

Progetto Basi di Dati 2024-25

“FANTASANREMO”

Parte III

Team 35 Alessio Molinas 5339413
Ettore Romano 5644926

1 Progettazione Fisica

1.1 CARICO DI LAVORO

1.1.1 Q1 - QUERY CON SINGOLA SELEZIONE E NESSUN JOIN

LINGUAGGIO NATURALE

Seleziona tutte le informazioni degli artisti dalla tabella artisti_cl il cui nome è esattamente 'Ana'

```
SELECT *  
FROM artisti_cl  
WHERE nome = 'Ana';
```

1.1.2 Q2 - QUERY CON CONDIZIONE DI SELEZIONE COMPLESSA E NESSUN JOIN

LINGUAGGIO NATURALE

Seleziona tutte le informazioni degli artisti dalla tabella artisti_cl che si chiamano Ana e sono nati dopo il 1° gennaio 1992.

```
SELECT *  
FROM artisti_cl  
WHERE nome = 'Ana' AND dataNascita >= DATE '1992-01-01';
```

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

LINGUAGGIO NATURALE

Seleziona tutte le informazioni relative alle leghe chiamate 'Anthony', insieme alle squadre che vi partecipano e ai dettagli della partecipazione a quelle leghe

```
SELECT * FROM leghe_cl l
JOIN partecipazione_leghe_cl pl ON l.codLega = pl.codLega
JOIN squadre_cl s ON pl.codSquadra = s.codSquadra
WHERE l.nome = 'Anthony';
```

1.2 1D-PROGETTO FISICO

Id query	Relazione	Chiave ricerca	di	Tipo (ordinato/hash, clusterizzato/non clusterizzato)	Motivazione
Q1	artisti	nome		ordinato e clusterizzato	Considerato l'elevato numero di query che filtrano risultati in base all'attributo nome, e considerato che tale attributo non è UNIQUE, si è optato per la clusterizzazione.
Q2	artisti	nome, dataNascita		ordinato e clusterizzato	Poiché le query coinvolgono ricerche per range sulla data di nascita, si esclude l'utilizzo di un indice hash. Si è quindi optato per un indice composto su nome e dataNascita, in modo da ottimizzare sia le ricerche puntuali sul nome che quelle basate su intervalli temporali. Questa scelta consente di migliorare le performance complessive delle interrogazioni più frequenti.
Q3	leghe, partecipazione_leghe, squadre	leghe.nome		ordinato	Oltre all'indice sull'attributo leghe.nome, si propone l'introduzione di indici sugli attributi chiave utilizzati nelle operazioni di join. Questa scelta consente al motore di ottimizzazione delle query di sfruttare algoritmi più efficienti, come l'indexed nested loop join, migliorando così le performance complessive delle interrogazioni che coinvolgono più tabelle."

Schema fisico complessivo per il carico di lavoro	Motivazione
- indice ordinato e clusterizzato su (nome, dataNascita), tabella artisti	Le query più frequenti prevedono la ricerca per uguaglianza sull'attributo nome e per intervallo su dataNascita, ma sono comuni anche interrogazioni basate esclusivamente sul nome. Per questo motivo, si è scelto di creare un indice composto su (nome, dataNascita), che ottimizza entrambe le tipologie di ricerca
- indice ordinato su nome, tabella leghe	Ricerca per nome

1.3 1G-ANALISI PIANI DI ESECUZIONE SCELTI DAL SISTEMA

1.3.1 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 as artisti_cl
Filter: ((nome)::text = 'Ana'::text)

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

Index Scan using idx_artisti_cl_nome_datanascita on artisti_cl as artisti_cl
Index Cond: ((nome)::text = 'Ana'::text)

CONFRONTO TRA I DUE PIANI

Tempo esecuzione PRIMA	Tempo esecuzione DOPO	Motivazione
00:00:00.153	00:00:00.054	Anche nel caso in cui la query effettui una ricerca solo sull'attributo nome, senza condizioni su dataNascita, l'indice composto (nome, dataNascita) risulta comunque efficace. Questo perché il motore può sfruttare la parte iniziale dell'indice per ottimizzare le interrogazioni basate esclusivamente su nome

1.3.2 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 artisti_cl as artisti_cl
Filter: ((datanascita ≥ '1992-01-01'::date) AND ((nome)::text = 'Ana'::text))

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

Index Scan using idx_artisti_cl_nome_datanascita on artisti_cl as artisti_cl
Index Cond: (((nome)::text = 'Ana'::text) AND (datanascita ≥ '1992-01-01'::date))

CONFRONTO TRA I DUE PIANI

Tempo esecuzione PRIMA	Tempo esecuzione DOPO	Motivazione
00:00:00.079	00:00:00.052	L'indice composto su (nome, dataNascita) viene utilizzato efficacemente nelle query che filtrano per uguaglianza sul nome e per intervallo su dataNascita. Questa combinazione consente al motore di sfruttare appieno la struttura dell'indice, migliorando le performance

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

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

```
Hash Inner Join
Hash Cond: (s.codsquadra = pl.codsquadra)
Seq Scan on squadre_cl as s
Hash
Hash Inner Join
Hash Cond: (pl.codlega = l.codlega)
Seq Scan on partecipazione_leghe_cl as pl
Hash
Seq Scan on leghe_cl as l
Filter: ((nome)::text = 'Anthony'::text)
```

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

```
Hash Inner Join
Hash Cond: (s.codsquadra = pl.codsquadra)
Seq Scan on squadre_cl as s
Hash
Hash Inner Join
Hash Cond: (pl.codlega = l.codlega)
Seq Scan on partecipazione_leghe_cl as pl
Hash
Bitmap Heap Scan on leghe_cl as l
Recheck Cond: ((nome)::text = 'Anthony'::text)
Bitmap Index Scan using idx_leghe_cl_nome
Index Cond: ((nome)::text = 'Anthony'::text)
```

CONFRONTO TRA I DUE PIANI

Tempo esecuzione PRIMA	Tempo esecuzione DOPO	Motivazione
00:00:00.043	00:00:00.052	Nonostante la creazione di un indice sul nome della lega — testato sia in versione clusterizzata che non — la query non ha ottenuto benefici significativi in termini di performance. Sono stati inoltre effettuati alcuni test con la creazione di indici sui campi utilizzati per le join (i codici delle relazioni), ma anche in questo caso il database ha preferito utilizzare l'hash join, ignorando gli indici disponibili.

2 CONTROLLO DELL'ACCESSO

2.1 GERARCHIA TRA I RUOLI

2.1.1 GERARCHIA

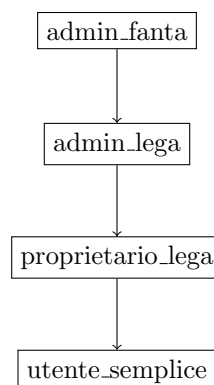
La gerarchia scelta è la seguente:

`admin_fanta` → `admin_lega` → `proprietario_lega` → `utente_semplice`

2.1.2 MOTIVAZIONE GERARCHIA

Gerarchia dei ruoli e motivazioni

La gerarchia dei ruoli nel sistema FantaSanremo è stata progettata seguendo i principi del **minimo privilegio** e della **modularità dei compiti**. La struttura gerarchica è la seguente:



admin_fanta È il super amministratore del sistema, con pieni privilegi su tutte le tabelle del FantaSanremo.

Motivazione: è responsabile della gestione complessiva del sistema e delle attività di manutenzione legate al gioco. Le tabelle relative alla competizione "Festival di Sanremo" vengono invece popolate da un sistema esterno, tramite un utente tecnico dotato di permessi specifici aggiuntivi.

admin_lega È il responsabile della gestione del FantaSanremo, con privilegi di modifica su leghe, squadre e partecipanti delle leghe.

Motivazione: è pensato per supervisionare la gestione del gioco FantaSanremo.

proprietario_lega Questo ruolo rappresenta l'utente che crea e gestisce una propria lega. Ha il diritto di gestire la partecipazione alla lega e modificare i dati della propria squadra.

Motivazione: ha responsabilità amministrative limitate alla propria lega.

utente_semplice È il partecipante comune al FantaSanremo. Può iscriversi alle leghe, creare e gestire le proprie squadre.

Motivazione: rappresenta l'utente finale che partecipa al gioco, con accesso strettamente limitato ai propri dati. Questa gerarchia consente di:

- garantire la **sicurezza**, evitando accessi non autorizzati a utenti con ruoli minori;
- mantenere la **chiarezza**, grazie all'ereditarietà dei privilegi tra ruoli;
- migliorare la **manutenibilità**, poiché i privilegi possono essere gestiti centralmente tramite i ruoli.

2.2 ASSEGNAZIONE PRIVILEGI SPECIFICI AI RUOLI

Relazione	Amministratore del FantaSanremo	Utente	Amministratore lega	Proprietario lega
artisti	SELECT	SELECT	SELECT	SELECT
brani	SELECT	SELECT	SELECT	SELECT
contributi.brani	SELECT	SELECT	SELECT	SELECT
esibizioni	SELECT	SELECT	SELECT	SELECT
serate	SELECT	SELECT	SELECT	SELECT
voti	SELECT	SELECT	SELECT	SELECT
bonus.assegnati	SELECT, INSERT, UPDATE, DELETE	SELECT	SELECT	SELECT
bonus.malus	SELECT, INSERT, UPDATE, DELETE	SELECT	SELECT	SELECT
formazioni	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)
gestione_leghe	SELECT, INSERT, UPDATE, DELETE	-	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)
leghe	SELECT, INSERT, UPDATE, DELETE	SELECT	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)
partecipazione_leghe	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)
squadre	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)	SELECT, INSERT, UPDATE, DELETE	SELECT, INSERT, UPDATE, DELETE (own)
utenti	SELECT, INSERT, UPDATE, DELETE	-	SELECT	-