Basi di dati II

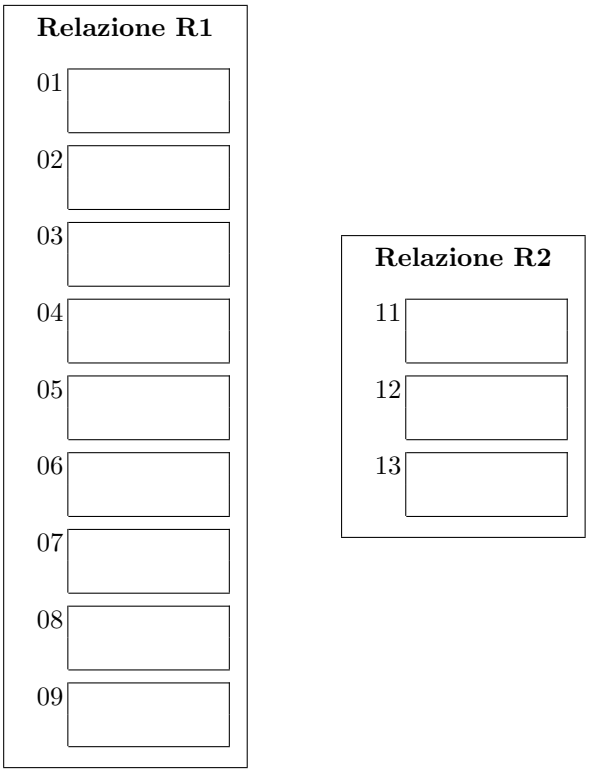
Esame — 18 settembre 2020

Tempo a disposizione: due ore.

Cognome _____ **Nome** _____ **Matricola** _____

Basi di dati II — 18 settembre 2020

Domanda 1 (20%)
Considerare le relazioni R1 ed R2 schematizzate sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco.

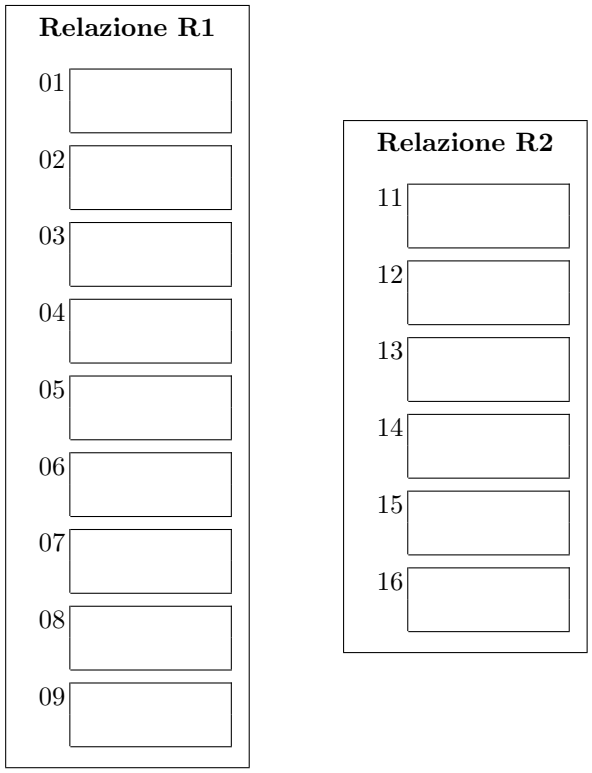


Supponendo di disporre di un buffer di quattro pagine, considerare l'esecuzione di un join con nested loop (senza indici) e rispondere ai quesiti seguenti.

Indicare, nell'ordine, gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per eseguire l'intero join (riportando più volte gli indirizzi su cui si eseguono eventualmente più pin).

Indicare simbolicamente il costo complessivo dell'operazione, in termini di accessi a blocchi, denotando con L_1 ed L_2, rispettivamente, il numero di blocchi di R1 e R2 e con P il numero di pagine di buffer disponibili.

Considerare ora le relazioni R1 ed R2 schematizzate sotto.



Supponendo di disporre sempre di un buffer di quattro pagine, considerare ancora l'esecuzione di un join con nested loop (senza indici) e rispondere ai quesiti seguenti (uguali a quelli posti nella domanda precedente).

Indicare, nell'ordine, gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per eseguire l'intero join (riportando più volte gli indirizzi su cui si eseguono eventualmente più pin).

Indicare simbolicamente il costo complessivo dell'operazione, in termini di accessi a blocchi, denotando con L_1 ed L_2, rispettivamente, il numero di blocchi di R1 e R2 e con P il numero di pagine di buffer disponibili

Domanda 2 (25%)
Considerare i seguenti scenari in cui due client inviano richieste ad un gestore del controllo di concorrenza. Per il secondo scenario si utilizza una notazione intuitiva anche se non ammissibile in Postgres (con le variabili `XX` e `YY`).

`start transaction`
`  isolation level serializable;`
`select * from impiegati;`	

	`start transaction`
	`  isolation level serializable;`
	`select * from impiegati;`

`insert into impiegati`
`  (matricola, cognome, conteggio)`
`  values (101,'Rossi',1);`
`commit`	

	`insert into impiegati`
	`  (matricola, cognome, conteggio)`
	`  values (102,'Bruni',2);`
	`commit`

`start transaction`
`  isolation level serializable;`
`XX = select count(*) from impiegati;`	

	`start transaction`
	`  isolation level serializable;`
	`YY = select count(*) from impiegati;`

`insert into impiegati`
`  (matricola, cognome, conteggio)`
`  values (101,'Rossi', XX+1);`
`commit`	

	`insert into impiegati`
	`  (matricola, cognome, conteggio)`
	`  values (102,'Bruni', YY+1));`
	`commit`

In entrambi i casi, la seconda transazione viene abortita al momento del commit.

- Spiegare brevemente perché

(continua a pagina seguente)

Se invece il livello di isolamento fosse stato `repeatable read`, allora entrambi in entrambi gli scenari le transazioni si sarebbero concluse con accettazione del commit.

- Spiegare brevemente perché

- Commentare i risultati ottenuti nei due scenari e spiegare perché uno dei due va considerato indesiderabile. Allo scopo, mostrare il contenuto della relazione dopo l'inserimento, supponendola vuota all'inizio. Per comprendere meglio il comportamento, tenere presente che l'attributo `conteggio`, in ciascuna ennupla, serve in sostanza ad indicare la cardinalità della relazione subito dopo l'inserimento della ennupla stessa.

In sostanza, si può osservare che, per ciascun livello di isolamento, i due scenari vengono trattati nello stesso modo: con `serializable` vengono rifiutati entrambi, anche se uno dei due è accettabile, mentre con `repeatable read` vengono accettati entrambi, anche se uno dei due è indesiderabile.

- Spiegare brevemente perché

Basi di dati II — 18 settembre 2020

Domanda 3 (30%) Si consideri la seguente porzione dello schema dell'archivio delle carriere degli studenti dei corsi **di laurea magistrale (biennale)** di una anagrafe ministeriale:

- STUDENTI (CodiceFiscale, Cognome, Nome, DataNascita, CodiceLaurea, VotoLaurea, AnnoAccademicoLaurea), in cui CodiceLaurea è il codice del corso di laurea triennale presso cui lo studente si è laureato
- IMMATRICOLAZIONI (CodiceFiscale, AnnoAccademico, CodiceCorsoLM), in cui CodiceCorsoLM è il codice del corso di laurea magistrale cui lo studente si è immatricolato (si noti che la chiave include anche AnnoAccademico perché, in anni diversi, lo studente potrebbe iscriversi a diverse lauree magistrali)
- CORSIDISTUDIO (CodiceCdS, Titolo, Livello, Classe, CodiceUniv), che contiene informazioni su tutti i corsi di laurea, triennali e magistrali
- UNIVERSITÀ (CodiceUniv, NomeUniversità), che contiene informazioni su tutte le università

Supporre (i) che l'anno accademico sia rappresentato in modo semplice e sempre nello stesso modo; (ii) che il titolo di un corso di studio possa cambiare da un anno all'altro.
Progettare uno schema dimensionale che permetta di rispondere, fra le altre, alle seguenti interrogazioni:

- calcolare il numero di studenti immatricolati in un certo corso di laurea magistrale (inteso come corso di studio presso una università) in un certo anno accademico (distinti in vario modo, ad esempio per corso di laurea di provenienza, oppure per università di provenienza o per fasce di voto di laurea o per numero di anni intercorsi fra laurea triennale e immatricolazione)

Assumere che, per ragioni di privatezza e di compattezza, sia opportuno limitare la cardinalità della tabella dei fatti, a patto di permettere la risposta alle precedenti interrogazioni. Per le fasce di voto, supporre che interessino fasce di 5 in 5 oppure di 10 in 10.
Indicare esplicitamente la grana dei fatti.

Grana dei fatti:

Schema dimensionale:

Basi di dati II — 18 settembre 2020

Descrivere, informalmente, ma in modo strutturato e comprensibile, il processo di ETL che porta alla tabella dei fatti mostrata in risposta alla domanda precedente

Basi di dati II — 18 settembre 2020

Domanda 4 (25%) Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione *R(ID, CodiceFiscale, Cognome, ...)* di cardinalità pari circa a $L = 400.000$, con ennuple di $e = 80$ byte, con due chiavi, *ID* e *CodiceFiscale* (cioè il valore di ciascuna di esse, da solo, identifica univocamente una ennupla). Supporre che il sistema offra

- strutture primarie disordinate
- indici di tipo B-tree

Considerare un carico applicativo che preveda le seguenti operazioni

1. inserimento di una ennupla, con verifica dei due vincoli di chiave (su *CodiceFiscale* e su *ID*) con frequenza oraria $f_1 = 10.000$;
2. ricerca di una ennupla sulla base del valore completo di *ID*, frequenza oraria $f_2 = 10$
3. ricerca di ennuple sulla base del *CodiceFiscale*, eventualmente parziale, con frequenza oraria $f_3 = 10$; supporre che il valore parziale sia molto selettivo e porti alla identificazione, in media, di $s = 2$ ennuple;
4. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) dell'attributo *Cognome*, con frequenza oraria $f_4 = 1$; supporre che il valore parziale sia poco selettivo e porti alla identificazione, in media, di $s = 40$ ennuple.

Progettare l'organizzazione fisica della relazione, individuando gli eventuali indici (da nessuno a tre). Ragionare in termini di numero di accessi a memoria secondaria, assumendo che: (i) gli indici abbiano profondità $p = 4$, (ii) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice, (iii) lettura e scrittura abbiano lo stesso costo. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Indici utilizzati			I due indici sono necessari per la verifica delle chiavi all'inserimento l'inserimento
Costo Op. 1			
Costo Op. 2			
Costo Op. 3			
Costo Op. 4			
Costo tot			

Basi di dati II

Esame — 18 settembre 2020

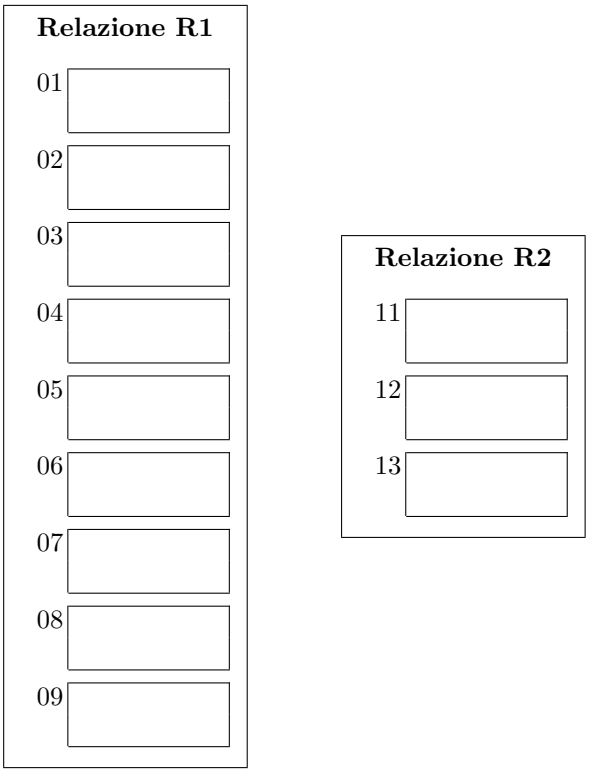
Cenni sulle soluzioni

Tempo a disposizione: due ore.

Cognome _____ **Nome** _____ **Matricola** _____

Basi di dati II — 18 settembre 2020

Domanda 1 (20%)
Considerare le relazioni R1 ed R2 schematizzate sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco.



Supponendo di disporre di un buffer di quattro pagine, considerare l'esecuzione di un join con nested loop (senza indici) e rispondere ai quesiti seguenti.

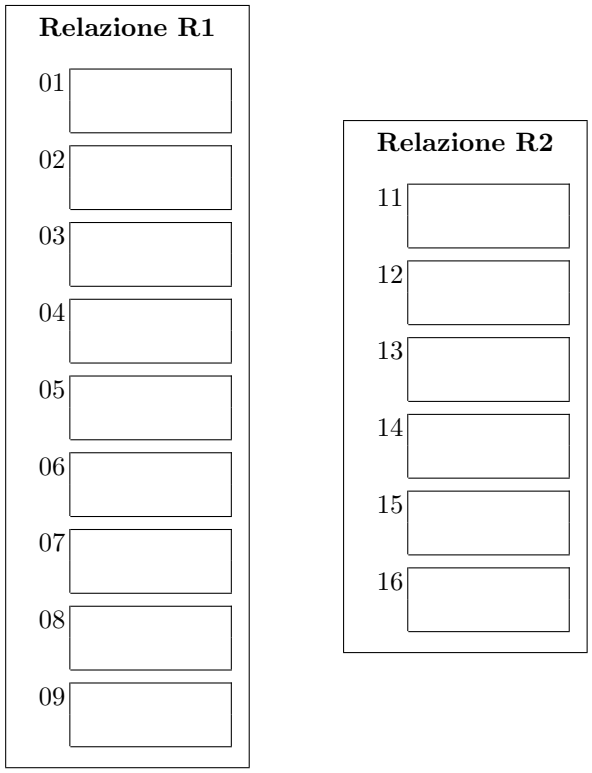
Indicare, nell'ordine, gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per eseguire l'intero join (riportando più volte gli indirizzi su cui si eseguono eventualmente più pin).

11, 12, 13, 01, 02, 03, ...,09

Indicare simbolicamente il costo complessivo dell'operazione, in termini di accessi a blocchi, denotando con L_1 ed L_2, rispettivamente, il numero di blocchi di R1 e R2 e con P il numero di pagine di buffer disponibili.

$L_2 + L_1 = 12$

Considerare ora le relazioni R1 ed R2 schematizzate sotto.



Supponendo di disporre sempre di un buffer di quattro pagine, considerare ancora l'esecuzione di un join con nested loop (senza indici) e rispondere ai quesiti seguenti (uguali a quelli posti nella domanda precedente).

Indicare, nell'ordine, gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per eseguire l'intero join (riportando più volte gli indirizzi su cui si eseguono eventualmente più pin).

11, 12, 13, 01, 02, 03, ...,09, 14, 15, 16, 01, 02, 03, ...,09 (per i compiti A e C)

Indicare simbolicamente il costo complessivo dell'operazione, in termini di accessi a blocchi, denotando con L_1 ed L_2, rispettivamente, il numero di blocchi di R1 e R2 e con P il numero di pagine di buffer disponibili

$L_2 + (L_2/(P-1)) \times L_1 = 24$ (idem)

Domanda 2 (25%)
Considerare i seguenti scenari in cui due client inviano richieste ad un gestore del controllo di concorrenza. Per il secondo scenario si utilizza una notazione intuitiva anche se non ammissibile in Postgres (con le variabili `XX` e `YY`).

```
start transaction
  isolation level serializable;
select * from impiegati;
```

```
start transaction
  isolation level serializable;
select * from impiegati;
```

```
insert into impiegati
  (matricola, cognome, conteggio)
  values (101,'Rossi',1);
commit
```

```
insert into impiegati
  (matricola, cognome, conteggio)
  values (102,'Bruni',2);
commit
```

```
start transaction
  isolation level serializable;
XX = select count(*) from impiegati;
```

```
start transaction
  isolation level serializable;
YY = select count(*) from impiegati;
```

```
insert into impiegati
  (matricola, cognome, conteggio)
  values (101,'Rossi', XX+1);
commit
```

```
insert into impiegati
  (matricola, cognome, conteggio)
  values (102,'Bruni', YY+1));
commit
```

In entrambi i casi, la seconda transazione viene abortita al momento del commit.

- Spiegare brevemente perché

  Possibile risposta: con `serializable`, viene rilevato un ciclo fra le letture e scritture delle due transazioni, sugli stessi dati. Nota bene: l'inserimento fantasma non è rilevante (e anche con il tipico esempio: lettura-inserimento-lettura, tutto funzionerebbe, perché verrebbe letta la versione a inizio transazione)

(continua a pagina seguente)

Se invece il livello di isolamento fosse stato `repeatable read`, allora entrambi in entrambi gli scenari le transazioni si sarebbero concluse con accettazione del commit.

- Spiegare brevemente perché
  con `repeatable read`, non vengono cercati i cicli, ma solo gli aggiornamenti sui dati scritti da ciascuna transazione, che sono in questo caso diversi; non si fa alcun riferimento alle letture

- Commentare i risultati ottenuti nei due scenari e spiegare perché uno dei due va considerato indesiderabile. Allo scopo, mostrare il contenuto della relazione dopo l'inserimento, supponendola vuota all'inizio. Per comprendere meglio il comportamento, tenere presente che l'attributo `conteggio`, in ciascuna ennupla, serve in sostanza ad indicare la cardinalità della relazione subito dopo l'inserimento della ennupla stessa.

  Nel primo scenario, conteggio viene inserito "a mano" e quindi i valori sono rispettivamente 1 e 2, cosa corretta (nell'esempio banale): in effetti, letture e scritture non interferiscono, perchè il valore della lettura non viene utilizzato. Nel secondo scenario, la concorrenza mal gestita fa inserire in entrambi i casi 1: le letture e le scritture interferiscono realmente, in quanto ciascuna scrittura scrive un dato calcolato a partire dalla corrispondente lettura

In sostanza, si può osservare che, per ciascun livello di isolamento, i due scenari vengono trattati nello stesso modo: con `serializable` vengono rifiutati entrambi, anche se uno dei due è accettabile, mentre con `repeatable read` vengono accettati entrambi, anche se uno dei due è indesiderabile.

- Spiegare brevemente perché

  Il controllo di concorrenza ignora la "semantica delle operazioni": sa solo che ci sono letture e scritture, ma non sa se le letture influenzano le scritture, cosa che avviene in uno scenario (che quindi andrebbe evitato) ma non nell'altro (che non crea problemi).

Basi di dati II — 18 settembre 2020

Domanda 3 (30%) Si consideri la seguente porzione dello schema dell'archivio delle carriere degli studenti dei corsi **di laurea magistrale (biennale)** di una anagrafe ministeriale:

- STUDENTI (CodiceFiscale, Cognome, Nome, DataNascita, CodiceLaurea, VotoLaurea, AnnoAccademicoLaurea), in cui CodiceLaurea è il codice del corso di laurea triennale presso cui lo studente si è laureato
- IMMATRICOLAZIONI (CodiceFiscale, AnnoAccademico, CodiceCorsoLM), in cui CodiceCorsoLM è il codice del corso di laurea magistrale cui lo studente si è immatricolato (si noti che la chiave include anche AnnoAccademico perché, in anni diversi, lo studente potrebbe iscriversi a diverse lauree magistrali)
- CORSIDISTUDIO (CodiceCdS, Titolo, Livello, Classe, CodiceUniv), che contiene informazioni su tutti i corsi di laurea, triennali e magistrali
- UNIVERSITÀ (CodiceUniv, NomeUniversità), che contiene informazioni su tutte le università

Supporre (i) che l'anno accademico sia rappresentato in modo semplice e sempre nello stesso modo; (ii) che il titolo di un corso di studio possa cambiare da un anno all'altro.
Progettare uno schema dimensionale che permetta di rispondere, fra le altre, alle seguenti interrogazioni:

- calcolare il numero di studenti immatricolati in un certo corso di laurea magistrale (inteso come corso di studio presso una università) in un certo anno accademico (distinti in vario modo, ad esempio per corso di laurea di provenienza, oppure per università di provenienza o per fasce di voto di laurea o per numero di anni intercorsi fra laurea triennale e immatricolazione)

Assumere che, per ragioni di privatezza e di compattezza, sia opportuno limitare la cardinalità della tabella dei fatti, a patto di permettere la risposta alle precedenti interrogazioni. Per le fasce di voto, supporre che interessino fasce di 5 in 5 oppure di 10 in 10.
Indicare esplicitamente la grana dei fatti.

Grana dei fatti: la grana scelta è "studenti immatricolati in un certo CdL Magistrale, provenienti da un certo CdL triennale, con voto in una certa fascia e con laurea triennale conseguita da un certo numero di anni"

Schema dimensionale: Cenni sulla soluzione

- FattiImmatricolazioni(KCdLM, KAA, KCdLT, KFasciaVotoLaurea, KAnniIntercorsi, NumeroStudenti)
- CdlMagistrale(KCdLM, Titolo, Classe, CodiceUniv, Università) modificata come slowly changing dimension
- AnnoAccademico(KAA, AnnoAccademico, ...)
- CdlTriennale(KCdLT, Titolo, Classe, CodiceUniv, Università) modificata come slowly changing dimension
- FasciaVotoLaurea(KFasciaVotoLaurea, Fascia5Voti, Fascia10Voti)
- AnniIntercorsi(KAnniIntercorsi, ...)

Basi di dati II — 18 settembre 2020

Descrivere, informalmente, ma in modo strutturato e comprensibile, il processo di ETL che porta alla tabella dei fatti mostrata in risposta alla domanda precedente

Basi di dati II — 18 settembre 2020

Domanda 4 (25%) Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione *R(ID, CodiceFiscale, Cognome, ...)* di cardinalità pari circa a $L = 400.000$, con ennuple di $e = 80$ byte, con due chiavi, *ID* e *CodiceFiscale* (cioè il valore di ciascuna di esse, da solo, identifica univocamente una ennupla). Supporre che il sistema offra

- strutture primarie disordinate
- indici di tipo B-tree

Considerare un carico applicativo che preveda le seguenti operazioni

1. inserimento di una ennupla, con verifica dei due vincoli di chiave (su *CodiceFiscale* e su *ID*) con frequenza oraria $f_1 = 10.000$;
2. ricerca di una ennupla sulla base del valore completo di *ID*, frequenza oraria $f_2 = 10$
3. ricerca di ennuple sulla base del *CodiceFiscale*, eventualmente parziale, con frequenza oraria $f_3 = 10$; supporre che il valore parziale sia molto selettivo e porti alla identificazione, in media, di $s = 2$ ennuple;
4. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) dell'attributo *Cognome*, con frequenza oraria $f_4 = 1$; supporre che il valore parziale sia poco selettivo e porti alla identificazione, in media, di $s = 40$ ennuple.

Progettare l'organizzazione fisica della relazione, individuando gli eventuali indici (da nessuno a tre). Ragionare in termini di numero di accessi a memoria secondaria, assumendo che: (i) gli indici abbiano profondità $p = 4$, (ii) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice, (iii) lettura e scrittura abbiano lo stesso costo. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere negli spazi sottostanti, in forma sia simbolica sia numerica.

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. strutt.	Indici su ID, CodiceFiscale e Cognome	Indici su ID e Cognome	I due indici sono necessari per la verifica delle chiavi all'inserimento l'inserimento
Costo Op. 1	3 × (p-2+1) + 2 = ca. 11: visita (p-2) e aggiornamento (1) per ciascun indice e lettura e scrittura del blocco con il record	2 × (p-2+1) + 2 = ca. 8: visita (p-2) e aggiornamento (1) per ciascun indice e lettura e scrittura del blocco con il record	
Costo Op. 2	p-2+1 = 3	p-2+1 =3	
Costo Op. 3	p-2+2 = 4; i 2 record sono quasi sempre in blocchi diversi	p-2+2 = 4	
Costo Op. 4	p-2+40 = ca. 40 — sono in generale in blocchi diversi	$(L \times e)/B$ = 8000 scansione sequenziale	
Tot	11×10.000 + 3×10 + 4×10 + 40×1 = ca 110.000	8×10.000 + 3×10 + 4×10 + 1×8000 = ca 88.000	