
Basi di Dati
Modulo Tecnologie

Transazioni

DR. SARA MIGLIORINI

A.A. 2024-2025

Tecnologie per le basi di dati

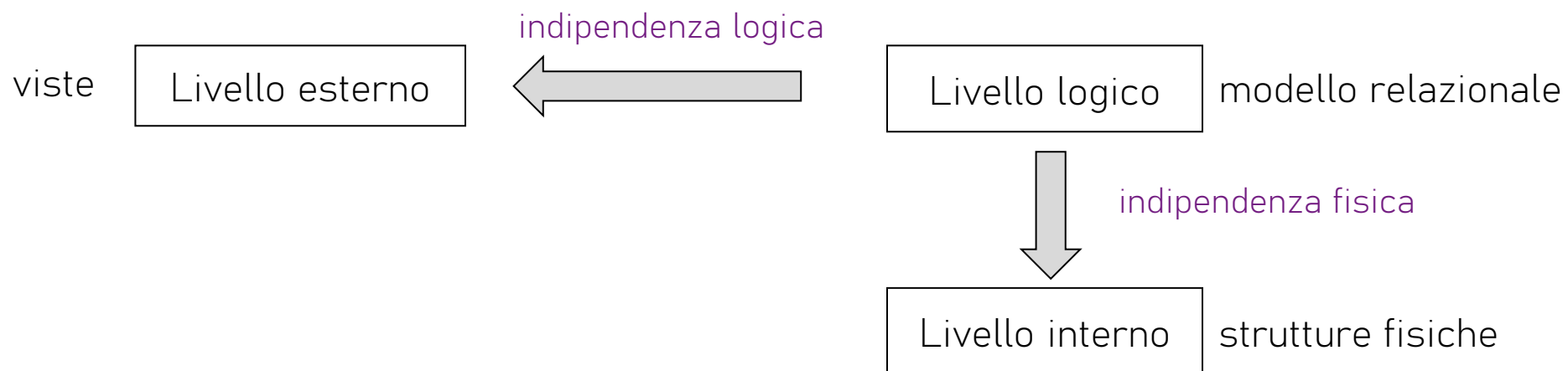
- Argomenti visti nel Modulo di Teoria:
 - Progettazione concettuale
 - Progettazione logica
 - Linguaggi di interrogazione (algebra relazionale e calcolo relazionale)
- Argomenti del Modulo di Tecnologie per basi di dati
 - Tecniche per l'implementazione dei sistemi che gestiscono basi di dati (DBMS)
 - DBMS: sistema per la gestione di basi di dati (DataBase Management System)
 - NoSQL: sistemi document-based

Tecnologia dei DBMS

- Un DBMS (**Database Management System**) o sistema di gestione di basi di dati è un sistema software in grado di gestire collezioni di dati che siano **grandi, condivise e persistenti**, assicurando la loro **affidabilità e privatezza**.
- Un **DBMS** in quanto sistema software deve essere **efficiente ed efficace**.
 - Dipendenza con la bontà della progettazione della base di dati.
- Un DMBS estende le funzionalità del file system
 - Gestione in memoria secondaria collezioni di dati.
- Una base di dati è una collezione di dati gestita da un DBMS.

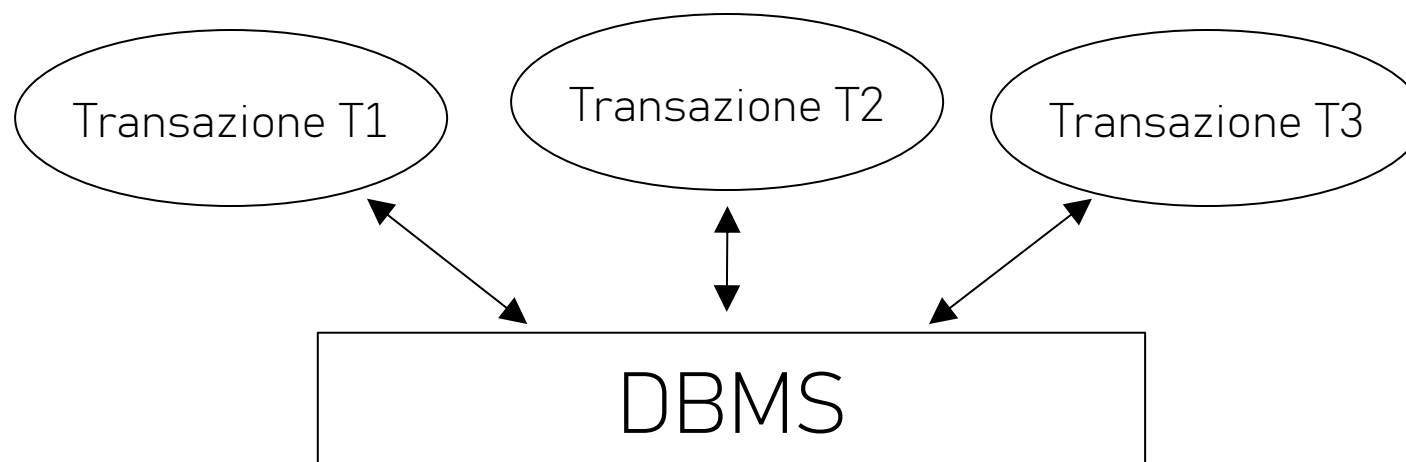
Tecnologia dei DBMS

- Architettura su 3 livelli dei DBMS



Tecnologia dei DBMS

- Un DBMS basato sul **modello relazionale** è nella maggior parte dei casi anche un **sistema transazionale**: fornisce un meccanismo per la definizione ed esecuzione di transazioni.



Tecnologie per le basi di dati

- In sintesi, durante il corso si affronteranno i seguenti argomenti:
 - Transazioni: definizione e proprietà
 - Architettura di un DBMS:
 - Gestore della memoria secondaria: strutture fisiche per la memorizzazione dei dati in memoria secondari
 - Gestore del buffer: interazione tra memoria secondaria e memoria centrale
 - Gestore dei metodi di accesso: indici primari e secondari, indici B+-tree e indici Hash
 - Gestore delle interrogazioni: ottimizzazione delle interrogazioni
 - Gestore della concorrenza: gestione dell'esecuzione concorrente (VSR, CSR, locking a due fasi)
 - Gestore dell'affidabilità: gestione dei guasti
 - Architettura di un sistema document-based

Transazione

- Definizione di transazione
 - È un'unità di lavoro svolta da un programma applicativo (che interagisce con una base di dati) per la quale si vogliono garantire proprietà di correttezza, robustezza e isolamento.
- 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! Non sono ammesse esecuzioni parziali

Transazione

- Sintassi per definire una transazione in SQL

<transazione> → begin transaction

<programma>

end transaction

<programma> → {<istruzione> | commit work | rollback work}

{<istruzione> | commit work | rollback work}

- La transazione va a buon fine all'esecuzione di un commit work.
- La transazione non ha alcun 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.

Transazione

- 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 = 205;
```

```
    commit work;
```

```
end transaction;
```

Proprietà delle transazioni

- Una transazione ha quattro proprietà:
 - ATOMICITÀ Atomicity
 - CONSISTENZA Consistency
 - ISOLAMENTO Isolation
 - PERSISTENZA Durability
- 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 **dopo** l'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:

- Verifica immediata:
 - Fatta nel corso della transazione
 - Viene abortita solo 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 con meccanismi di controllo dell'accesso alle risorse.

Proprietà delle transazioni

Persistenza

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

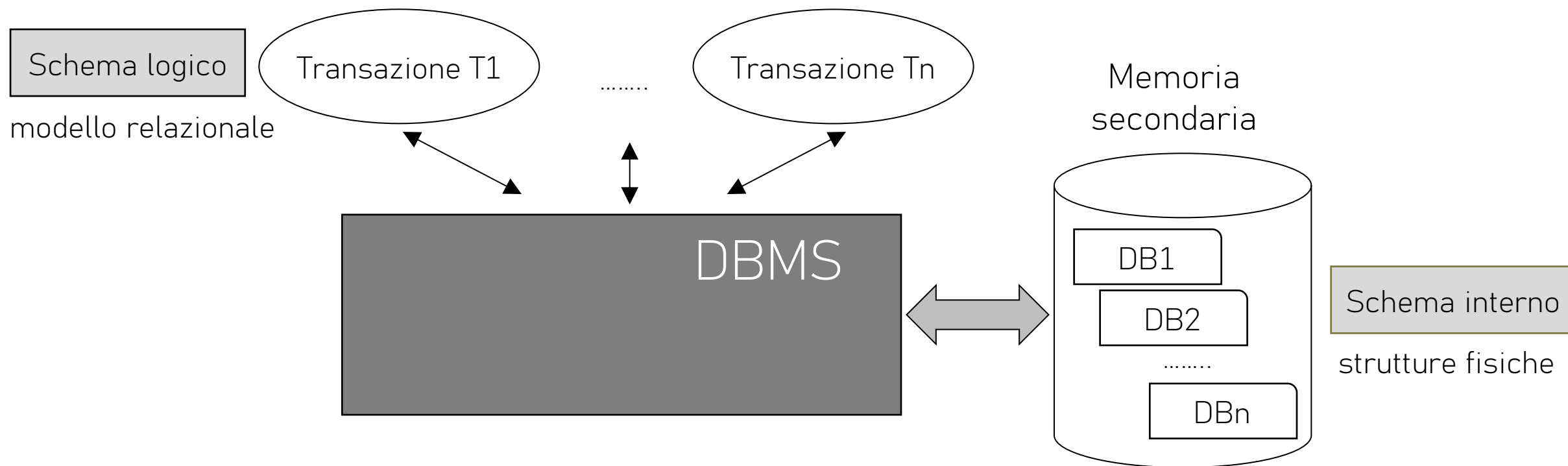
Implicazioni:

- 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.

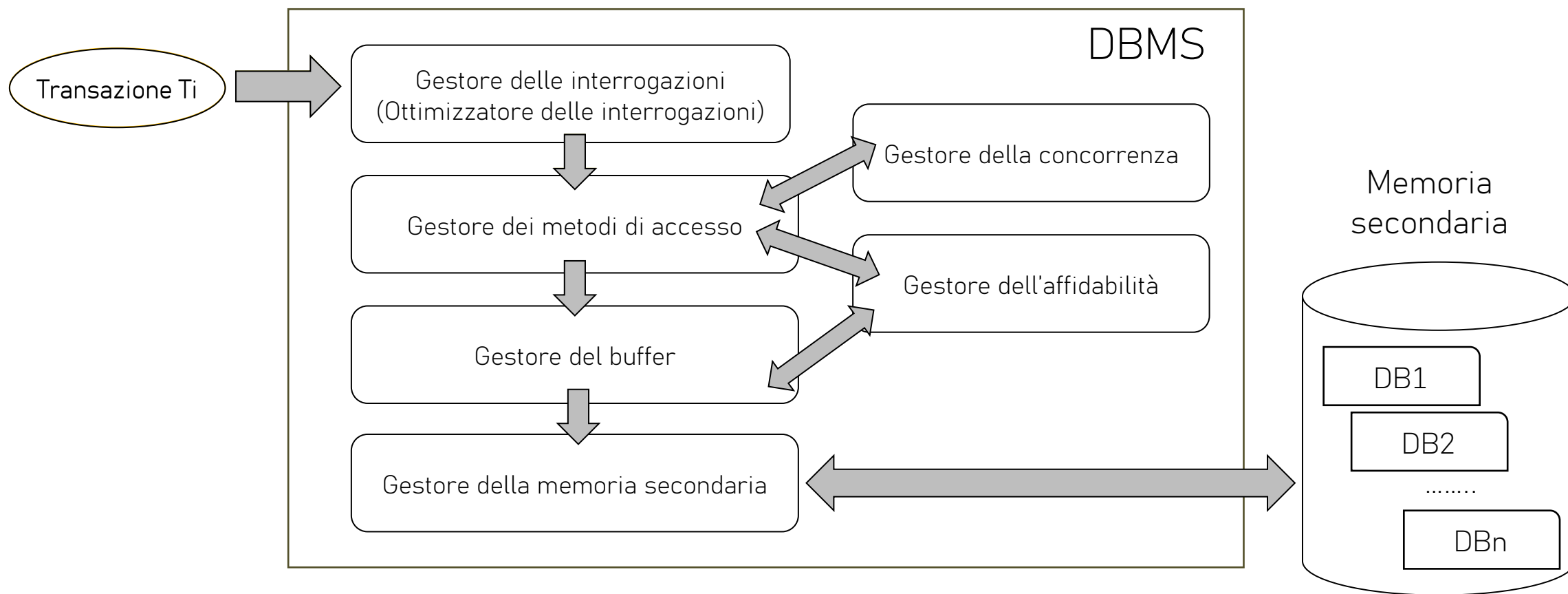
Architettura di un DBMS

- 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.

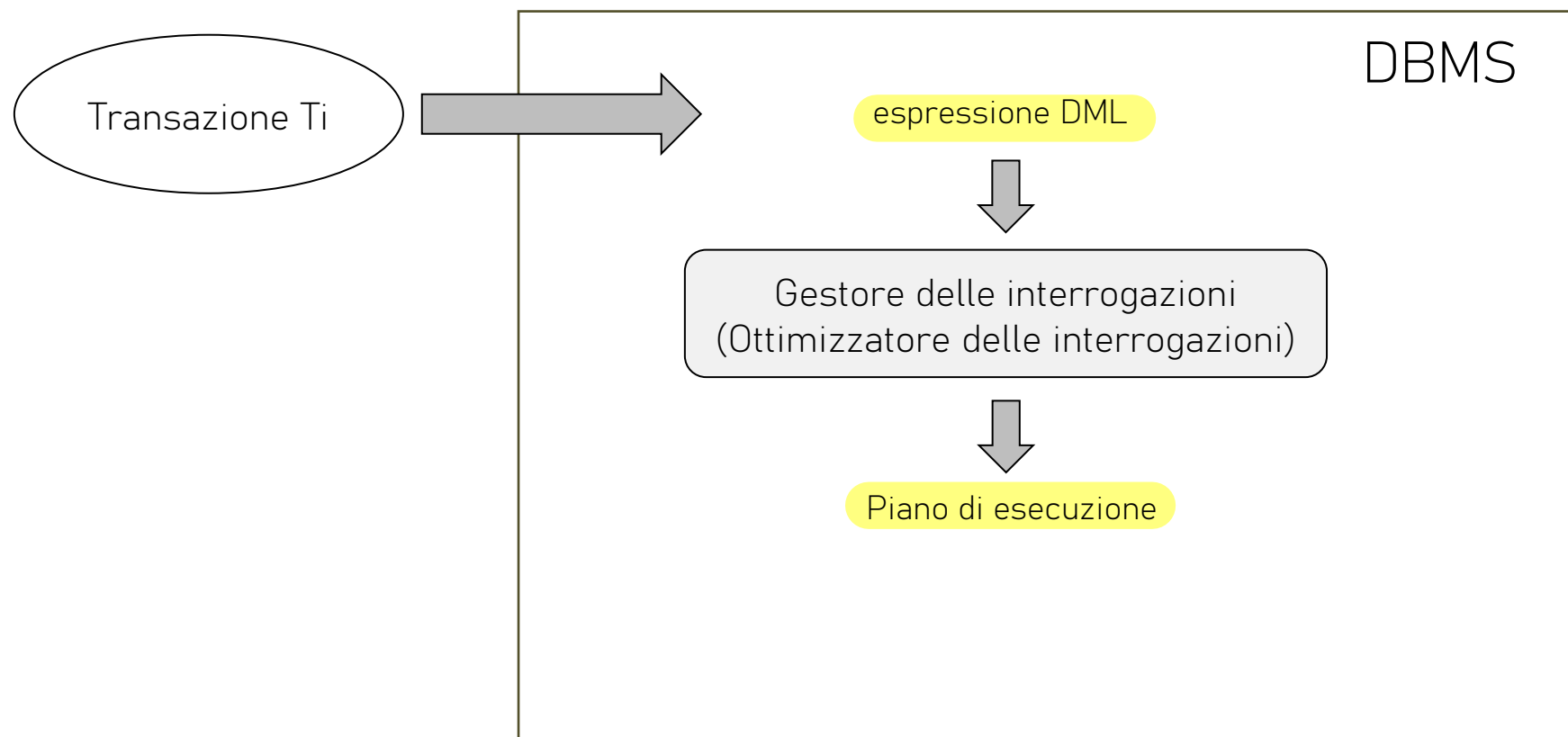
Architettura di riferimento di un DBMS



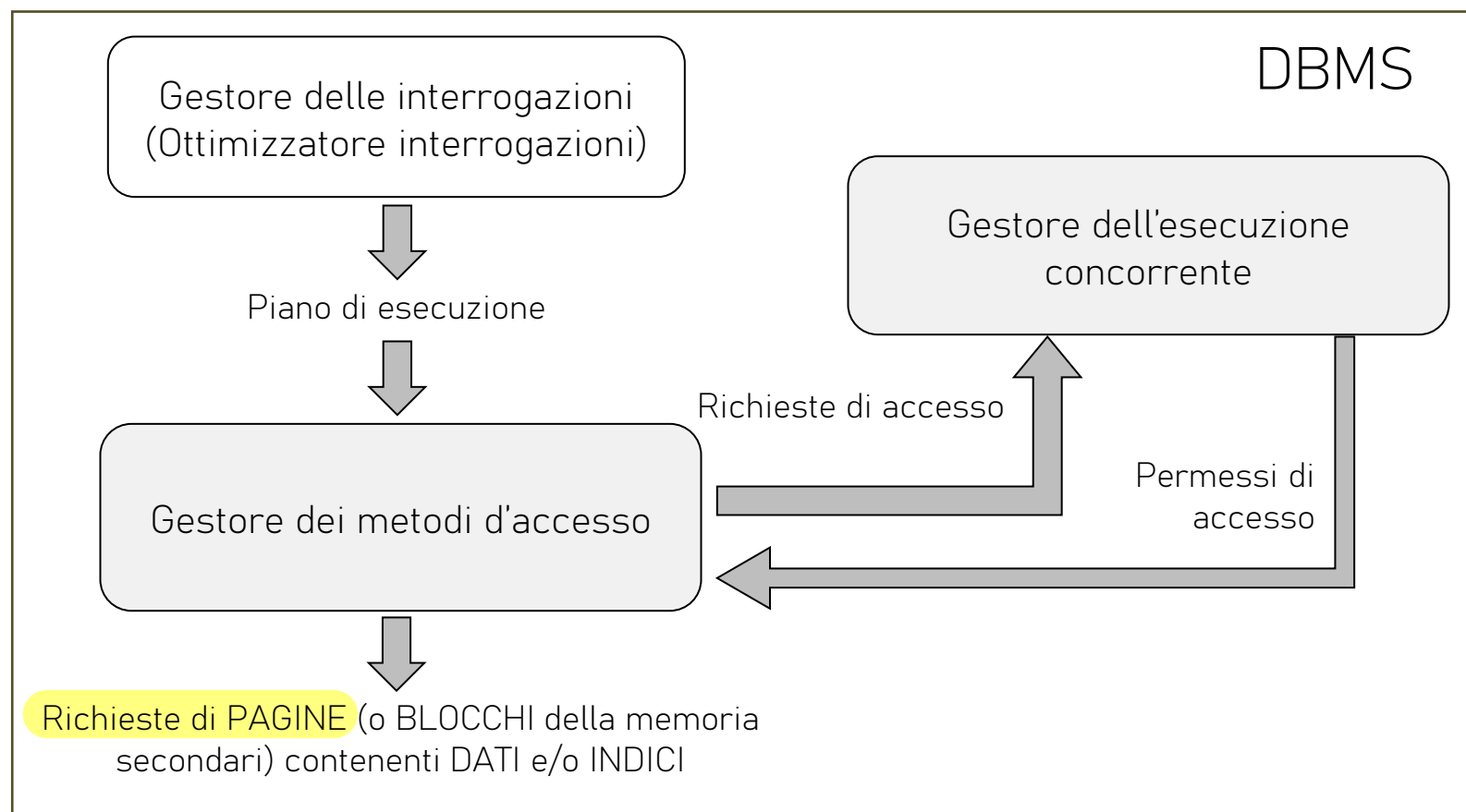
Architettura di riferimento di un DBMS



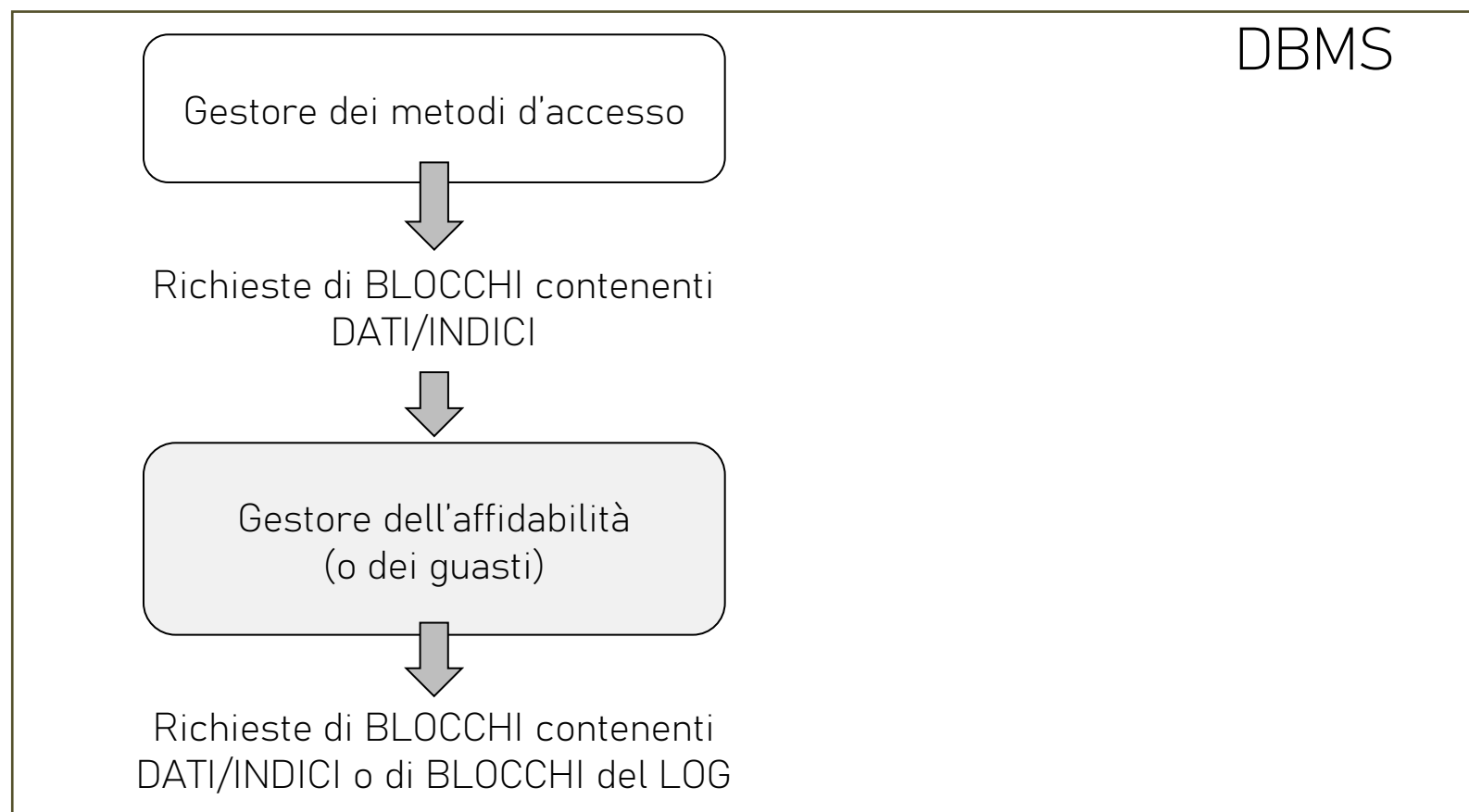
Architettura di riferimento di un DBMS



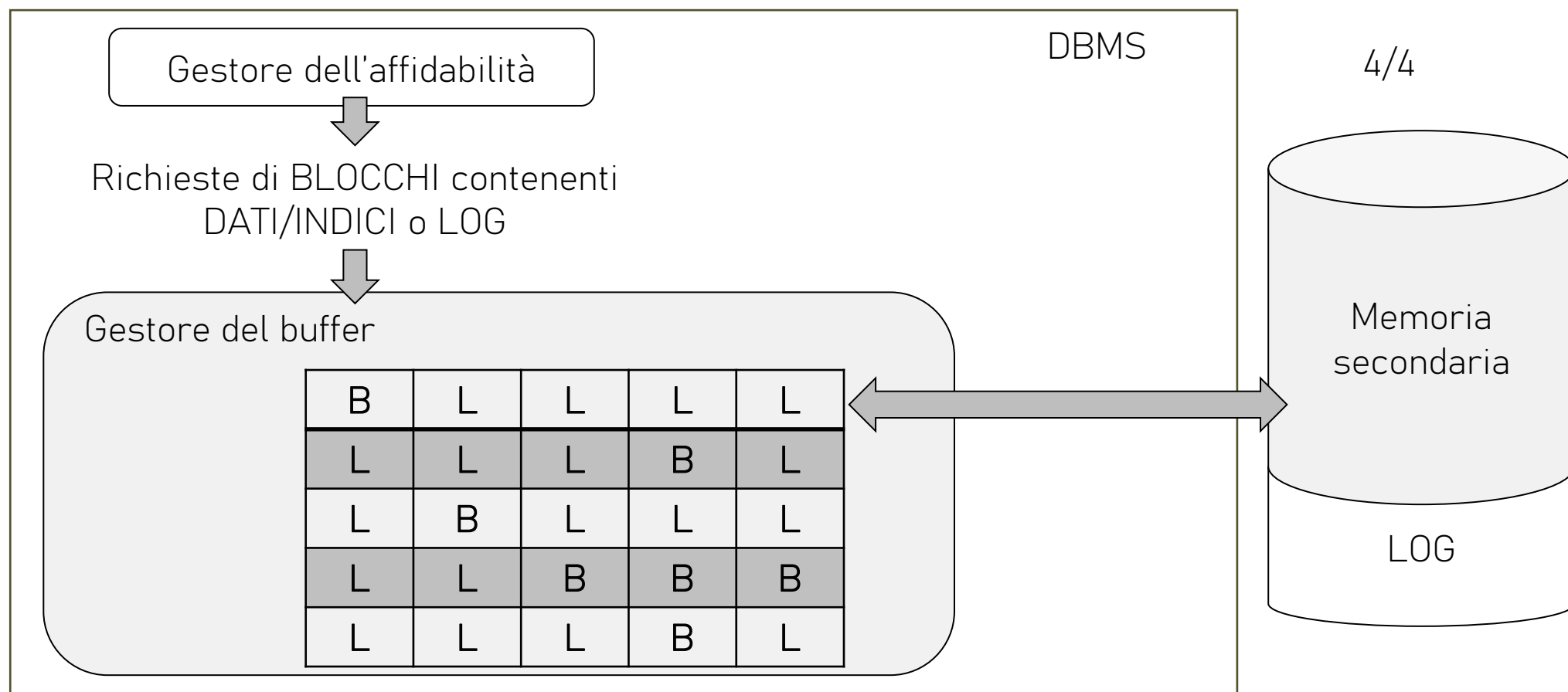
Architettura di riferimento di un DBMS



Architettura di riferimento di un DBMS



Architettura di riferimento di un DBMS



Moduli e proprietà delle transazioni

- Quali moduli contribuiscono a garantire le proprietà delle transazioni?

Gestore dei metodi d'accesso	Consistenza
Gestore dell'esecuzione concorrente	Atomicità e Isolamento
Gestore dell'affidabilità	Atomicità e Persistenza