

Progetto Basi di Dati 2023-24

“UNIGE SOCIAL SPORT”

Parte III

[Numero gruppo: 43]

[Eugenio Vassallo 5577783, Mario Madalin Biberia 5608210, Umeshan Panchabalasingham 5606614]

1. PROGETTAZIONE FISICA

1A+1C CARICO DI LAVORO

Q1 - QUERY CON SINGOLA SELEZIONE E NESSUN JOIN

LINGUAGGIO NATURALE

[Riportare in questa sezione l'interrogazione del carico di lavoro con singola selezione e nessun join, in linguaggio naturale]

Selezioniamo l'utente con nome “Nome1000”

SQL

[Riportare in questa sezione l'interrogazione del carico di lavoro con singola selezione e nessun join, in SQL]

```
EXPLAIN ANALYZE
SELECT *
FROM Utenti_CL
WHERE nome = 'Nome1000';
```

Q2 - QUERY CON CONDIZIONE DI SELEZIONE COMPLESSA E NESSUN JOIN

LINGUAGGIO NATURALE

[Riportare in questa sezione l'interrogazione del carico di lavoro con condizione di selezione complessa e nessun join, in linguaggio naturale]

Il risultato della query è una tabella di utenti di genere femminile con un id > di 50000 e nate nel 1990

SQL

[Riportare in questa sezione l'interrogazione del carico di lavoro con condizione di selezione complessa e nessun join, in SQL]

```
EXPLAIN ANALYZE
SELECT *
FROM Utenti_CL
WHERE anno_nascita = 1990 and utente_id > 50000
AND genere = 'F';
```

Q3 - QUERY CON ALMENO UN JOIN E ALMENO UNA CONDIZIONE DI SELEZIONE

LINGUAGGIO NATURALE

[Riportare in questa sezione l'interrogazione del carico di lavoro con almeno un join e almeno una condizione di selezione, in linguaggio naturale]

La query ci restituisce informazioni su eventi e categoria, solo degli eventi con stato "aperto"

SQL

[Riportare in questa sezione l'interrogazione del carico di lavoro con almeno un join e almeno una condizione di selezione, in SQL]

```
EXPLAIN ANALYZE
SELECT e.evento_id, e.data_evento, c.nome AS categoria_nome
FROM Eventi_CL e
JOIN Categorie_CL c ON e.categoria_id = c.categoria_id
WHERE e.stato = 'APERTO';
```

1D-PROGETTO FISICO

[Riportare nella seguente tabella l'elenco degli indici che si intendono creare per: (1) ciascuna query del carico di lavoro individualmente; (2) l'insieme delle query del carico di lavoro, motivando opportunamente, in modo sintetico, le scelte effettuate]

Id query	Relazione	Chiave di ricerca	Tipo	Motivazione
----------	-----------	-------------------	------	-------------

			(ordinato/hash, clusterizzato/non clusterizzato)	
Q1	Utenti_cl	Nome	Ordinati	È nel SELECT
Q2	Utenti_cl	Anno nascita	Ordinati	È nel WHERE
Q3	Eventi_cl	Stato	Ordinati	È nel WHERE
Q3	Eventi_cl	Categoria_ID	Ordinati	È nel JOIN
Q3	Catego- ria_cl	Categoria_ID	Ordinati	È nel JOIN

Schema fisico complessivo per il carico di lavoro	Motivazione
Utente_cl, Categoria_cl, Eventi_cl	Erano le tabelle su cui provare un carico di lavoro realistico

1G-ANALISI PIANI DI ESECUZIONE SCELTI DAL SISTEMA

Q1 - QUERY CON SINGOLA SELEZIONE E NESSUN JOIN

PIANO DI ESECUZIONE SCELTO DAL SISTEMA PRIMA DELLA CREAZIONE DELLO SCHEMA FISICO

[Riportare in questa sezione il piano di esecuzione scelto da PostgreSQL prima della creazione dello schema fisico, in formato testuale]

"Gather (cost=1000.00..24629.43 rows=1 width=134) (actual time=4.400..168.933 rows=1 loops=1)"

" Workers Planned: 2"

" Workers Launched: 2"

" -> Parallel Seq Scan on utenti_cl (cost=0.00..23629.33 rows=1 width=134) (actual time=29.136..79.047 rows=0 loops=3)"

" Filter: ((nome)::text = 'Nome10000'::text)"

" Rows Removed by Filter: 333333"

"Planning Time: 1.741 ms"

"Execution Time: 168.978 ms"

PIANO DI ESECUZIONE SCELTO DAL SISTEMA DOPO DELLA CREAZIONE DELLO SCHEMA FISICO

[Riportare in questa sezione il piano di esecuzione scelto da PostgreSQL dopo la creazione dello schema fisico, in formato testuale]

"Index Scan using idx_nome on utenti_cl (cost=0.42..8.44 rows=1 width=134) (actual time=0.514..0.517 rows=1 loops=1)"

" Index Cond: ((nome)::text = 'Nome10000'::text)"

"Planning Time: 1.992 ms"

"Execution Time: 0.557 ms"

CONFRONTO TRA I DUE PIANI

[Riportare nella seguente tabella i tempi di esecuzione per i piani ottenuti prima e dopo la creazione dello schema fisico, giustificando i piani e i tempi ottenuti]

Tempo esecuzione, PRIMA	Tempo esecuzione DOPO	Motivazione
168.978 ms	0.557 ms	<i>l'indice ha migliorato notevolmente la performance, eliminando la necessità di scansionare l'intera tabella utenti</i>

Q2 - QUERY CON CONDIZIONE DI SELEZIONE COMPLESSA E NESSUN JOIN

PIANO DI ESECUZIONE SCELTO DAL SISTEMA PRIMA DELLA CREAZIONE DELLO SCHEMA FISICO

[Riportare in questa sezione il piano di esecuzione scelto da PostgreSQL prima della creazione dello schema fisico, in formato testuale]

"Gather (cost=1000.00..28313.47 rows=16008 width=134) (actual time=140.723..152.332 rows=0 loops=1)"

" Workers Planned: 2"

" Workers Launched: 2"

" -> Parallel Seq Scan on utenti_cl (cost=0.00..25712.67 rows=6670 width=134) (actual time=84.995..84.996 rows=0 loops=3)"

" Filter: ((utente_id > 50000) AND (anno_nascita = 1990) AND ((genere)::text = 'F'::text))"

" Rows Removed by Filter: 333333"

"Planning Time: 0.245 ms"

"Execution Time: 152.366 ms"

PIANO DI ESECUZIONE SCELTO DAL SISTEMA DOPO DELLA CREAZIONE DELLO SCHEMA FISICO

[Riportare in questa sezione il piano di esecuzione scelto da PostgreSQL dopo la creazione dello schema fisico, in formato testuale]

"Bitmap Heap Scan on utenti_cl (cost=362.60..19834.51 rows=15720 width=134) (actual time=4.888..4.889 rows=0 loops=1)"

" Recheck Cond: (anno_nascita = 1990)"

" Filter: ((utente_id > 50000) AND ((genere)::text = 'F'::text))"

" Rows Removed by Filter: 33333"

" Heap Blocks: exact=603"

" -> Bitmap Index Scan on idx_anno_nascita (cost=0.00..358.67 rows=32833 width=0) (actual time=0.747..0.748 rows=33333 loops=1)"

" Index Cond: (anno_nascita = 1990)"

"Planning Time: 1.276 ms"

"Execution Time: 4.918 ms"

CONFRONTO TRA I DUE PIANI

[Riportare nella seguente tabella i tempi di esecuzione per i piani ottenuti prima e dopo la creazione dello schema fisico, giustificando i piani e i tempi ottenuti]

Tempo esecuzione PRIMA	Tempo esecuzione DOPO	Motivazione
152.366 ms	4.918 ms	L'indice ci aiuta a guardare meno elementi, ma comunque dobbiamo guardare alcuni elementi nello heap(elementi non in ordine)

Q3 - QUERY CON ALMENO UN JOIN E ALMENO UNA CONDIZIONE DI SELEZIONE

PIANO DI ESECUZIONE SCELTO DAL SISTEMA PRIMA DELLA CREAZIONE DELLO SCHEMA FISICO

[Riportare in questa sezione il piano di esecuzione scelto da PostgreSQL prima della creazione dello schema fisico, in formato testuale]

"Hash Join (cost=1641.00..14526.56 rows=250950 width=22) (actual time=22.897..314.219 rows=250000 loops=1)"

```

" Hash Cond: (e.categoria_id = c.categoria_id)"

" -> Seq Scan on eventi_cl e (cost=0.00..9435.00 rows=250950 width=12) (actual time=0.141..172.831
rows=250000 loops=1)"

"   Filter: ((stato)::text = 'APERTO'::text)"

"   Rows Removed by Filter: 250000"

" -> Hash (cost=1016.00..1016.00 rows=50000 width=18) (actual time=22.384..22.386 rows=50000 loops=1)"

"   Buckets: 65536 Batches: 1 Memory Usage: 3051kB"

"   -> Seq Scan on categorie_cl c (cost=0.00..1016.00 rows=50000 width=18) (actual time=0.052..12.437
rows=50000 loops=1)"

"Planning Time: 4.154 ms"

"Execution Time: 331.522 ms"

```

PIANO DI ESECUZIONE SCELTO DAL SISTEMA DOPO DELLA CREAZIONE DELLO SCHEMA FISICO

[Riportare in questa sezione il piano di esecuzione scelto da PostgreSQL dopo la creazione dello schema fisico, in formato testuale]

```

" Hash Cond: (e.categoria_id = c.categoria_id)"

" -> Bitmap Heap Scan on eventi_cl e (cost=2801.29..9123.16 rows=250950 width=12) (actual
time=7.801..40.042 rows=250000 loops=1)"

"   Recheck Cond: ((stato)::text = 'APERTO'::text)"

"   Heap Blocks: exact=3185"

"   -> Bitmap Index Scan on idx_eventi_stato (cost=0.00..2738.55 rows=250950 width=0) (actual
time=7.440..7.440 rows=250000 loops=1)"

"     Index Cond: ((stato)::text = 'APERTO'::text)"

" -> Hash (cost=1016.00..1016.00 rows=50000 width=18) (actual time=13.209..13.211 rows=50000 loops=1)"

"   Buckets: 65536 Batches: 1 Memory Usage: 3051kB"

"   -> Seq Scan on categorie_cl c (cost=0.00..1016.00 rows=50000 width=18) (actual time=0.014..5.693
rows=50000 loops=1)"

"Planning Time: 5.293 ms"

"Execution Time: 178.084 ms"

```

CONFRONTO TRA I DUE PIANI

[Riportare nella seguente tabella i tempi di esecuzione per i piani ottenuti prima e dopo la creazione dello schema fisico, giustificando i piani e i tempi ottenuti]

<i>Tempo esecuzione PRIMA</i>	<i>Tempo esecuzione DOPO</i>	<i>Motivazione</i>
<i>331.522 ms</i>	<i>178.084 ms</i>	<i>Abbiamo aggiunto tre indici per aiutare con la JOIN e in WHERE, comunque il guadagno di prestazione non è come nella Q1 e Q2.</i>

2. CONTROLLO DELL'ACCESSO

GERARCHIA TRA I RUOLI

GERARCHIA

[Riportare in questa sezione la gerarchia che si intende definire tra i quattro ruoli]

Amministratore > Utente Premium > Utente Semplice

Amministratore > Gestore Impianti

MOTIVAZIONE GERARCHIA

[Riportare in questa sezione una motivazione per la gerarchia proposta]

L'amministratore ha potere su tutto il database, ha più privilegi di tutti

Utente Premium può fare più cose di un utente semplice (creazione eventi, squadre, tornei ecc)

Il Gestore impianti e l'utente premium non sono "paragonabili" perché hanno privilegi su tabelle diverse

ASSEGNAZIONE PRIVILEGI SPECIFICI AI RUOLI

[Riportare nella prima colonna della seguente tabella le relazioni considerate e in ciascuna altra cella (i,j) i privilegi specifici (quindi non acquisiti tramite la gerarchia) che si intendono assegnare al ruolo j sulla tabella i].

RELAZIONE	Amministratore di UniGE social sport	Utente premium	Gestore impianto	Utente semplice
Impianti	Tutti i permessi amministratore		Tutti i permessi	
Eventi	Tutti i permessi amministratore	Tutti i permessi		View only
Tornei	Tutti i permessi Amministratore	Tutti i permessi		View only
Squadre	Tutti i permessi amministratore	Tutti i permessi		View only
Valutazioni	Tutti i permessi amministratore	Tutti i permessi		View and insert only