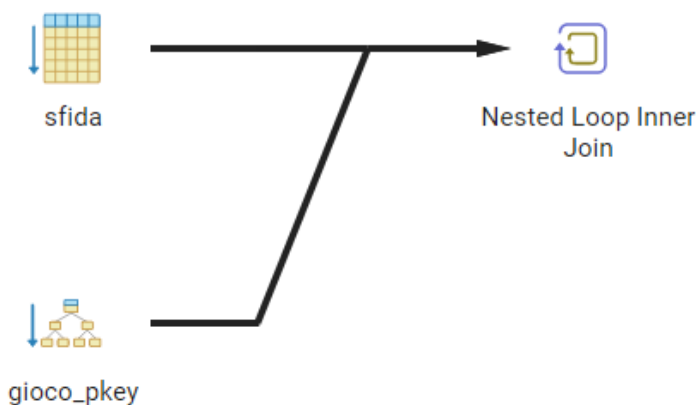
2. La seconda query è la seguente:

```
SELECT IDSfida
```

```
FROM Sfida JOIN Gioco ON Sfida.IDgioco = Gioco.IDgioco
```

```
WHERE Sfida.IDGioco = 'g1' AND Sfida.Data BETWEEN '01/01/2021' AND '01/31/2021' AND Durata > '02:00'  
OR Sfida.Data BETWEEN '03/01/2021' AND '03/31/2021' AND Durata = '00:30';
```

Per questa Query il piano di esecuzione risulta come raffigurato di seguito:



Con l'aggiunta dei due seguenti indici:

```
CREATE INDEX DataSfida ON Sfida (Data);
```

```
CREATE INDEX DurataSfida ON Sfida (Durata);
```

Il piano di esecuzione effettivamente non cambia siccome si ha un Nested Loop Join, ovvero il tipo di join fisico che consuma meno risorse RAM, esegue il minor numero di confronti ed è il più veloce. Tuttavia questo tipo di join fisico perde efficienza nel caso la quantità di dati sia 'grande'; quindi con l'aggiunta degli indici permetto al BDSM di considerare un Hash Inner Join che performa meglio all'aumentare dei dati.

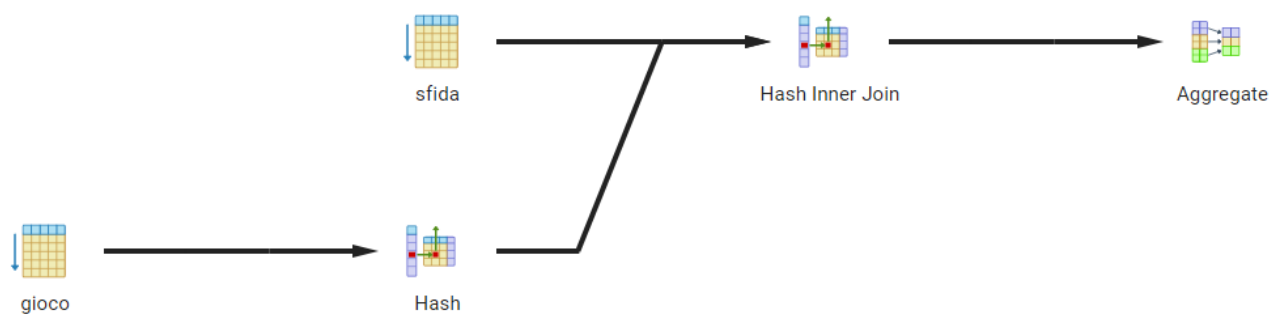
3. La terza query è formulata in questo modo:

```
SELECT DISTINCT IDSfida, Sfida.IDGioco
```

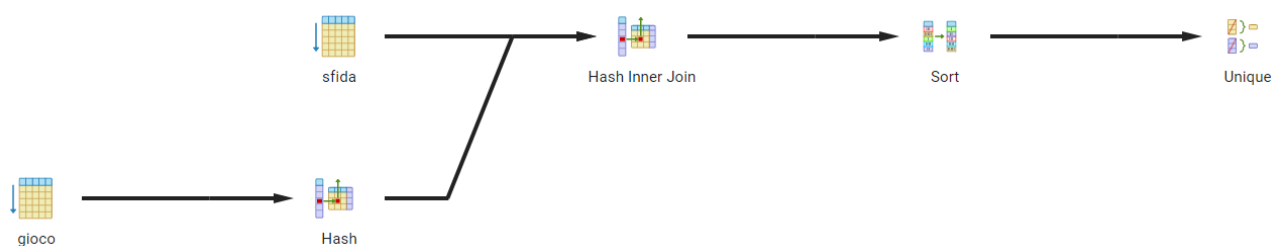
```
FROM Sfida JOIN Gioco ON Sfida.IDGioco = Gioco.IDGioco
```

```
WHERE Gioco.IDGioco=Sfida.IDGioco AND Gioco.NumDadi >=2 AND Durata > '02:00';
```

Senza l'aggiunta degli indici presentati sopra, che sono sufficienti all'ottimizzazione della query, si ha il piano di esecuzione:



Invece andando a creare i quattro indici per le query 1 e 2 si ottiene il seguente piano di esecuzione:



In questo caso possiamo vedere come il distinct viene ottimizzato, infatti ordinando la tabella prima di operare l'aggregazione voluta da distinct si ottiene una migliore gestione della memoria RAM infatti non occorre tenere in memoria tutta la tabella per poter aggregare i dati.

CONTROLLO DEGLI ACCESSI

	Gameadmin	Gamecreator	Giocatore	Utente
Gioco	ALL	ALL	S	S
Dado	ALL	ALL	S	
Casella	ALL	ALL	S	
Turno	ALL	S	S	
Squadra	ALL	S	S	S
Icona	ALL	ALL	S	
Usa	ALL	S	S	
Quiz	ALL	ALL	S	
Lancio	ALL	S	SI	
Task	ALL	ALL	S	
Svolge	ALL	S	S	
Utente	ALL	S	SI	
Risposta	ALL	ALL	S	
RispostaData	ALL	S	SI	
Sfida	ALL	S	S	S
Scelta	ALL	S	SI	
Consegna	ALL	S	SI	
Fornita	ALL	S	S	
Partecipa	ALL	S	S	

La gerarchia è pensata in modo da dare i privilegi di modifica e inserimento al Gameadmin che impersonifica il responsabile del funzionamento del gioco stesso, pertanto non ha limiti sui privilegi.

Il Gamecreator è il responsabile della creazione di nuovi giochi o della modifica strutturale di quelli già presenti, quindi ha completi diritti sulle entità che definiscono e caratterizzano un gioco. Inoltre il gamecreator deve poter vedere le informazioni relative ai giochi per osservare le dinamiche di gioco e trarne spunto in modo da creare giochi sempre più interessanti e di controllare quelli che sono più utilizzati.

Il giocatore deve poter avere una visione complessiva rispetto al gioco o ai giochi a cui partecipa o che suscitano il suo interesse, quindi non ha limiti di select. Invece ha diritto di insert solo nelle tabelle che corrispondono ad una sua interazione con il gioco e che quindi gli permettono di partecipare alle sfide.

Infine abbiamo l'utente, che in questa gerarchia è pensato come uno spettatore che vuole capire se i giochi offerti possano interessargli o meno; quindi un utente ha diritti di select sulle tabelle che contengono le informazioni generali di un gioco, sulle squadre e sulle sfide.