Basi di dati II, primo modulo — prova breve
22 giugno 2010
Cenni sulle soluzioni

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

Rispondere in modo ordinato su un foglio protocollo diverso da quello utilizzato per le risposte alle domande relative al secondo modulo. Nell'ambito di questa prova, le domande sul primo modulo avranno un peso del 30% e quelle sul secondo del 70%. Qui (come nell'altro foglio con le tracce) sono indicati i pesi relativi.

Domanda 1 (40%) (Rispondere sul foglio protocollo) Rispondere a questa domanda **dopo** avere risposto alla domanda 1 relativa al secondo modulo.
Illustrare brevemente una alternativa di memorizzazione che favorisca interrogazioni volte a conoscere solo codici fiscali (`cf`) dei clienti e costi dei noleggi. Fornire una breve spiegazione, con riferimento alle problematiche di gestione dei dati in memoria secondaria.
Risposta (cenni)
Si può partizionare la tabella utilizzata per i noleggi, separando codice fiscale e costo dagli altri attributi (con una opportuna chiave per eseguire il join quando necessario).

Domanda 2 (60%) (Rispondere sul foglio protocollo) Illustrare brevemente, ma con chiarezza, gli aspetti salienti delle trasformazioni utilizzate per alimentare uno degli schemi dimensionali utilizzati nel progetto a partire dalla base di dati sorgente fornita.

Basi di dati II, primo modulo — prova lunga
22 giugno 2010 — Compito A
Cenni sulle soluzioni

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

Rispondere alle ultime due domande su un foglio protocollo diverso da quello utilizzato per le risposte alle domande relative al secondo modulo. Rispondere su questo foglio alle altre domande.
Nell'ambito di questa prova, le domande sul primo modulo avranno un peso del 55% e quelle sul secondo del 45%. Qui (come nell'altro foglio con le tracce) sono indicati i pesi relativi.

Domanda 1 (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.

In ciascuno dei seguenti casi:

(a) indice primario (sparso) su A realizzato con B+-tree;
(b) indice secondario su A realizzato con B+-tree;
(c) struttura primaria hash su A.

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`

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 sarà 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 al blocco del file (il resto del file è nel buffer, quindi non costa niente) più a quelli dei record: $\lceil (N/F)/F'_I \rceil + N/F =$ circa 20
(b)1 : l'indice è sparso, quindi per N record abbiamo N riferimenti nell'indice, che occupano N/F'_I blocchi; è poi necessario accedere ai record; costo: $N/F'_I + N =$ circa 2000
(c)1 : la struttura hash non è molto utile in questo caso (servono $N = 2000$ accessi diversi di costo poco più che unitario), ma è comunque preferibile utilizzarla rispetto all'accesso sequenziale che costerebbe $L/F = 100.000$
(a)2 : come a(1) (l'indice è sparso e quindi per contare è necessario accedere ai record)
(b)2 : poiché l'indice è denso, l'indice permette di sapere quanti record soddisfano la condizione; costo: $N/F'_I =$ circa 13
(c)2 : come (c)1
(a)3 : accesso diretto: una foglia dell'indice più un blocco del file, totale 2
(b)3 : accesso diretto: una foglia dell'indice più un blocco del file, totale 2
(c)3 : accesso diretto al blocco del file, totale 1

Domanda 2 (20%) Specificare, per ciascuno dei seguenti schedule, la sua appartenenza alle varie classi (scrivere "sì" o "no" nelle caselle della tabella; assumere che gli identificatori delle transazioni corrispondano ai timestamp).

Risposte

	Seriale	VSR	CSR	TS	2PL
$r_2(x)w_2(x)r_1(x)w_1(x)$	`seriale`	`VSR`	`CSR`	`no`	`2PL`
$r_1(x)w_1(x)r_2(x)w_2(x)r_0(y)w_1(y)$	`no`	`VSR`	`CSR`	`TS`	`no`
$r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$	`no`	`VSR`	`CSR`	`TS`	`2PL`
$r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$	`no`	`no`	`no`	`no`	`no`

Domanda 3 (15%) Spiegare perché, scrivendo un programma che accede a due basi di dati diverse utilizzando JDBC, non è possibile garantire il commit a due fasi. Indicare quali funzionalità aggiuntive sono necessarie per i server locali che vogliano realizzare tale servizio e come si può procedere con un sistema che non disponga di tali funzionalità.

Risposta (cenni)

Con JDBC non è possibile implementare sui partecipanti lo stato di ready, che affidi la responsabilità della decisione ad un coordinatore. L'unica soluzione alternativa prevede la disponibilità di "transazioni compensative" che annullino l'effetto delle transazioni effettuate sui singoli nodi.

Domanda 4 (15%) (Rispondere sul foglio protocollo) Rispondere a questa domanda **dopo** avere risposto alla domanda 1 relativa al secondo modulo.

Illustrare brevemente una alternativa di memorizzazione che favorisca interrogazioni volte a conoscere solo codici fiscali (`cf`) dei clienti e costi dei noleggi. Fornire una breve spiegazione, con riferimento alle problematiche di gestione dei dati in memoria secondaria.

Risposta (cenni)

Si può partizionare la tabella utilizzata per i noleggi, separando codice fiscale e costo dagli altri attributi (con una opportuna chiave per eseguire il join quando necessario).

Domanda 5 (25%) (Rispondere sul foglio protocollo) Illustrare brevemente, ma con chiarezza, gli aspetti salienti delle trasformazioni utilizzate per alimentare uno degli schemi dimensionali utilizzati nel progetto a partire dalla base di dati sorgente fornita.

Basi di dati II, primo modulo — prova lunga
22 giugno 2010 — Compito B
Cenni sulle soluzioni

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

Rispondere alle ultime due domande su un foglio protocollo diverso da quello utilizzato per le risposte alle domande relative al secondo modulo. Rispondere su questo foglio alle altre domande.
Nell'ambito di questa prova, le domande sul primo modulo avranno un peso del 55% e quelle sul secondo del 45%. Qui (come nell'altro foglio con le tracce) sono indicati i pesi relativi.

Domanda 1 (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.

In ciascuno dei seguenti casi:

(a) indice primario (sparso) su A realizzato con B+-tree;
(b) indice secondario su A realizzato con B+-tree;
(c) struttura primaria hash su A.

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`

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 sarà 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 al blocco del file (il resto del file è nel buffer, quindi non costa niente) più a quelli dei record: $\lceil (N/F)/F'_I \rceil + N/F =$ circa 20
(b)1 : l'indice è sparso, quindi per N record abbiamo N riferimenti nell'indice, che occupano N/F'_I blocchi; è poi necessario accedere ai record; costo: $N/F'_I + N =$ circa 2000
(c)1 : la struttura hash non è molto utile in questo caso (servono $N = 2000$ accessi diversi di costo poco più che unitario), ma è comunque preferibile utilizzarla rispetto all'accesso sequenziale che costerebbe $L/F = 100.000$
(a)2 : come a(1) (l'indice è sparso e quindi per contare è necessario accedere ai record)
(b)2 : poiché l'indice è denso, l'indice permette di sapere quanti record soddisfano la condizione; costo: $N/F'_I =$ circa 13
(c)2 : come (c)1
(a)3 : accesso diretto: una foglia dell'indice più un blocco del file, totale 2
(b)3 : accesso diretto: una foglia dell'indice più un blocco del file, totale 2
(c)3 : accesso diretto al blocco del file, totale 1

Domanda 2 (20%) Specificare, per ciascuno dei seguenti schedule, la sua appartenenza alle varie classi (scrivere "sì" o "no" nelle caselle della tabella; assumere che gli identificatori delle transazioni corrispondano ai timestamp).

Risposte

	Seriale	VSR	CSR	TS	2PL
$r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$	`no`	`no`	`no`	`no`	`no`
$r_1(x)w_1(x)r_2(x)w_2(x)r_0(y)w_1(y)$	`no`	`VSR`	`CSR`	`TS`	`no`
$r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$	`no`	`VSR`	`CSR`	`TS`	`2PL`
$r_2(x)w_2(x)r_1(x)w_1(x)$	`seriale`	`VSR`	`CSR`	`no`	`2PL`

Domanda 3 (15%) Spiegare perché, scrivendo un programma che accede a due basi di dati diverse utilizzando JDBC, non è possibile garantire il commit a due fasi. Indicare quali funzionalità aggiuntive sono necessarie per i server locali che vogliano realizzare tale servizio e come si può procedere con un sistema che non disponga di tali funzionalità.

Risposta (cenni)

Con JDBC non è possibile implementare sui partecipanti lo stato di ready, che affidi la responsabilità della decisione ad un coordinatore. L'unica soluzione alternativa prevede la disponibilità di "transazioni compensative" che annullino l'effetto delle transazioni effettuate sui singoli nodi.

Domanda 4 (15%) (Rispondere sul foglio protocollo) Rispondere a questa domanda **dopo** avere risposto alla domanda 1 relativa al secondo modulo.

Illustrare brevemente una alternativa di memorizzazione che favorisca interrogazioni volte a conoscere solo codici fiscali (`cf`) dei clienti e costi dei noleggi. Fornire una breve spiegazione, con riferimento alle problematiche di gestione dei dati in memoria secondaria.

Risposta (cenni)

Si può partizionare la tabella utilizzata per i noleggi, separando codice fiscale e costo dagli altri attributi (con una opportuna chiave per eseguire il join quando necessario).

Domanda 5 (25%) (Rispondere sul foglio protocollo) Illustrare brevemente, ma con chiarezza, gli aspetti salienti delle trasformazioni utilizzate per alimentare uno degli schemi dimensionali utilizzati nel progetto a partire dalla base di dati sorgente fornita.

Tecnologia delle basi di dati
22 giugno 2010
Cenni sulle soluzioni

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

Domanda 1 (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.

In ciascuno dei seguenti casi:

(a) indice primario (sparso) su A realizzato con B+-tree;
(b) indice secondario su A realizzato con B+-tree;
(c) struttura primaria hash su A.

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`

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 sarà 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 al blocco del file (il resto del file è nel buffer, quindi non costa niente) più a quelli dei record: $\lceil (N/F)/F'_I \rceil + N/F =$ circa 20

(b)1 : l'indice è sparso, quindi per N record abbiamo N riferimenti nell'indice, che occupano N/F'_I blocchi; è poi necessario accedere ai record; costo: $N/F'_I + N =$ circa 2000

(c)1 : la struttura hash non è molto utile in questo caso (servono $N = 2000$ accessi diversi di costo poco più che unitario), ma è comunque preferibile utilizzarla rispetto all'accesso sequenziale che costerebbe $L/F = 100.000$

(a)2 : come a(1) (l'indice è sparso e quindi per contare è necessario accedere ai record)

(b)2 : poiché l'indice è denso, l'indice permette di sapere quanti record soddisfano la condizione; costo: $N/F'_I =$ circa 13

(c)2 : come (c)1

(a)3 : accesso diretto: una foglia dell'indice più un blocco del file, totale 2

(b)3 : accesso diretto: una foglia dell'indice più un blocco del file, totale 2

(c)3 : accesso diretto al blocco del file, totale 1

Domanda 2 (20%) Specificare, per ciascuno dei seguenti schedule, la sua appartenenza alle varie classi (scrivere "sì" o "no" nelle caselle della tabella; assumere che gli identificatori delle transazioni corrispondano ai timestamp).

Risposte

	Seriale	VSR	CSR	TS	2PL
$r_1(y)r_2(z)r_2(y)w_2(y)w_2(z)r_1(z)$	no	no	no	no	no
$r_1(x)w_1(x)r_2(x)w_2(x)r_0(y)w_1(y)$	no	VSR	CSR	TS	no
$r_1(x)r_2(z)w_2(z)w_1(x)r_2(x)w_2(x)$	no	VSR	CSR	TS	2PL
$r_2(x)w_2(x)r_1(x)w_1(x)$	seriale	VSR	CSR	no	2PL

Domanda 3 (15%) Spiegare perché, scrivendo un programma che accede a due basi di dati diverse utilizzando JDBC, non è possibile garantire il commit a due fasi. Indicare quali funzionalità aggiuntive sono necessarie per i server locali che vogliano realizzare tale servizio e come si può procedere con un sistema che non disponga di tali funzionalità.

Risposta (cenni)

Con JDBC non è possibile implementare sui partecipanti lo stato di ready, che affidi la responsabilità della decisione ad un coordinatore. L'unica soluzione alternativa prevede la disponibilità di "transazioni compensative" che annullino l'effetto delle transazioni effettuate sui singoli nodi.

Domanda 4 (15%) Commentare brevemente la seguente affermazione: "le tecniche per il controllo di concorrenza basate su 2PL e su timestamp si basano su condizioni sufficienti ma non necessarie per la view-serializzabilità." In particolare, chiarire se essa è vera o falsa, spiegare perché e motivare le ragioni pratiche per le quali si utilizzano tali tecniche anziché la view-serializzabilità stessa.

Risposta (cenni)

La view-serializzabilità è inutilizzabile in pratica per vari motivi, a cominciare dal costo computazionale (e dalel ipotesi restrittive). 2PL e TS sono condizioni sufficienti e non necessarie (vedi libro). Sono realizzabili in pratica in modo efficiente, al prezzo della rinuncia ad alcuni schedule che sarebbero VS.

Domanda 5 (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:

KStudente	KCorso	KData	Voto	...
301	201	405	25	...
301	202	406	28	...
302	201	407	30	...
302	203	407	22	...
...	...	...	...	...

KCorso	Titolo	Crediti	...
201	Fisica I	6	...
202	Chimica	9	...
203	Geometria	6	...
...	...	...	...

Supporre che si presentino le seguenti esigenze di modifica:

- i corsi cambiano nome nel tempo: per esempio, il corso nella prima ennupla potrebbe ad un certo punto cambiare nome da "Fisica I" in "Meccanica"; 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;
- 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).

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).

Possibile soluzione

La tabella dei fatti può essere estesa aggiungendo una dimensione supplementare, che dipende da Corso e Data e può quindi essere facilmente aggiunta anche ad una tabella dei fatti esistente (i dati, come detto, sono disponibili):

KStudente	KCorso	KData	KDocente	Voto	...
301	201	405	701	25	...
301	202	406	701	28	...
302	201	407	702	30	...
302	203	407	703	22	...
...	...	...	...	...	...

Per la dimensione può essere utile la tecnica della"slowly changing dimension," con un nuovo elemento (quindi una ennupla nella relazione) per ogni modifica. Viste le specifiche, qui potrebbe essere utile introdurre un codice, che non cambia nel tempo, e avere due attirbuti per il nome, con quello attuale e quello "dell'epoca."

KCorso	Codice	Titolo	TitoloAttuale	Crediti	...
201	FIS01	Fisica I	Meccanica	6	...
202	CHIM01	Chimica	Chimica	9	...
203	GEOM01	Geometria	Geometria	6	...
...	...	...	...	...	...
209	FIS01	Meccanica	Meccanica	6	...