



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Introduzione ai DBMS

Basi di Dati

Corso di Laurea in Informatica per il Management

Alma Mater Studiorum - Università di Bologna

Prof. Marco Di Felice

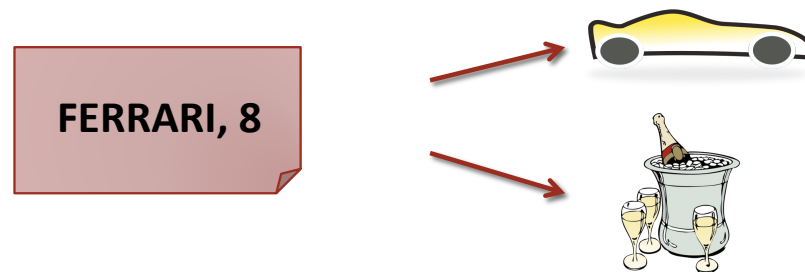
Dipartimento di Informatica – Scienza e Ingegneria

marco.difelice3@unibo.it

Dato vs Informazione

- **Informazione** (def): notizia, o elemento che consente di avere conoscenza piu' o meno esatta di fatti, situazioni e modi d'essere.
- **Dato** (def): elementi di informazione costituiti da simboli che devono essere elaborati.

NOTA: Senza conoscenza del contesto, il dato non è molto utile!



Dato vs Informazione

ESEMPI DI TIPOLOGIE DI DATO



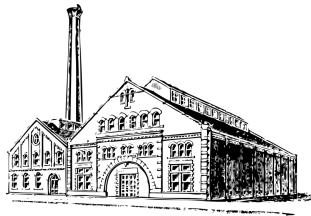
- **Numerico** (intero) → 40
- **Numerico** (virgola mobile) → 40.1212
- **Booleano** → True
- **Data** → 21/10/2022 15:00:00
- **Stringa** (sequenza di caratteri) → 'Basi di dati'
- **Testo** → 'bla bla bla'
- **Binario** (sequenza di 0/1) → 0100010010



Sistemi Informativi

Un **Sistema Informativo (SI)** è una componente di un'organizzazione il cui scopo è quello di **gestire le informazioni** utili ad i fini dell'organizzazione stessa.

Organizzazione → Azienda, Ufficio, Ente, Università, etc



Sistemi Informativi

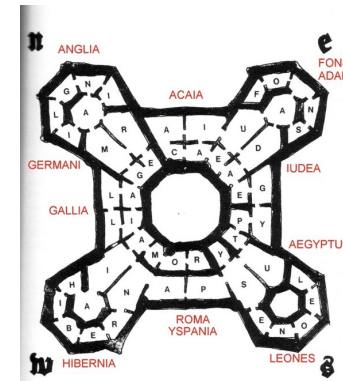
L'esistenza di un **Sistema Informativo** è indipendente dalla sua automatizzazione.



Biblioteca reale di Alessandria d'Egitto

IV-I secolo a.C

400000 rotoli presenti.



Biblioteche medievali

Sistemi Informativi

L'esistenza di un **Sistema Informativo** è indipendente dalla sua automatizzazione.

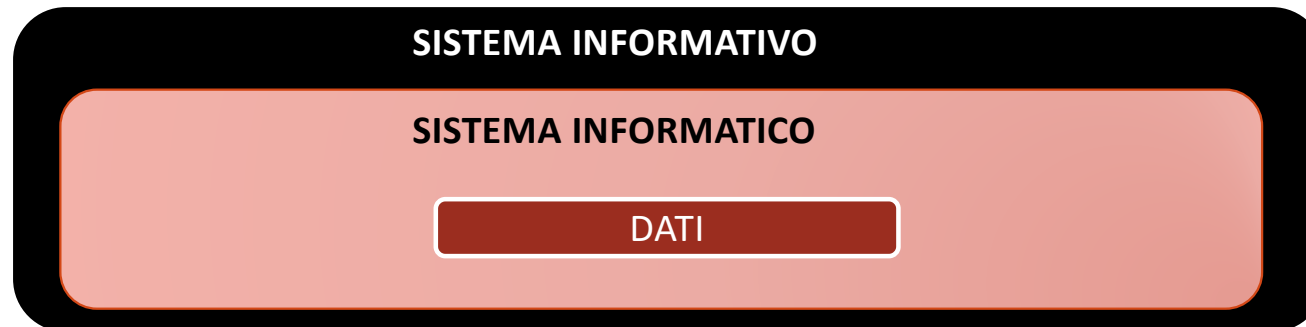
Censimenti e Registro Anagrafe.



Nell'Antica Roma, i **censimenti** venivano effettuati dalla fine del IV secolo a.c. Gli **elenchi dei censiti**, distinti secondo il possesso o meno dei diritti civili e politici, la classe patrimoniale e l'età, venivano utilizzati come liste elettorali e per determinare la ruoli per l'esenzione dei tributi e le liste di leva.

Sistemi Informativi

- La porzione **automatizzata** di un sistema informativo prende il nome di **Sistema Informatico**.
- All'interno di un sistema informatico, le informazioni sono rappresentate da **dati** ...



Approcci di gestione dei dati

Gran parte dei sistemi informatici ha necessità di gestire **dati in maniera persistente**.



Persistente → Dati memorizzati su memoria secondaria

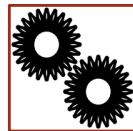
APPROCCI di GESTIONE

- Approccio **convenzionale** (basato su software non specializzati)
- Approccio **specializzato** (basato su software di gestione dei dati)

Approccio convenzionale

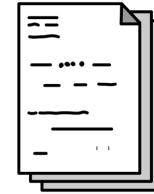
Approccio **Convenzionale** (basato su **software non specializzati**)

APPLICAZIONE



----->
Operazioni di **Lettura/Scrittura**
su file mediante supporto del
Sistema Operativo

FILES



- Nessuna chiara distinzione tra **dati ed applicazioni**.
- L'applicazione contiene al suo interno la **logica di gestione e memorizzazione** dei dati stessi (es. formato interno dei dati).
- Il **Sistema Operativo** offre le primitive di base per l'accesso ai files ed i meccanismi di sicurezza del **file-system**.

Approccio convenzionale

Approccio **Convenzionale** (basato su **software non specializzati**)

PROBLEMA1: Gestione di **grandi quantità di dati**?

Qualche esempio “estremo”:



> 200 milioni di abbonati



> 2 miliardi di utenti iscritti



> 50 miliardi di foto pubblicate

Ovvi problemi di **scalabilità** ...

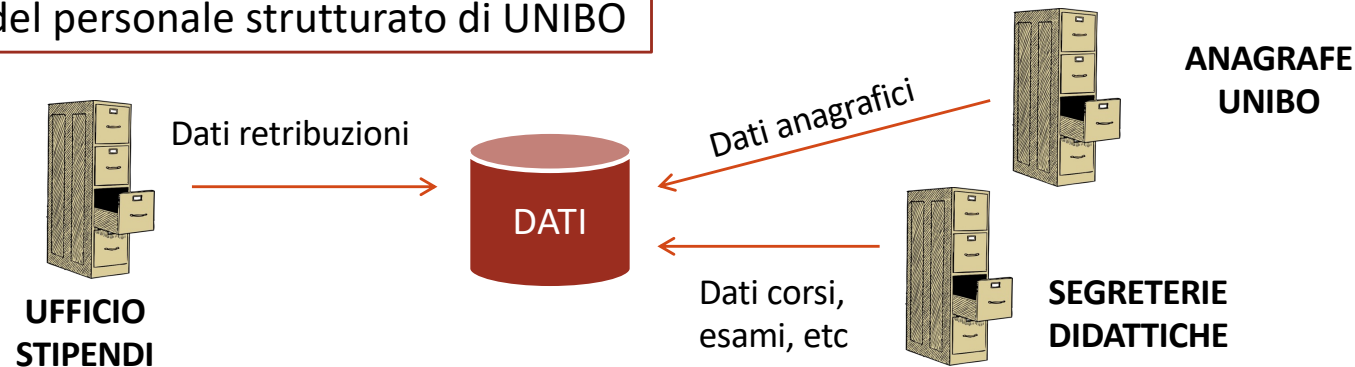
Approccio convenzionale

Approccio **Convenzionale** (basato su **software non specializzati**)

PROBLEMA2: **Condivisione** ed **accesso** concorrente?

In molti scenari pratici, i dati devono essere a disposizione di una moltitudine di utenti/applicazioni per accessi concorrenti.

Es. Dati del personale strutturato di UNIBO



Approccio convenzionale

Approccio **Convenzionale** (basato su **software non specializzati**)

- L'accesso a file condivisi avviene attraverso le politiche di accesso del file-system → Alcune applicazioni implementano il lock a livello di file, **bassa granularità di concorrenza**, prestazioni limitate!
- Applicazioni diverse devono conoscere l'esatta collocazione e formato dei dati → **Aggiornamento del formato dei dati?**
- In alternativa: replica dei dati presso i vari sistemi/utenti che ne fanno utilizzo → **Consistenza delle repliche?**

Approcci di gestione dei dati

Gran parte dei sistemi informatici ha necessità di gestire **dati in maniera persistente**.



Persistente → Dati memorizzati su memoria secondaria

APPROCCI di GESTIONE

- Approccio **convenzionale** (basato su software non specializzati)
- Approccio **specializzato** (basato su software di gestione dei dati)

Caratteristiche dei DBMS

Un **DBMS** è un sistema software che è in grado di gestire collezioni di dati *grandi, condivise e persistenti, in maniera efficiente e sicura.*

(ALCUNE) FUNZIONALITA':

- Creazione di una collezione di dati e sua memorizzazione su memoria secondaria
- Accesso in lettura/scrittura ad i dati
- Condivisione di dati tra diversi utenti/applicazioni
- Protezione dei dati da accessi non autorizzati
- Reliability dei dati in caso di guasti (hardware/software)
- ...

Caratteristiche dei DBMS

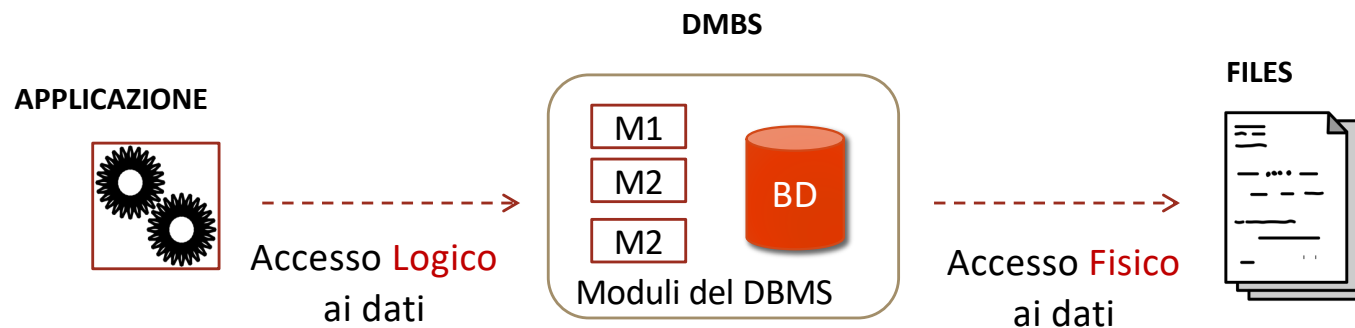
Un **DBMS** è un sistema software che è in grado di gestire collezioni di dati *grandi, condivise e persistenti, in maniera efficiente e sicura.*

(ALCUNE) FUNZIONALITA':

- Creazione di una collezione di dati e sua memorizzazione su memoria secondaria
- Accesso in lettura/scrittura ad i dati
- Co
- Pro
- Reliability dei dati in caso di guasti (hardware/software)
- ...

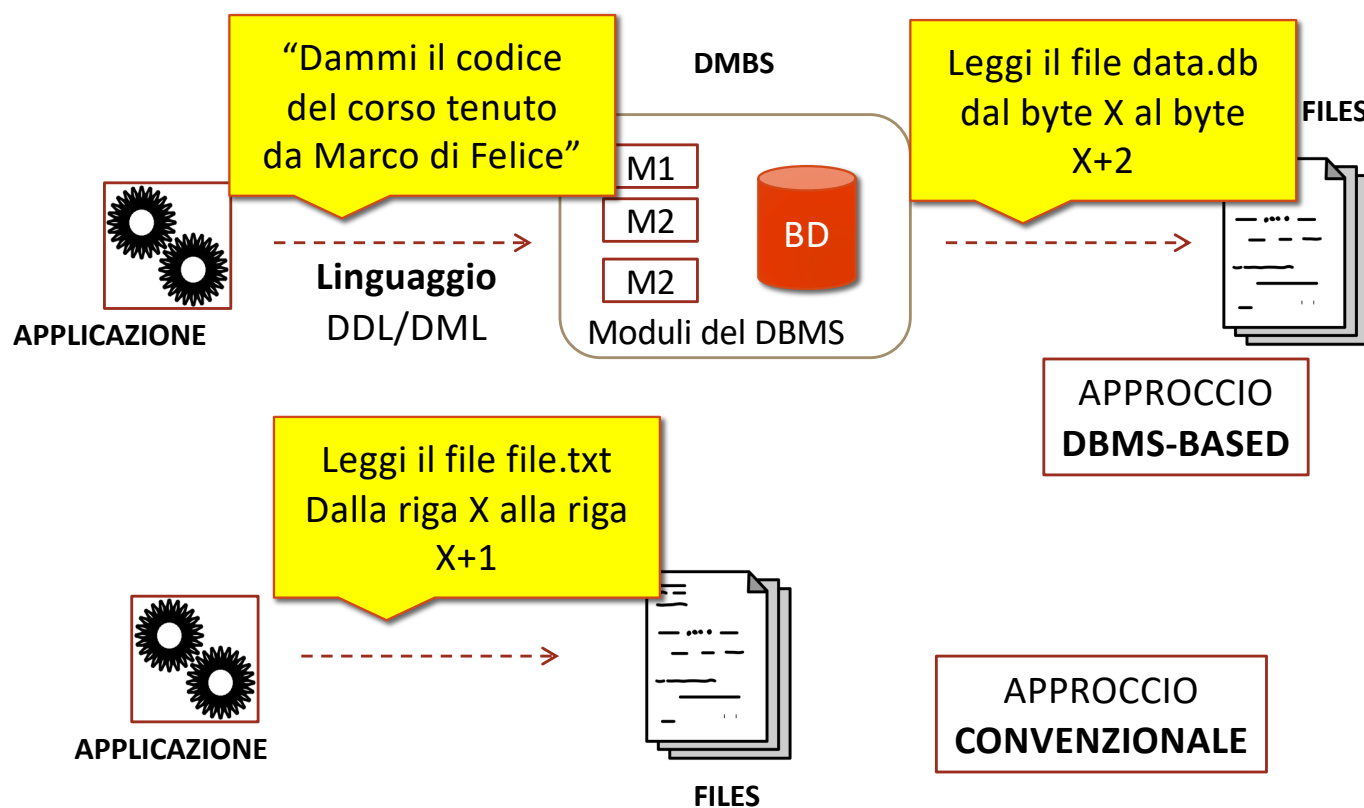
Da qui in avanti nel corso, una base di dati è una collezione di dati gestita da un DBMS!

Separazione Dati/Applicazioni con DBMS



- Tramite i DBMS, è possibile implementare un **paradigma di separazione di dati ed applicazioni** ...
- Le applicazioni non necessitano di conoscere la struttura fisica dei dati (es. come e dove sono memorizzati su disco) ma **solo la struttura logica** (cosa rappresentano).

Separazione Dati/Applicazioni con DBMS



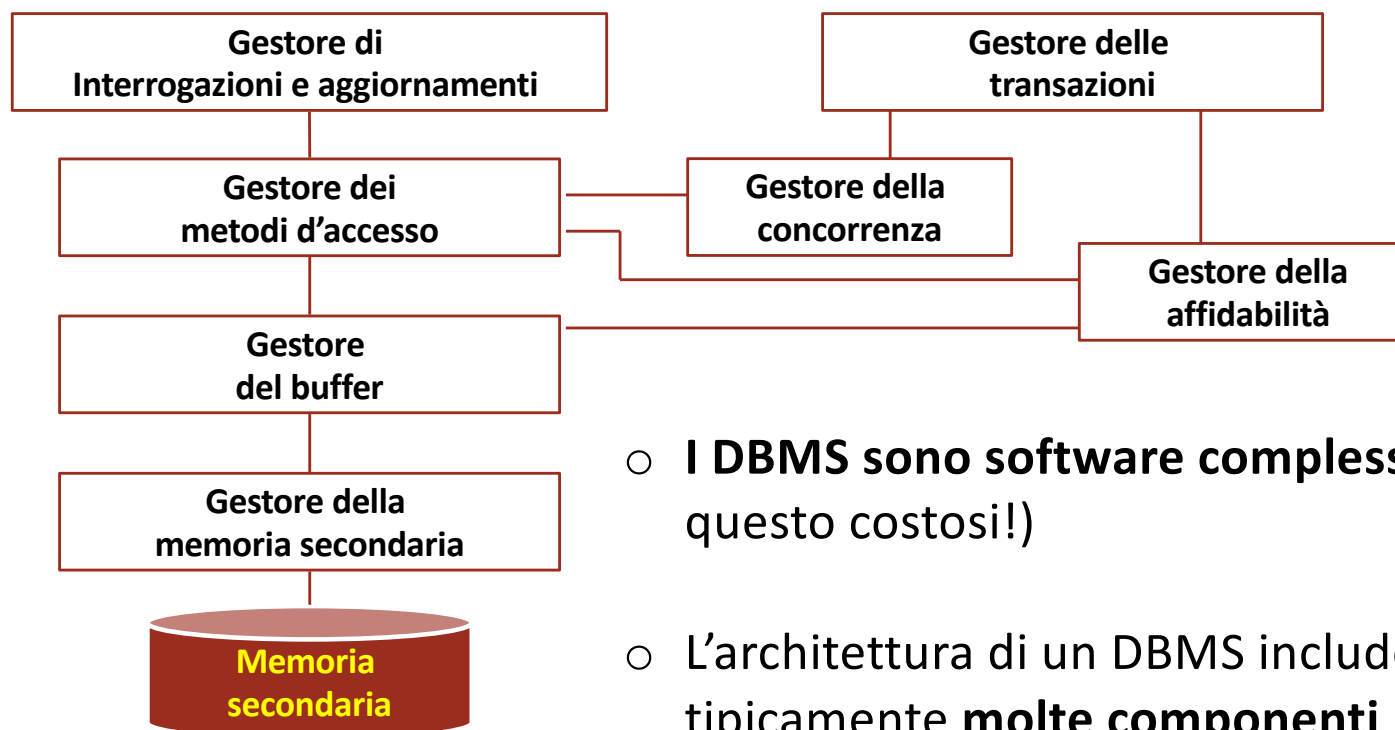
DBMS: Quali?

Lista (parziale) dei DBMS più popolari (sorgente: Wikipedia)

Principali sistemi esistenti [\[modifica \]](#) [modifica wikitesto \]](#)

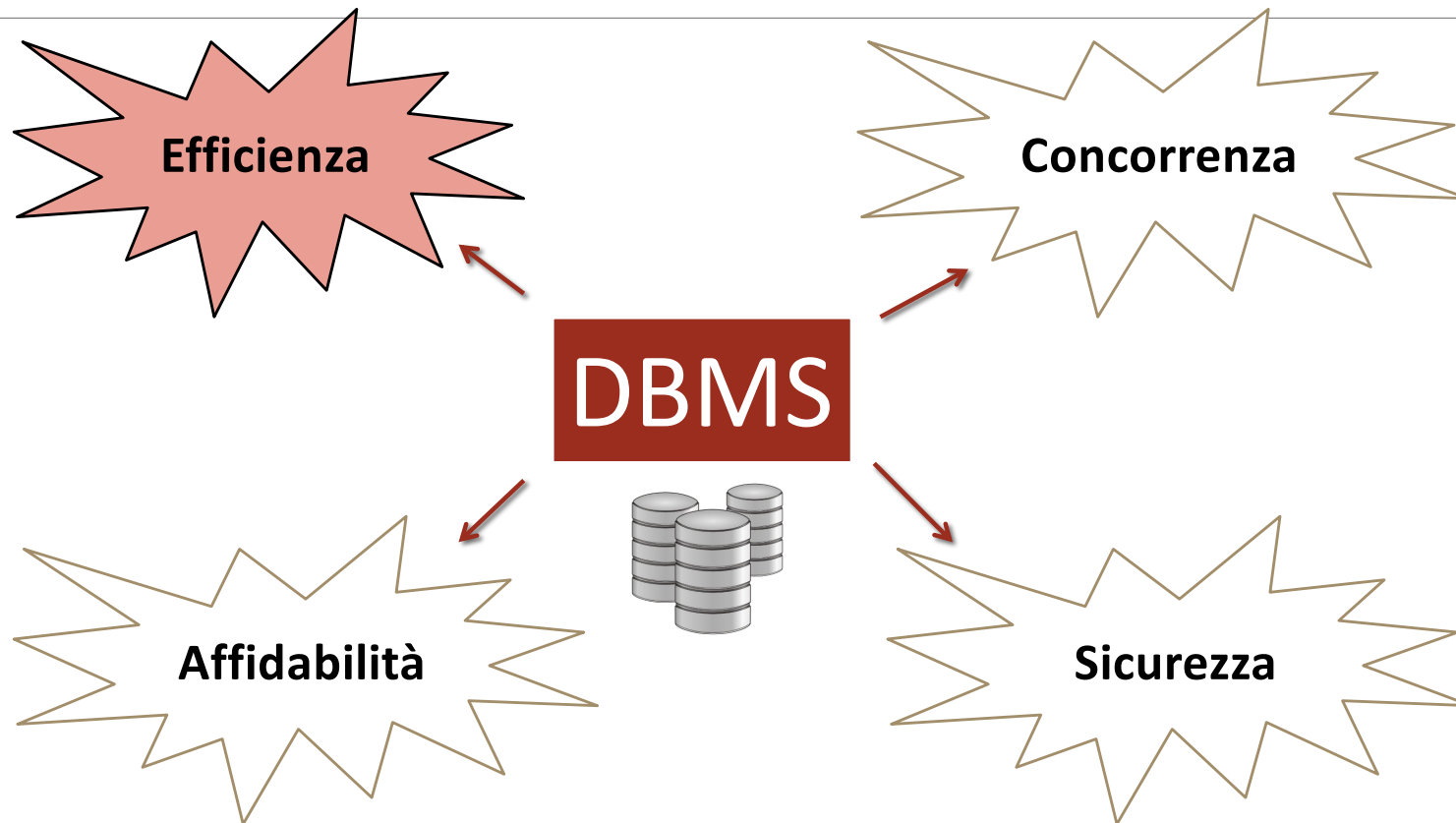
- **Software** proprietario
 - 4th Dimension
 - CUBRID
 - dBase
 - IBM DB1/2
 - Caché (InterSystems)
 - Dataphor un RDBMS "veramente-relazionale" che rispetta tutte le regole di Codd.
 - FileMaker Pro (FileMaker Inc., posseduta da Apple)
 - Informix (IBM)
 - INGRES (Computer Associates)
 - InterBase (Borland)
 - Microsoft SQL Server (Microsoft)
 - Microsoft Access (Microsoft)
 - mSQL
 - Netezza
 - Oracle (Oracle Corporation)
 - Paradox (Borland)
 - SQL Anywhere Studio
 - Sybase SQL Server
 - Teradata (NCR Corporation)
 - TimesTen (TimesTen, Inc.)
- **Open source** o **free software**
 - Apache Derby
 - Berkeley DB (Sleepycat Software)
 - Drizzle
 - Firebird SQL (The FirebirdSQL Foundation; derivato da InterBase)
 - GigaBASE
 - HSQL Database Engine
 - HyperNetDatabase
 - MaxDB (prima SAP DB e Adabas)
 - Metakit
 - MySQL (MySQL AB)
 - MariaDB (MariaDB Foundation)
 - Percona Server (Percona)
 - Orient ODBMS
 - Ozone
 - PicoSQL (un Dbms italiano)
 - PostgreSQL (in precedenza *Postgres*) (PostgreSQL Global Development Group)
 - SQLite (di pubblico dominio)
 - Visual FoxPro (FoxBase)
 - ZODB
 - SADAS

Componenti di un DBMS



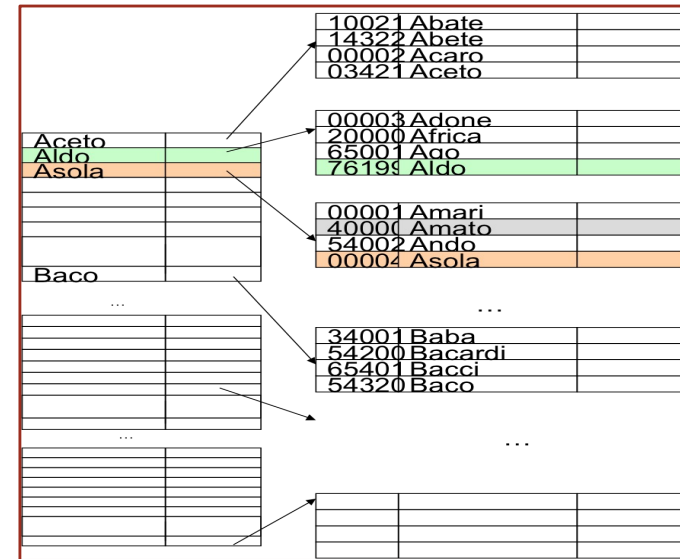
- **I DBMS sono software complessi** (e per questo costosi!)
- L'architettura di un DBMS include tipicamente **molte componenti ...**

Componenti di un DBMS



Caratteristiche di un DBMS: Efficienza

- DBMS forniscono adeguate **strutture dati** per organizzare i dati all'interno dei file, e per supportare le operazioni di ricerca, aggiornamento, inserimento, modifica (operazioni **CRUD**).
- In genere, parliamo di **strutture dati ad albero o tabelle hash**.



Caratteristiche di un DBMS: **Efficienza**

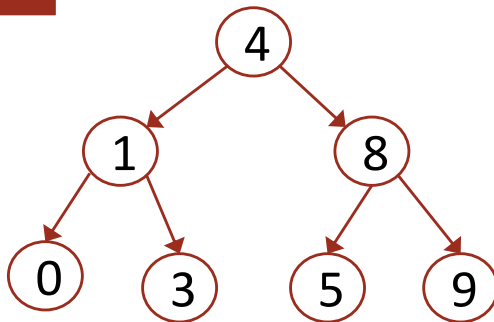
Indice → struttura che contiene **informazioni sulla posizione dei contenuti di interesse** sulla base del valore di un campo **chiave**.



Caratteristiche di un DBMS: **Efficienza**

Efficienza di un DBMS nella gestione dei dati ...

B-tree

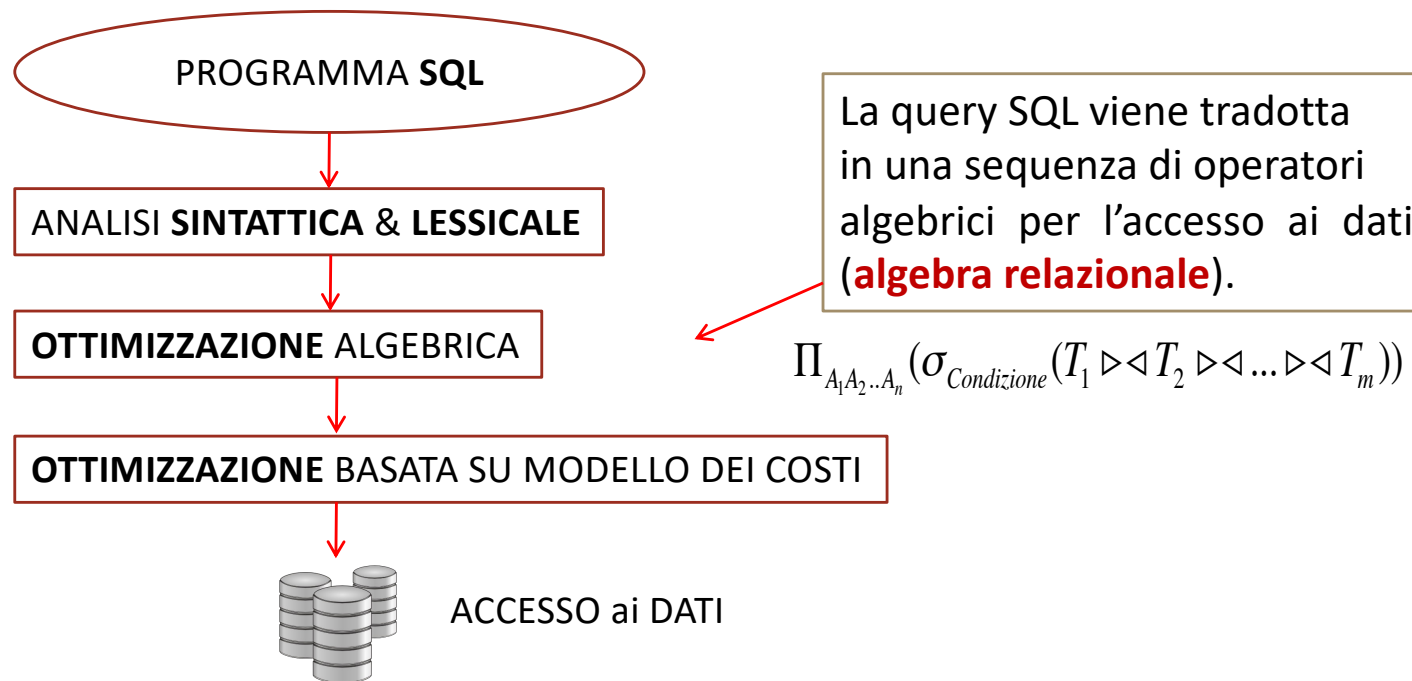


- **Ricerca** → $O(\log(N))$
- **Inserimento** → $O(\log(n))$
- **Cancellazione** → $O(\log(n))$

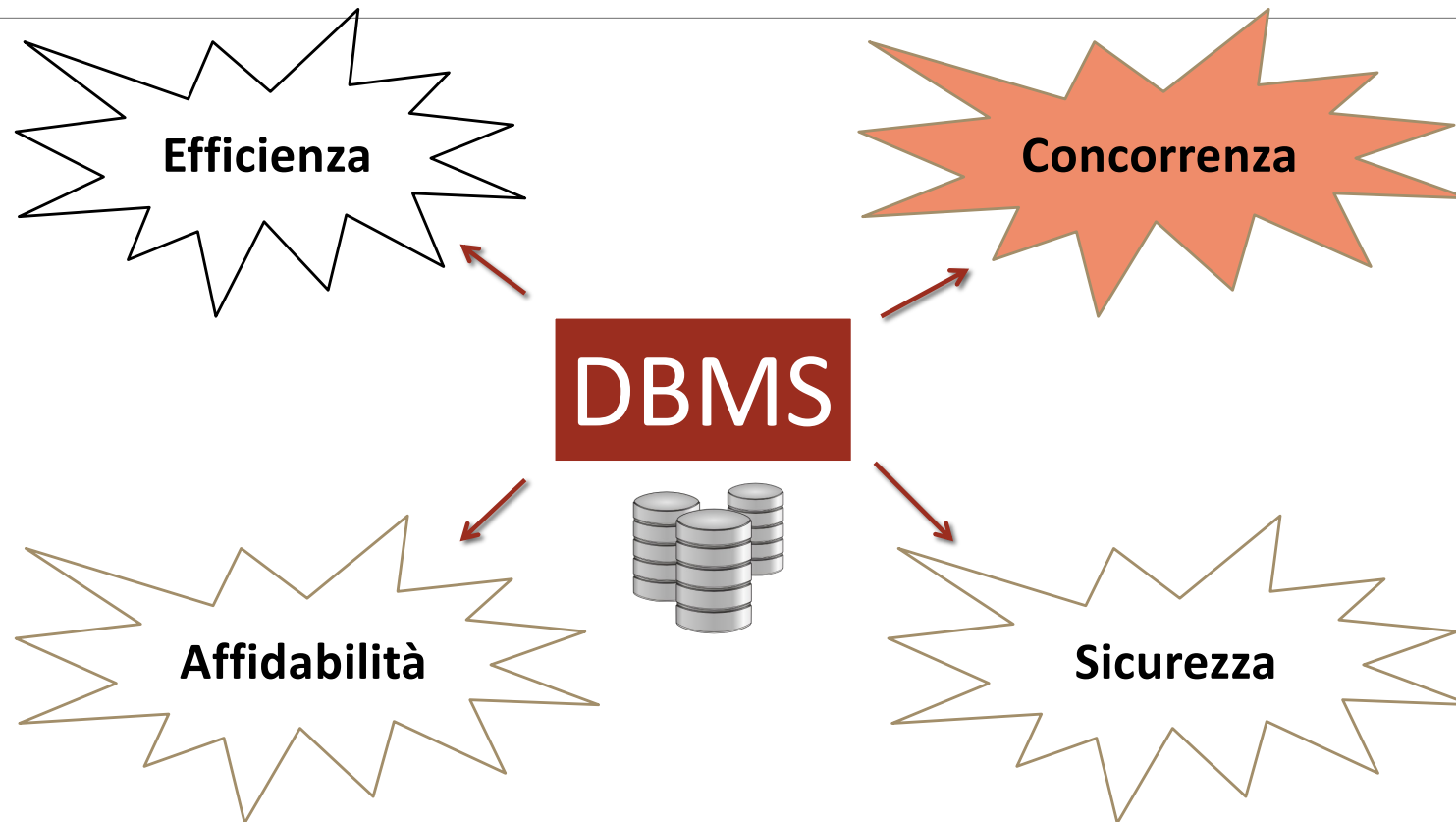
- Nel contest dei DBMS, le strutture ad albero dinamiche di tipo B (B-tree) e B+ (B+-tree) sono quelle più frequentemente utilizzate per la realizzazione di indici.

Caratteristiche di un DBMS: **Efficienza**

- **Ottimizzazione** di operazioni di ricerca (**interrogazioni**)



Componenti di un DBMS



Caratteristiche di un DBMS: Concorrenza

- In molti sistemi è fondamentale **gestire operazioni concorrenti** di accesso ai dati ...



Processing di più di 7.7 milioni
di pagamenti ogni giorno

La maggior parte dei DBMS forniscono un **livello di granularità di locking** più fine di quello convenzionale (a livello di tabella, pagina, o singola entry).

Caratteristiche di un DBMS: Concorrenza

Al tempo stesso, un DBMS deve garantire il fatto che **accessi da parte di applicazioni diverse non interferiscano tra loro**, lasciando il sistema in uno **stato inconsistente** ...

Es. Sistema informativo dei conti bancari

- 2 richieste da gestire al tempo t:
 - Prelievo di 100 euro dal conto X
 - Prelievo di 80 euro dal conto X
- Saldo del conto X at tempo t: 120 euro
- Quale deve essere il risultato atteso?

Caratteristiche di un DBMS: Concorrenza

ESEMPIO di ESECUZIONE (**non corretta!!**)

OP1

Leggi X da disco
Calcola X-100
Scrivi X su disco

OP2

Leggi X da disco
Calcola X-80
Scrivi X su disco

Schedule:

OP1: Leggi X
OP2: Leggi X
OP1: Calcola X-100
OP2: Calcola X-80
OP1: Scrivi X
OP2: Scrivi X

Valore X:

120
120
120
120
20
40 (????)

Per prevenire tali situazioni, i DBMS implementano **algoritmi di controllo della concorrenza** che garantiscono che **operazioni sui dati (transazioni) concorrenti producano lo stesso risultato di un'esecuzione seriale.**

Caratteristiche di un DBMS: Concorrenza

Lock Manager → componente del DBMS responsabile di **gestire i lock alle risorse del DB, e di implementare le transazioni.**

OP1

Lock(x)

Leggi X

Calcola X-100

Scrivi X

Unlock(x)

OP2

Lock(x)

Leggi X

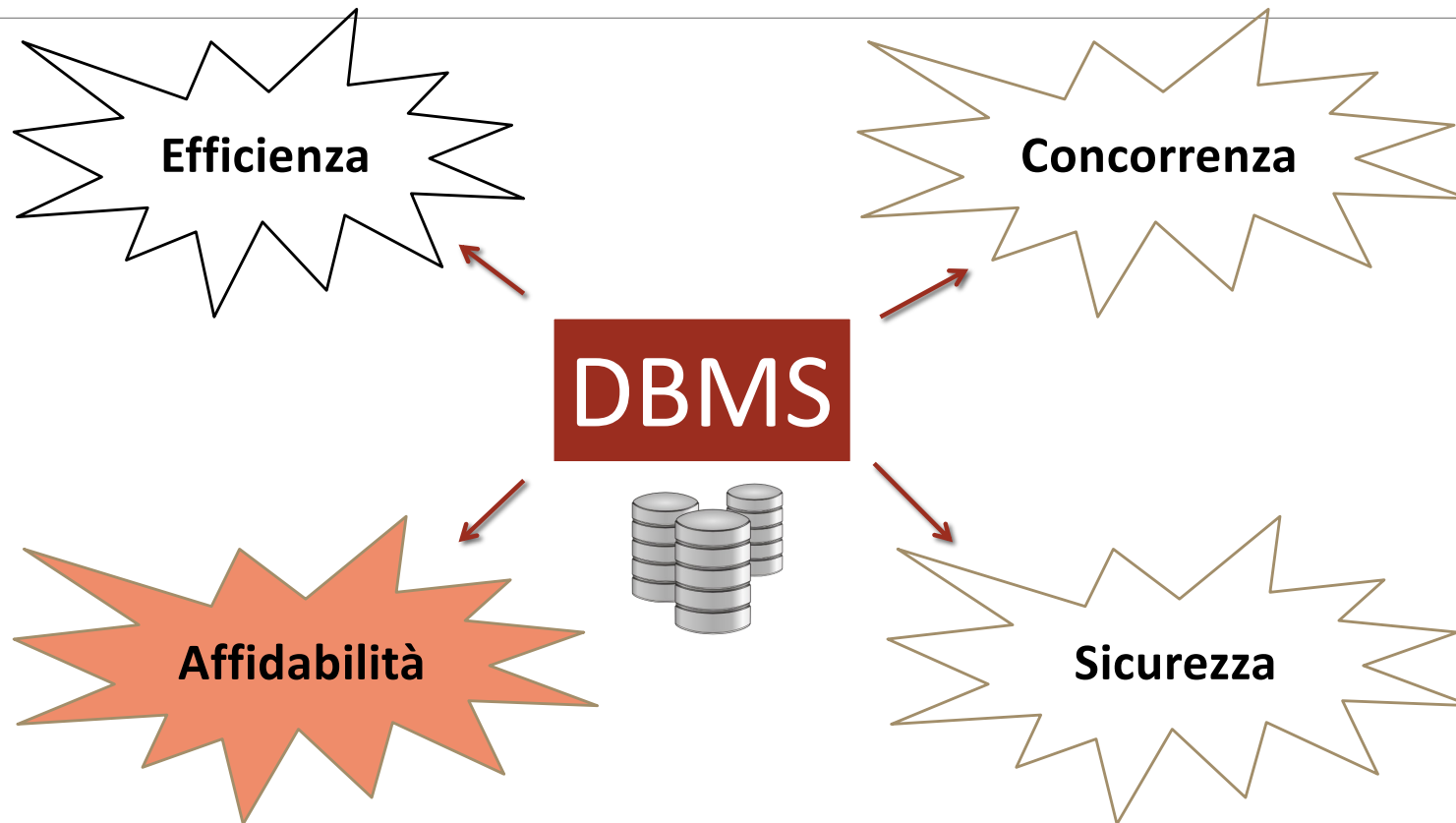
Calcola X-80

Scrivi X

Unlock(x)

- Utilizzo di **lock** in lettura/scrittura per accesso a risorse condivise (dati).
- Algoritmi (**2FL**, **S2FL**) per gestire l'ordine di acquisizione dei lock.

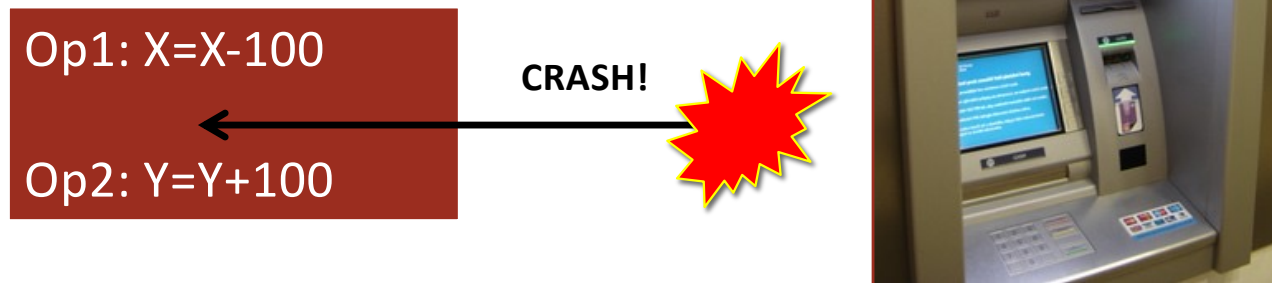
Componenti di un DBMS



Caratteristiche di un DBMS: **Atomicità**

Alcune operazioni sui dati sono **particolarmente delicate**, e devono essere gestite in maniera opportuna, secondo la regola del **tutto o niente**.

Es. Trasferimento di denaro (100\$) dal conto X al conto Y.



Caratteristiche di un DBMS: **Atomicità**

- Alcune operazioni sui dati sono **particolarmente delicate**, e devono essere gestite in maniera opportuna, secondo la regola del **tutto o niente**.



- Per questo, i DBMS devono fornire appositi strumenti per **annullare operazioni non completate** e fare **roll-back** dello stato del sistema ...

Caratteristiche di un DBMS: Affidabilità

In molti casi i DBMS mettono a disposizione appositi **strumenti ed algoritmi per garantire la persistenza dei dati** anche in presenza di **malfunzionamenti hardware/software**.

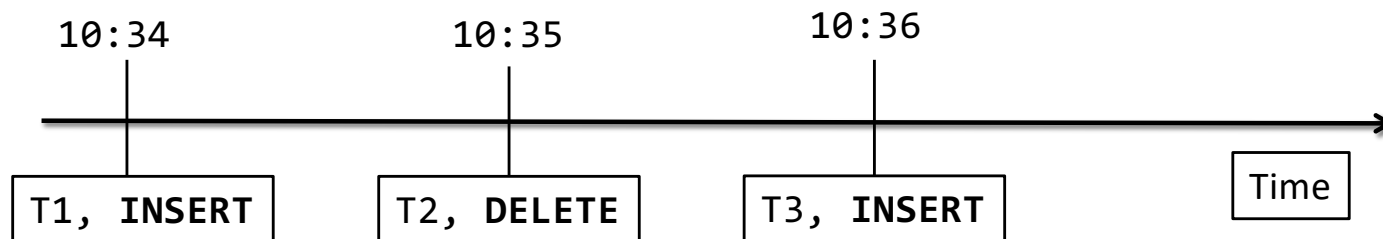
Il controllore di affidabilità utilizza dei file di **log**, nei quali sono indicate tutte le **operazioni svolte dal DBMS**.

- Algoritmi ad-hoc (es. algoritmo di **ripresa a caldo/a freddo**) per ripristinare lo stato dei dati a partire dai log del DBMS.

Caratteristiche di un DBMS: **Atomicità**

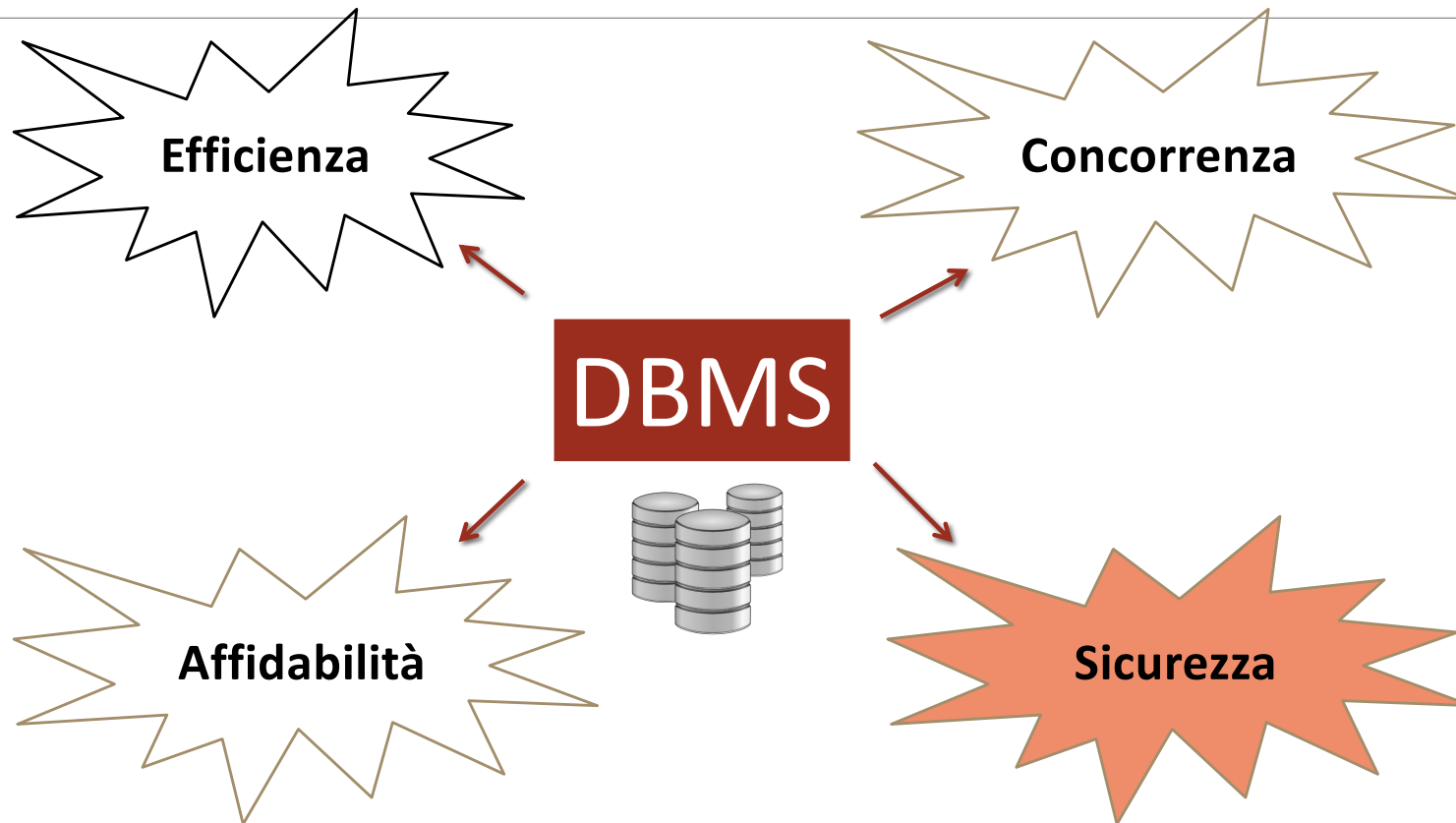
Il controllore di affidabilità utilizza file di **log**, nei quali sono indicate tutte le **operazioni svolte dal DBMS**.

LOG, struttura logica



Tramite il log, e' possibile fare **do/undo** delle operazioni...

Componenti di un DBMS



Caratteristiche di un DBMS: Multi-utenza

La maggior parte dei DBMS implementa **politiche di controllo degli accessi** ad i dati mediante **sistemi di permessi**:

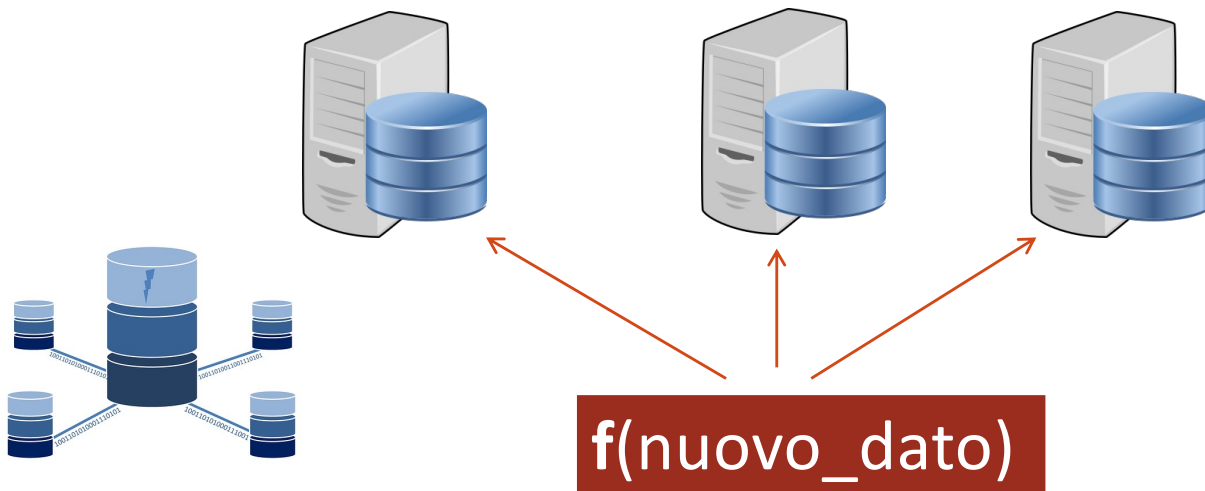
- Quali **operazioni sono consentite** all'utente X?
- Quali **dati appartengono** all'utente X?



Utente	OPERAZIONE	DATO	PERMESSO
Utente X	Lettura	Stipendio di X	Consentito
Utente X	Lettura	Stipendio di Y	Consentito
Utente Y	Scrittura	Stipendio di Y	Negato

Caratteristiche di un DBMS: **Scalabilità**

Possibilità di gestire grandi moli di dati aumentando il **numero di istanze del DBMS presenti** nel sistema (**database distribuito**).



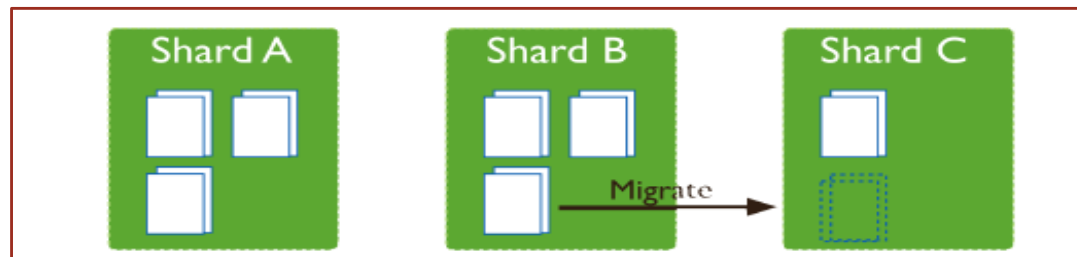
✧ La funzione di **sharding** determina la politica di distribuzione dei dati tra i nodi del sistema.

Argomento NON trattato nel corso ...

Caratteristiche di un DBMS: Scalabilità

Ulteriori funzionalità di un database distribuito.

1. Meccanismi di **Load-balancing**
2. Meccanismi di **Gestione delle repliche dati**

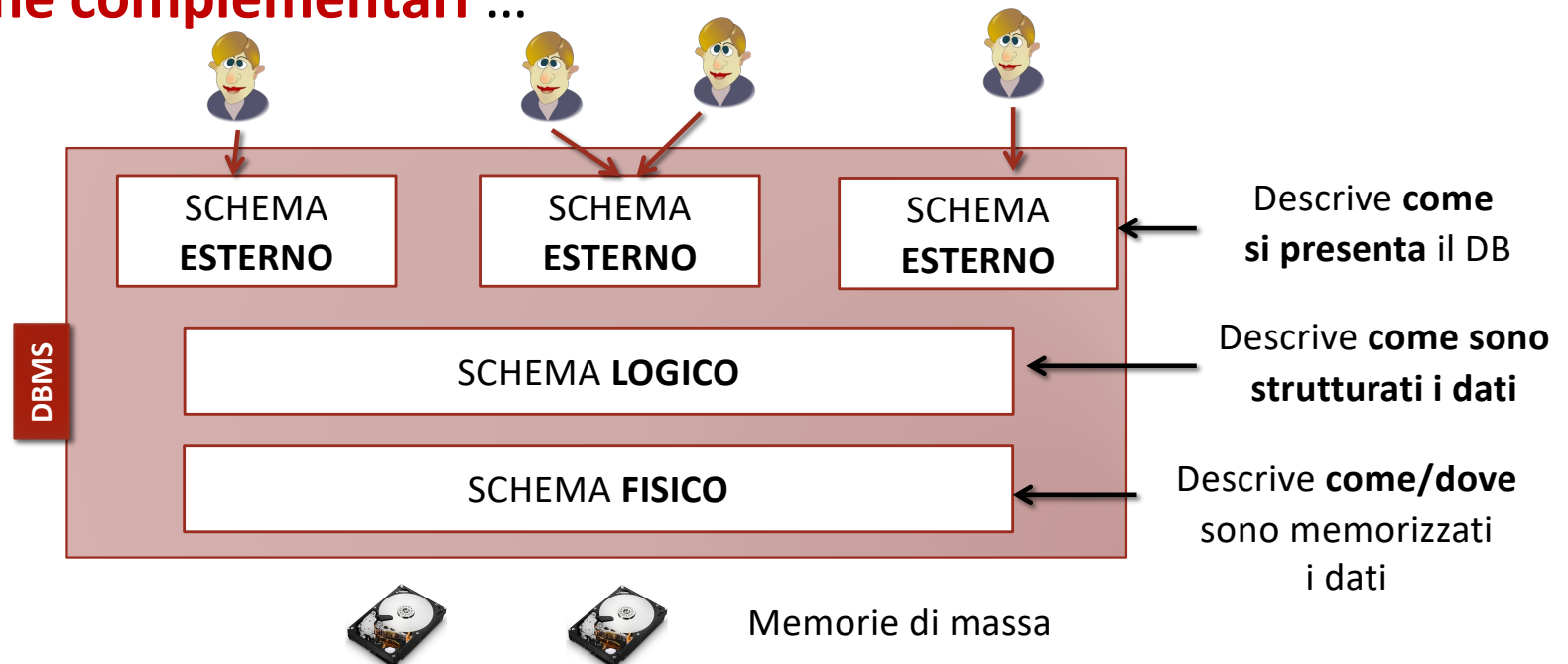


PROBLEMA. Come gestire la **consistenza** delle repliche dati in presenza di partizionamenti della rete e perdita di messaggi?

Argomento NON trattato nel corso ...

DBMS: Architettura a livelli

In pratica, un DBMS organizza i dati secondo **tre livelli di astrazione complementari** ...



Livello **Logico** di un DBMS

- Descrive come sono rappresentati/organizzati i dati (sempre dal punto di vista logico) e le relazioni tra gli stessi.
- DBMS possono differire sulla base del modello logico dei dati che supportano:
 - Modello **Relazionale** (di fatto, il più usato)
 - Modello Gerarchico
 - Modello Reticolare
 - Modello ad Oggetti
 - Approcci NoSQL (diversi)

Livello **Logico** di un DBMS

ESEMPIO: Modello Relazionale

Base di dati che gestisce le informazioni relative alla programmazione didattica di un Corso di Laurea: elenco corsi, con numero ore, semestre, crediti, nome e codice identificativo di ciascun corso.

Codice	Nome	NumOre	Semestre	Crediti
010	Basi di Dati	72	1	9
001	Algoritmi	90	1	12

→ **SCHEMA**

} **ISTANZE**

- Nel modello relazionale, i dati sono organizzati in tabelle ...

Livello **Esterno** di un DBMS

Il livello esterno consente di avere **viste** personalizzate della stessa base di dati **da parte di diversi utenti/applicazioni!**

Es. Base di dati condivisa tra diversi uffici di una stessa organizzazione.
Solo alcuni uffici possono accedere a tutto il contenuto del DB!

Codice	Nome	Cognome	Data Nascita	Livello	Stipendio
001	Marco	Rossi	10/10/1970	1	24000
002	Michele	Bianchi	10/10/1970	1	32000

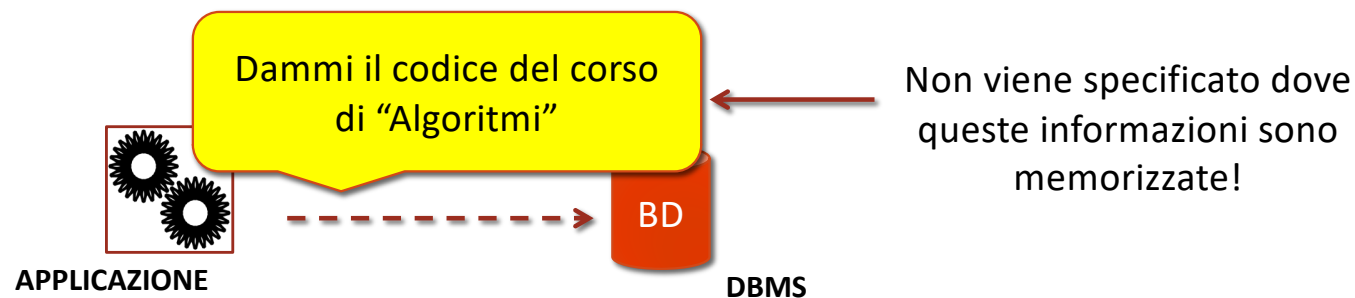
VISTA Ufficio **Anagrafe**

VISTA Ufficio **Stipendi**

Proprietà di **indipendenza** tra i livelli

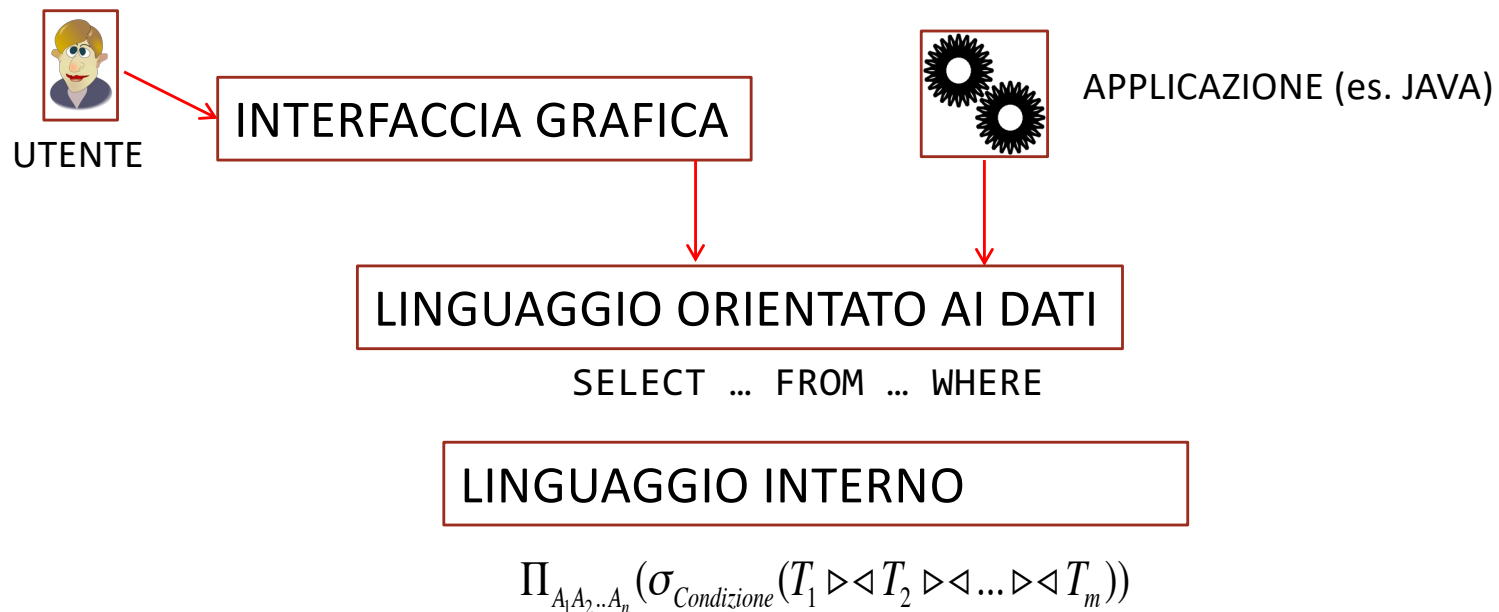
INDIPENDENZA MODELLO LOGICO – MODELLO FISICO

- L'organizzazione logica dei dati **non dipende** dalle strutture dati usate per l'effettiva memorizzazione degli stessi su disco!
- In pratica, **le applicazioni accedono al DBMS specificando i concetti logici** del modello dei dati, piuttosto che i dettagli relativi alla loro memorizzazione.



Interazione con un DBMS

Come possono utenti /applicazioni interagire con un DBMS?



Interazione con un DBMS

Come possono utenti /applicazioni interagire con un DBMS?

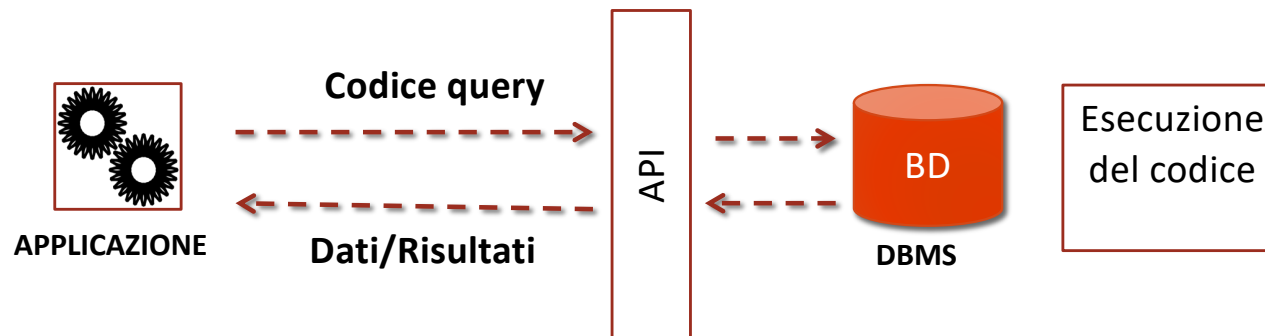
Quasi tutti i DBMS mettono a disposizione dei **linguaggi**:

- Definizione dello schema logico (**Linguaggio DDL**)
 - Manipolazione delle istanze (**Linguaggio DML**)
-
- Linguaggi **orientati ad i dati**, molto diversi da linguaggi di programmazione “tradizionali” (es. C/C++/Java/etc)!
 - Noi vederemo il linguaggio (DDL+DML) **SQL per il modello relazionale**.

Interazione con un DBMS

Le applicazioni che si interfacciano con un DBMS:

- integrano **codice delle operazioni sul DB (es. SQL)** all'interno del loro codice, o di quello di una libreria/framework
- utilizzano le **API** (fornite dal DBMS) per la connessione dati.



Vantaggi nell'uso di DBMS

Quando **usare** un DBMS in un progetto SW?

- Necessità di gestire **volumi consistenti di dati**
- Necessità di costruire **sistemi data-centric** con molte operazioni di accesso ai dati
- Necessità di **condividere dati**, fornendo l'accesso a diversi sistemi SW/applicazioni
- Necessità di garantire la **persistenza dei dati** anche a fronte di possibili guasti e malfunzionamenti HW/SW
- Necessità di implementare **meccanismi di sicurezza** per l'accesso ad i dati in un ambiente multi-utente

Svantaggi nell'uso di DBMS

Quando **NON usare** un DBMS in un progetto SW?

- Pochi dati o **persistenza** non necessaria.
- **Prestazioni:** In alcuni sistemi con richieste di efficienza sull'elaborazione (es. **real-time**), l'overhead computazionale introdotto dal DBMS può essere eccessivo ...
- **Costo:** Spese per l'acquisto di DBMS, formazione del personale, amministrazione del DB, etc
- **Complessità:** Applicazioni/sistemi di dimensioni ridotte, o dispositivi hardware con risorse limitate (es. microcontrollori).

Quale DBMS usare?

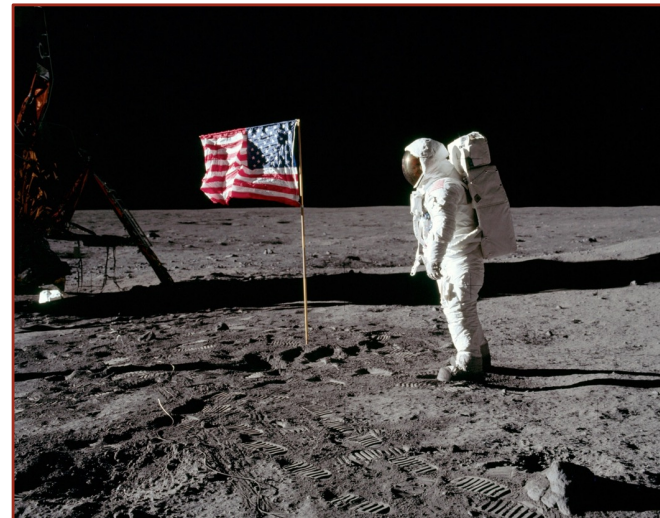
- Fin'ora abbiamo parlato in generale delle caratteristiche dei DBMS ...
- ... ma **i DBMS sono tutti uguali? NO!**

Differenze sostanziali, ad esempio in termini di:

- Modello logico supportato (relazionale? → RDBMS)
- Linguaggio DDL/DML (SQL-2? SQL-3? varianti?)
- Algoritmi di indicizzazione (es. R+ tree?)
- Supporto alla transazioni (es. proprietà ACID?)
- Gestione della concorrenza
- ...

DBMS: Un po' di storia

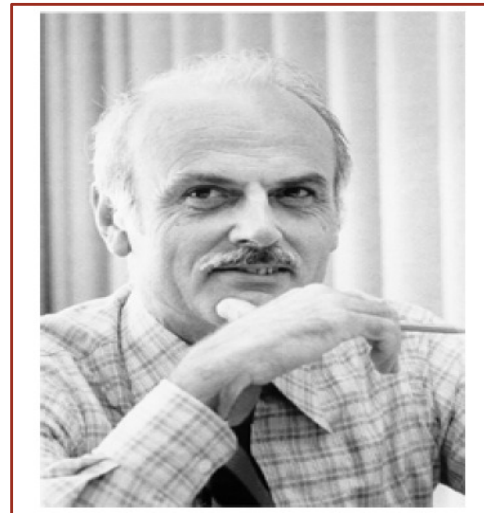
- *Information Management System (IMS)* è il nome di un software sviluppato da IBM nel 1968.
- Utilizzato come supporto alle missioni *Apollo* per la gestione dei dati tecnici/amministrativi e delle forniture di materiali.
- Modello **gerarchico** di gestione dei dati, **motore transazionale** per la concorrenza.



DBMS: Un po' di storia

Nel 1970, un ricercatore della IBM (**Edgar Codd**) pubblica la sua visione di **modello relazionale** dei dati, basato sul concetto matematico di relazione tra insiemi.

Edgar F. Codd *A relational model of data for large shared data banks*
Communications of the ACM 13 (6),
377-387. 1970.



DBMS: Un po' di storia

- Negli anni 1970, IBM lavora allo sviluppo di un linguaggio basato sul modello relazionale (SQL), ed all'implementazione di un RDBMS sperimentale (**System R → SQL/DS → DB2**), ma continua anche lo sviluppo del sistema IMS.
- Nel 1979, una piccola startup (**Relational Software Inc**) produce un primo esempio di RDBMS commerciale ... Nel 1982, Relational Software cambia il proprio nome in **Oracle Corporation**.

DBMS: Un po' di storia

- Negli anni '80, compaiono i primi **DBMS basati sul modello ad oggetto (ORDBMS)**, che cercano di emulare il successo del paradigma di programmazione ad oggetti, e facilitare l'integrazione tra DBMS e linguaggi di alto livello (es. C++/Java)
- Viene sviluppata **OQL** – omologo di SQL per il paradigma ad oggetti ... Nonostante ciò gli **ORDBMS restano poco diffusi!**

DBMS: Un po' di storia

Oggi il mercato dei DBMS è dominato da pochi vendor.

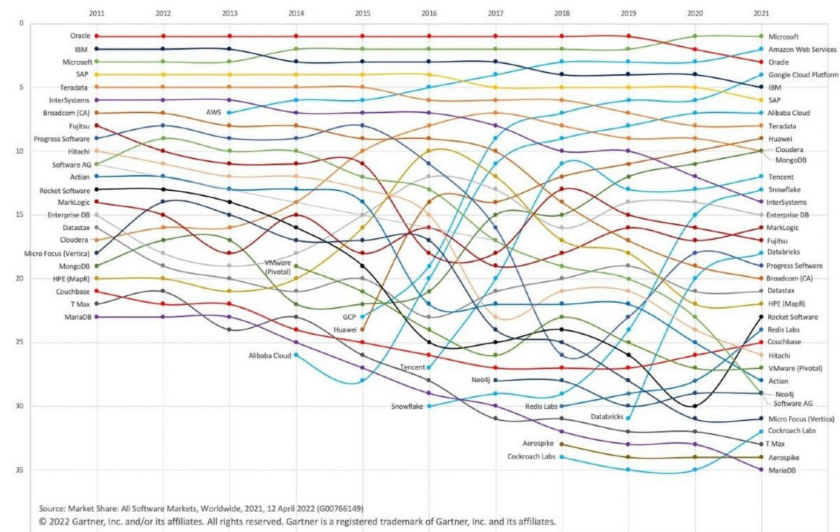


Image: Gartner via Adam Ronthal (@aronthal) on Twitter.

<https://blogs.gartner.com/merv-adrian/2022/04/16/dbms-market-transformation-2021-the-big-picture/>

DBMS: Un po' di **storia**

- Oggi, una delle nuove linee evolutive dei DBMS è rappresentata dall'approccio **NoSQL**.
- **Idea di base:** superare la rigidità del modello relazionale nella definizione dello schema, consentendo una più facile espansione del DB in termini di dati, e di computazione distribuita.
- Molti approcci sotto la definizione NoSQL:
 - **Database chiave-valore,**
 - **Database document-based**
 - **Database a grafi**