

Basi di dati II

Organizzazione del corso

Orario

- Martedì 9.30 -11.00
 - Aula VA3 (Via Valleggio, 11), Como
 - Aula 6 TM (Padiglione Morselli), Varese
- Venerdì 11.00 -12.30
 - Aule: Aula VP1 (Via Valleggio, 11), Como
 - Aula 10 MTG (Pad. Monte generoso), Varese

Controllare aule
assegnate
nell'orario online

Materiale

- *Sistemi di gestione dati. Concetti e architetture* – B. Catania – E. Ferrari –G. Guerrini - CittàStudi editore
- *Basi di Dati 5/ED* P. Atzeni, S. Ceri, P. Fraternali, S. Paraboschi, R. Torlone Mc Graw Hill.

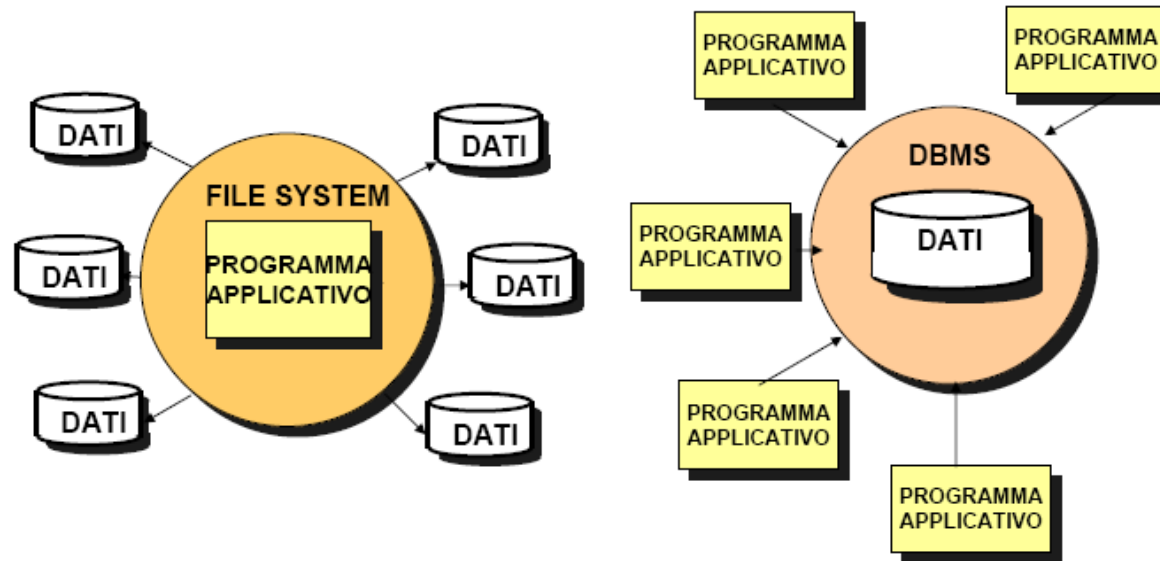
Lucidi tratti da entrambi i libri

Obiettivi del corso

- O1. Conoscere le principali strutture fisiche e relative gestioni (indici, metodi di accesso) utilizzati nei DBMS.
- O2. Essere in grado di ottimizzare la progettazione e le interrogazioni da eseguire.
- O3. Riconoscere e gestire problemi legati alla gestione di transazioni
- O4. Conoscere i principali modelli di dati NoSQL.

Basi di dati I - dove ci siamo lasciati

Base di dati vs. file system



I DBMS estendono le funzionalità dei file system, fornendo più servizi ed in maniera **integrata**

La carta vincente dei DBMS

- Il meccanismo fondamentale di un DBMS è lo **schema logico** della base di dati
- Lo schema logico descrive il contenuto della base di dati tramite un formalismo ad alto livello che esula dai dettagli della sua effettiva implementazione fisica, detto **modello dei dati**

Il modello relazionale

- Modello dei dati più diffuso
- E' basato su una singola struttura dati - la *relazione*
- Una relazione può essere vista come una tabella con righe, dette *tuple*, e colonne contenenti dati di tipo specificato (ad esempio interi e stringhe)

Esempio

Relazione

Attributo

Film				
titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

Tupla

Livelli nella rappresentazione dei dati

- **Livello logico:**

- descrizione della base di dati mediante il modello logico del DBMS (**schema logico**):
 - Quali sono i dati memorizzati nella base di dati
 - Eventuali associazioni tra di essi
 - Vincoli di integrità semantica e di autorizzazione

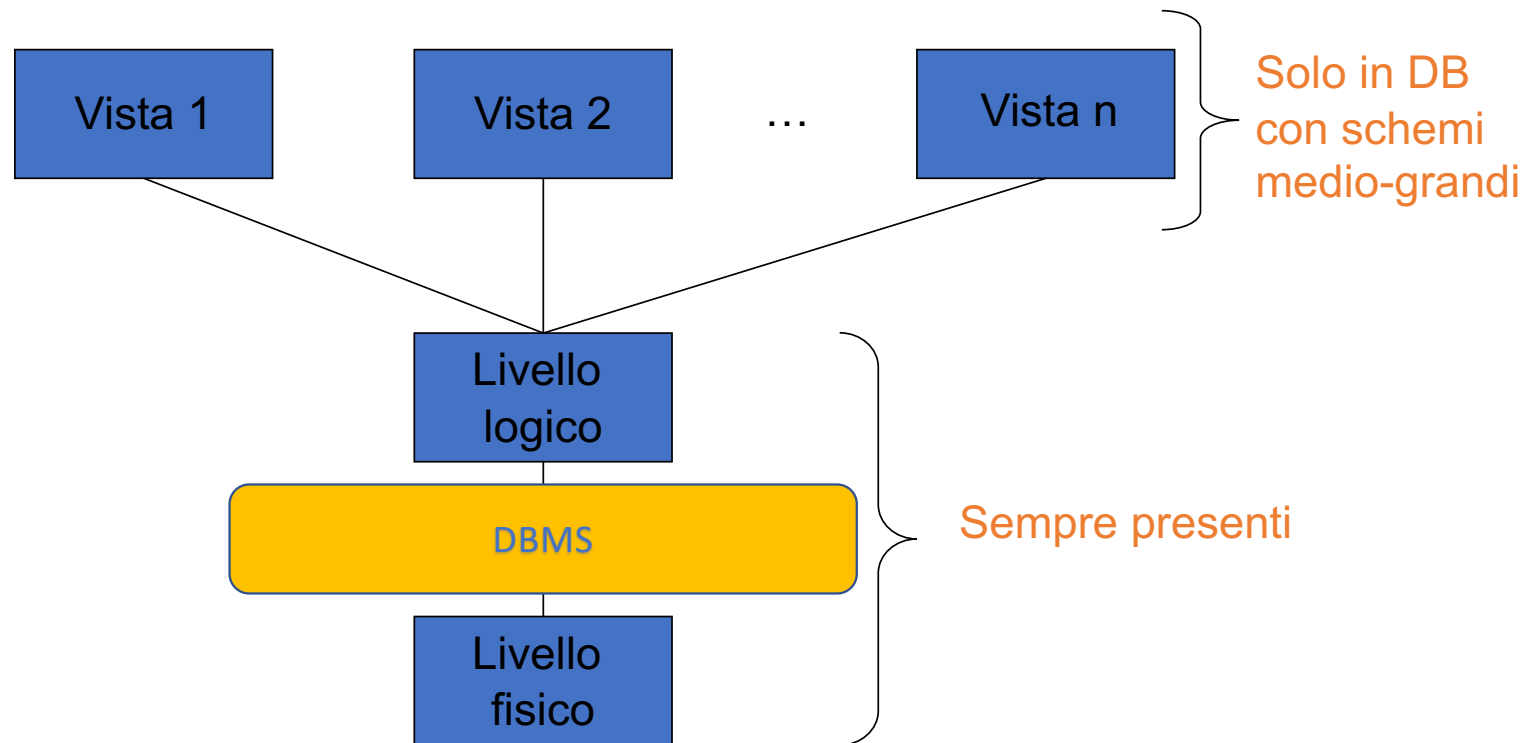
- **Livello fisico:**

- E' il livello più basso in cui viene definito lo **schema fisico** della base di dati, precisando come i dati nello schema logico sono effettivamente memorizzati tramite strutture di memorizzazione

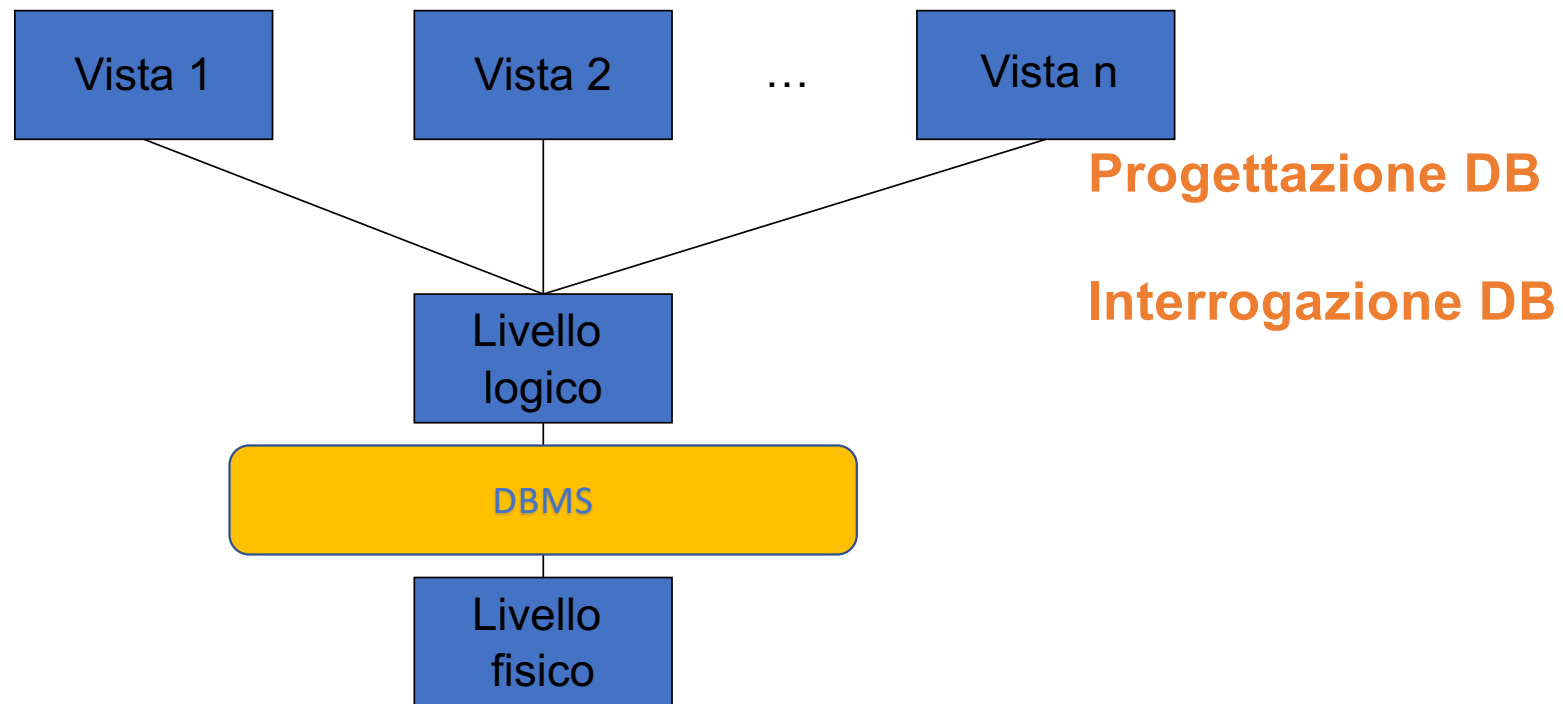
- **Livello esterno o livello delle viste:**

- E' il livello di astrazione più alto:
 - Descrive una porzione dell'intero schema della base di dati (**vista**)
 - Possono essere definite più viste di una stessa base di dati
 - E' definito sullo schema logico

Livelli nella rappresentazione dei dati



Basi di dati I - dove ci siamo lasciati



Basi di dati II

- I DBMS offrono i loro servizi in modo "trasparente":
 - per questo in Basi dati I abbiamo potuto ignorare molti aspetti realizzativi
 - abbiamo considerato il DBMS come una "scatola nera"
- Perché aprirla?
 - capire come funziona può essere utile per un migliore utilizzo

DataBase Management System — DBMS

- **prodotto software** in grado di gestire **collezioni di dati** che siano (anche):

- **grandi**
- **persistenti**
- **condivise**

garantendo

- **affidabilità**
- **confidenzialità**
- Come ogni prodotto informatico, un DBMS deve essere:
 - **efficiente**
 - **efficace**

DataBase Management System — DBMS

- **prodotto software** in grado di gestire **collezioni di dati** che siano (anche):

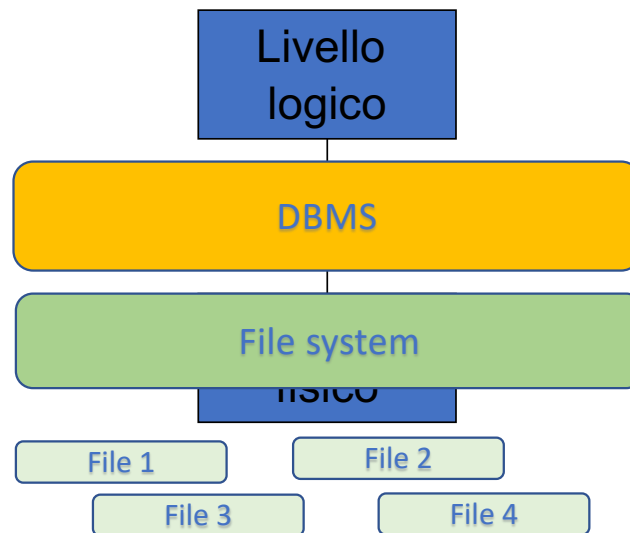
- **grandi**
- **persistenti**
- **condivise**

garantendo

- **affidabilità**
- **confidenzialità**
- Come ogni prodotto informatico, un DBMS deve essere:
 - **efficiente**
 - **efficace**

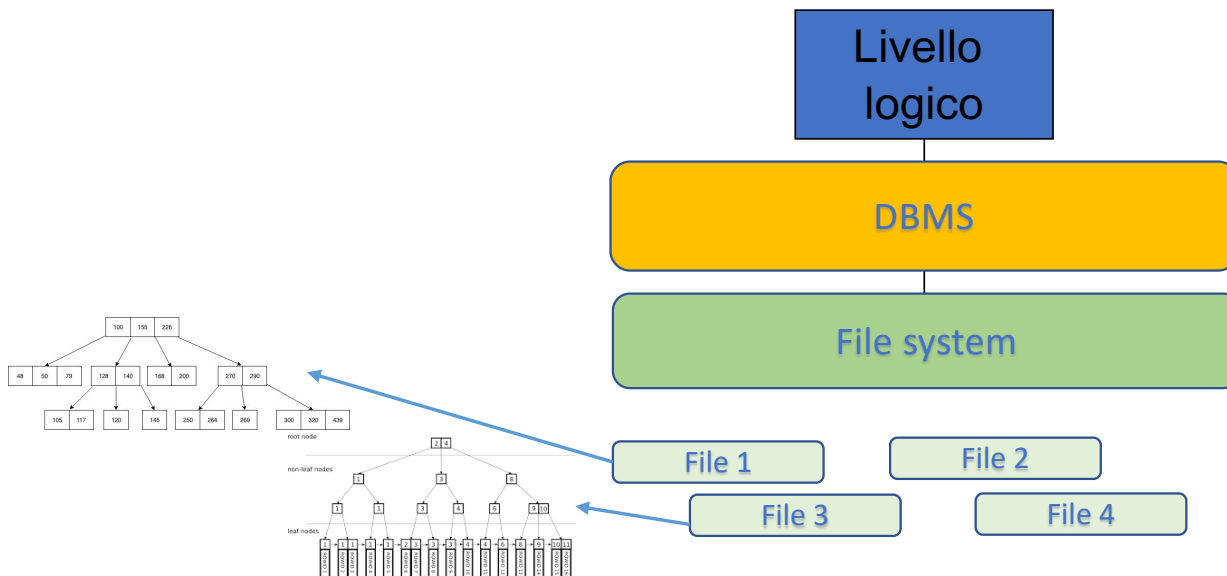
Le basi di dati sono grandi e persistenti

- La persistenza richiede una gestione in memoria secondaria



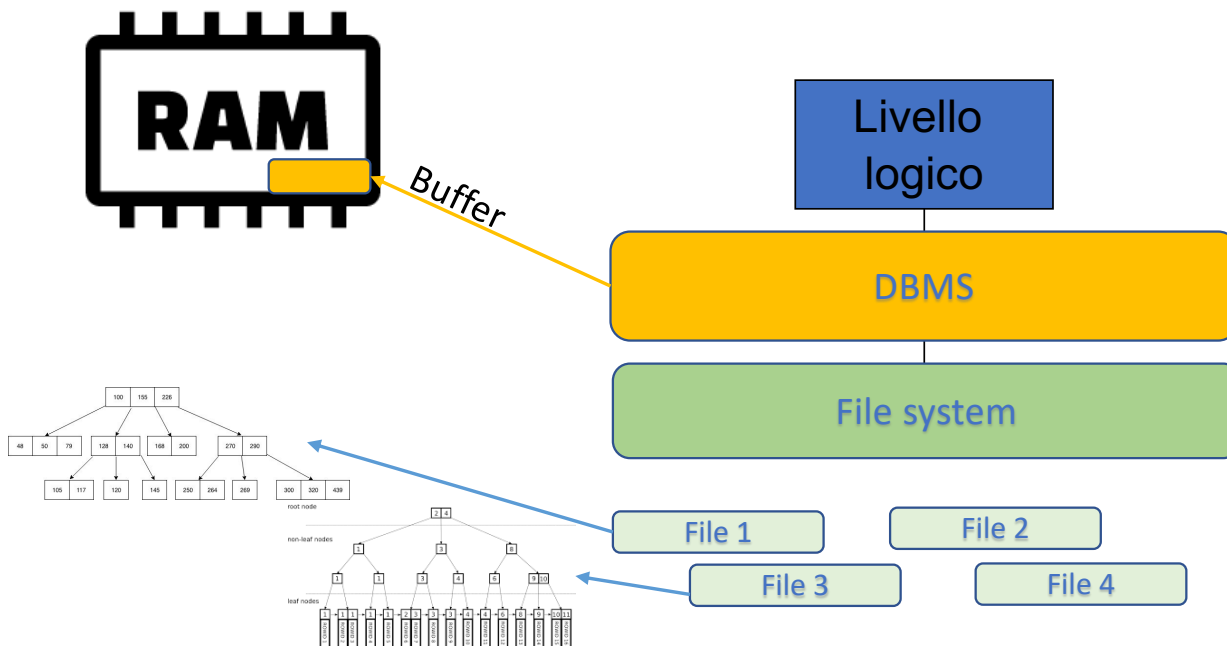
Le basi di dati sono grandi e persistenti

- La grandezza richiede che tale gestione sia sofisticata
- Le strutture logiche (tabelle) non sarebbero efficienti in memoria secondaria:
 - servono strutture fisiche opportune



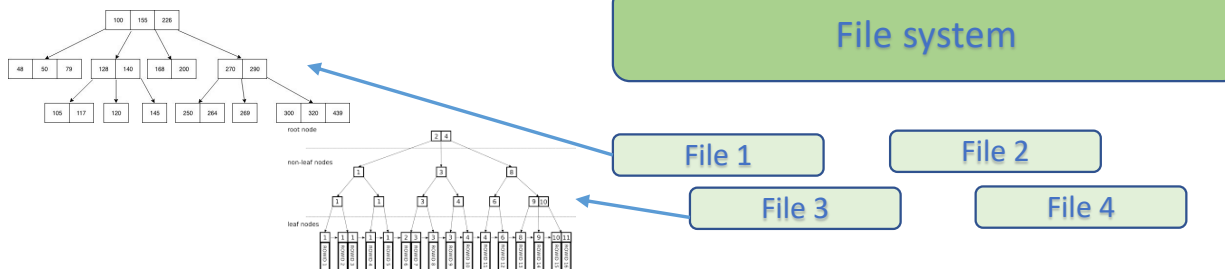
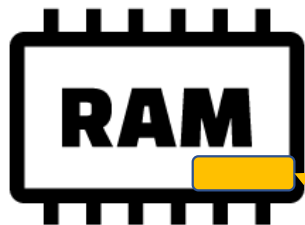
Le basi di dati sono grandi e persistenti

- La memoria secondaria è molto più lenta della memoria principale:
 - serve un'interazione fra memoria principale e secondaria che limiti il più possibile gli accessi alla secondaria



Le basi di dati vengono interrogate ...

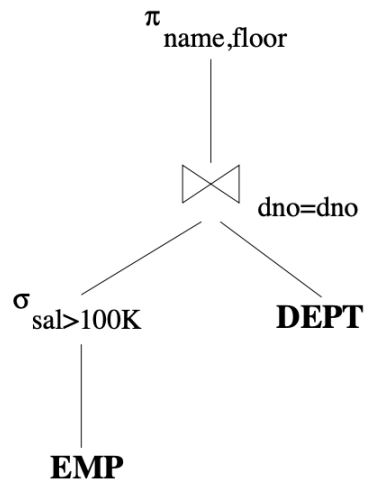
- Gli utenti vedono il modello logico (relazionale), esprimono le query in SQL (linguaggio dichiarativo), che devono essere tradotte in accessi alle memorie



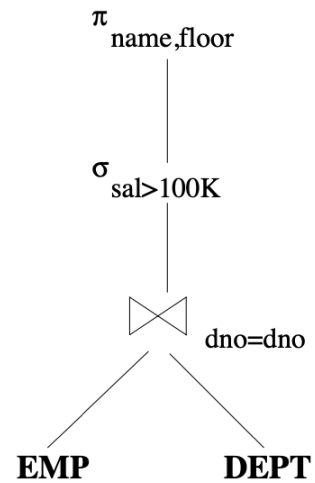
```
SELECT name, floor  
FROM emp, dept  
WHERE emp.dno=dept.dno AND sal>100
```

?

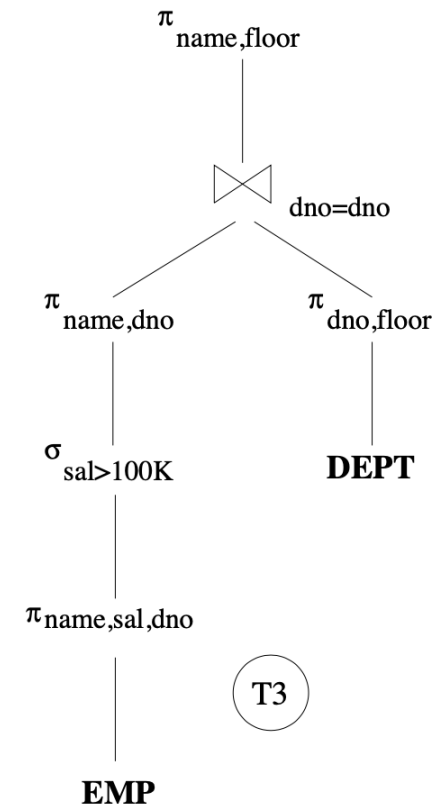
SELECT name, floor
 FROM emp, dept
 WHERE emp.dno=dept.dno AND sal>100



T1

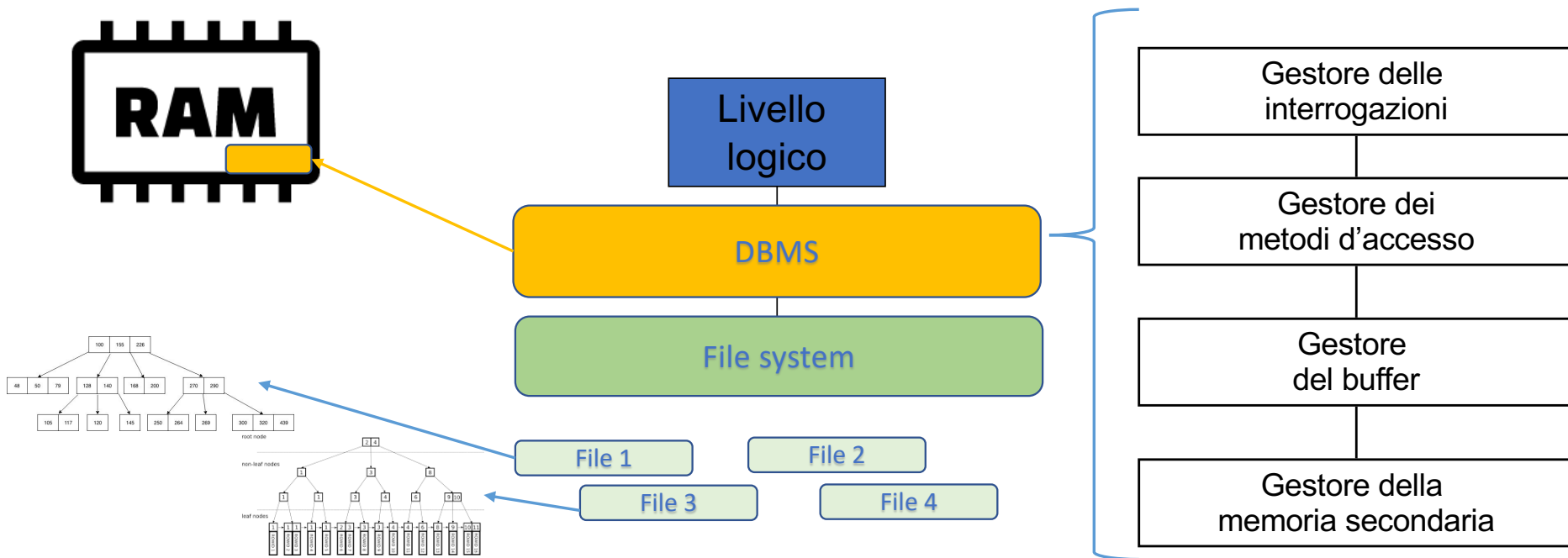


T2



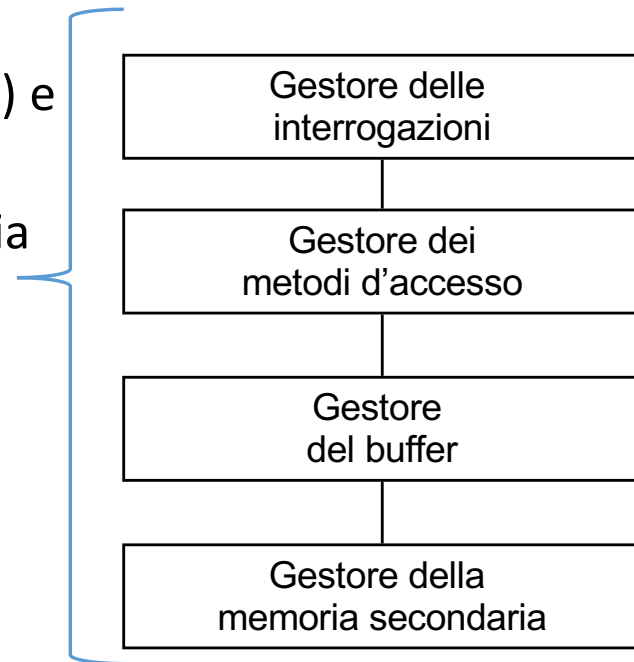
T3

DataBase Management System — DBMS



DataBase Management System — DBMS

- Traduce le query in forma interna (**algebra relazionale**) e le trasforma per renderle più efficienti
- Le traduce in operazioni di lettura/scrittura in memoria secondaria
- Rende le operazioni di lettura/scrittura più efficienti andando a memorizzare una parte del db in memoria centrale



DataBase Management System — DBMS

- **prodotto software** in grado di gestire **collezioni di dati** che siano (anche):

- **grandi**
- **persistenti**
- **condivise**

garantendo

- **affidabilità**
- **confidenzialità**

- Come ogni prodotto informatico, un DBMS deve essere:
 - **efficiente**
 - **efficace**

Le basi di dati sono affidabili

- Le basi di dati sono una risorsa per chi le possiede, e debbono essere conservate anche in presenza di malfunzionamenti
- L'affidabilità è impegnativa per via degli aggiornamenti frequenti e della necessità di gestire il buffer
- Transizione: unità di elaborazione che gode delle proprietà "ACIDE"
 - Atomicità
 - Consistenza
 - Isolamento
 - Durata (persistenza)

Esempio transazione

start transaction;

update ContoCorrente

set Saldo = Saldo + 10 where NumConto = 12202;

update ContoCorrente

set Saldo = Saldo – 10 where NumConto = 42177;

select Saldo into A

from ContoCorrente

where NumConto = 42177;

if (A>=0) then commit work

else rollback work;



- **Atomicità**

- La transazione è una unità atomica di elaborazione
 - Non può lasciare la base di dati in uno stato intermedio

- **Consistenza**

- La transazione rispetta i vincoli di integrità

- **Isolamento**

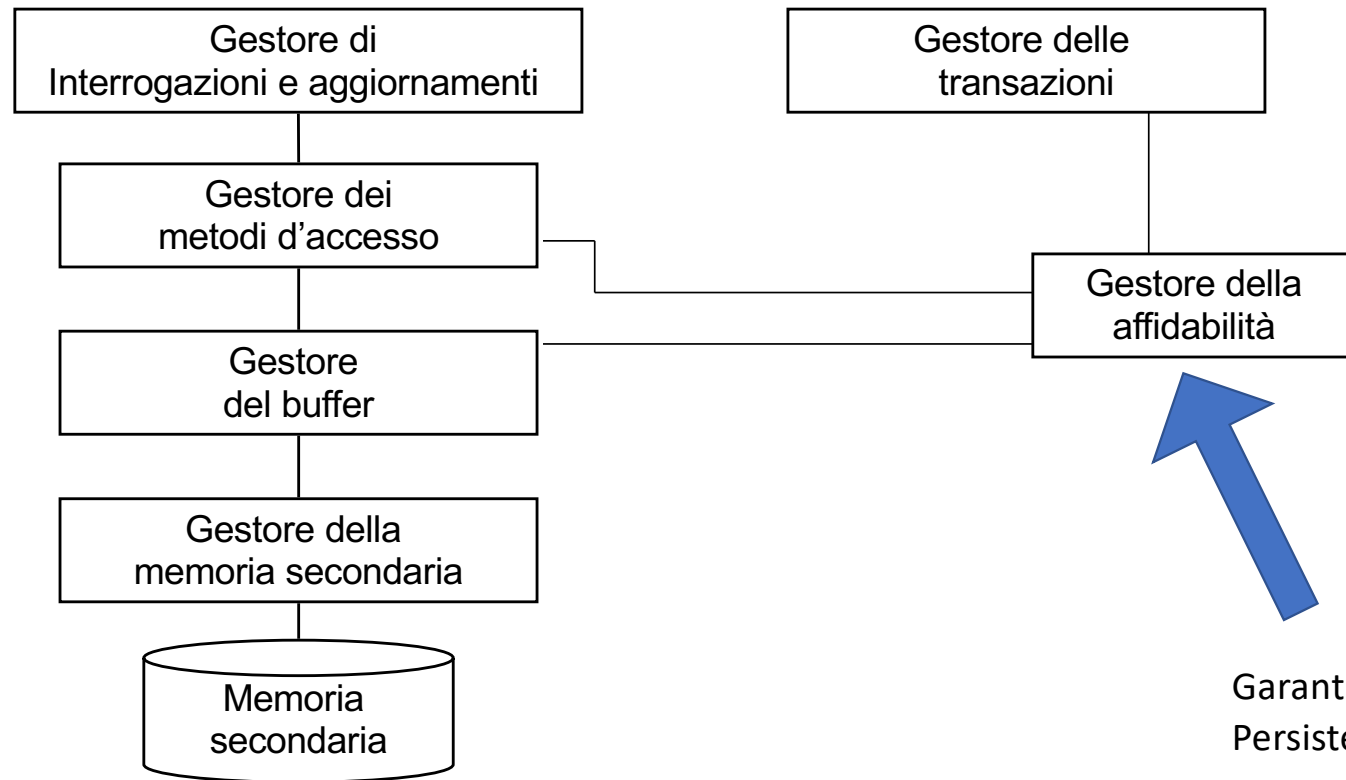
- La transazione non risente degli effetti delle altre transazioni concorrenti
 - l'esecuzione concorrente di una collezione di transazioni deve produrre un risultato che si potrebbe ottenere con una esecuzione sequenziale

- **Persistenza**

- Gli effetti di una transazione andata in commit non devono andare perduti ("durano per sempre")

Gestore degli accessi e delle interrogazioni

Gestore delle transazioni



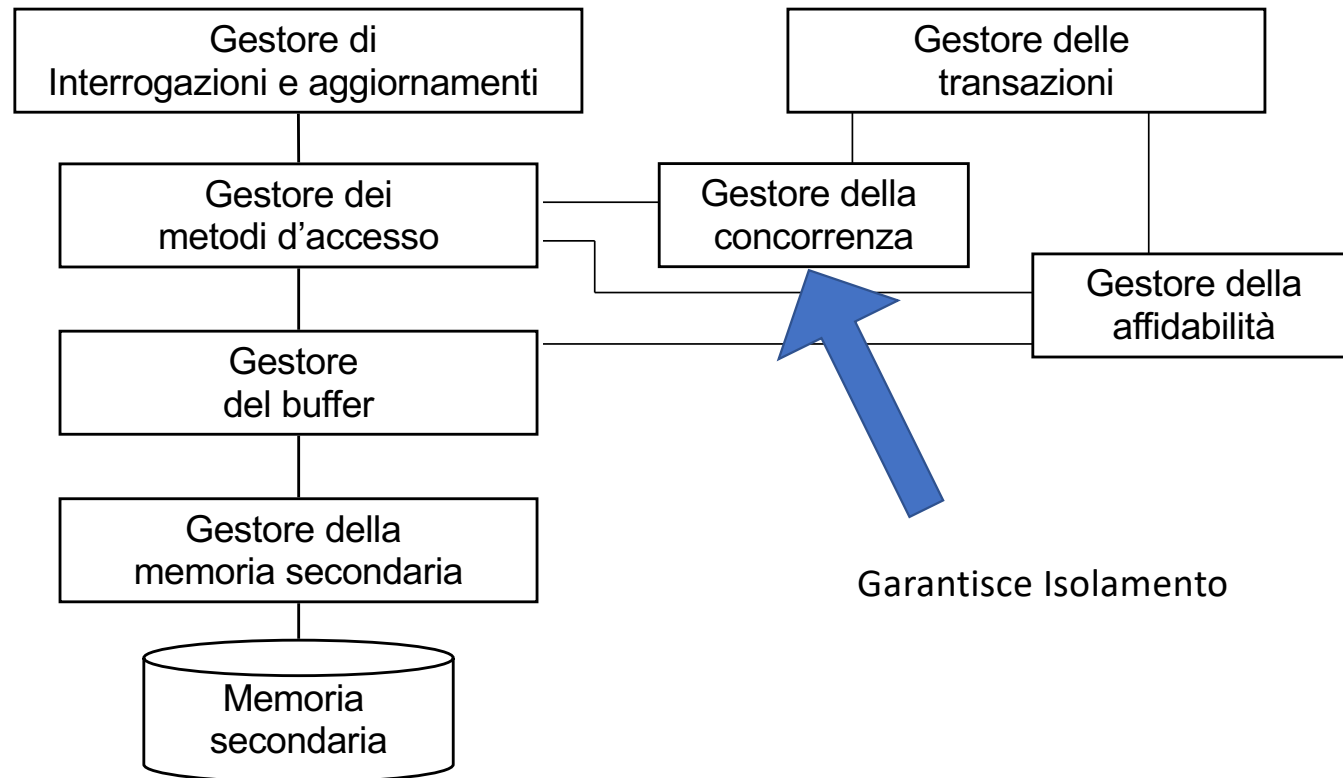
Garantisce Atomicità e Persistenza

Le basi di dati sono **condivise**

- Una base di dati è una risorsa **integrata, condivisa** fra le varie applicazioni
- Attività multi-utente su dati condivisi:
 - Esempio:
 - due prelevamenti (quasi) contemporanei sullo stesso conto corrente
- Intuitivamente, le transazioni sono corrette se **seriali** (prima una e poi l'altra)
- Ma in molti sistemi reali l'efficienza sarebbe penalizzata troppo se le transazioni fossero seriali:
 - il **controllo della concorrenza** permette un ragionevole compromesso

Gestore degli accessi e delle interrogazioni

Gestore delle transazioni



Basi dati II: programma (to be cont'd)

- Gestione della memoria secondaria e del buffer
- Organizzazione fisica dei dati
- Gestione ("ottimizzazione") delle interrogazioni
- Controllo della affidabilità
- Controllo della concorrenza

Basi dati II: programma (to be cont'd)

- Algebra relazionale
- Gestione della memoria secondaria e del buffer
- Organizzazione fisica dei dati
- Gestione ("ottimizzazione") delle interrogazioni
- Controllo della affidabilità
- Controllo della concorrenza

Obiettivi del corso

Al termine del corso, lo studente sarà in grado di:

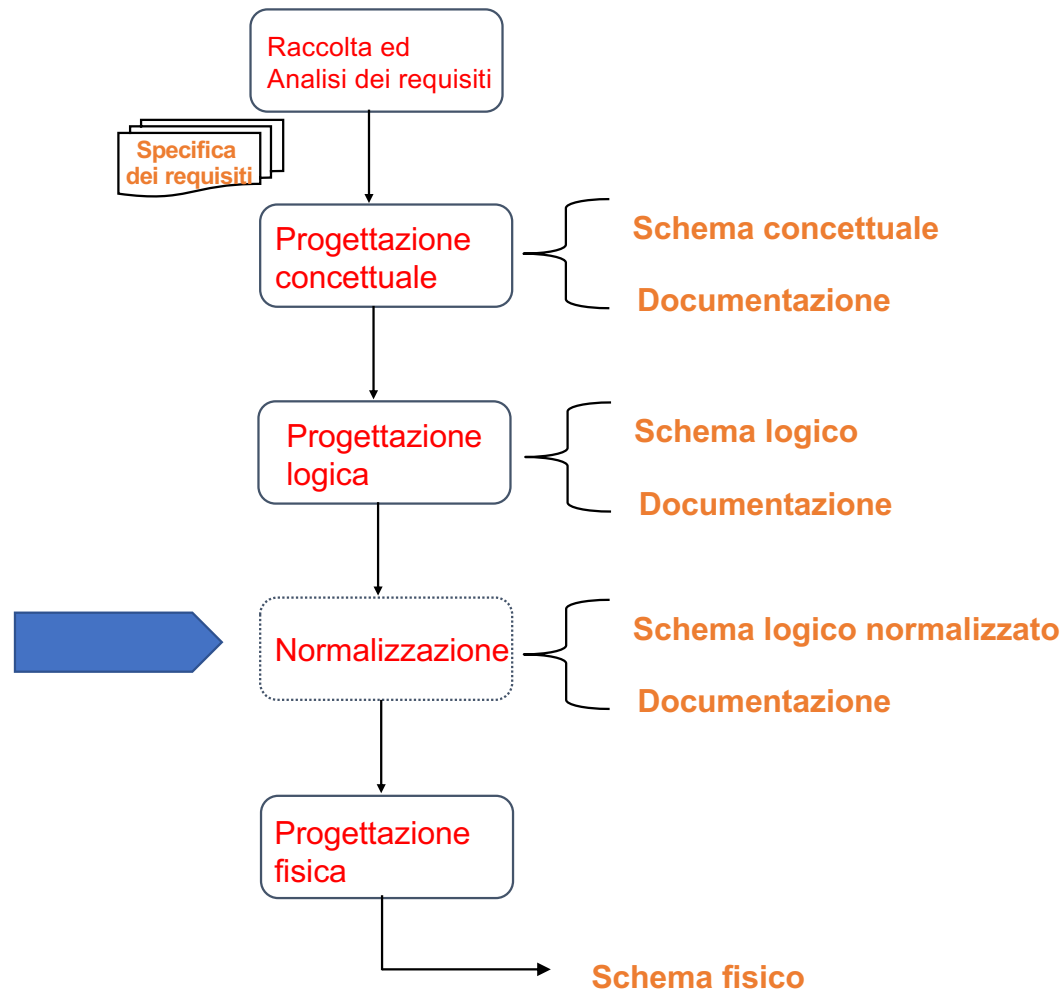
- O1. Conoscere le principali strutture fisiche e relative gestioni (indici, metodi di accesso) utilizzati nei DBMS.
- O2. Essere in grado di ottimizzare la progettazione e le interrogazioni da eseguire.
- O3. Riconoscere e gestire problemi legati alla gestione di transazioni
- O4. Conoscere i principali modelli di dati NoSQL.

Obiettivi del corso

Al termine del corso, lo studente sarà in grado di:

- O1. Conoscere le principali strutture fisiche e relative gestioni (indici, metodi di accesso) utilizzati nei DBMS.
- O2. Essere in grado di ottimizzare la progettazione e le interrogazioni da eseguire.
- O3. Riconoscere e gestire problemi legati alla gestione di transazioni
- O4. Conoscere i principali modelli di dati NoSQL.

Fasi della progettazione



Normalizzazione

- Verifica di qualità dello schema logico prodotto, effettuata tramite opportuni strumenti formali
- Prende il nome di normalizzazione nel caso di basi di dati relazionali
- Una forma **normale** è una proprietà di una base di dati relazionale che ne garantisce la “qualità”, cioè l'assenza di determinate **anomalie**
- Quando una relazione non è normalizzata:
 - presenta ridondanze,
 - si presta a comportamenti poco desiderabili durante gli aggiornamenti

Esempio: Una relazione con anomalie

<u>Impiegato</u>	Stipendio	<u>Progetto</u>	Bilancio	Funzione
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	55	Venere	15	direttore
Neri	55	Giove	15	consulente
Neri	55	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

Anomalie

- Lo stipendio di ciascun impiegato è ripetuto in tutte le ennuple relative
 - **ridondanza**
- Se lo stipendio di un impiegato varia, è necessario andarne a modificare il valore in diverse ennuple
 - **anomalia di aggiornamento**
- Se un impiegato interrompe la partecipazione a tutti i progetti, dobbiamo cancellarlo
 - **anomalia di cancellazione**
- Un nuovo impiegato senza progetto non può essere inserito
 - **anomalia di inserimento**

Basi dati II: programma (to be cont'd)

- Algebra relazionale
- Gestione della memoria secondaria e del buffer
- Organizzazione fisica dei dati
- Gestione ("ottimizzazione") delle interrogazioni
- Controllo della affidabilità
- Controllo della concorrenza
- Normalizzazione
- Introduzione NoSQL

esercizi

esercizi

esercizi

esercizi

esercizi

esercizi

esercizi

Modalità d'esame

- L'esame consiste in una prova scritta della durata di 2 ore in cui lo studente è tenuto a rispondere a domande aperte e/o esercizi.