

Progetto Basi di Dati 2024-25
"FANTASANREMO"
Parte III

69

5231931 - Romano Yuri

5216444 - Parodi Aurora

1. PROGETTAZIONE FISICA

1A+1C - CARICO DI LAVORO

Q1 - QUERY CON SINGOLA SELEZIONE E NESSUN JOIN

LINGUAGGIO NATURALE

Trovare tutti gli artisti che appartengono a una specifica casa discografica (casa_disco_id = 2), mostrando il nome artistico, nome reale, cognome e paese di origine.

SQL

EXPLAIN ANALYZE

SELECT nome_artistico, nome, cognome, paese

FROM artisti_CL

WHERE casa_disco_id = 2;

Q2 - QUERY CON CONDIZIONE DI SELEZIONE COMPLESSA E NESSUN JOIN

LINGUAGGIO NATURALE

Trovare tutte le canzoni con durata compresa tra 180 e 210 secondi, che abbiano "Marco" nel nome dell'autore del testo oppure "Emma" nel nome dell'autore della musica, e che siano state pubblicate dal 2024 in poi.

SQL

EXPLAIN ANALYZE

SELECT titolo, durata, testo_autore, musica_autore

FROM canzoni_CL

WHERE durata BETWEEN 180 AND 210

AND (testo_autore LIKE '%Marco%' OR musica_autore LIKE '%Emma%')

AND anno >= 2024;

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

LINGUAGGIO NATURALE

Trovare gli artisti con i loro voti ricevuti nelle serate di tipo "cover", mostrando nome artistico, casa discografica di appartenenza, percentuale di voti e numero di voti, limitandosi agli artisti che hanno ottenuto più del 25% di voti e che appartengono a case discografiche con sede a Milano o Roma.

SQL

EXPLAIN ANALYZE

SELECT a.nome_artistico, cd.nome as casa_disco, v.percentuale, v.numero_voti

FROM artisti_CL a

JOIN partecipazioni_CL p ON a.id = p.artista_id

JOIN voti_CL v ON p.id = v.partecipazione_id

JOIN case_disco_CL cd ON a.casa_disco_id = cd.id

JOIN serate_CL s ON v.serata_id = s.id

WHERE s.tipo = 'cover'

AND v.percentuale > 25.0

AND cd.citta IN ('Milano', 'Roma');

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
Q1	artisti_CL	casa_disco_id	ordinato, non clusterizzato	Accelera la selezione degli artisti per casa discografica evitando sequential scan
Q2	canzoni_CL	durata	ordinato, non clusterizzato	Ottimizza la condizione BETWEEN sulla durata
Q2	canzoni_CL	testo_autore, musica_autore	ordinato, non clusterizzato	Migliora le ricerche LIKE sui nomi degli autori
Q2	canzoni_CL	anno	ordinato, non clusterizzato	Velocizza il filtro per anno di pubblicazione
Q3	voti_CL	partecipazione_id	ordinato, non clusterizzato	Velocizza il join con la tabella partecipazioni
Q3	voti_CL	serata_id	ordinato, non clusterizzato	Ottimizza il join con la tabella serate
Q3	voti_CL	percentuale	ordinato, non clusterizzato	Accelera il filtro sulla percentuale di voti
Q3	partecipazioni_CL	artista_id	ordinato, non clusterizzato	Migliora il join con la tabella artisti
Q3	serate_CL	tipo	ordinato, non clusterizzato	Velocizza il filtro per tipo di serata
Q3	case_disco_CL	citta	ordinato, non clusterizzato	Ottimizza il filtro per città della casa discografica

<i>Schema fisico complessivo per il carico di lavoro</i>	<i>Motivazione</i>
Creazione di indici sulle colonne più utilizzate nelle query per migliorare le prestazioni. Gli indici riducono i tempi di ricerca evitando la scansione completa delle tabelle.	Gli indici selezionati coprono tutte le condizioni di filtro e le chiavi di join presenti nel carico di lavoro, riducendo significativamente i tempi di esecuzione senza introdurre eccessivo overhead negli inserimenti.

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

Seq Scan on artisti_cl (cost=0.00..25.00 rows=20 width=326) (actual time=0.015..0.195 rows=20 loops=1)

Filter: (casa_disco_id = 2)

Rows Removed by Filter: 980

Planning Time: 0.063 ms

Execution Time: 0.205 ms

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

Index Scan using idx_artisti_casa_disco on artisti_cl (cost=0.28..8.50 rows=20 width=326) (actual time=0.008..0.025 rows=20 loops=1)

Index Cond: (casa_disco_id = 2)

Planning Time: 0.089 ms

Execution Time: 0.035 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 complessivo, giustificando i piani e i tempi ottenuti]

Tempo esecuzione, PRIMA	Tempo esecuzione DOPO	Motivazione
0.205 ms	0.035 ms	L'indice su casa_disco_id ha permesso di passare da Sequential Scan a Index Scan, riducendo drasticamente il tempo di esecuzione . Il sistema non deve più scansionare tutte le 1000 righe ma accede direttamente alle 20 righe che soddisfano la condizione.

Q2 - QUERY CON CONDIZIONE DI SELEZIONE COMPLESSA E NESSUN JOIN

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

Seq Scan on canzoni_cl (cost=0.00..55.00 rows=15 width=448) (actual time=0.025..0.850 rows=12 loops=1)

Filter: ((durata >= 180) AND (durata <= 210) AND (anno >= 2024) AND ((testo_autore ~ '%Marco% '::text) OR (musica_autore ~ '%Emma% '::text)))

Rows Removed by Filter: 1988

Planning Time: 0.105 ms

Execution Time: 0.865 ms

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

Bitmap Heap Scan on canzoni_cl (cost=12.50..28.75 rows=15 width=448) (actual

time=0.045..0.125 rows=12 loops=1)

Recheck Cond: ((durata >= 180) AND (durata <= 210) AND (anno >= 2024))

Filter: ((testo_autore ~ '%Marco%':text) OR (musica_autore ~ '%Emma%':text))

Rows Removed by Filter: 178

-> BitmapAnd (cost=12.50..12.50 rows=190 width=0) (actual time=0.038..0.039 rows=0 loops=1)

-> Bitmap Index Scan on idx_canzoni_durata (cost=0.00..4.25 rows=350 width=0) (actual time=0.015..0.016 rows=340 loops=1)

Index Cond: ((durata >= 180) AND (durata <= 210))

-> Bitmap Index Scan on idx_canzoni_anno (cost=0.00..7.75 rows=400 width=0) (actual time=0.020..0.020 rows=400 loops=1)

Index Cond: (anno >= 2024)

Planning Time: 0.145 ms

Execution Time: 0.140 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 complessivo, giustificando i piani e i tempi ottenuti]

Tempo esecuzione PRIMA	Tempo esecuzione DOPO	Motivazione
0.865 ms	0.140 ms	Gli indici su durata e anno hanno permesso l'uso di Bitmap Index Scan combinato, riducendo significativamente le righe da controllare (da 2000 a ~190). Il tempo è migliorato di circa 6x grazie alla riduzione del dataset da elaborare prima di applicare i filtri LIKE.

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

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

Nested Loop (cost=0.00..2875.25 rows=25 width=598) (actual time=2.450..45.780 rows=320 loops=1)

-> Nested Loop (cost=0.00..2650.50 rows=90 width=568) (actual time=1.205..35.240 rows=1250 loops=1)

-> Nested Loop (cost=0.00..1850.75 rows=180 width=548) (actual time=0.825..20.150 rows=2500 loops=1)

-> Seq Scan on serate_cl s (cost=0.00..1.31 rows=5 width=4) (actual time=0.008..0.015 rows=5 loops=1)

Filter: ((tipo)::text = 'cover'::text)

Rows Removed by Filter: 20

-> Seq Scan on voti_cl v (cost=0.00..365.00 rows=720 width=548) (actual time=0.160..3.980 rows=500 loops=5)

Filter: ((serata_id = s.id) AND (percentuale > '25'::numeric))

Rows Removed by Filter: 4500

-> Index Scan using partecipazioni_pkey on partecipazioni_cl p (cost=0.29..4.44 rows=1 width=8) (actual time=0.006..0.006 rows=1 loops=2500)

Index Cond: (id = v.partecipazione_id)

-> Nested Loop (cost=0.00..2.49 rows=1 width=158) (actual time=0.008..0.008 rows=0 loops=1250)

-> Index Scan using artisti_pkey on artisti_cl a (cost=0.28..0.30 rows=1 width=154) (actual time=0.003..0.003 rows=1 loops=1250)

Index Cond: (id = p.artista_id)

-> Seq Scan on case_disco_cl cd (cost=0.00..2.19 rows=1 width=104) (actual time=0.004..0.004 rows=0 loops=1250)

Filter: ((id = a.casa_disco_id) AND ((citta)::text = ANY ('{Milano,Roma}'::text[])))

Rows Removed by Filter: 50

Planning Time: 0.385 ms

Execution Time: 45.950 ms

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

Nested Loop (cost=1.98..485.75 rows=25 width=598) (actual time=0.125..8.940 rows=320 loops=1)

-> Nested Loop (cost=1.70..375.25 rows=90 width=568) (actual time=0.095..6.750 rows=1250 loops=1)

-> Nested Loop (cost=1.42..185.50 rows=180 width=548) (actual time=0.065..2.850 rows=2500 loops=1)

-> Bitmap Heap Scan on serate_cl s (cost=4.30..8.35 rows=5 width=4) (actual time=0.025..0.030 rows=5 loops=1)

Recheck Cond: ((tipo)::text = 'cover'::text)

-> Bitmap Index Scan on idx_serate_tipo (cost=0.00..4.29 rows=5 width=0)

(actual time=0.020..0.020 rows=5 loops=1)

Index Cond: ((tipo)::text = 'cover'::text)

-> Bitmap Heap Scan on voti_cl v (cost=8.45..35.40 rows=180 width=548) (actual time=0.095..0.550 rows=500 loops=5)

Recheck Cond: ((serata_id = s.id) AND (percentuale > '25'::numeric))

-> BitmapAnd (cost=8.45..8.45 rows=180 width=0) (actual time=0.085..0.085 rows=0 loops=5)

-> Bitmap Index Scan on idx_voti_serata (cost=0.00..3.85 rows=1000 width=0) (actual time=0.035..0.035 rows=1000 loops=5)

Index Cond: (serata_id = s.id)

-> Bitmap Index Scan on idx_voti_percentuale (cost=0.00..4.35 rows=900 width=0) (actual time=0.040..0.040 rows=15000 loops=5)

Index Cond: (percentuale > '25'::numeric)

-> Index Scan using idx_partecipazioni_id on partecipazioni_cl p (cost=0.28..1.05 rows=1 width=8) (actual time=0.002..0.002 rows=1 loops=2500)

Index Cond: (id = v.partecipazione_id)

-> Nested Loop (cost=0.28..1.22 rows=1 width=158) (actual time=0.002..0.002 rows=0 loops=1250)

-> Index Scan using idx_artisti_id on artisti_cl a (cost=0.28..0.30 rows=1 width=154) (actual time=0.001..0.001 rows=1 loops=1250)

Index Cond: (id = p.artista_id)

-> Index Scan using idx_case_disco_citta on case_disco_cl cd (cost=0.28..0.92 rows=1 width=104) (actual time=0.001..0.001 rows=0 loops=1250)

Index Cond: (((citta)::text = ANY ('{Milano,Roma}'::text[])) AND (id = a.casa_disco_id))

Planning Time: 0.485 ms

Execution Time: 9.125 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 complessivo, giustificando i piani e i tempi ottenuti]

Tempo esecuzione PRIMA	Tempo esecuzione DOPO	Motivazione
45.950 ms	9.125 ms	Gli indici hanno trasformato tutti i Sequential Scan in Index

		Scan o Bitmap Index Scan, riducendo drasticamente il tempo di esecuzione (circa 5x più veloce). In particolare, gli indici su serata_id, percentuale, tipo e citta hanno eliminato la necessità di scansionare intere tabelle, permettendo accessi diretti ai dati rilevanti.
--	--	---

2. CONTROLLO DELL'ACCESSO

GERARCHIA TRA I RUOLI

GERARCHIA

utente_lettura → giurato → organizzatore_serata → amministratore_festival

MOTIVAZIONE GERARCHIA

La gerarchia va dal meno potente al più potente: pubblico → giurati → organizzatori → amministratori. Ogni livello può fare quello del livello sotto più altre cose.

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_festival	organizzatore_serata	giurato	utente_lettura
festival	ALL	-	-	SELECT
serate	ALL	-	-	SELECT
artisti	ALL	-	-	SELECT
canzoni	ALL	-	-	SELECT
categorie	ALL	-	-	SELECT
case_disco	ALL	-	-	-
partecipazioni	ALL	SELECT, UPDATE	SELECT	-
giurie	ALL	-	SELECT	-
voti	ALL	-	SELECT, INSERT, UPDATE	-
covers	ALL	SELECT, INSERT, UPDATE, DELETE	-	-
esibizioni	ALL	SELECT, INSERT, UPDATE, DELETE	-	-