

#### 4. Progetto Fisico

Svolgiamo l'attività di tuning fisico, che si preoccupa di progettare uno schema fisico, in termini di insieme di indici creati; lo schema deve essere realizzato al fine di rendere possibile un'esecuzione efficiente delle interrogazioni contenute nel **workload**.

Consideriamo quindi le interrogazioni relative al workload, andandole ad analizzare una alla volta; siccome esse hanno la stessa frequenza le consideriamo in ordine numerico:

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

Gli attributi che appaiono in una clausola WHERE sono candidati come chiavi di ricerca per un indice; in questo caso gli attributi che ci interessano sono maxSquadre e dadiRichiesti.

Proviamo quindi a ipotizzare un potenziale piano di esecuzione ottimale per l'interrogazione:

- Cerchiamo di eseguire la selezione sulla condizione del WHERE nel modo più efficiente possibile; *maxSquadre = 4 AND dadiRichiesti = 2* è un **fattore booleano**, ovvero una condizione che, se falsa, rende falsa tutta l'interrogazione; sfruttiamo quindi questa condizione per creare un piano di esecuzione ottimale.

Individuiamo gli indici che permettono al sistema di prendere in considerazione il piano individuato nello spazio dei piani; per la selezione possiamo utilizzare vari indici:

- Indici su singolo attributo, ad albero o hash, su maxSquadre e/o su dadiRichiesti; entrambi i tipi di indici sono possibili, poiché entrambi gli attributi sono coinvolti in ricerche per uguaglianza;
- Multi-attributo ad albero o hash su maxSquadre e dadiRichiesti,
- Multi-attributo ad albero o hash su dadiRichiesti e maxSquadre.

Valutiamo, dopo aver considerato le altre interrogazione, quale opzione scegliere ed anche se **clusterizzare** o meno questo/i indice/i.

- 2) Determinare l'identificatore delle sfide relative a un gioco A di vostra scelta (ovvero '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.

Gli attributi che appaiono in una clausola WHERE sono candidati come chiavi di ricerca per un indice; in questo caso sono codGioco, dataOra e durataMax.

Proviamo quindi a ipotizzare un potenziale piano di esecuzione ottimale per l'interrogazione e individuiamo gli indici necessari:

- Per la selezione consideriamo tre indici su codGioco, dataOra e durataMax: quelli su codGioco e dataOra possono essere indici hash, quello su durataMax è per forza un indice ordinato. Purtroppo, in questo caso non vi è un fattore booleano a causa dell'OR all'interno della condizione; si può quindi cercare di creare vari indici sugli attributi interessati per poi unire i loro risultati.

- 3) Determinare le sfide, di durata massima superiore a due ore, dei giochi che richiedono almeno due dadi. Restituire sia l'identificatore della sfida sia l'identificatore del gioco.

Gli attributi che appaiono in una clausola WHERE sono candidati come chiavi di ricerca per un indice; in questo caso l'unico attributo in questione è durataMax.

Inoltre, dobbiamo considerare un piano efficiente per effettuare il join.

Proviamo quindi a ipotizzare un potenziale piano di esecuzione ottimale per l'interrogazione:

- Per eseguire il join nel modo più efficiente possibile consideriamo l'algoritmo di merge join;
- Per la selezione consideriamo un indice ordinato su durataMax, che viene creato ed utilizzato nel secondo punto.

Individuiamo gli indici che permettono al sistema di prendere in considerazione il piano individuato nello spazio dei piani:

- Per eseguire il join mediante merge join servono vari indici ordinati clusterizzati: quello su Sfida.codGioco esiste già (dovremmo clusterizzarlo) mentre ne serve uno su Gioco.codGioco ed un terzo su Dado.codGioco.
- Per la selezione ci serve l'indice ad albero su durataMax, già creato nel punto precedente.

#### **Indici Scelti:**

Andiamo quindi a riassumere quali potrebbero essere degli indici efficienti per i vari piani di esecuzione delle tre interrogazioni nel workload:

- Per la prima interrogazione creiamo un indice multi-attributo ordinato su maxSquadre e dadiRichiesti.
- Per la seconda interrogazione creiamo indici su codGioco, dataOra e durataMax; quello su codGioco lo clusterizziamo poiché ci servirà anche nella terza interrogazione.
- Per la terza interrogazione creiamo due indici ordinati clusterizzati rispettivamente su Dado.codGioco e Gioco.codGioco (per il merge join); il terzo indice clusterizzato, su Sfida.codGioco esiste già.

Per la selezione abbiamo già l'indice ad albero su durataMax creato nel punto precedente.