

Basi di dati II — 14 maggio 2018 — Compito A

Domanda 4 (20%)

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 (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè 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 |
|--|---|-----|------|----|
| $s_2, s_1, r_2(x), w_2(x), r_1(x), w_1(x), c_2, c_1$ | | | | |
| $s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$ | | | | |
| $s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$ | | | | |
| $s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$ | | | | |

Basi di dati II — 14 maggio 2018 — Compito A

Domanda 4 (20%)

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 (cioè generabile da uno scheduler basato su 2PL stretto), MV (cioè 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 |
|--|----|-----|------|----|
| $s_2, s_1, r_2(x), w_2(x), r_1(x), w_1(x), c_2, c_1$ | sì | sì | no | no |
| $s_2, r_2(x), w_2(x), c_2, s_1, r_1(x), w_1(x), c_1$ | sì | sì | sì | sì |
| $s_1, r_1(x), s_2, r_2(x), w_1(x), c_1, w_2(x), c_2$ | no | no | no | no |
| $s_1, r_1(x), s_2, r_2(x), w_2(x), r_2(y), w_2(y), c_2, r_1(y), c_1$ | no | no | no | sì |

Basi di dati II — 12 maggio 2015 — Compito A

Domanda 3 (15%)

Si consideri una base di dati su cui una applicazione effettua moltissimi inserimenti e aggiornamenti, con operazioni tutte molto semplici ma estremamente numerose. Considerare le due situazioni seguenti:

- A. concorrentemente alle operazioni sopra citate non viene eseguita nessun'altra operazione
- B. concorrentemente alle operazioni sopra citate vengono eseguite molte interrogazioni (e nessun altro aggiornamento)

Commentare brevemente per ciascuna delle due situazioni quali potrebbero essere i vantaggi e gli svantaggi (con riferimento alle prestazioni in termini di tempo di risposta sia per la normale operatività sia in caso di guasto e conseguente necessità di recovery) delle seguenti tre scelte per l'applicazione che esegue inserimenti e aggiornamenti:

- i. eseguire ciascuna operazione in una transazione separata
- ii. riunire le operazioni in transazioni di medie dimensioni (alcune decine di operazioni per ciascuna)
- iii. riunire tutte le operazioni in un'unica transazione

Nota bene: non è detto che esista una soluzione ideale, ciò che si deve fornire sono riflessioni critiche, che, sinteticamente, illustrino spunti interessanti.

Domanda 3 (15%)

Si consideri una base di dati su cui una applicazione effettua moltissimi inserimenti e aggiornamenti, con operazioni tutte molto semplici ma estremamente numerose. Considerare le due situazioni seguenti:

- A. concorrentemente alle operazioni sopra citate non viene eseguita nessun'altra operazione
- B. concorrentemente alle operazioni sopra citate vengono eseguite molte interrogazioni (e nessun altro aggiornamento)

Commentare brevemente per ciascuna delle due situazioni quali potrebbero essere i vantaggi e gli svantaggi (con riferimento alle prestazioni in termini di tempo di risposta sia per la normale operatività sia in caso di guasto e conseguente necessità di recovery) delle seguenti tre scelte per l'applicazione che esegue inserimenti e aggiornamenti:

- i. eseguire ciascuna operazione in una transazione separata
- ii. riunire le operazioni in transazioni di medie dimensioni (alcune decine di operazioni per ciascuna)
- iii. riunire tutte le operazioni in un'unica transazione

Nota bene: non è detto che esista una soluzione ideale, ciò che si deve fornire sono riflessioni critiche, che, sinteticamente, illustrino spunti interessanti.

- Una transazione
 - Vantaggi: (concorrenza) con MV, c'è una sola versione precedente e quindi può essere efficiente; (affidabilità) poche scritture nel log
 - Svantaggi: (concorrenza) con S2PL, le interrogazioni potrebbero essere rallentate; (affidabilità) in caso di guasto, si deve ricominciare da capo; inoltre, recovery potenzialmente costoso (molte undo)
- Moltissime transazioni piccole
 - Vantaggi: (concorrenza) i lock interferiscono poco; (affidabilità) in caso di guasto si perde poco e i recovery sono semplici
 - Svantaggi: (concorrenza) se più transazioni modificano lo stesso dato, si possono bloccare a vicenda, problema comunque breve e minore; (affidabilità) molte scritture nel log
- Soluzione intermedia: da valutare sperimentalmente