Basi di dati II

Esame — 18 luglio 2019

Cenni sulle soluzioni

Tempo a disposizione: un'ora per la prova breve, due ore e trenta minuti per la prova completa.

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

Basi di dati II — 18 luglio 2019

Domanda 1 (15% per la prova completa, 40% per la prova breve)
Si consideri uno schema dimensionale con la seguente tabella dei fatti relativa alle carriere degli studenti presso una università:

- FattiEsami(KMateria, KSessione, KCorsoDiStudio, NumeroEsami, Media)

Con riferimento a tale schema, si supponga che

- complessivamente la tabella contenga 50.000 ennuple
- vi siano 1000 materie e 10 sessioni e per ogni sessione e per ogni materia ci sia almeno un esame verbalizzato
- vi siano 50 corsi di studio
- ogni materia abbia studenti di mediamente 5 corsi di studio (e che in ogni sessione e per ogni materia ci sia almeno uno studente per ciascuno di tali corsi di studio)
- le operazioni più frequenti siano quelle che producono:
  1. numero esami per una materia e ciascuna sessione (ad esempio, Basi di dati II, in ciascuna delle 10 sessioni; quindi il risultato contiene 10 ennuple) con frequenza $f_1 = 10$
  2. numero esami complessivo per ciascuna materia (quindi il risultato contiene 1000 ennuple) con frequenza $f_2 = 1000$
  3. numero esami in una sessione (ad esempio, la sessione estiva del 2017) per ciascun corso di studio (quindi vanno prodotte 50 ennuple, una per corso di studio) con frequenza $f_3 = 10$
- sia possibile realizzare una sola vista materializzata
- nelle selezioni, il costo sia pari al numero di ennuple estratte dalla relazione (prima dell'eventuale aggregazione)

Scegliere, fra le tre viste sostanzialmente corrispondenti alle tre interrogazioni, quella che si ritiene supporti meglio il carico applicativo. Indicare lo schema di ciascuna vista.

Indichiamo con $c = 50$ il numero di corsi di studio, $s = 10$ il numero di sessioni, $i = 1000$ il numero di materie e $N = 50.000$ il numero di ennuple nella tabella dei fatti.
Le tre viste:

1. EsamiMateriaSessione(KMateria, KSessione, NumeroEsami, Media), cardinalità $i \times s = 1000 \times 10 = 10.000$, supporta l'interrogazione 1 e 2
2. EsamiMateria(KMateria, NumeroEsami,Media), cardinalità $i = 1000$, supporta l'interrogazione 2
3. EsamiSessioneCdS(KSessione, KCorsoDiStudio, NumeroEsami, Media), cardinalità $s \times c = 10 \times 50 = 500$, supporta l'interrogazione 3

Notare che le operazioni 1 e 3 chiedono un sottoinsieme delle ennuple della vista (rispettivamente, quelle relative ad una materia e quelle relative ad una sessione)

Costo unitario per ogni operazione con ciascuna vista e costo complessivo:

	con vista 1	con vista 2	con vista 3
c_1	Si deve accedere alle ennuple della vista relative a una materia, per tutte le sessioni; costo unitario: $s = 10$	Si deve usare la relazione base, accedendo a tutte le ennuple di una materia: $N/i = 50.000/1000 = 50$	Si deve usare la relazione base, accedendo a tutte le ennuple di una materia: $N/i = 50.000/1000 = 50$
c_2	Si deve accedere a tutte le ennuple della vista 1: $i \times s = 1000 \times 10 = 10.000$	Si deve accedere a tutte le ennuple della vista 2: $i = 1000$	Si deve accedere a tutte le ennuple della relazione base: $N = 50.000$
c_3	Si deve usare la relazione base, accedendo a tutte le ennuple di una sessione: $N/s = 50.000/10 = 5.000$	Si deve usare la relazione base, accedendo a tutte le ennuple di una sessione: $N/s = 50.000/10 = 5.000$	Si deve accedere alle ennuple della vista relative alla sessione, una per corso di studio: $c = 50$
$\Sigma_i\ c_i \times f_i$	$10 \times 10 + 10.000 \times 1000 + 5.000 \times 10$	$50 \times 10 + 1000 \times 1000 + 5.000 \times 10$	$50 \times 10 + 50.000 \times 1000 + 50 \times 10$

Conviene quindi scegliere la vista 2

Basi di dati II — 18 luglio 2019

Domanda 2 (15% per la prova completa, 60% per la prova breve)
Si consideri la seguente porzione dello schema dell'archivio delle carriere degli studenti di una anagrafe ministeriale:

- Studenti(CodiceFiscale, Cognome, Nome, DataNascita, TipoMaturità)
- Iscrizioni(CodiceFiscale, AnnoAccademico, CorsoDiStudio, AnnoDiCorso, AnnodiImmatricolazione)
- CorsiDiStudio(CodiceCdS, Titolo, Livello, Classe, Università)
- Lauree(CodiceFiscale, CodiceCdS, Data, Voto)

Progettare uno schema dimensionale che permetta di rispondere, fra le altre, alle seguenti interrogazioni:

- calcolare il numero di studenti (con la relativa media dei voti) che si sono laureati in un certo corso di studio (inteso come corso di studio presso una università) in un certo anno accademico (si supponga che, per la data di laurea, l'unico dettaglio rilevante sia l'anno accademico e che esista un modo univoco per associare un anno accademico ad una data di laurea)
- calcolare il numero di laureati per una classe di corsi studio, distinto per tipo di maturità e per numero di anni impiegati per conseguire il titolo (ad esempio, 2, 3, 4, 5, 6, più di 6)
- calcolare il numero di laureati per una classe di corsi studio, distinto per "età alla laurea" (ad esempio, 21, 22, 23, ...26, più di 26)

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.

Mostrare lo schema dimensionale, specificando la grana scelta.

Grana: insieme di studenti laureati in corso di studio, in un anno accademico, con un tipo di maturità, che ha impiegato uno stesso numero di anni, con una stessa età.

Tabella dei fatti:
FattiLauree(KCdS, KAnnoAccademico, KTipoMaturità, KDurataStudi, KEtàAllaLaurea,NumStudenti, VotoMedio)

Dimensioni:
CorsoDiStudio(KCdS, CodiceCdS, NomeCdS,Università,..., Livello, CodiceClasse, NomeClasse...)
AnnoAccademico(KAnnoAccademico, ...)
TipoMaturità(KTipoMaturità, ...)
DurataStudi(KDurataStudi, ...)
EtàAllaLaurea(KEtàAllaLaurea, ...)

Basi di dati II — 18 luglio 2019

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

- join delle relazioni STUDENTI, ISCRIZIONI, e LAUREE (peraltro, ISCRIZIONI serve solo per l'anno di immatricolazione e va gestita con attenzione per evitare duplicazioni, è opportuna una proiezione che elimini l'anno acacdemico; si può fare la proiezione dopo il join)
- aggiunta di attributi con l'età dello studente alla laurea, l'anno accademico di laurea e la durata degli studi (in modo coerente con le scelte fatte nelle dimensioni corrispondenti)
- proiezione sugli attributi rilevanti: CodiceCdS, AnnoAccademicoDiLaurea, TipoMaturità, DurataStudi, EtàAllaLaurea, Voto
- aggregazione sui primi cinque attributi, con conteggio del numeri di laureati e calcolo del voto medio
- sostituzione degli identificatori o dei valori (a seconda dei casi) con le chiavi surrogate delle dimensioni

Basi di dati II — 18 luglio 2019

Domanda 3 (20%) Si considerino un sistema con blocchi di dimensione $B = 4000$ byte e una relazione *R(Matricola, Codice, Cognome, ...)* di cardinalità pari circa a $L = 400.000$, con ennuple di $e = 80$ byte, con due chiavi, *Matricola* e *Codice* (cioè il valore di ciascuna di esse, da solo, identifica univocamente una ennupla). Supporre che il sistema offra

- strutture primarie disordinate oppure strutture hash
- 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 *Codice* e su *Matricola*) con frequenza oraria $f_1 = 100$;
2. ricerca di una ennupla sulla base del valore completo di *Matricola*, frequenza oraria $f_2 = 10$
3. ricerca di ennuple sulla base del codice *Codice*, 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 $f_4 = 1000$; supporre che il valore parziale sia poco selettivo e porti alla identificazione, in media, di $s = 80$ ennuple.

Progettare l'organizzazione fisica della relazione, individuando la struttura primaria (disordinata o hash) e gli eventuali indici (da nessuno a tre) Ragionare in termini di numero di accessi a memoria secondaria, assumendo che (i) il costo dell'accesso hash sia costante e pari a 1; (ii) gli indici abbiano profondità $p = 4$, (iii) il buffer disponibile permetta di mantenere stabilmente in memoria due livelli di indice, (iv) 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.	Hash su Matricola e indici su Codice e Cognome	Indici su Matricola, Codice e Cognome	Hash su Matricola e indice su Codice
Costo Op. 1	(2) + (p-2+1)+ (p-2-1) = ca. 8: lettura e scrittura con hash (2), visita (p-2) e aggiornamento (1) per entrambi gli indici	3×(p-2+1) + 2= ca. 11 lettura e scrittura di tre indici e lettura e scrittura del blocco con il record	(2) + (p-2+1) = ca. 5: lettura e scrittura con hash (2), visita (p-2) e aggiornamento (1) per l'indice
Costo Op. 2	1	(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	p-2+2 = 4
Costo Op. 4	p-2+80 = ca. 80 i sono in generale in blocchi diversi	p-2+80 = ca. 80	$(L \times e)/B$ = 8000 scansione sequenziale
Tot	8×100 + 1×10 + 4×10 + 80×1000 = ca 80.000	11×100 + 3×10 + 4×10 + 80×1000 = ca 80.000	5×100 + 3×10 + 4×10 + 8000×1000 = ca 8.000.000

Basi di dati II — 18 luglio 2019

Domanda 4 (10%)
Considerare ancora il caso illustrato nella domanda precedente, ma con riferimento ad una fase in cui le frequenze siano completamente diverse:

1. $f_1 = 10.000$
2. $f_2 = 10$
3. $f_3 = 10$
4. $f_4 = 1$

Indicare quale soluzione si sceglierebbe in questo caso

	Alternativa 1	Alternativa 2	Alternativa 3 (eventuale)
Descr. strutt.	Hash su Matricola e indici su Codice e Cognome	Hash su Matricola e indice su Codice	
Costo Op. 1	(2) + (p-2+1)+ (p-2-1) = ca. 8: lettura e scrittura con hash (2), visita (p-2) e aggiornamento (1) per entrambi gli indici	(2) + (p-2+1) = ca. 5: lettura e scrittura con hash (2), visita (p-2) e aggiornamento (1) per l'indice	
Costo Op. 2	1	(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+80 = ca. 80 i sono in generale in blocchi diversi	$(L \times e)/B$ = 8000 scansione sequenziale	
Tot	8×10.000 + 1×10 + 4×10 + 80×1 = ca 80.000	5×10.000 + 3×10 + 4×10 + 1×8000 = ca 60.000	

Basi di dati II — 18 luglio 2019

Domanda 5 (10%)
Considerare una relazione definita con il seguente comando:

```
create table conti (numero integer primary key, saldo integer not null);
```

Considerare il seguente scenario in cui due client 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. Supporre che, in caso di abort, le transazioni non vengano rilanciate

`start transaction`
`  isolation level serializable;`
`select * from conti where numero = 3;`	

	`start transaction`
	`  isolation level serializable;`
	`select * from conti where numero = 3;`

`update conti` `  set saldo = 20 where numero = 3;`	
	`update conti` `  set saldo = 10 where numero = 3;`
`commit;`	
	`commit;`

Considerare uno scheduler con controllo di concorrenza basato su **Multversioni** (come in Postgres). Mostrare il comportamento dello scheduler,

Sarebbe stato utile riportare il dettaglio degli eventi.
La seconda transazione va prima in attesa, perché la prima detiene il lock sul conto numero 3 e poi viene abortita, perché vorrebbe scrivere su un dato modificato da un'altra transazione dopo l'inizio della transazione stessa.
Messaggio `ERROR: could not serialize access due to concurrent update`

Basi di dati II — 18 luglio 2019

Domanda 6 (10%)
Considerare ancora la relazione considerata nella domanda precedetnte:

```
create table conti (numero integer primary key, saldo integer not null);
```

e il seguente scenario

`start transaction`
`  isolation level repeatable read;`
`insert into conti values (8, 1000) ;`
 	
`commit;`
	
	`start transaction`
	`  isolation level repeatable read;`
	`insert into conti values (8, 3000) ;`
	`commit`

Considerare ancora uno scheduler con controllo di concorrenza basato su **Multversioni** (come in Postgres). Mostrare il comportamento dello scheduler,

La seconda transazione va prima in attesa, perché la prima detiene il lock sul conto numero 8 e poi viene abortita, per la violazione del vincolo di chiave.
Messaggio `duplicate key value violates unique constraint "conti_pkey"`

Basi di dati II — 18 luglio 2019

Domanda 7 (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. Quindi R1 occupa $B_1 = 6$ blocchi e R2 ne occupa $B_2 = 8$.

Relazione R1

20			21			22			23			24			25		
	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

40			41			42			43			44			45			46			47		
	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 $p = 3$ 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, una notazione del tipo 'pin(30)' e 'unpin(30),' tutte le operazioni di pin (o fix) e unpin necessarie per eseguire l'intero join.

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

$B_1 + (B_1/(p-1)) \times B_2 = 30$

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.

20, 21, 43