Transazioni

DOCENTE
PROF. ALBERTO BELUSSI

Anno accademico 2011/12

Tecnologia dei DBMS

DBMS: sistema per la gestione di basi di dati

In questa seconda parte del modulo di TEORIA del corso di Basi di dati, si presentano le tecniche per l'implementazione dei sistemi DBMS.

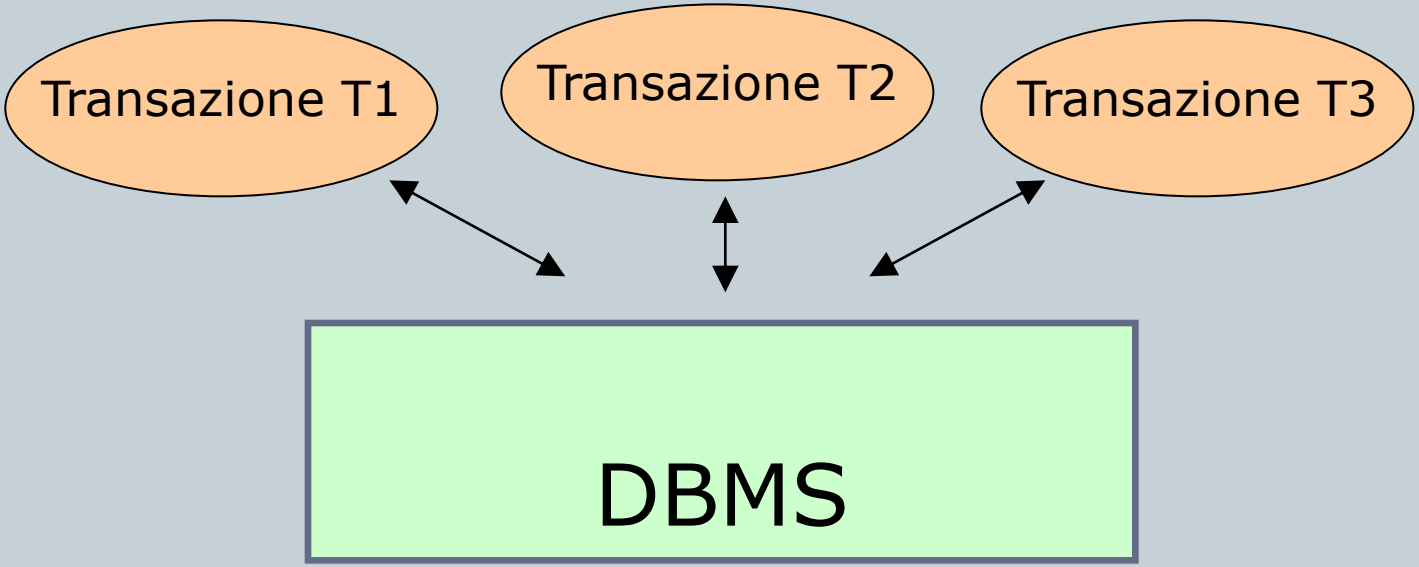
Principalmente si affronteranno i seguenti argomenti:

- Transazioni: definizione, proprietà ed esecuzione concorrente di transazioni
- Architettura di un DBMS
- Gestione dei guasti (cenni)
- Strutture fisiche: pagine della memoria secondaria
- Strutture d'accesso ai dati: indici primari e secondari
- B+-tree e strutture ad accesso calcolato (Hashing)

Tecnologia dei DBMS

Principale caratteristica di un DBMS

Un DBMS è un sistema TRANSAZIONALE: fornisce un meccanismo per la definizione ed esecuzione di TRANSAZIONI.



Transazione

Definizione di transazione

E' un'unità di lavoro svolto da un programma applicativo (che interagisce con una base di dati) per la quale si vogliono garantire alcune proprietà.

Principale caratteristica di una transazione:

Una transazione o va a buon fine e ha effetto sulla base di dati o abortisce e non ha nessun effetto sulla base di dati.

O tutto o niente!

Transazione

Sintassi per la definire di una transazione in SQL

<transazione> → `begin transaction`
<programma>
`end transaction`

<programma> → ⟨<istruzioni> | `commit work` | `rollback work`⟩
{⟨<istruzioni> | `commit work` | `rollback work`⟩}

La transazione va a buon fine all'esecuzione di un `commit work`. Non ha invece effetto se viene eseguito un `rollback work`.

Transazione

Una transazione è ben formata se:

- Inizia con un `begin transaction.`
- Termina con un `end transaction.`
- La sua esecuzione comporta il raggiungimento di un `commit` o di un `rollback work` e dopo il `commit/rollback` non si eseguono altri accessi alla base di dati.

Esempio di transazione ben formata

```
begin transaction;
  update CONTO set saldo = saldo - 1200
         where filiale = '005' and numero = 15;
  update CONTO set saldo = saldo + 1200
         where filiale = '005' and numero = 105;
commit work;
end transaction;
```

Proprietà delle transazioni

Una transazione ha quattro proprietà:

ATOMICITA'	**A**tomicity
CONSISTENZA	**C**onsistency
ISOLAMENTO	**I**solation
PERSISTENZA	**D**urability

Un DBMS che gestisce transazioni dovrebbe garantire per ogni transazione che esegue tutte queste proprietà.

Proprietà delle transazioni

Atomicità

Una transazione è una unità di esecuzione INDIVISIBILE. O viene eseguita completamente o non viene eseguita affatto.

Implicazioni:

- Se una transazione viene interrotta prima del commit, il lavoro fin qui eseguito dalla transazione deve essere disfatto ripristinando la situazione in cui si trovava la base di dati prima dell'inizio della transazione.
- Se una transazione viene interrotta all'esecuzione del commit (commit eseguito con successo), il sistema deve assicurare che la transazione abbia effetto sulla base di dati.

Proprietà delle transazioni

Consistenza

L'esecuzione di una transazione non deve violare i vincoli di integrità.

Implicazioni:

- Al verificarsi della violazione di un vincolo il sistema può:
  VERIFICA IMMEDIATA:
  - Viene abortita l'ultima operazione e il sistema restituisce all'applicazione una segnalazione d'errore
  - L'applicazione può quindi reagire alla violazione.

  VERIFICA DIFFERITA:
  - Al commit se un vincolo di integrità viene violato la transazione viene abortita senza possibilità da parte dell'applicazione di reagire alla violazione.

Proprietà delle transazioni

Isolamento

L'esecuzione di una transazione deve essere INDIPENDENTE dalla contemporanea esecuzione di altre transazioni.

Implicazioni:

- Il rollback di una transazione non deve creare rollback a catena di altre transazioni che si trovano in esecuzione contemporaneamente.
- Il sistema deve regolare l'esecuzione concorrente

Proprietà delle transazioni

Persistenza

L'effetto di una transazione che ha eseguito il commit non deve andare perso.

Implicazioni:

- Il sistema deve essere in grado, in caso di guasto, di garantire gli effetti delle transazioni che al momento del guasto avevano già eseguito un commit.

Architettura di un DBMS

L'architettura mostra i moduli principali che possiamo individuare nei DBMS attuali, considerando le diverse funzionalità che il DBMS svolge durante l'esecuzione delle transazioni

Per ogni modulo dell'architettura presentiamo le funzionalità che esso svolge e alcune delle tecniche che applica.

Alla lavagna