

Alessandro Patrone 4249874
Sabrina Polidoro 4682624

9. [D] Progetto fisico e sua validazione

Interrogazione 1

--Determinare i clienti con saldo maggiore o uguale di 45 e punti mensili maggiori di 37

```
SELECT *  
FROM socialmarket.cliente  
WHERE saldo >= 45 AND puntimensili > 37;
```

Eseguendo più volte le query con e senza indici, possiamo notare che:

- senza indici: 71 ms
- con (I_CF(codicefiscale)): 43 ms
- con CLUSTER (I_CF(codicefiscale)): 37 ms
- con (I_saldo_punti(saldo, puntimensili)): 35 ms

#	Node	Timings	
		Exclusive	Inclusive
1.	→ Seq Scan on cliente as cliente (actual=0.011..0.086 rows=102 loops=1) Filter: ((saldo >= '45'::numeric) AND (puntimensili > '37'::numeric)) Rows Removed by Filter: 398	0.086 ms	0.086 ms

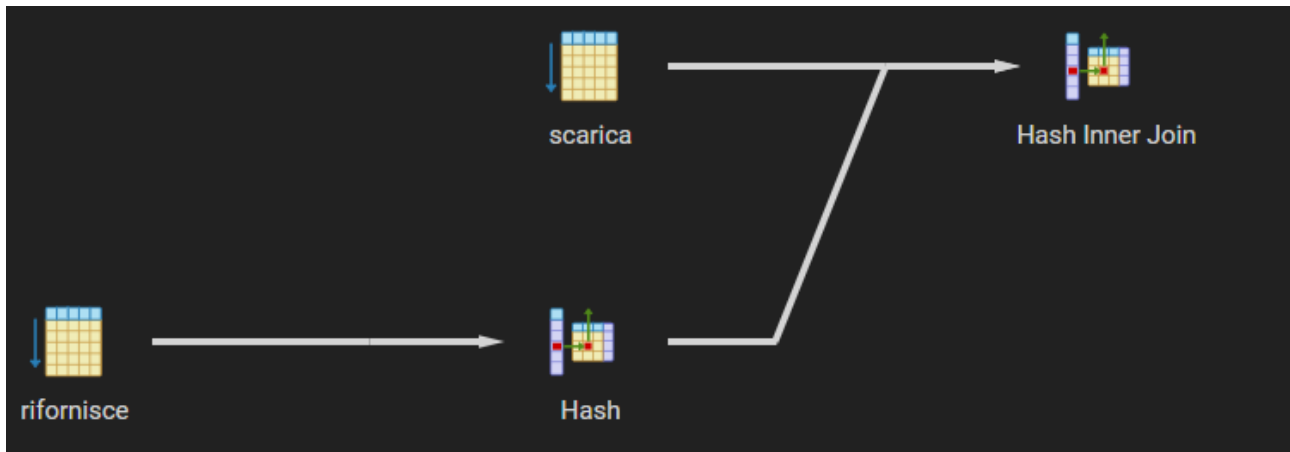
Interrogazione 2

--Determinare i prodotti il cui rifornimento è uguale allo scarico

```
SELECT rifornisce.nome, rifornisce.quantita AS rQ, scarica.quantita AS sQ  
FROM socialmarket.scarica  
JOIN socialmarket.rifornisce ON rifornisce.quantita = scarica.quantita;
```

Anche con questa interrogazione possiamo notare come il tempo di esecuzione senza l'utilizzo di indici è nettamente maggiore rispetto al loro mancato utilizzo.

- senza indici: 52 ms
- con (I_Prodotto(nome)): 43 ms
- con CLUSTER (I_prodotto(nome)): 32 ms



#	Node	Timings	
		Exclusive	Inclusive
1.	→ Hash Inner Join (actual=0.032..0.061 rows=54 loops=1) Hash Cond: (scarica.quantita = rifornisce.quantita)	0.031 ms	0.061 ms
2.	→ Seq Scan on scarica as scarica (actual=0.008..0.013 rows=100 loops=1)	0.013 ms	0.013 ms
3.	→ Hash (actual=0.018..0.018 rows=50 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 11 kB	0.01 ms	0.018 ms
4.	→ Seq Scan on rifornisce as rifornisce (actual=0.004..0.008 rows=50 loops=1)	0.008 ms	0.008 ms

Interrogazione 3

--Determinare i membri autorizzati all'utilizzo della tessera appartenenti alla fascia di età identificata dalla lettera 't'

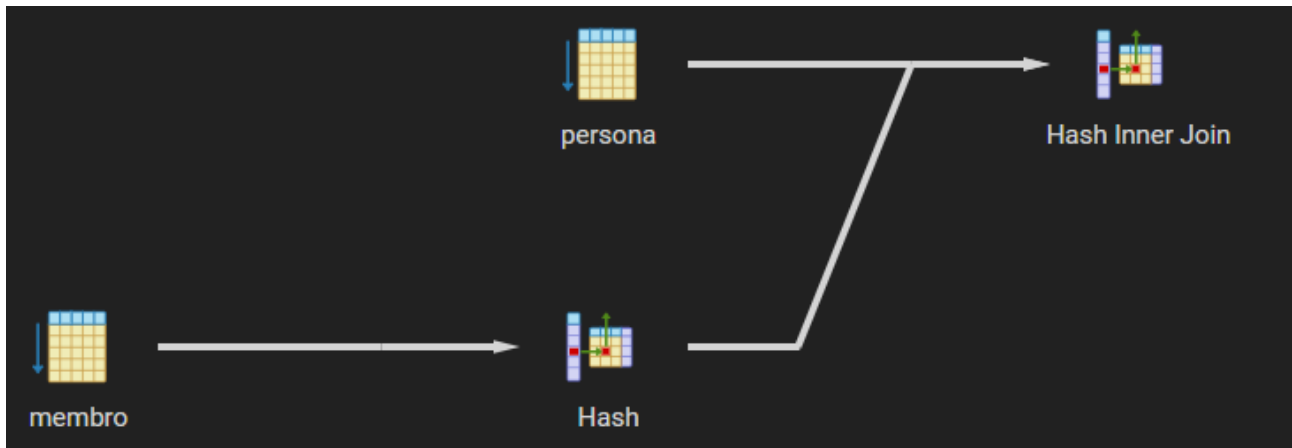
```
SELECT membro.codicefiscale, persona.nome, persona.cognome
```

```
FROM socialmarket.membro
```

```
JOIN socialmarket.persona ON persona.codicefiscale = membro.codicefiscale
```

```
WHERE membro.fasciaeta = 't' AND membro.autorizzato = true;
```

- senza indici: 41 ms
- con CLUSTER (I_CF(codicefiscale)): 33 ms



#	Node	Timings	
		Exclusive	Inclusive
1.	→ Hash Inner Join (actual=0.059..0.22 rows=5 loops=1) Hash Cond: ((persona.codicefiscale)::text = (membro.codicefiscale)::text)	0.113 ms	0.22 ms
2.	→ Seq Scan on persona as persona (actual=0.01..0.086 rows=1500 loops=1)	0.086 ms	0.086 ms
3.	→ Hash (actual=0.022..0.022 rows=5 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 9 kB	0.003 ms	0.022 ms
4.	→ Seq Scan on membro as membro (actual=0.011..0.019 rows=5 loops=1) Filter: (autorizzato AND (fasciaeta = 'Y'::bpchar)) Rows Removed by Filter: 95	0.019 ms	0.019 ms

La scelta è ricaduta sull'hash join a prescindere dagli indici perché non è richiesto l'utilizzo di tabelle ordinate per il suo funzionamento.

Per tutte e tre le interrogazioni la scelta del sistema è stata quella di usare scansioni sequenziali, probabilmente dovuta alla dimensione delle tabelle. Purtroppo, non è stato possibile usare il programma Datanamic causa problemi con le chiavi esterne per il popolamento del database, portandoci alla decisione di utilizzare programmi terzi limitati nelle dimensioni massime di generazione.

10. [D] Descrizione in linguaggio naturale e codice PL/pgSQL della transazione considerata e giustificazione per il livello di isolamento prescelto.

La transazione consiste in tre operazioni:

la prima seleziona tutti i prodotti 'Assorted Desserts', la seconda aggiorna la quantità disponibile di cinque unità e la terza mostra tutti i prodotti chiamati 'Assorted Desserts' oppure tutti i prodotti con quantità uguale a 100.

```
BEGIN TRANSACTION;
```

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

```
SELECT *  
FROM socialmarket.prodotto  
WHERE nome = 'Assorted Desserts';
```

```
UPDATE socialmarket.prodotto  
SET quantita= quantita + 5  
WHERE nome = 'Assorted Desserts';
```

```
SELECT nome, quantita  
FROM socialmarket.prodotto  
WHERE nome = 'Assorted Desserts' OR quantita = '100';
```

```
COMMIT;
```

L'anomalia che si potrebbe generare è di tipo "unrepeatable read", in quanto le due letture potrebbero essere inconsistenti se effettuate senza lock sul processo di update.

I livelli di isolamento possibili che garantiscono l'assenza di questa anomalia sono di tipo SERIALIZABLE oppure REPEATABLE READ.

La scelta è ricaduta su SERIALIZABLE in quanto nessun valore letto o scritto dalla transazione viene modificato da altre transazioni prima che abbia terminato.

Inoltre, se si utilizzasse REPEATABLE READ, la condizione di ricerca "quantita='100'" potrebbe portare a non selezionare tutte le tuple nel caso in cui si verificasse un'anomalia "Phantom Row", situazione invece non verificabile con livello di isolamento SERIALIZABLE.

11.[D] Descrizione della politica di controllo dell'accesso scelta, motivando le scelte effettuate. Si suggerisce di riassumere la politica di controllo dell'accesso con una tabella contenente una riga per ogni tabella, una colonna per ogni utente

Autorizzazioni di roberto:

- Insert su "compra" e "rifornisce" così da poter svolgere le sue mansioni di volontario: fare la spesa con il cliente e riacquistare ciò che manca nel market.
- Select su "turni" per visione i turni che gli sono stati assegnati
- Select e Update su "prodotto" così da poter aggiornare le quantità dopo aver aiutato un cliente all'acquisto, dopo aver fatto rifornimento degli articoli mancanti oppure dopo aver ricevuto una donazione.

Autorizzazioni di alice:

- Select su "recapito" e "volontario" così da poterne visionare i dati
- Select, Update e Insert su "rifornisce", "cliente", "turni", "donatore", "prodotto" così da poter gestire al meglio il Social Market ed eseguire le proprie mansioni come, ad esempio, aggiornare i turni dei volontari, fornire le tessere ai clienti, assegnare punti mensili. Queste autorizzazioni sono con GRANT OPTION in modo da poter delegare i propri compiti.

	roberto	alice
Compra	insert	
Rifornisce	insert	select,insert,update
Recapito		select
Volontario		select
Cliente		select,insert,update
Turni	select	select,insert,update
Donatore		select,insert,update
Prodotto	select, update	select,insert,update