Basi di dati II

Esame — 19 settembre 2019

Tempo a disposizione: due ore.

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

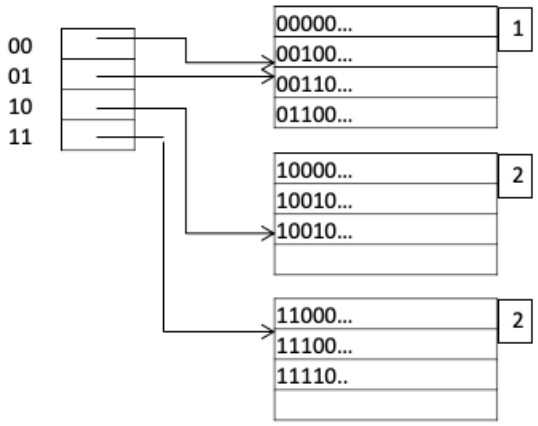
Basi di dati II — 19 settembre 2019

Domanda 1 (15%)
Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** nelle varie caselle, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (generabile da uno scheduler basato su 2PL stretto), MV (generabile da uno scheduler multiversion con controllo di serializzabilità: "a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began"). Negli schedule, s_i indica l'inizio della transazione i e c_i il suo commit.

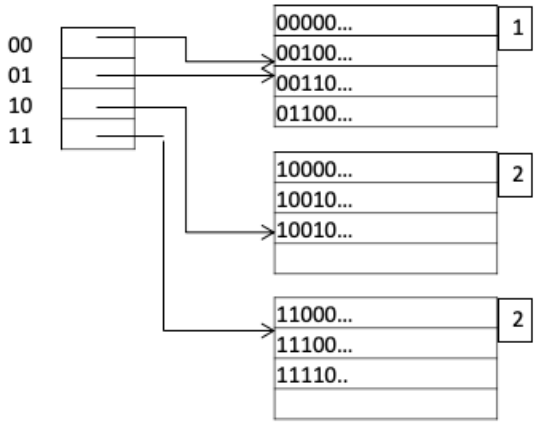
	S	CSR	S2PL	MV
$s_2, r_2(x), w_2(x), s_1, c_2, r_1(x), w_1(x), c_1$				
$s_1, s_2, r_1(x), r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$				
$s_1, s_2, r_1(x), r_2(x), w_1(x), c_1, w_2(x), c_2$				
$s_2, r_2(x), s_1, w_2(x), r_1(x), c_2, w_1(x), c_1$				

Basi di dati II — 19 settembre 2019

Domanda 2 (20%) Considerare la figura qui sotto, che schematizza una struttura con hashing estendibile, composta dalla directory, a sinistra, e dai blocchi del file, a destra, in cui sono mostrate solo le parti iniziali dei valori hash associati alle chiavi. Nel riquadro a destra, mostrare come si modifica la struttura se viene inserito un record il cui valore della funzione di hash inizia con 00111.

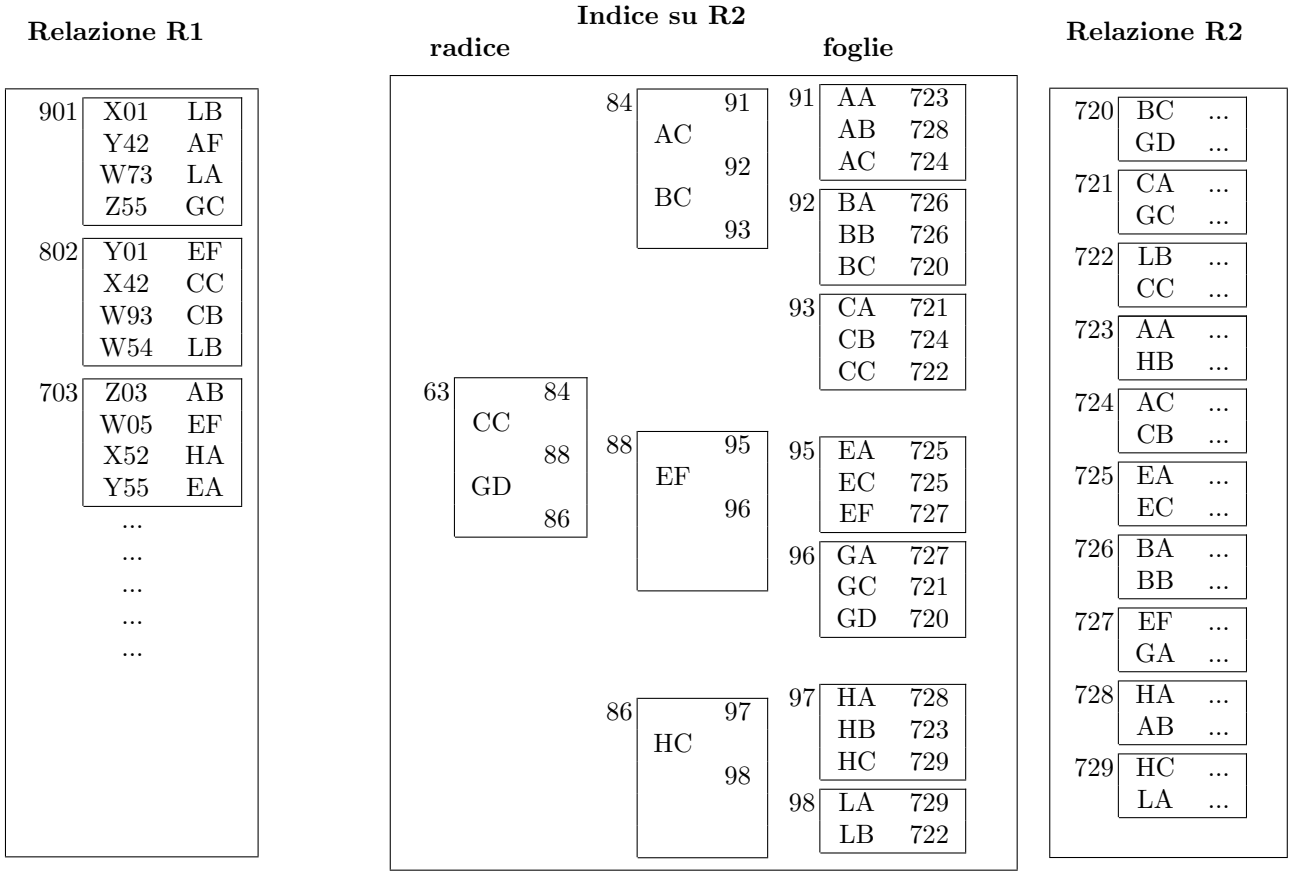


Mostrare poi come si modifica la stessa struttura, se, invece dell'inserimento sopra discusso, si hanno due inserimenti, di record con valori della funzione di hash che iniziano rispettivamente con 11001 e 11111.



Basi di dati II — 19 settembre 2019

Domanda 3 (20%)
Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **otto** 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 un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime quattro ennuple del risultato.

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime quattro ennuple del risultato.

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la quarta ennupla.

Basi di dati II — 19 settembre 2019

Domanda 4 (15%)
Considerare un sistema distribuito con tre nodi N_1, N_2 e N_3, che eseguono due transazioni T_x e T_y che coinvolgono i tre nodi, in modo diverso. Per la prima transazione N_1 è il coordinatore, mentre per la seconda il coordinatore è N_2. I due coordinatori inviano, come riportato nello schema sottostante, le richieste di `prepare`. Il nodo N_3 va in crash subito dopo aver risposto alla prima richiesta (senza avere il tempo di ricevere il messaggio di `commit`) e prima di ricevere la seconda. Poi va in crash anche il nodo N_1. Indicare, nello schema sottostante, una possibile sequenza di scritture sui log e invio di messaggi (che includa anche i passi sopra illustrati), supponendo che entrambi i nodi siano ripristinati abbastanza presto (ma che vengano persi alcuni messaggi di risposta, ad esempio inviati a seguito di una decisione). Per i messaggi si usi la notazione *tipo(transaz)→destinatari* (come nell'esempio: `prepare(`T_x`)→`N_2`,`N_3). Supporre che nel log del coordinatore si scrivano solo i record di `prepare`, `commit` e `complete`, con i messaggi gestiti invece in memoria. Indicare ragionevoli istanti per i timeout, che permettano di concludere il protocollo per entrambe le transazioni.

Nodo N_1		Nodo N_2		Nodo N_3	
Log	Messaggi	Log	Messaggi	Log	Messaggi
`prep(`T_x`,`N_2`,`N_3`)`	`prep(`T_x`)→`N_2`,`N_3				
				crash	
		`prep(`T_y`,`N_1`,`N_3`)`			
			`prep(`T_y`)→`N_1`,`N_3		
crash					
restart					
				restart	

Basi di dati II — 19 settembre 2019

Domanda 5 (30%) Si consideri la seguente base di dati, relativa alle ricette acquisite da un insieme di farmacie:

- Ricette(Numero, CodFarmacia, CFPaziente, Data)
- Farmacie(CodFarmacia, Nome, CodIndirizzo)
- ElementiRicetta(NumeroRicetta, CodFarmaco, Quantità)
- Farmaci(Codice, Descrizione, CodMolecola, CodCasa, Prezzo, Fascia)
- Molecole(CodMolecola, Descrizione)
- Pazienti(CF, Cognome, Nome, DataNascita, CodIndirizzo)
- CaseFarmaceutiche(CodCasa, Nome)
- ASL(Codice, Nome)
- Territorio(CodIndirizzo,Via, NumeroCivico, Comune, ASL)

Ci sono dati che cambiano nel tempo fra cui prezzi e fasce ('A', 'B' o 'C') dei farmaci e indirizzi dei pazienti. Costruire, in tale contesto, uno schema a stella che permetta di analizzare le prescrizioni (quantità e prezzi complessivi) rispetto a

- data (dimensione standard i cui dettagli possono essere omessi);
- farmaci, con le loro proprietà (molecola e casa farmaceutica);
- prescrizione di farmaci nella stessa ricetta
- ASL di residenza e fascia d'età (ad esempio, 0-3,4-17, 18-30, ...; ma potrebbero variare) dei pazienti;
- ASL della farmacia

Supporre che, per ovvie ragioni di privacy, non possano essere riportati dati che permettano di risalire alle identità dei pazienti (CF, cognome, nome, data di nascita e indirizzo) **Indicare esplicitamente la grana dei fatti.**

Grana dei fatti:

Schema dimensionale:

Basi di dati II — 19 settembre 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

Basi di dati II

Esame — 19 settembre 2019

Cenni sulle soluzioni

Tempo a disposizione: due ore.

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

Basi di dati II — 19 settembre 2019

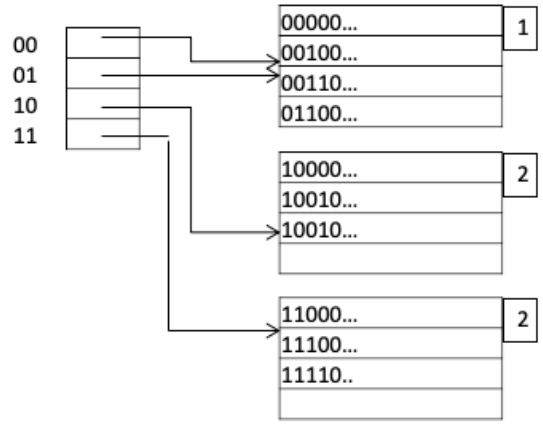
Domanda 1 (15%)
Per ciascuno degli schedule sotto riportati, indicare, scrivendo **sì** o **no** nelle varie caselle, a quali classi appartiene: S (seriale, rispetto a letture e scritture, ignorare commit e abort), CSR (conflict-serializzabile), S2PL (generabile da uno scheduler basato su 2PL stretto), MV (generabile da uno scheduler multiversion con controllo di serializzabilità: "a serializable transaction cannot modify or lock rows changed by other transactions after the serializable transaction began"). Negli schedule, s_i indica l'inizio della transazione i e c_i il suo commit.

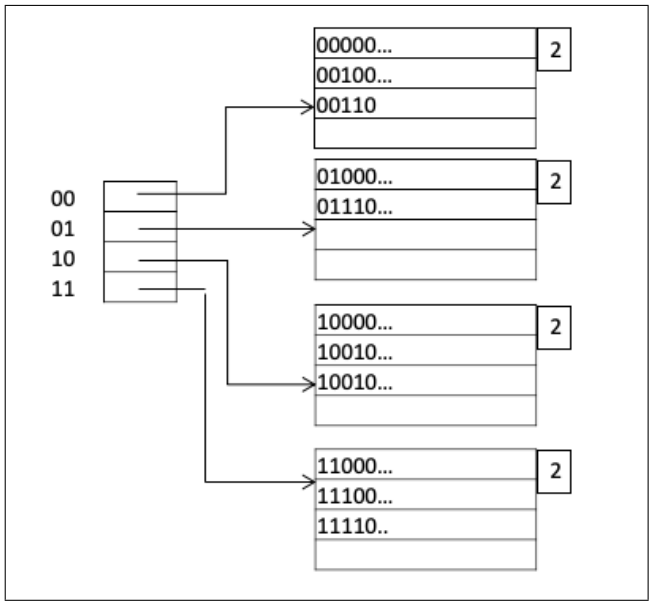
	S	CSR	S2PL	MV
$s_2, r_2(x), w_2(x), s_1, c_2, r_1(x), w_1(x), c_1$	sì	sì	sì	no (*)
$s_1, s_2, r_1(x), r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$	no	no	no	sì
$s_1, s_2, r_1(x), r_2(x), w_1(x), c_1, w_2(x), c_2$	no	no	no	no
$s_2, r_2(x), s_1, w_2(x), r_1(x), c_2, w_1(x), c_1$	sì	sì	no	no

(*) Postgres considera come inizio della transazione la prima operazione e non la "start", perciò anche la risposta "sí" può essere considerata corretta

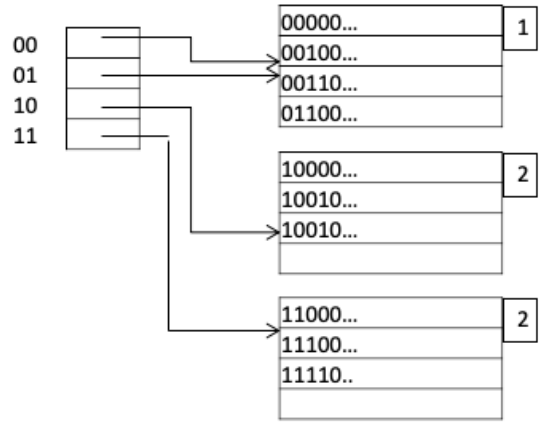
Basi di dati II — 19 settembre 2019

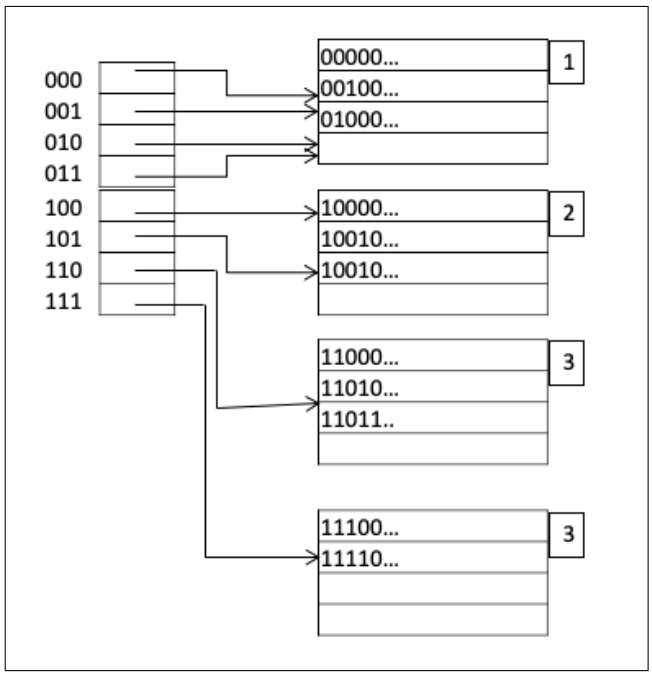
Domanda 2 (20%) Considerare la figura qui sotto, che schematizza una struttura con hashing estendibile, composta dalla directory, a sinistra, e dai blocchi del file, a destra, in cui sono mostrate solo le parti iniziali dei valori hash associati alle chiavi. Nel riquadro a destra, mostrare come si modifica la struttura se viene inserito un record il cui valore della funzione di hash inizia con 00111.





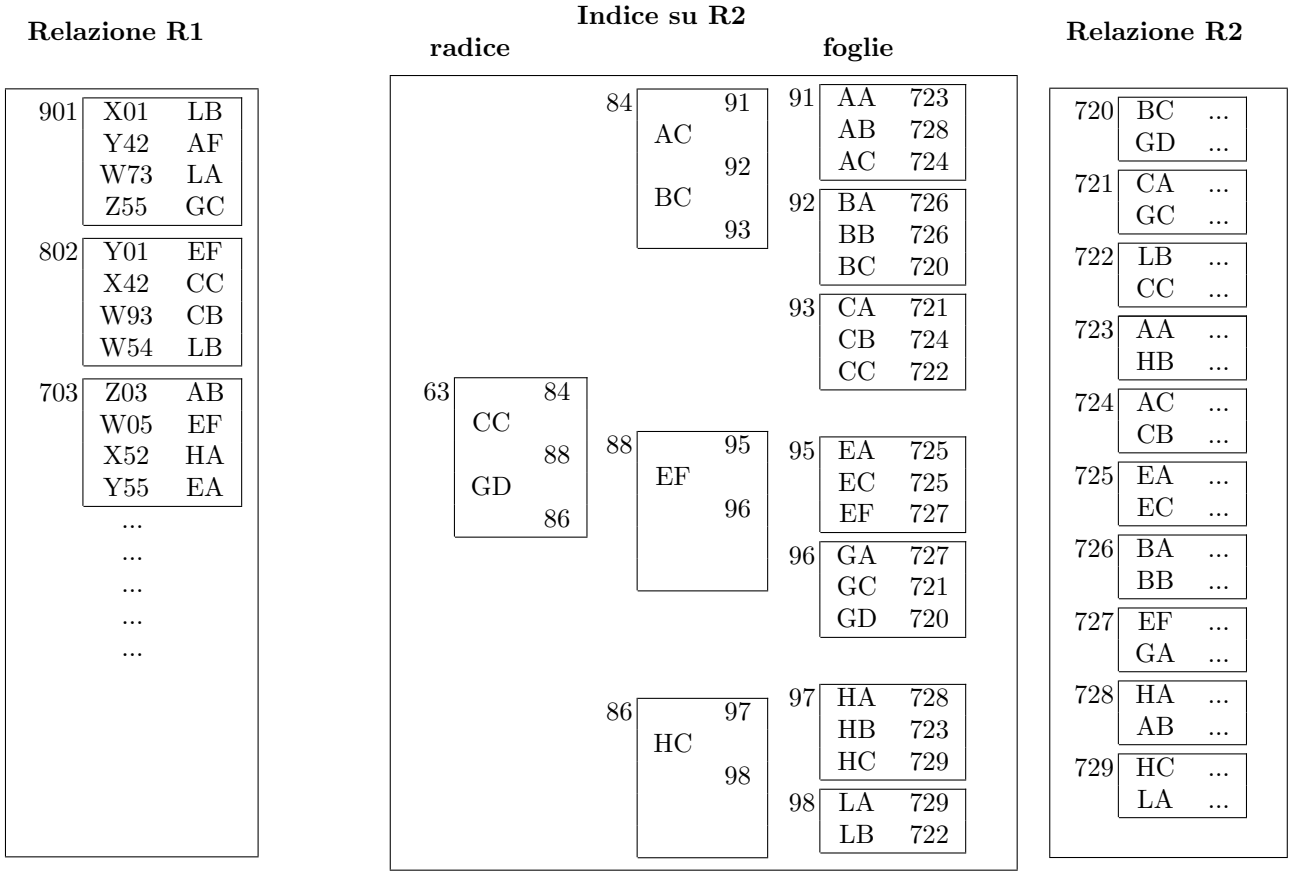
Mostrare poi come si modifica la stessa struttura, se, invece dell'inserimento sopra discusso, si hanno due inserimenti, di record con valori della funzione di hash che iniziano rispettivamente con 11001 e 11111.





Basi di dati II — 19 settembre 2019

Domanda 3 (20%)
Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **otto** 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 un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime quattro ennuple del risultato.
901, 63, 86, 98, 722, — produce la ennupla (X01, LB, ...)
63, 84, 92, — cerca AF in R2
63, 86, 98, 729, — produce la ennupla (W73, LA, ...)
63, 88, 96, 721, — produce la ennupla (Z55, GC, ...)
802, 63, 88, 95, 727 — produce la ennupla (Y01, EF, ...)

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime quattro ennuple del risultato.
901, 63, 86, 98, 722,
84, 92,
729,
88 (rimpiazza 722), 96 (rimpiazza 84), 721(rimpiazza 92)
802 (rimpiazza 86), 95 (rimpiazza 98), 727 (rimpiazza 729)
In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la quarta ennupla.
901, 63, 88, 96, 721, 802 , 95, 727

Basi di dati II — 19 settembre 2019

Domanda 4 (15%)
Considerare un sistema distribuito con tre nodi N_1, N_2 e N_3, che eseguono due transazioni T_x e T_y che coinvolgono i tre nodi, in modo diverso. Per la prima transazione N_1 è il coordinatore, mentre per la seconda il coordinatore è N_2. I due coordinatori inviano, come riportato nello schema sottostante, le richieste di `prepare`. Il nodo N_3 va in crash subito dopo aver risposto alla prima richiesta (senza avere il tempo di ricevere il messaggio di `commit`) e prima di ricevere la seconda. Poi va in crash anche il nodo N_1. Indicare, nello schema sottostante, una possibile sequenza di scritture sui log e invio di messaggi (che includa anche i passi sopra illustrati), supponendo che entrambi i nodi siano ripristinati abbastanza presto (ma che vengano persi alcuni messaggi di risposta, ad esempio inviati a seguito di una decisione). Per i messaggi si usi la notazione *tipo(transaz)→destinatari* (come nell'esempio: `prepare(`T_x`)→`N_2`,`N_3). Supporre che nel log del coordinatore si scrivano solo i record di `prepare`, `commit` e `complete`, con i messaggi gestiti invece in memoria. Indicare ragionevoli istanti per i timeout, che permettano di concludere il protocollo per entrambe le transazioni.

Nodo N_1		Nodo N_2		Nodo N_3	
Log	Messaggi	Log	Messaggi	Log	Messaggi
`prep(`T_x`,`N_2`,`N_3`)`					
	`prep(`T_x`)→`N_2`,`N_3				
		`ready(`T_x`)`		`ready(`T_x`)`	
			`ready(`T_x`)→`N_1		`ready(`T_x`)→`N_1
				crash	
`commit(`T_x`)`					
	`commit(`T_x`)→`N_2`,`N_3				
		`commit(`T_x`)`			
			`ack(`T_x`)→`N_1		
		`prep(`T_y`,`N_1`,`N_3`)`			
			`prep(`T_y`)→`N_1`,`N_3		
`ready(`T_y`)`					
	`ready(`T_y`)→`N_2				
	crash				
		`abort(`T_y`)`			
			`abort(`T_y`)→`N_1`,`N_3		
	restart				
	`commit(`T_x`)→`N_2`,`N_3				
			`ack(`T_x`)→`N_1		
			`abort(`T_y`)→`N_1`,`N_3		
`abort(`T_y`)`					
	`ack(`T_y`)→`N_2				
				restart	
			`abort(`T_y`)→`N_3		
				`abort(`T_y`)`	
					`ack(`T_y`)→`N_2
		`complete(`T_y`)`			
	`commit(`T_x`)→`N_3				
				`commit(`T_x`)`	
					`ack(`T_x`)→`N_1
`complete(`T_x`)`					

Basi di dati II — 19 settembre 2019

Domanda 5 (30%) Si consideri la seguente base di dati, relativa alle ricette acquisite da un insieme di farmacie:

- Ricette(Numero, CodFarmacia, CFPaziente, Data)
- Farmacie(CodFarmacia, Nome, CodIndirizzo)
- ElementiRicetta(NumeroRicetta, CodFarmaco, Quantità)
- Farmaci(Codice, Descrizione, CodMolecola, CodCasa, Prezzo, Fascia)
- Molecole(CodMolecola, Descrizione)
- Pazienti(CF, Cognome, Nome, DataNascita, CodIndirizzo)
- CaseFarmaceutiche(CodCasa, Nome)
- ASL(Codice, Nome)
- Territorio(CodIndirizzo,Via, NumeroCivico, Comune, ASL)

Ci sono dati che cambiano nel tempo fra cui prezzi e fasce ('A', 'B' o 'C') dei farmaci e indirizzi dei pazienti. Costruire, in tale contesto, uno schema a stella che permetta di analizzare le prescrizioni (quantità e prezzi complessivi) rispetto a

- data (dimensione standard i cui dettagli possono essere omessi);
- farmaci, con le loro proprietà (molecola e casa farmaceutica);
- prescrizione di farmaci nella stessa ricetta
- ASL di residenza e fascia d'età (ad esempio, 0-3,4-17, 18-30, ...; ma potrebbero variare) dei pazienti;
- ASL della farmacia

Supporre che, per ovvie ragioni di privacy, non possano essere riportati dati che permettano di risalire alle identità dei pazienti (CF, cognome, nome, data di nascita e indirizzo) **Indicare esplicitamente la grana dei fatti.**

Grana dei fatti: la grana scelta è "singole prescrizioni"

Schema dimensionale:

- FattiPrescrizioni(KData, KVersioneFarmaco, KRicetta, KASLfarmacia, KASLpaziente, KEtà, Quantità, Importo)
- DimFarmaci(KVersioneFarmaco, Codice, Descrizione, CodMolecola, DescrizioneMolecola, CodCasa, NomeCasa, Fascia)
- DimASL(KASL, CodiceASL, NomeASL)
- DimEtà(KEtà, Età, Fascia ...)
- DimData(KData, ...)

Commenti:

- sono indicate chiavi ad hoc per le dimensioni
- *Ricetta* è una dimensione "degenere," cioè senza attributi
- per la privatezza, si eliminano tutte le informazioni personali, indicando solo ASL del paziente e fascia d'età; quindi la grana scelta è "singole prescrizioni;" in effetti, vista la presenza di *KRicetta*, tutte le dimensioni, a parte *Ricetta* e *Farmaci*, sono secondarie
- *Farmaci* è una slowly changing dimension rispetto alla fascia
- si usa *Età* invece di fascia di età, per dare un po' di flessibilità
- è opportuno avere due viste su DimASL, poiché la dimensione è utilizzata due volte

Basi di dati II — 19 settembre 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 Ricette, ElementiRicetta, Pazienti, Farmacie, due volte Territorio (per la ASL) e Farmaci (per il prezzo)
- calcolo dell'età del paziente
- proiezione sugli attributi rilevanti:
- calcolo dell'importo (moltiplicazione del prezzo con la quantià)
- sostituzione degli identificatori o dei valori (a seconda dei casi) con le chiavi surrogate delle dimensioni