

## ELABORAZIONE DELLE INTERROGAZIONI

### QUERY 1

```
SELECT Gioco.IdG FROM Gioco WHERE Gioco.MaxS <= 4 AND Gioco.NumDadi = 2;
```

Piano di esecuzione e tempi prima della creazione dello schema fisico



ProgettoBD.gioco

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Seq Scan on ProgettoBD.gioco as gioco (cost=0..975.11 rows=283... Filter: ((gioco.maxs <= 4) AND (gioco.numdadi = 2)) Rows Removed by Filter: 24721	3.462 ms	3.462 ms	1 1.02	286	283	1

Piano di esecuzione e tempi dopo la creazione dello schema fisico



ProgettoBD.dadisq-  
uadregioco

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Index Scan using dadisquadregioco on ProgettoBD.gioco as gioco ... Index Cond: ((gioco.numdadi = 2) AND (gioco.maxs <= 4))	0.071 ms	0.071 ms	1 1.02	286	283	1

Si può notare come utilizzando l'indice DadiSquadreGioco il tempo di esecuzione della query è notevolmente inferiore, infatti utilizzando un indice clusterizzato le tuple della tabella saranno memorizzate in modo ordinato e si riuscirà ad accedere solo alle tuple con **NumDadi = 2** e con **MaxS <= 4** (invece che dover accedere a tutte le tuple con una scansione sequenziale)

### QUERY 2

```
SELECT IdSf FROM Sfida WHERE IdG=4 AND (
    (Durata > '02:00:00' AND Inizio BETWEEN '01-01-2021' AND '01-31-2021') OR
    (Durata = '00:30:00' AND Inizio BETWEEN '03-01-2021' AND '03-31-2021')
);
```

Piano di esecuzione e tempi prima della creazione dello schema fisico



ProgettoBD.sfida

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Seq Scan on ProgettoBD.sfida as sfida (cost=0..1191.7 rows=42 wi... Filter: ((sfida.inizio >= '2021-01-01 00:00:00+01':timestamp with time zon e) AND (sfida.inizio <= '2021-01-31 00:00:00+01':timestamp with time zon e) AND (sfida.idg = 10) AND ((sfida.durata > '02:00:00':time without time z one) OR (sfida.durata = '00:30:00':time without time zone))) Rows Removed by Filter: 24962	3.82 ms	3.82 ms	↓ 1.12	47	42	1

Piano di esecuzione e tempi dopo la creazione dello schema fisico



ProgettoBD.idgini-  
ziosfida

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Index Scan using idginiziosfida on ProgettoBD.sfida as sfida (cost... Filter: ((sfida.durata > '02:00:00':time without time zone) OR (sfida.durata = '00:30:00':time without time zone)) Index Cond: ((sfida.idg = 10) AND (sfida.inizio >= '2021-01-01 00:00:00+0 1':timestamp with time zone) AND (sfida.inizio <= '2021-01-31 00:00:00+ 01':timestamp with time zone)) Rows Removed by Filter: 9	0.046 ms	0.046 ms	↓ 1.12	47	42	1

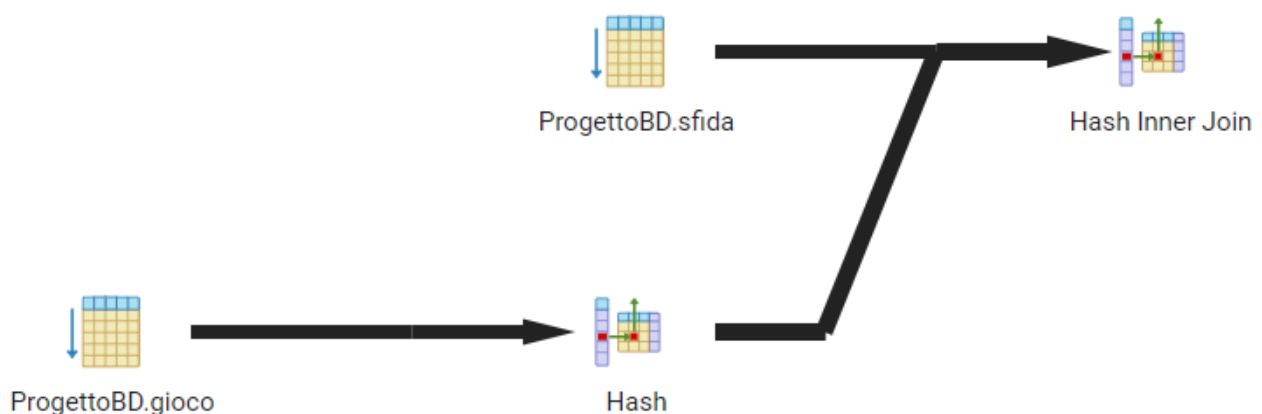
Come nel caso precedente, utilizzando un indice ordinato clusterizzato, in questo caso IdGInizioSfida, il tempo di esecuzione sarà notevolmente inferiore.

Non viene creato un indice su Durata siccome è stato notato che non veniva mai usato, infatti se dovessero esserci tante duple che rispettano la condizione `Durata > '02:00:00'`, diventa più conveniente effettuare una scansione sequenziale.

### QUERY 3

```
SELECT Sfida.IdSf, Gioco.IdG FROM Sfida NATURAL JOIN Gioco
WHERE Durata > '02:00:00' AND NumDadi >= 2;
```

Piano di esecuzione e tempi prima della creazione dello schema fisico



#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Hash Inner Join (cost=1209.39..2421.61 rows=19399 width=8) (...) Hash Cond: (sfida.idg = gioco.idg)	4.786 ms	16.534 ms	↓ 1.01	19539	19399	1
2.	→ Seq Scan on ProgettoBD.sfida as sfida (cost=0..941.61 row...) Filter: (sfida.durata > '02:00:00':time without time zone) Rows Removed by Filter: 4576	3.917 ms	3.917 ms	↓ 1.01	20433	20431	1
3.	→ Hash (cost=912.59..912.59 rows=23744 width=4) (actual=7...) Buckets: 32768 Batches: 1 Memory Usage: 1091 kB	3.128 ms	7.831 ms	↑ 1	23744	23744	1
4.	→ Seq Scan on ProgettoBD.gioco as gioco (cost=0..912.5...) Filter: (gioco.numdadi >= 2) Rows Removed by Filter: 1263	4.703 ms	4.703 ms	↑ 1	23744	23744	1

Piano di esecuzione e tempi dopo la creazione dello schema fisico



#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Seq Scan on ProgettoBD.sfidagioco2oredadi as sfidagioco2oreda...	0.786 ms	0.786 ms	↑ 1	19539	19539	1

Per ottimizzare la terza query si è scelto di penalizzare l'inserimento di una nuova sfida, infatti ogni volta che viene inserito un nuovo record sarà necessario effettuare il refresh della vista.

In questo modo però è stato possibile velocizzare notevolmente la query, infatti utilizzando gli indici, come avveniva nella query precedente, se c'erano tante tuple che rispettavano la condizione `NumDadi >= 2` o la condizione `Durata >= '02:00:00'`, diventava più conveniente effettuare una scansione sequenziale comportando quindi tempi di esecuzione molto più elevati.

# CONTROLLO DELL'ACCESSO

## DEFINIZIONE DEI RUOLI

- **Utente:** si tratta del ruolo più basso, con meno permessi e che avranno tutti gli utenti
- **Giocatore:** si tratta sempre di un ruolo con pochi permessi in quanto lo avranno tutti gli utenti che in quel momento partecipano ad una sfida
- **Gameadmin:** apparterrà agli utenti che in quel momento stanno amministrando una sfida consentendogli di attivare le sfide e di approvare i task
- **Gamecreator:** apparterrà ad un numero ristretto di utenti, i quali possono creare i nuovi giochi

Considerando i permessi che dovranno avere i vari ruoli è possibile strutturare la seguente gerarchia:

Gamecreator  $\geq$  Gameadmin  $\geq$  Giocatore  $\geq$  Utente

Viene creata la gerarchia Gamecreator  $\geq$  Gameadmin siccome è possibile immaginare che un utente che crea un gioco deve avere anche la possibilità di controllare eventuali sfide legate a quel gioco.

## TABELLA DEI RUOLI

Tabella	Utente	Giocatore	Gameadmin	Gamecreator
Appartiene (Utente, Squadra)	SI	SI	ALL	ALL
ApprovaRispostaQuiz		SI	ALL	ALL
CambiaDadiRispostaQuiz				ALL
CambiaDadiRispostaTask		S	S	ALL
Casella		S	S	ALL
Dado		S	S	ALL
Gioco	S	S	S	ALL
Icone		S	S	ALL
Plancia		S	S	ALL
Podio		S	S	ALL
Possiede (Dadi, Turno, Squadra)		S	ALL	ALL
Quiz		S	S	ALL
Richiede (Dadi, Gioco)		S	S	ALL
RispostaQuiz		S	S	ALL
RispostaTask		SI	ALL	ALL
SceltaRispostaQuiz		SI	ALL	ALL
SetIcone		S	S	ALL
Sfida	S	S	ALL	ALL
SiTrova (Icona, Turno, Casella)		S	ALL	ALL
SiTrovaPodio (Icona, Turno, Podio)		S	ALL	ALL
Squadra	S	SI	ALL	ALL
Task		S	S	ALL
Tiro		SI	ALL	ALL

Turno		S	ALL	ALL
Utente	SUI	SUI	SUI	SUI

Siccome si tratta di un'operazione molto delicata, nessuno potrà effettuare la delete sulla tabella utente (tranne il DBA).

S = SELECT

U = UPDATE

I = INSERT

D = DELETE

ALL = TUTTE LE OPERAZIONI