Basi di dati II — 30 gennaio 2015

Tempo a disposizione: due ore.

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

Domanda 1 (20%) Una tecnica per il controllo di concorrenza diversa da quelle viste nel corso va sotto il nome di "concorrenza basata sulla validazione" ed è basata sulle seguenti ipotesi e princìpi.

- Ogni transazione è divisa in tre fasi: (i) prima esegue tutte le proprie letture e svolge le proprie elaborazioni "privatamente" (cioè in memoria, senza scrivere sulla base di dati); (ii) poi chiede allo scheduler il permesso di andare in commit (fase di "validazione") e (iii) se autorizzata, scrive.
- Lo scheduler ricorda per ogni transazione (1) gli insiemi di dati che essa ha letto $\textsc{rSet}(T)$ e ha scritto (o vuole scrivere) $\textsc{wSet}(T)$, (2) in quale "stato" si trova: INIZ (iniziata e non ancora validata), VAL (validata, ma con scritture da completare), CPL (completata, con l'esecuzione di tutte le scritture) e (3) i corrispondenti istanti di inizio $\textsc{iniz}(T)$, validazione $\textsc{val}(T)$ e completamento $\textsc{cpl}(T)$.
- Ad una transazione T la validazione viene concessa in tutti i casi esclusi i seguenti, in cui viene negata:
  1. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata o completata, che non era completata quando T è iniziata (e quindi risulterebbe $\textsc{cpl}(T') > \textsc{iniz}(T)$)
     - un dato $x \in \textsc{rSet}(T) \cap \textsc{wSet}(T')$
  2. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata ma non completata (e quindi risulterebbe $\textsc{cpl}(T') > t = \textsc{val}(T)$)
     - un dato $x \in \textsc{wSet}(T) \cap \textsc{wSet}(T')$

Dimostrare (almeno intuitivamente) che la classe di schedule prodotta è propriamente contenuta in CSR.

Domanda 2 (15%)
Si supponga si abbiano documenti che contengono informazioni su opere teatrali e autori, come il seguente:

```
<opere teatrali>
   <opera>
      <titolo>Riccardo III</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
   </opera>
   <opera>
      <titolo>Amleto</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
   </opera>
   <opera>
      <titolo>La locandiera</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
   </opera>
   <opera>
      <titolo>La tempesta</titolo> <autore>Shakespeare</autore> <tipo>commedia</tipo>
   </opera>
   <opera>
      <titolo>I rusteghi</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
   </opera>
</opere teatrali>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli (mostrare solo le stringhe, non i nodi XML) delle opere il cui autore è Goldoni e che sono commedie

In XQuery, produrre un documento che contiene, per ogni autore che abbia scritto almeno una commedia, la lista di tali commedie. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<autore>
   Goldoni
      La locandiera
      I rusteghi
</autore>
<autore>
   Shakespeare
      La tempesta
</autore>
```

Basi di dati II — 30 gennaio 2015

Domanda 3 (15%) Una grande azienda ha una base di dati dei propri clienti che dispongono di una "tessera fedeltà," con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l'uso della tessera e (b) la filiale cui fa riferimento (ad esempio, quella presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l'interrogazione che calcola, per ciascuna filiale, il numero dei clienti, la somma dei punti fedeltà dei clienti e la relativa media per cliente. Indicare quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascuna filiale pochi rispetto a quelli già presenti), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L'operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di individuare tre filiali da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
4. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascuna filiale pochi rispetto a quelli già presenti), con la finalità di individuare tre filiali da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di individuare tre filiali da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.

Risposte				
1.	2.	3.	4.	5.

Basi di dati II — 30 gennaio 2015

Domanda 4 (25%) Sia data una relazione $R(\underline{A}, B, C)$ contenente circa $L = 10.000.000$ ennuple di $r = 20$ byte ciascuna, di cui $a = 4$ per la chiave A, che contiene valori interi quasi consecutivi, da 1 a poco più di 10.000.000. Supporre che i blocchi abbiano dimensione $B = 2$KB, approssimabile come 2.000, che i puntatori ai record abbiano lunghezza $p = 6$; e che i nodi intermedi degli indici possano essere contenuti nei buffer.
Indicare il costo prevedibile per le seguenti operazioni

1. `SELECT * FROM R WHERE A >= 1000 AND A <=3000`
2. `SELECT COUNT(*) FROM R WHERE A >= 1000 AND A <=3000`
3. `SELECT * FROM R WHERE A = 2000`

in ciascuno dei seguenti casi:

(a) indice primario (sparso) su A realizzato con B+-tree;
(b) indice secondario (denso) su A realizzato con B+-tree.

Riportare le risposte nella tabella sottostante, indicando formula e valore numerico (con brevissimo commento, se necessario)

	Risposte
(a) 1	
(b) 1	
(a) 2	
(b) 2	
(a) 3	
(b) 3	

Basi di dati II — 30 gennaio 2015

Domanda 5 (25%) 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. Quindi R1 occupa $N_1 = 6$ blocchi e R2 ne occupa $N_2 = 8$.

Relazione R1

50		51		52		53		54		55	
X01	AA	Y01	DA	Z03	AB	K03	AB	Z03	AB	Z03	AB
Y42	CA	X42	CC	W05	EF	W07	EF	W08	EF	W09	EF
W73	CC	W93	CB	X52	HA	X59	HA	X50	HA	X56	HA
Z55	GC	W54	LB	Y55	EA	Y54	EA	Y51	EA	Y57	EA

Relazione R2

60			61			62			63			64			65			66			67		
AA	3		BC	4		LB	7		AA	8		AC	3		EA	7		BA	5		EF	6	
DA	7		GB	7		HB	3		EC	2		CB	5		LB	8		BB	4		GA	8	

Si supponga di disporre di un buffer di $p = 4$ pagine.
Considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con il metodo nested loop senza utilizzo di indici. Supporre che non serva memorizzare il risultato e che quindi esso possa essere prodotto una ennupla alla volta (approccio "pipelining")

Indicare, nell'ordine, le prime quattro ennuple che vengono prodotte

Indicare gli indirizzi dei blocchi che si trovano nel buffer dopo che sono state prodotte le prime quattro ennuple.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime quattro ennuple del risultato.

Indicare il numero complessivo di accessi a memoria secondaria necessari per eseguire il join (indicare formula e numero)

Basi di dati II — 30 gennaio 2015

Cenni sulle soluzioni

Tempo a disposizione: due ore.

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

Domanda 1 (20%) Una tecnica per il controllo di concorrenza diversa da quelle viste nel corso va sotto il nome di "concorrenza basata sulla validazione" ed è basata sulle seguenti ipotesi e princìpi.

- Ogni transazione è divisa in tre fasi: (i) prima esegue tutte le proprie letture e svolge le proprie elaborazioni "privatamente" (cioè in memoria, senza scrivere sulla base di dati); (ii) poi chiede allo scheduler il permesso di andare in commit (fase di "validazione") e (iii) se autorizzata, scrive.
- Lo scheduler ricorda per ogni transazione (1) gli insiemi di dati che essa ha letto $\textsc{rSet}(T)$ e ha scritto (o vuole scrivere) $\textsc{wSet}(T)$, (2) in quale "stato" si trova: INIZ (iniziata e non ancora validata), VAL (validata, ma con scritture da completare), CPL (completata, con l'esecuzione di tutte le scritture) e (3) i corrispondenti istanti di inizio $\textsc{iniz}(T)$, validazione $\textsc{val}(T)$ e completamento $\textsc{cpl}(T)$.
- Ad una transazione T la validazione viene concessa in tutti i casi esclusi i seguenti, in cui viene negata:
  1. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata o completata, che non era completata quando T è iniziata (e quindi risulterebbe $\textsc{cpl}(T') > \textsc{iniz}(T)$)
     - un dato $x \in \textsc{rSet}(T) \cap \textsc{wSet}(T')$
  2. esistono (nell'istante t in cui si esamina la validazione di T):
     - una transazione T' validata ma non completata (e quindi risulterebbe $\textsc{cpl}(T') > t = \textsc{val}(T)$)
     - un dato $x \in \textsc{wSet}(T) \cap \textsc{wSet}(T')$

Dimostrare (almeno intuitivamente) che la classe di schedule prodotta è propriamente contenuta in CSR.

Idea generale:
si deve mostrare che uno schedule di questa classe è CSR: si deve trovare lo schedule seriale equivalente ad esso e lo si può trovare sulla base dell'ordine di validazione e vedendo che i conflitti sono in questo caso tutti nello stesso ordine nello schedule ammesso e in quello seriale

Domanda 2 (15%)
Si supponga si abbiano documenti che contengono informazioni su opere teatrali e autori, come il seguente:

```
<opere teatrali>
   <opera>
      <titolo>Riccardo III</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
   </opera>
   <opera>
      <titolo>Amleto</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
   </opera>
   <opera>
      <titolo>La locandiera</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
   </opera>
   <opera>
      <titolo>La tempesta</titolo> <autore>Shakespeare</autore> <tipo>commedia</tipo>
   </opera>
   <opera>
      <titolo>I rusteghi</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
   </opera>
</opere teatrali>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli (mostrare solo le stringhe, non i nodi XML) delle opere il cui autore è Goldoni e che sono commedie

```
//opera[autore/text()="Goldoni"][tipo/text()="commedia"]/titolo/text()
```

In XQuery, produrre un documento che contiene, per ogni autore che abbia scritto almeno una commedia, la lista di tali commedie. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<autore>
   Goldoni
      La locandiera
      I rusteghi
</autore>
<autore>
   Shakespeare
      La tempesta
</autore>
```

Possibile risposta:

```
let $nl := "&#10;"
for $a in distinct-values(//opera[tipo="commedia"]/autore/text())
return
<autore>
   {$a} {for $o in //opera[tipo="commedia"]
      where $o/autore = $a
      return ($nl, " ", $o/titolo/text())
   }
</autore>
```

Basi di dati II — 30 gennaio 2015

Domanda 3 (15%) Una grande azienda ha una base di dati dei propri clienti che dispongono di una "tessera fedeltà," con varie informazioni su ciascun cliente, fra cui (a) il totale dei punti acquisiti attraverso l'uso della tessera e (b) la filiale cui fa riferimento (ad esempio, quella presso cui ha inizialmente richiesto la tessera). Si vuole eseguire su di essa l'interrogazione che calcola, per ciascuna filiale, il numero dei clienti, la somma dei punti fedeltà dei clienti e la relativa media per cliente. Indicare quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascuna filiale pochi rispetto a quelli già presenti), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
2. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di acquisire informazioni approssimate ma ragionevolmente indicative sugli andamenti complessivi.
3. L'operazione è eseguita in un momento in cui non ci sono aggiornamenti di alcun genere, con la finalità di individuare tre filiali da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
4. L'operazione è eseguita mentre vengono inseriti alcuni nuovi clienti (per ciascuna filiale pochi rispetto a quelli già presenti), con la finalità di individuare tre filiali da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.
5. L'operazione è eseguita mentre vengono modificati i valori dei punti fedeltà di tutti i clienti (a seguito di una ridefinizione dei criteri di assegnazione dei punti stessi), con la finalità di individuare tre filiali da premiare in una campagna promozionale sulla base dei punti acquisiti dai rispettivi clienti.

Risposte				
1.	2.	3.	4.	5.
RC	RR	RU	S	RR

Basi di dati II — 30 gennaio 2015

Domanda 4 (25%) Sia data una relazione $R(\underline{A}, B, C)$ contenente circa $L = 10.000.000$ ennuple di $r = 20$ byte ciascuna, di cui $a = 4$ per la chiave A, che contiene valori interi quasi consecutivi, da 1 a poco più di 10.000.000. Supporre che i blocchi abbiano dimensione $B = 2$KB, approssimabile come 2.000, che i puntatori ai record abbiano lunghezza $p = 6$; e che i nodi intermedi degli indici possano essere contenuti nei buffer.
Indicare il costo prevedibile per le seguenti operazioni

1. `SELECT * FROM R WHERE A >= 1000 AND A <=3000`
2. `SELECT COUNT(*) FROM R WHERE A >= 1000 AND A <=3000`
3. `SELECT * FROM R WHERE A = 2000`

in ciascuno dei seguenti casi:

(a) indice primario (sparso) su A realizzato con B+-tree;
(b) indice secondario (denso) su A realizzato con B+-tree.

Riportare le risposte nella tabella sottostante, indicando formula e valore numerico (con brevissimo commento, se necessario)

Possibile soluzione

Indichiamo con

- $F = B/r = 100$ il fattore di blocco del file
- $F_I = B/(a+p) = 200$ il fattore bi blocco massimo dell'indice; trattandosi di B+-tree, quello reale F'_I sar‡ minore (assumiamo del 30%) $F'_I = 150$ circa
- $N = 2000$ (o poco meno) il numero di record con valore di A compreso fra 1000 e 3000

(a)1 : poiché il file è ordinato, gli N record da trovare si trovano in $N/F = 20$ blocchi, che sono quindi accessibili (visto che l'indice è sparso), attraverso $\lceil (N/F)/F'_I \rceil = 1$ blocchi dell'indice; il costo è pari all'accesso alla foglia dell'indice (i livelli più alti sono nel buffer, quindi non costano niente) più a quelli dei record: $\lceil (N/F)/F'_I \rceil + N/F =$ circa 20
(b)1 : il file non è ordinato, i record sono sparpagliati, vanno acceduti a uno a uno e quindi servono N accessi più le foglie dell'indice (che è denso) $\lceil N/F'_I \rceil =$ circa 15 quindi trascurabile; costo totale circa $N = 2000$
(a)2 : come a(1) (l'indice è sparso e quindi per contare è necessario accedere ai record)
(b)2 : l'indice è denso, bastano le foglie dell'indice: $\lceil N/F'_I \rceil =$ circa 15
(a)3 : accesso diretto: una foglia dell'indice più un blocco del file, totale 2
(b)3 : come (a)3

Basi di dati II — 30 gennaio 2015

Domanda 5 (25%) 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. Quindi R1 occupa $N_1 = 6$ blocchi e R2 ne occupa $N_2 = 8$.

Relazione R1

50			51			52			53			54			55		
	X01	AA		Y01	DA		Z03	AB		K03	AB		Z03	AB		Z03	AB
	Y42	CA		X42	CC		W05	EF		W07	EF		W08	EF		W09	EF
	W73	CC		W93	CB		X52	HA		X59	HA		X50	HA		X56	HA
	Z55	GC		W54	LB		Y55	EA		Y54	EA		Y51	EA		Y57	EA

Relazione R2

60			61			62			63			64			65			66			67		
	AA	3		BC	4		LB	7		AA	8		AC	3		EA	7		BA	5		EF	6
	DA	7		GB	7		HB	3		EC	2		CB	5		LB	8		BB	4		GA	8

Si supponga di disporre di un buffer di $p = 4$ pagine.
Considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con il metodo nested loop senza utilizzo di indici. Supporre che non serva memorizzare il risultato e che quindi esso possa essere prodotto una ennupla alla volta (approccio "pipelining")

Indicare, nell'ordine, le prime quattro ennuple che vengono prodotte

(X01, AA, 3), (Y01, DA, 7), (W54, LB, 7), (X01, AA, 8)

Indicare gli indirizzi dei blocchi che si trovano nel buffer dopo che sono state prodotte le prime quattro ennuple.

50, 51, 52, 63

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime quattro ennuple del risultato.

50, 51, 52, 60, 61, 62, 63

Indicare il numero complessivo di accessi a memoria secondaria necessari per eseguire il join (indicare formula e numero)

$N_1 + (N_1/(p-1)) \times N_2 = 22$