

Basi di dati II, primo modulo Prova parziale — 18 aprile 2011— Compito A

Rispondere su questo fascicolo.
Tempo a disposizione: un'ora e trenta minuti.

Cognome _____ Nome _____ Matricola _____ Ordin. _____

Domanda 1 (25%)

Si consideri una base di dati con le relazioni

- $R1(\underline{A}, B, C, D)$ con
 - vincolo di integrità referenziale fra l'attributo D e la chiave E della relazione R2
 - $N_1=2.000.000$ ennuple e fattore di blocco $f_1=20$
 - $b=200$ valori diversi sull'attributo B (tutti gli interi compresi fra 1 e b)
 - $c=200.000$ valori diversi sull'attributo C (tutti gli interi compresi fra 1 e c)
 - una struttura disordinata, un indice sulla chiave primaria A e un altro sull'attributo C;
- $R2(\underline{E}, F, G)$ con
 - $N_2=1.000.000$ ennuple e fattore di blocco $f_2=10$
 - una struttura disordinata, un indice sulla chiave primaria E

Supponendo che:

- gli indici abbiano tutti $p=4$ livelli (contando anche radice e foglie) e fattore di blocco massimo $f_i=100$
- il sistema esegua sempre i join come nested loop
- ogni operazione possa contare su un numero di pagine di buffer pari a circa $q=150$,

valutare il costo (indicandolo in modo sia simbolico sia numerico) di ciascuna delle interrogazioni seguenti:

```
select *
from R1 join R2 on D=E
```

```
select *
from R1 join R2 on D=E
where B=20
```

```
select *
from R1 join R2 on D=E
where C=4
```

Domanda 2 (25%)

Considerare i due seguenti scenari in ciascuno dei quali due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si ignorino le successive richieste della transazione che ha abortito (senza rilanciarla).

scenario 1		scenario 2	
client 1	client 2	client 1	client 2
read(x)		read(x)	
x = x + 10	read(x)		read(x)
write(x)			x = x + 20
	x = x + 20		write(x)
	write(x)		commit
commit		read(x)	
	commit	commit	

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 100. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

Scenario 1 SERIALIZABLE	Scenario 2 READ COMMITTED

Domanda 3 (25%) Considerare uno schema dimensionale relativo agli esami, che utilizzi, come tabella dei fatti e come una delle dimensioni, le relazioni come quelle qui schematizzate:

<u>K</u> Studente	<u>K</u> Corso	<u>K</u> Data	Voto	...
301	201	405	25	...
301	202	406	28	...
302	201	405	25	...
302	203	407	22	...
...

<u>K</u> Corso	Titolo	Crediti	...
201	Geometria	6	...
202	Chimica	6	...
203	Fisica	10	...
...

Supporre che si presentino le seguenti esigenze di modifica:

- per ogni corso, interessa rappresentare anche il docente, per supportare analisi sugli esami svolti da ciascun docente; i docenti cambiano nel tempo e passano da un corso all'altro (e possono anche tenere più corsi nello stesso momento, ma ogni corso ha, in un certo giorno, un solo docente); è disponibile l'informazione relativa ai docenti dei corsi nel tempo (per tutto il periodo, anche passato, di interesse);
- i corsi cambiano nome nel tempo: per esempio, il corso nella prima ennupla potrebbe ad un certo punto cambiare nome da "Geometria" in "Algebra lineare"; interessano selezioni e aggregazioni relative agli esami tanto con riferimento al nome del corso (al momento dell'esame) quanto alla sua identità (un codice che viene introdotto allo scopo, ma non sempre viene utilizzato, perché alcuni analisti preferiscono fare riferimento al nome corrente del corso); le modifiche sono rare, ma è possibile che ci siano corsi con vari cambiamenti di nome.

Mostrare nuove versioni delle due tabelle che permettano di soddisfare le esigenze sopra citate (mostrare anche i dati, con riferimento a quelli presenti negli esempi sopra, aggiungendo nuovi dati ragionevoli, che permettano di comprendere le modifiche). Indicare in particolare quali attributi e quali ennuple vadano aggiunti alle due relazioni.

Domanda 4 (25%) Come noto, esistono varie strategie utilizzabili da parte del gestore del buffer per scegliere la pagina libera (unpinned) da utilizzare (e quindi il blocco da rimpiazzare) per eseguire una pin, fra cui le seguenti:

- *naif*: si sceglie una qualunque pagina libera (ad esempio la prima che si incontra scandendo un array o una lista sempre dall'inizio)
- *FIFO*: si sceglie, fra le pagine libere, quella caricata da più tempo
- *LRU* (least recently used): si sceglie, fra le pagine libere, quella utilizzata meno di recente (cioè quella liberata da più tempo)

Nella figura seguente è schematizzato un piccolissimo buffer con quattro pagine (numerare da 0 a 3), il cui stato viene descritto, per ciascuna pagina, da (i) un intero che indica il numero di pin su di essa (quindi 0 indica che la pagina è libera) (ii) un riferimento al blocco che per ultimo è stato caricato nella pagina; (iii) l'istante in cui è stato effettuato l'ultimo caricamento; (iv) l'istante in cui la pagina è stata per l'ultima volta liberata (se è libera).

Pagina del buffer:	0	1	2	3
numero di pin sulla pagina	1	2	0	0
blocco	70	33	35	47
istante load	1	7	3	5
istante unpin			8	6

Si supponga ora che vengano eseguite (a partire dall'istante 10) le seguenti operazioni (in cui l'argomento del metodo è il blocco di interesse):

unpin(70), pin(60), unpin(60), pin(70), unpin(70), pin(60), unpin(60), pin(70)

Riempendo la tabella seguente, indicare quale pagina del buffer viene utilizzata per ciascuna delle pin e se viene effettivamente eseguita una lettura su disco (ignoriamo le scritture e infatti non abbiamo indicato se le pagine sono sporche), in ciascuna delle strategie. Commentare brevemente il comportamento che si osserva.

Istante		naif		FIFO		LRU	
		Pagina	Lettura?	Pagina	Lettura?	Pagina	Lettura?
11	pin(60)						
13	pin(70)						
15	pin(60)						
17	pin(70)						

Commento:

Basi di dati II, primo modulo Prova parziale — 18 aprile 2011— Compito B

Rispondere su questo fascicolo.
Tempo a disposizione: un'ora e trenta minuti.

Cognome _____ Nome _____ Matricola _____ Ordin. _____

Domanda 1 (25%)

Si consideri una base di dati con le relazioni

- R1(A,B,C,D) con
 - vincolo di integrità referenziale fra l'attributo D e la chiave E della relazione R2
 - $L_1=1.000.000$ ennuple e fattore di blocco $f_1=10$
 - $b=100$ valori diversi sull'attributo B (tutti gli interi compresi fra 1 e b)
 - $c=100.000$ valori diversi sull'attributo C (tutti gli interi compresi fra 1 e c)
 - una struttura disordinata, un indice sulla chiave primaria A e un altro sull'attributo C;
- R2(E,F,G) con
 - $L_2=2.000.000$ ennuple e fattore di blocco $f_2=20$
 - una struttura disordinata, un indice sulla chiave primaria E

Supponendo che:

- gli indici abbiano tutti $i=4$ livelli (contando anche radice e foglie) e fattore di blocco massimo $f_i=100$
- il sistema esegua sempre i join come nested loop
- ogni operazione possa contare su un numero di pagine di buffer pari a circa $q=150$,

valutare il costo (indicandolo in modo sia simbolico sia numerico) di ciascuna delle interrogazioni seguenti:

```
select *
from R1 join R2 on D=E
```

```
select *
from R1 join R2 on D=E
where B=20
```

```
select *
from R1 join R2 on D=E
where C=4
```

Domanda 2 (25%)

Considerare i due seguenti scenari in ciascuno dei quali due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si ignorino le successive richieste della transazione che ha abortito (senza rilanciarla).

scenario 1		scenario 2	
client 1	client 2	client 1	client 2
read(x)		read(x)	
x = x + 10	read(x)		read(x)
write(x)			x = x + 20
	x = x + 20		write(x)
	write(x)		commit
commit		read(x)	
	commit	commit	

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 100. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

Scenario 1 READ COMMITTED	Scenario 2 SERIALIZABLE

Domanda 3 (25%) Considerare uno schema dimensionale relativo agli esami, che utilizzi, come tabella dei fatti e come una delle dimensioni, le relazioni come quelle qui schematizzate:

<u>K</u> Studente	<u>K</u> Corso	<u>K</u> Data	Voto	...
201	301	405	26	...
201	302	406	28	...
202	301	405	26	...
202	303	407	22	...
...

<u>K</u> Corso	Titolo	Crediti	...
301	Geometria	6	...
302	Chimica	6	...
303	Fisica	10	...
...

Supporre che si presentino le seguenti esigenze di modifica:

- per ogni corso, interessa rappresentare anche il docente, per supportare analisi sugli esami svolti da ciascun docente; i docenti cambiano nel tempo e passano da un corso all'altro (e possono anche tenere più corsi nello stesso momento, ma ogni corso ha, in un certo giorno, un solo docente); è disponibile l'informazione relativa ai docenti dei corsi nel tempo (per tutto il periodo, anche passato, di interesse);
- i corsi cambiano nome nel tempo: per esempio, il corso nella prima ennupla potrebbe ad un certo punto cambiare nome da "Geometria" in "Algebra lineare"; interessano selezioni e aggregazioni relative agli esami tanto con riferimento al nome del corso (al momento dell'esame) quanto alla sua identità (un codice che viene introdotto allo scopo, ma non sempre viene utilizzato, perché alcuni analisti preferiscono fare riferimento al nome corrente del corso); le modifiche sono rare, ma è possibile che ci siano corsi con vari cambiamenti di nome.

Mostrare nuove versioni delle due tabelle che permettano di soddisfare le esigenze sopra citate (mostrare anche i dati, con riferimento a quelli presenti negli esempi sopra, aggiungendo nuovi dati ragionevoli, che permettano di comprendere le modifiche). Indicare in particolare quali attributi e quali ennuple vadano aggiunti alle due relazioni.

Domanda 4 (25%) Come noto, esistono varie strategie utilizzabili da parte del gestore del buffer per scegliere la pagina libera (unpinned) da utilizzare (e quindi il blocco da rimpiazzare) per eseguire una pin, fra cui le seguenti:

- *naif*: si sceglie una qualunque pagina libera (ad esempio la prima che si incontra scandendo un array o una lista sempre dall'inizio)
- *FIFO*: si sceglie, fra le pagine libere, quella caricata da più tempo
- *LRU* (least recently used): si sceglie, fra le pagine libere, quella utilizzata meno di recente (cioè quella liberata da più tempo)

Nella figura seguente è schematizzato un piccolissimo buffer con quattro pagine (numerata da 0 a 3), il cui stato viene descritto, per ciascuna pagina, da (i) un intero che indica il numero di pin su di essa (quindi 0 indica che la pagina è libera) (ii) un riferimento al blocco che per ultimo è stato caricato nella pagina; (iii) l'istante in cui è stato effettuato l'ultimo caricamento; (iv) l'istante in cui la pagina è stata per l'ultima volta liberata (se è libera).

Pagina del buffer:	0	1	2	3
numero di pin sulla pagina	1	0	2	0
blocco	70	35	33	47
istante load	1	3	7	5
istante unpin		8		6

Si supponga ora che vengano eseguite (a partire dall'istante 10) le seguenti operazioni (in cui l'argomento del metodo è il blocco di interesse):

unpin(70), pin(60), unpin(60), pin(70), unpin(70), pin(60), unpin(60), pin(70)

Riempendo la tabella seguente, indicare quale pagina del buffer viene utilizzata per ciascuna delle pin e se viene effettivamente eseguita una lettura su disco (ignoriamo le scritture e infatti non abbiamo indicato se le pagine sono sporche), in ciascuna delle strategie. Commentare brevemente il comportamento che si osserva.

Istante		naif		FIFO		LRU	
		Pagina	Lettura?	Pagina	Lettura?	Pagina	Lettura?
11	pin(60)						
13	pin(70)						
15	pin(60)						
17	pin(70)						

Commento:

Basi di dati II, primo modulo Prova parziale — 18 aprile 2011— Compito C

Rispondere su questo fascicolo.
Tempo a disposizione: un'ora e trenta minuti.

Cognome _____ Nome _____ Matricola _____ Ordin. _____

Domanda 1 (25%)

Si consideri una base di dati con le relazioni

- T1(A,B,C,D) con
 - vincolo di integrità referenziale fra l'attributo D e la chiave E della relazione T2
 - $N_1=2.000.000$ ennuple e fattore di blocco $f_1=20$
 - $b=200$ valori diversi sull'attributo B (tutti gli interi compresi fra 1 e b)
 - $c=200.000$ valori diversi sull'attributo C (tutti gli interi compresi fra 1 e c)
 - una struttura disordinata, un indice sulla chiave primaria A e un altro sull'attributo C;
- T2(E,F,G) con
 - $N_2=1.000.000$ ennuple e fattore di blocco $f_2=10$
 - una struttura disordinata, un indice sulla chiave primaria E

Supponendo che:

- gli indici abbiano tutti $i=4$ livelli (contando anche radice e foglie) e fattore di blocco massimo $f_i=100$
- il sistema esegua sempre i join come nested loop
- ogni operazione possa contare su un numero di pagine di buffer pari a circa $q=150$,

valutare il costo (indicandolo in modo sia simbolico sia numerico) di ciascuna delle interrogazioni seguenti:

```
select *
from T1 join T2 on D=E
```

```
select *
from T1 join T2 on D=E
where B=20
```

```
select *
from T1 join T2 on D=E
where C=4
```

Domanda 2 (25%)

Considerare i due seguenti scenari in ciascuno dei quali due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si ignorino le successive richieste della transazione che ha abortito (senza rilanciarla).

scenario 1		scenario 2	
client 1	client 2	client 1	client 2
read(x)		read(x)	
x = x + 10	read(x)		read(x)
write(x)			x = x + 20
commit			write(x)
	x = x + 20	read(x)	commit
	write(x)	commit	
	commit		

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 100. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

Scenario 1 SERIALIZABLE	Scenario 2 READ COMMITTED

Domanda 3 (25%) Considerare uno schema dimensionale relativo agli esami, che utilizzi, come tabella dei fatti e come una delle dimensioni, le relazioni come quelle qui schematizzate:

<u>K</u> Studente	<u>K</u> Corso	<u>K</u> Data	Voto	...
401	201	305	27	...
401	202	306	28	...
402	201	305	27	...
402	203	307	22	...
...

<u>K</u> Corso	Titolo	Crediti	...
201	Geometria	6	...
202	Chimica	6	...
203	Fisica	10	...
...

Supporre che si presentino le seguenti esigenze di modifica:

- per ogni corso, interessa rappresentare anche il docente, per supportare analisi sugli esami svolti da ciascun docente; i docenti cambiano nel tempo e passano da un corso all'altro (e possono anche tenere più corsi nello stesso momento, ma ogni corso ha, in un certo giorno, un solo docente); è disponibile l'informazione relativa ai docenti dei corsi nel tempo (per tutto il periodo, anche passato, di interesse);
- i corsi cambiano nome nel tempo: per esempio, il corso nella prima ennupla potrebbe ad un certo punto cambiare nome da "Geometria" in "Algebra lineare"; interessano selezioni e aggregazioni relative agli esami tanto con riferimento al nome del corso (al momento dell'esame) quanto alla sua identità (un codice che viene introdotto allo scopo, ma non sempre viene utilizzato, perché alcuni analisti preferiscono fare riferimento al nome corrente del corso); le modifiche sono rare, ma è possibile che ci siano corsi con vari cambiamenti di nome.

Mostrare nuove versioni delle due tabelle che permettano di soddisfare le esigenze sopra citate (mostrare anche i dati, con riferimento a quelli presenti negli esempi sopra, aggiungendo nuovi dati ragionevoli, che permettano di comprendere le modifiche). Indicare in particolare quali attributi e quali ennuple vadano aggiunti alle due relazioni.

Domanda 4 (25%) Come noto, esistono varie strategie utilizzabili da parte del gestore del buffer per scegliere la pagina libera (unpinned) da utilizzare (e quindi il blocco da rimpiazzare) per eseguire una pin, fra cui le seguenti:

- *naif*: si sceglie una qualunque pagina libera (ad esempio la prima che si incontra scandendo un array o una lista sempre dall'inizio)
- *FIFO*: si sceglie, fra le pagine libere, quella caricata da più tempo
- *LRU* (least recently used): si sceglie, fra le pagine libere, quella utilizzata meno di recente (cioè quella liberata da più tempo)

Nella figura seguente è schematizzato un piccolissimo buffer con quattro pagine (numerare da 0 a 3), il cui stato viene descritto, per ciascuna pagina, da (i) un intero che indica il numero di pin su di essa (quindi 0 indica che la pagina è libera) (ii) un riferimento al blocco che per ultimo è stato caricato nella pagina; (iii) l'istante in cui è stato effettuato l'ultimo caricamento; (iv) l'istante in cui la pagina è stata per l'ultima volta liberata (se è libera).

Pagina del buffer:	0	1	2	3
numero di pin sulla pagina	1	2	0	0
blocco	70	33	35	47
istante load	1	7	3	5
istante unpin			8	6

Si supponga ora che vengano eseguite (a partire dall'istante 10) le seguenti operazioni (in cui l'argomento del metodo è il blocco di interesse):

unpin(70), pin(60), unpin(60), pin(70), unpin(70), pin(60), unpin(60), pin(70)

Riempendo la tabella seguente, indicare quale pagina del buffer viene utilizzata per ciascuna delle pin e se viene effettivamente eseguita una lettura su disco (ignoriamo le scritture e infatti non abbiamo indicato se le pagine sono sporche), in ciascuna delle strategie. Commentare brevemente il comportamento che si osserva.

Istante		naif		FIFO		LRU	
		Pagina	Lettura?	Pagina	Lettura?	Pagina	Lettura?
11	pin(60)						
13	pin(70)						
15	pin(60)						
17	pin(70)						

Commento:

Basi di dati II, primo modulo Prova parziale — 18 aprile 2011— Compito D

Rispondere su questo fascicolo.
Tempo a disposizione: un'ora e trenta minuti.

Cognome _____ Nome _____ Matricola _____ Ordin. _____

Domanda 1 (25%)

Si consideri una base di dati con le relazioni

- T1(A,B,C,D) con
 - vincolo di integrità referenziale fra l'attributo D e la chiave E della relazione T2
 - $L_1=1.000.000$ ennuple e fattore di blocco $f_1=10$
 - $b=100$ valori diversi sull'attributo B (tutti gli interi compresi fra 1 e b)
 - $c=100.000$ valori diversi sull'attributo C (tutti gli interi compresi fra 1 e c)
 - una struttura disordinata, un indice sulla chiave primaria A e un altro sull'attributo C;
- T2(E,F,G) con
 - $L_2=2.000.000$ ennuple e fattore di blocco $f_2=20$
 - una struttura disordinata, un indice sulla chiave primaria E

Supponendo che:

- gli indici abbiano tutti $i=4$ livelli (contando anche radice e foglie) e fattore di blocco massimo $f_i=100$
- il sistema esegua sempre i join come nested loop
- ogni operazione possa contare su un numero di pagine di buffer pari a circa $q=150$,

valutare il costo (indicandolo in modo sia simbolico sia numerico) di ciascuna delle interrogazioni seguenti:

```
select *
from T1 join T2 on D=E
```

```
select *
from T1 join T2 on D=E
where B=20
```

```
select *
from T1 join T2 on D=E
where C=4
```

Domanda 2 (25%)

Considerare i due seguenti scenari in ciascuno dei quali due client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si ignorino le successive richieste della transazione che ha abortito (senza rilanciarla).

scenario 1		scenario 2	
client 1	client 2	client 1	client 2
read(x)	read(x)	read(x)	read(x)
x = x + 10			x = x + 20
write(x)			write(x)
commit			commit
	x = x + 20	read(x)	
	write(x)	commit	
	commit		

Considerare uno scheduler che utilizzi il controllo di concorrenza basato su 2PL e livelli di isolamento SERIALIZABLE e READ COMMITTED. Assumiamo che (come avviene di solito) 2PL preveda

- SERIALIZABLE: lock a due fasi stretto, con lock condivisi per letture e esclusivi per scritture.
- READ COMMITTED: lock condivisi per la lettura senza 2PL (possono essere rilasciati prima della acquisizione di altri lock) ed esclusivi per la scrittura con 2PL stretto (mantenuti fino a commit o abort).

Mostrare il comportamento dello scheduler nei due casi seguenti, supponendo che il valore iniziale dell'oggetto x sia 100. Indicare le operazioni che vengono eseguite nell'ordine con, per ciascuna, il valore che viene letto o scritto. In conclusione, per ciascun caso, dire se si verificano o meno anomalie.

Scenario 1 READ COMMITTED	Scenario 2 SERIALIZABLE

Domanda 3 (25%) Considerare uno schema dimensionale relativo agli esami, che utilizzi, come tabella dei fatti e come una delle dimensioni, le relazioni come quelle qui schematizzate:

<u>K</u> Studente	<u>K</u> Corso	<u>K</u> Data	Voto	...
401	301	105	28	...
401	302	106	28	...
402	301	105	28	...
402	303	107	22	...
...

<u>K</u> Corso	Titolo	Crediti	...
301	Geometria	6	...
302	Chimica	6	...
303	Fisica	10	...
...

Supporre che si presentino le seguenti esigenze di modifica:

- per ogni corso, interessa rappresentare anche il docente, per supportare analisi sugli esami svolti da ciascun docente; i docenti cambiano nel tempo e passano da un corso all'altro (e possono anche tenere più corsi nello stesso momento, ma ogni corso ha, in un certo giorno, un solo docente); è disponibile l'informazione relativa ai docenti dei corsi nel tempo (per tutto il periodo, anche passato, di interesse);
- i corsi cambiano nome nel tempo: per esempio, il corso nella prima ennupla potrebbe ad un certo punto cambiare nome da "Geometria" in "Algebra lineare"; interessano selezioni e aggregazioni relative agli esami tanto con riferimento al nome del corso (al momento dell'esame) quanto alla sua identità (un codice che viene introdotto allo scopo, ma non sempre viene utilizzato, perché alcuni analisti preferiscono fare riferimento al nome corrente del corso); le modifiche sono rare, ma è possibile che ci siano corsi con vari cambiamenti di nome.

Mostrare nuove versioni delle due tabelle che permettano di soddisfare le esigenze sopra citate (mostrare anche i dati, con riferimento a quelli presenti negli esempi sopra, aggiungendo nuovi dati ragionevoli, che permettano di comprendere le modifiche). Indicare in particolare quali attributi e quali ennuple vadano aggiunti alle due relazioni.

Domanda 4 (25%) Come noto, esistono varie strategie utilizzabili da parte del gestore del buffer per scegliere la pagina libera (unpinned) da utilizzare (e quindi il blocco da rimpiazzare) per eseguire una pin, fra cui le seguenti:

- *naif*: si sceglie una qualunque pagina libera (ad esempio la prima che si incontra scandendo un array o una lista sempre dall'inizio)
- *FIFO*: si sceglie, fra le pagine libere, quella caricata da più tempo
- *LRU* (least recently used): si sceglie, fra le pagine libere, quella utilizzata meno di recente (cioè quella liberata da più tempo)

Nella figura seguente è schematizzato un piccolissimo buffer con quattro pagine (numerare da 0 a 3), il cui stato viene descritto, per ciascuna pagina, da (i) un intero che indica il numero di pin su di essa (quindi 0 indica che la pagina è libera) (ii) un riferimento al blocco che per ultimo è stato caricato nella pagina; (iii) l'istante in cui è stato effettuato l'ultimo caricamento; (iv) l'istante in cui la pagina è stata per l'ultima volta liberata (se è libera).

Pagina del buffer:	0	1	2	3
numero di pin sulla pagina	1	0	2	0
blocco	70	35	33	47
istante load	1	3	7	5
istante unpin		8		6

Si supponga ora che vengano eseguite (a partire dall'istante 10) le seguenti operazioni (in cui l'argomento del metodo è il blocco di interesse):

unpin(70), pin(60), unpin(60), pin(70), unpin(70), pin(60), unpin(60), pin(70)

Riempendo la tabella seguente, indicare quale pagina del buffer viene utilizzata per ciascuna delle pin e se viene effettivamente eseguita una lettura su disco (ignoriamo le scritture e infatti non abbiamo indicato se le pagine sono sporche), in ciascuna delle strategie. Commentare brevemente il comportamento che si osserva.

Istante		naif		FIFO		LRU	
		Pagina	Lettura?	Pagina	Lettura?	Pagina	Lettura?
11	pin(60)						
13	pin(70)						
15	pin(60)						
17	pin(70)						

Commento: