UNIVERSITÀ
DEGLI STUDI
DI MILANO

LA STATALE

Introduzione alla transazioni

Problemi di concorrenza nelle basi di dati

Prof. Stefano Montanelli

Nozione di transazione

- L’idea di fondo di una transazione è di collocare in un’unica operazione atomica un insieme di operazioni sui dati in modo che si giunga a uno dei due seguenti esiti:
  - Tutte le operazioni vanno a buon fine e la transazione è approvata
  - Un’operazione va in errore e la transazione è annullata: nessuna delle operazioni nella transazione ha effetto sullo stato della base di dati
- Le operazioni interne alla transazione (e il loro esito) non sono visibili a altre transazioni

Utilità delle transazioni

- Mantenere il corretto il comportamento e l'integrità della base di dati anche in presenza di singole operazioni che falliscono
- Mantenere il corretto il comportamento della basi di dati anche in presenza di operazioni concorrenti, ovvero eseguite in tempi diversi sugli stessi dati
- Una transazione trasforma lo stato corretto di un database in un altro stato corretto del database. Durante l'esecuzione della transazione, lo stato può essere temporaneamente non corretto
- Al termine della transazione tuttavia gli esiti possibili sono solo due: raggiungimento di un nuovo stato corretto (**COMMIT**) o ritorno a uno stato corretto precedente (**ROLLBACK**)

Concorrenza nell’accesso ai dati

- Le operazioni di un DBMS sono eseguite tramite le primitive di *read* dei dati da memoria secondaria a memoria principale e *write* dei dati da memoria principale e memoria secondaria
- Quando più utenti o applicazioni lavorano in modo concorrente sui dati i tempi di lettura e scrittura possono provocare anomalie
  - Lost update
  - Dirty read
  - Incorrect summary

Concorrenza nell’accesso ai dati

- **Lost update**: due utenti U1, U2 leggono il medesimo dato X ed eseguono un’operazione di aggiornamento. Poiché U2 legge il dato X prima che U1 scriva, l’aggiornamento effettuato da U1 viene perso

- **Dirty read**: l’utente U1 aggiorna un dato X e successivamente fallisce. Il dato X (temporaneamente) aggiornato da U1 viene letto dall’utente U2 prima che X venga riportato al suo valore originale precedente la modifica effettuata da U1

- **Incorrect summary**: se un utente calcola una funzione aggregata su un insieme di record mentre un altro utente effettua un aggiornamento sui record, il primo client può calcolare la funzione aggregata considerando alcuni valori prima dell’aggiornamento e alcuni valori dopo l’aggiornamento

Proprietà delle transazioni

- Il DBMS esegue transazioni concorrenti in modo tale da garantire le seguenti proprietà:
  - **A**tomicity
  - **C**onsistency
  - **I**solation
  - **D**urability

Comportamento predefinito

- Modalità AUTOCOMMIT: ogni operazione SQL viene considerata una transazione (non servono né BEGIN né COMMIT)
- Quando viene lanciato il comando COMMIT/ROLLBACK la transazione corrente termina e ne può iniziare una nuova
- Il ROLLBACK può essere eseguito automaticamente in caso di errore o anche invocato esplicitamente dall'utente nella definizione della transazione per catturare eventuali condizioni logiche non soddisfatte

Atomicity

- L'esecuzione di una transazione deve essere per definizione totale o nulla e non sono ammesse esecuzioni parziali
- Se si verifica un errore durante l’esecuzione di una transazione prima del COMMIT, gli effetti parziali della transazione non sono memorizzati nel database
- La cancellazione delle modifiche parziali di una transazione si definisce ROLLBACK

Consistency

- Ogni transazione può assumere di lavorare su un DB dove tutti i vincoli di integrità sono verificati
- Ogni transazione deve lasciare un DB che soddisfa tutti i vincoli di integrità.
- Ai fini delle transazioni, i vincoli possono essere specificati come IMMEDIATE o DEFERRED
  - **IMMEDIATE**: verifica di integrità eseguita durante l'esecuzione della transazione. L'istruzione che causa violazione viene cancellata (UNDO) senza imporre un abort della transazione
  - **DEFERRED**: verifica di integrità eseguita al termine della transazione, dopo il COMMIT. Se qualche vincolo viene violato, viene eseguito il ROLLBACK della transazione disfacendo (UNDO) l'intera sequenza di comandi che costituiscono la transazione

Isolation

- Ogni transazione deve essere eseguita in modo isolato e indipendente dalle altre transazioni
- L'eventuale fallimento di una transazione non deve interferire con le altre transazioni in esecuzione
- Le operazioni sono alternate in modo che l’esecuzione sia equivalente a qualche ordine sequenziale (seriale) delle transazioni (nozione di *serializzabilità*)

Durability

- Anche detta **persistency**, stabilisce gli effetti delle operazioni di una transazione giunta al COMMIT non debbano essere persi.

- Il DBMS mantiene registri di log dove sono annotate tutte le operazioni su un DB

- Questo permette di evitare eventuali perdite di dati dovute a malfunzionamenti nell'intervallo di tempo fra l'impegno a memorizzare le modifiche e l'effettiva scrittura sui dischi da parte del DBMS