

Gestione della concorrenza (DBMS)

Un'esecuzione di un'insieme di transazioni si dice *seriale* se le operazioni di ciascuna transazione avvengono in sequenza senza intrecciarsi con altre transazioni.

Uno schedule seriale previene errori di concorrenza, come:

perdita di aggiornamento T_1 modifica x , T_2 lo modifica e scrive, T_1 scrive \rightarrow l'effetto di T_2 è annullato;

lettura sporca T_1 scrive, T_2 legge e fa commit, T_1 fa rollback;

lettura incoerente T_1 legge due volte x , T_2 lo modifica tra una lettura e l'altra.

Si può ottenere solo se non si ammette concorrenza o le transazioni sono completamente indipendenti, che è irrealistico. Perciò introduciamo il concetto di *serializzabilità*: un insieme di transazioni è serializzabile se la sua esecuzione concorrente produce lo stesso effetto di un'esecuzione seriale.

Il gestore della concorrenza garantisce la serializzabilità delle transazioni eseguite concorrentemente. Ci sono due classi di protocolli:

pessimistici ritardano l'esecuzione delle transazioni che potrebbero generare conflitti;

ottimistici lettura-validazione-scrittura: esecuzione senza coordinamento e controllo al momento del commit di eventuali conflitti verificati (\rightarrow rollback e riesecuzione).

Protocolli pessimistici

strict 2-phase locking lock a livello di riga, tabella o pagina (multigranularità) e in mutua esclusione per scrittura o accesso condiviso per lettura (multimodale). La transazione:

- acquisisce i lock di tutti i dati che le servono nella modalità appropriata — viene messa in attesa se questa non è compatibile con lo stato corrente del lock;
- li rilascia tutti dopo il commit.

Può portare a deadlock, risolti con:

timeout allo scadere si annulla e riesegue la transazione;

avoidance strategie: lock/unlock atomici, livelli di priorità o timestamp;

detection & recovery individuazione di cicli nel grafo richieste-risorse e abort delle transazioni coinvolte;

timestamp si tiene traccia del tempo di inizio di ogni transazione, e si impediscono:

- accessi a dati scritti da una transazione con timestamp maggiore;
- scritture di dati letti da una transazione con timestamp maggiore.

Livelli di isolamento

serializable accesso esclusivo in scrittura o condiviso in lettura della tabella;

repeatable read a livello di riga, quindi le letture sono ripetibili ma le inserzioni possono causare problemi;

read committed i dati letti sono sempre effetto di una transazione conclusa, ma i lock per le letture vengono rilasciati subito, quindi possibilità di letture non ripetibili;

read uncommitted possibile leggere dati modificati da una transazione non terminata (che quindi potrebbe fare rollback).