1 - Introduzione

Prima di cominciare

SI CONSIGLIA DI AVERE A PORTATA IL LIBRO

- Pur essendoci gli esercizi svolti dalla professoressa, sul libro ne sono presenti altri per potervi esercitare voi
- Qualche esempio potrebbe mancare volontariamente o involontariamente
- Ci sono approfondimenti che magari non ho trattato essendo questi riassunti ma sono carini da sapere (meno se stai usando questi file per studiare all'ultimo secondo)
- Il libro di testo su cui questi appunti son stati scritti è Basi di Dati, VI Edizione di Paolo Azteni
  SIETE STATI AVVERTITI :D

Sistema organizzativo

Un **sistema organizzativo** è un insieme di risorse e regole che consentono il funzionamento di una qualunque struttura sociale per il raggiungimento dei suoi obbiettivi (alcuni esempi di questi sono una biblioteca, studio medico, università etc...)

Sistema informativo

Ogni sistema organizzativo è dotato di **sistema informativo**, ossia l'insieme delle risorse e delle procedure che un'organizzazione utilizza per gestire le informazioni necessarie al perseguimento dei propri scopi.
Il sistema informativo assicura:

1. **Raccolta**
2. **Acquisizione**
3. **Archiviazione**

Un sistema informativo è molte volte indipendente dalle automatizzazioni che conosciamo, si basti ricordare l'era pre-calcolatori, come gli archivi delle banche o servizi anagrafici che esistono da vari secoli.

Un esempio di sistema informativo può essere una biblioteca:

2. Una *biblioteca* gestisce informazioni sui materiali raccolti, sui prestiti, sulle persone che prendono in prestito materiali, per svolgere varie attività, esempio:
   (a) attività rivolte alla raccolta dei documenti (e.g., abbonamento ai periodici);
   (b) attività rivolte alla conservazione e consultazione dei documenti (e.g., produzione cataloghi)
   (c) attività rivolte alla gestione della biblioteca (e.g., controllo della restituzione dei prestiti e dell'arrivo dei periodici)

La parte che quindi conosciamo di un sistema informativo per adesso è quella **non automatizzata**

Sistema Informatico

Andiamo a definire un **sistema informatico** come la parte automatizzata di un sistema informativo.
Un esempio di sistema informatico potrebbe essere quello di una banca:

Il sistema informatico di una banca è costituito da:

- Archivi elettronici contenenti i dati dei clienti, dei conti correnti, dei prestiti, ecc.
- I sistemi di calcolo (hardware e software) che consentono la gestione di tali archivi.
- La rete di terminali utilizzati dagli operatori agli sportelli delle agenzie.

Andando a riformulare quindi un **sistema informatico** è un sistema software orientato alla gestione dei dati, dove l'aspetto prevalente è rappresentato dai dati stessi (memorizzati, ricercati, modificati) che costituiscono il patrimonio informativo di un'organizzazione

Si distingue da:

- **Software orientato alla realizzazione di funzioni** per la complessità prevalente riguarda le funzioni da realizzare
- **Software orientato al controllo** per la complessità prevalente riguarda il controllo di attività che si sincronizzano e cooperano durante l'evoluzione del sistema

Componenti

Il sistema informatico prevede quattro componenti:

- I programmi applicativi che forniscono i servizi agli utenti eseguendo insiemi di operazioni sulla base informativa.
- Uno schema che descrive la struttura della base informativa, le operazioni per agire su di essa, le restrizioni sui valori memorizzabili e sui modi in cui essi possono evolvere nel tempo (vincoli d'integrità);
- La base informativa (o base di dati) che contiene una rappresentazione delle informazioni memorizzate;
- HW e SW di base.

Nei sistemi orientati ai dati le informazioni sono strutturate con un formato predeterminato rispetto ai linguaggi di programmazione classici.

Dato e informazioni

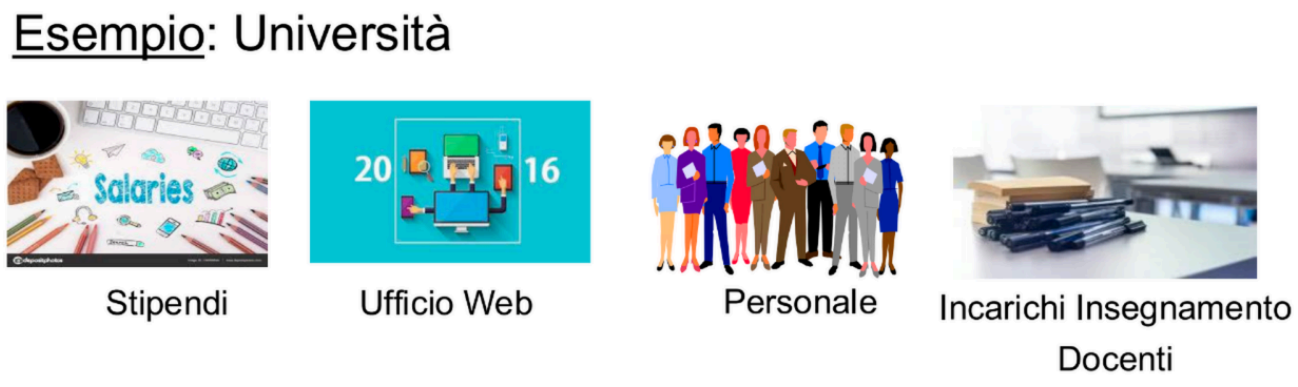
Nei sistemi informatici le informazioni sono rappresentate in modo essenziale attraverso i dati.
Ma qual'è la differenza tra un dato e un informazione?

- Il **dato** è un elemento di informazione costituito da simboli non ancora elaborati (in parole povere, sono fatti noti che possono essere registrati su un qualche supporto in una forma simbolica.)
- L'**informazione** è un processo di elaborazione e interpretazione che consente di attribuire un significato ad un dato.

I dati da soli quindi non hanno un significato, lo acquisiscono soltanto quando vengono interpretati e correlati, in quel caso si crea l'informazione che si riferisce ad un contesto

Evoluzione dei sistemi informatici

Un sistema organizzativo è composto da **settori** tra loro coordinati tramite la struttura organizzativa:



Ogni **settore** ha le proprie informazioni dove alcune sono comuni ad altri mentre altre sono di esclusiva competenza di alcuni sotto-settori.
Sempre nell'esempio dell'università troviamo:

Settore	Evento	Dati generati
Incarichi Insegnamento Docenti	Assegnazione di un nuovo insegnamento ad un docente	Codice Insegnamento, Nome Insegnamento, a.a. di attivazione, Docente affidatario
Personale	Assunzione nuovo Docente	Matr. Docente, CF, Nome Cognome, Indirizzo, posizione accademica
Amministrazione Sito Web	Pubblicazione insegnamenti per il nuovo a.a.	Nome insegnamento, Programma, Nome Docente, Numero Stanza

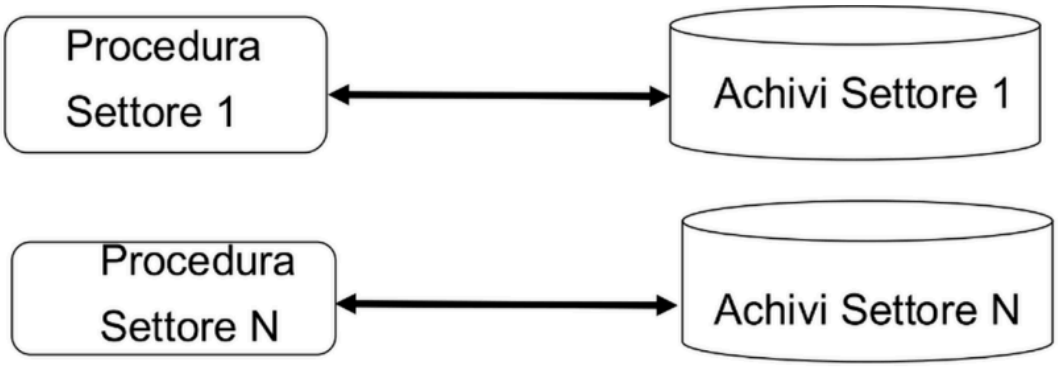
Tipicamente gli eventi posso determinare una nuova informazione o la morte di un dato.

Sistemi Informatici Settoriali

Si stabiliscono tra i settori flussi di informazioni che permettono ad ogni settore di procurarsi i dati di interesse dal settore originante, di conseguenza accade:

- **Una proliferazioni dell'informazioni**, ossia informazioni soggette a frequenti aggiornamenti con richiesta di definire regole organizzative per garantire una consistenza delle informazioni
- **Una relazione gerarchica** che si instaura fra i settori coinvolti nello scambio informativo (l settore che cede informazione controlla quantità e qualità dell’informazione fornita)

Nella fase iniziale del processo di automazione dei vari settori aziendali, le procedure tendono ad essere prodotte separatamente per ogni settore:



Questa procedura è più economica e permette maggiore produttività, ma:

- Si ha un alta ridondanza dei dati
- Per uno stesso dato ci possono essere diversi cicli di vita (in base al settore)
- Ogni settore sceglie il supporto di memorizzazione, la struttura di memorizzazione, la chiave di accesso, ecc. secondo criteri di ottimizzazione locali

- Le proprietà che il dato deve rispettare possono variare da settore a settore.

Questo tipo di approccio non va bene per un **sistema informatico complesso**

Requisiti di un sistema informatico complesso

- **Integrazione dei dati**: disporre di un'unica raccolta dati comuni e tanti programmi che realizzano le applicazioni operano solo sui dati di loro interesse, eliminando praticamente le ridondanze
- **Flessibilità di realizzazione**: possibilità di partire da un nucleo di funzioni essenziali e poter espandere il sistema (scoprendo possibili modalità d'impiego)

Sistemi di gestione di basi di dati (PT.1)

Il **DBMS** (sistemi di gestioni di basi di dati) è un sistema software in grado di gestire collezioni di dati che siano grandi, condivise e persistenti, garantendo affidabilità, privatezza, efficienza ed efficacia.

I DBMS mettono a disposizione strumenti avanzati di archiviazione e reperimento di informazioni, soddisfacendo i requisiti di un sistema informatico complesso

Proprietà delle basi di dati

Le **basi di dati** sono:

- **Grandi**, possono avere dimensioni enormi e in generale molto maggiori della della memoria centrale disponibile, di conseguenza i DBMS devono prevedere una gestione articolata dei dati in memoria secondaria
- **Condivise**, diverse applicazioni, settori e sottosistemi devono potervi accedere secondo le opportune modalità. In questa maniera è possibile ridurre la ridondanza di dati e, di conseguenza, inconsistenze.
  Nel caso un DBMS gestisca più operazioni in contemporanea vi è necessario un controllo di concorrenza.
- **Persistenti**, hanno un tempo di vita indipendente dalle singole esecuzioni dei programmi che lo utilizzano

Garanzie di un DBMS

Un DBMS deve garantire:

- **Affidabilità**, quindi una resistenza a malfunzionamenti HW e SW in modo da mantenere intatto il contenuto o permetterne la ricostruzione (backup e recovery).
  Una tecnica fondamentale è la gestione delle transazioni, cioè delle unità di lavoro

atomiche che non possono avere effetti parziali.

Esempio:
- due utenti/applicazioni hanno accesso all'anagrafica di una persona
- *Mentre una sta accedendo in modifica all'indirizzo di residenza, l'altra accede in lettura*

La gestione delle transazioni garantisce che l'indirizzo letto **non** sia inconsistente (es. in parte composto dal vecchio indirizzo)

- **Privatezza dei dati**, un sistema deve poter definire dei meccanismi di autorizzazione per utente (opportunamente riconosciuto)
  - l'utente A è autorizzato a leggere tutti i dati e a modificare X
  - l'utente B è autorizzato a leggere dati X e a modificare Y

Affidabilità

Un DBMS prevede che le interazioni con la base di dati avvengano per mezzo di **transazioni**.
Una **transazione** è una sequenza di azioni di lettura e scrittura del DB e di elaborazioni di dati in memoria temporanea, che il DBMS esegue garantendo le seguenti proprietà (ACID properties):

- **Atomicity**: La transazione è eseguita nella sua interezza oppure non è eseguita affatto (le transazioni che terminano prematuramente sono abortite)
- **Consistency preservation**: una esecuzione corretta della transazione porta il DB da uno stato consistente all'altro (i vincoli di integrità devono essere rispettati).
- **Isolation**: L'esecuzione di una transazione deve essere indipendente da quella di altre transazioni concorrenti. Il risultato deve essere analogo a quello che si otterrebbe eseguendo le transazioni una alla volta (serialmente).
- **Durability** : le modifiche su DB di una transazione terminata normalmente sono permanenti, cioè non sono alterabili da malfunzionamenti successivi alla terminazione

L'interruzione di transazione causa l'attivazione di procedure di ripristino o recovery (riportano il DB allo stato corretto precedente al malfunzionamento)

Integrità

I DBMS prevedono anche meccanismi per controllare che i dati inseriti, o modificati, siano conformi alle definizioni nello schema per garantire sempre la consistenza del DB
I linguaggi per la definizione dello schema logico consentono di definire le condizioni cui i dati devono sottostare per essere significativi (vincoli d'integrità), e cosa fare in caso di violazioni.

Esempio:

Esami

Studente	Voto	Corso
276545	28	01
290089	27	01
200768	27	04

Un vincolo di integrità riguarda l'univocità della coppia (codice studente, corso) (vedi UNIQUE in SQL) .

Un altro vincolo di integrità riguarda il voto:
(Voto≥18) And (Voto≤30)

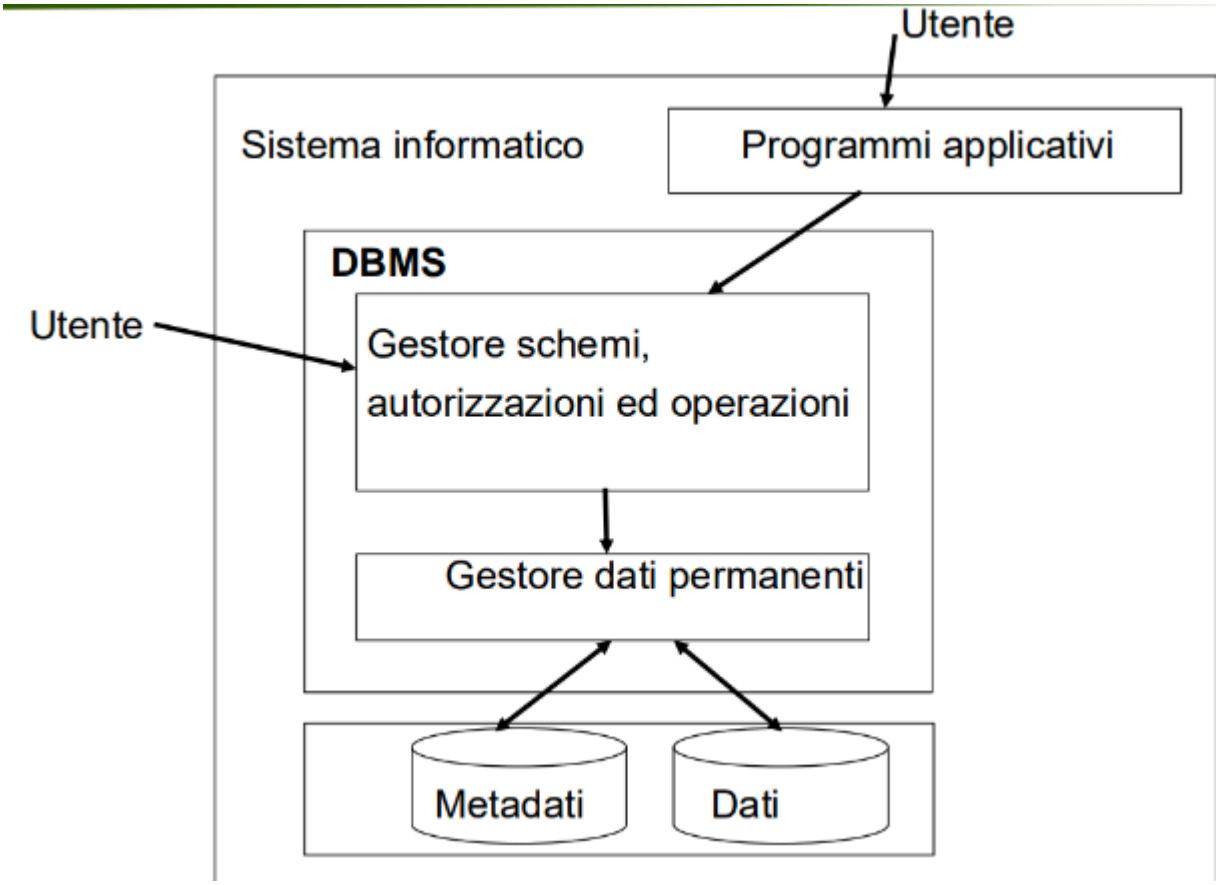
Sistemi di gestione di basi di dati (PT.2)

I dati di DB, gestiti da un elaboratore, si distinguono in:

- **Metadati**, ovvero lo schema di DB
  Def. raccolta di definizioni che descrivono la struttura dei dati, le restrizioni sui valori ammissibili (vincoli d'integrità), relazioni tra gli insiemi
- **Dati**: rappresentazione di fatti conformi alle definizioni dello schema

Un DBMS consente di:

- Definire schemi di basi di dati e vincoli di integrità;
- Scegliere le strutture dati per la memorizzazione e l'accesso ai dati;
- Memorizzare, interrogare e modificare i dati
  Interattivamente da utenti o programmi autorizzati



Alcuni DBMS sul mercato sono:

- MySQL
- Access
- DB2
- Oracle
- SQLServer
- PostgresSQL

Modelli di dati

Un **modello di dati** è un insieme di concetti (o costrutti) per organizzare i dati di interesse e descriverne la struttura in modo comprensibile ad un elaboratore.
Ogni modello di dati fornisce meccanismi di astrazione per definire nuovi tipi sulla base di tipi (elementari) predefiniti e costruttori di tipo.
Il modello relazione dei dati (più diffuso tra tutti) permette di definire tipi per mezzo del costrutto della **relazione**, che consente di organizzare i dati in insiemi di record a struttura fissa.

Un buon modello di dati è caratterizzato da:

- Espressività: rappresentazione in modo naturale e diretto del significato di ciò che si sta modellando.
- Semplicità d’uso: pochi meccanismi, semplici da usare e da comprendere.
- Efficienza di realizzabilità

Oltre al modello relazionale possiamo trovare altri modelli:

Modello	Struttura usata	Anni	DBMS
Modello gerarchico	basato sull'uso di strutture ad albero	'60	IMS System 2000
Modello reticolare	basato sull'uso di strutture a grafo	Inizio '70	IDMS, IDS II, DM IV
Modello relazionale	basato sull'uso di relazioni: insiemi di record a struttura fissa con campi di tipo primitivo	'80	System R, Ingres, Oracle, DB2
Modello ad oggetti	basato sull'uso di classi di oggetti e istanze	Fine '80 - '90	O2 ObjectStore
Modello XML	rivisitazione del modello gerarchico: dati presentati insieme alla loro descrizione e non devono sottostare rigidamente ad un'unica struttura logica	'90	BaseX
Modelli semi-strutturati e flessibili	non hanno rigidità nell'organizzazione dei dati ed hanno alte prestazioni	'00	Sistemi NoSQL

Modello relazionale

Il modello relazionale dei dati permette di definire tipi grazie al costruttore della **relazione**, che rappresenta l'organizzazione dei dati in insiemi omogenei.
Una relazione viene rappresentata generalmente tramite **tabella**, le cui righe rappresentano record **specifici** e le colonne ai vari campi dei record

Docenza

Corso	Docente
Basi di dati	Rossi
Reti	Neri
Liguaggi	Verdi

Manifesto

CdL	Insegnamento	Anno
Informatica	Basi di dati	2
Informatica	Reti	3
Informatica	Liguaggi	2
ITPS	Basi di dati	3
ITPS	Reti	3

Esempio:

- *Schema di Relazione*: Docenza(Corso, Docente)
- *Istanza*: tutte le righe di Docenza

Schemi e istanze

Nella base di dati esiste una parte invariata nel tempo, detta **schema della base di dati**, costituita dalle caratteristiche dei dati, e una parte variabile, chiamata **istanza** della base di dati, costituita dai valori effettivi.

Modelli dei dati vs concettuali

I modelli dei dati precedentemente elencati sono detti

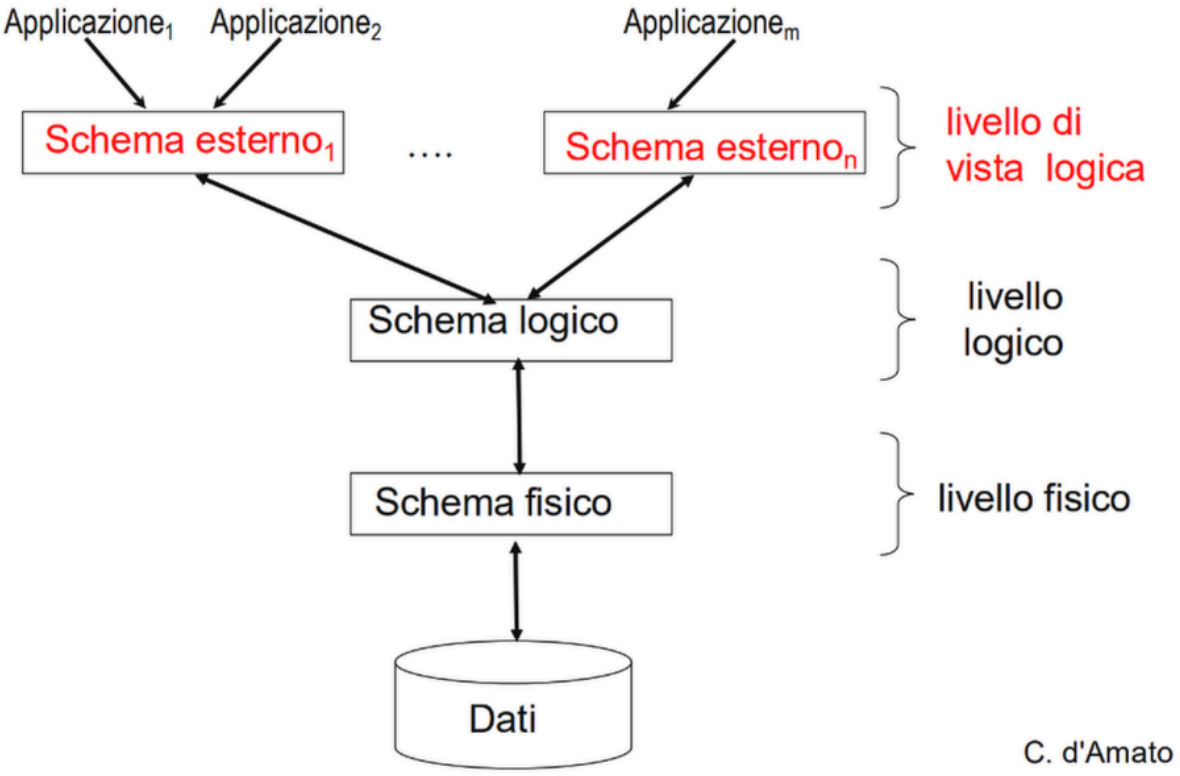
- **Logici**: le strutture usate da questi modelli, pur essendo astratte, riflettono una particolare organizzazione (alberi, grafi, a tabelle, oggetti etc).
  Sono adottati nei DBMS esistenti per l'organizzazione dei dati e indipendenti dalle strutture fisiche
- **Concettuali**: utilizzati per descrivere i dati in modo completamente indipendente dalla scelta del modello logico. Descrivono concetti del mondo reale, piuttosto che i dati utili a rappresentarli.
  Sono utilizzati nella fase preliminare di progettazione di un DB, per modellare la realtà indipendentemente da aspetti realizzativi, un esempio è il modello Entità-Relazioni.

I modelli concettuali, a differenza di quelli logici, non sono generalmente disponibili nei DBMS.

Livelli di astrazione nei DBMS

L'architettura di un DBMS è distinta in 3 livelli di descrizione di dati in schema logico:

- **Livello fisico**: costituisce una descrizione dell'organizzazione fisica dei dati nelle memorie permanenti e strutture dati ausiliare per facilitarne l'uso (es. file hash, sequenziale, sequenziale con indici)
- **Livello logico**: descrive la struttura degli insiemi di dati e delle relazioni fra loro, secondo un certo modello logico dei dati, senza nessun riferimento alla loro organizzazione fisica nella memoria permanente (es. modello relazionale o altro)
- **Livello di vista logica**: definisce come deve apparire la struttura del DB ad una certa applicazione e/o utente



Indipendenza dei dati

L'architettura dei livelli quindi garantisce **l'indipendenza** dei dati, permette a utenti e programmi di interagire con la base di dati a un livello astratto, prescindendo dai dettagli realizzativi.
L'indipendenza dei dati può essere caratterizzata in due stati:

- **Indipendenza fisica**: consente di interagire con il DBMS in modo indipendente dalla struttura fisica dei dati, senza influire sulle descrizioni e quindi sui programmi che usano i dati
- **Indipendenza logica**: consente di interagire con il livello esterno della base di dati in modo indipendente dal livello logico, per esempio come aggiungere un nuovo schema esterno senza modificare lo schema logico e perciò la sottostante organizzazione fisica dei dati

Il **modello relazionale** risponde perfettamente al requisito dell'indipendenza dei dati perché introduce una netta distinzione tra il livello fisico e quello logico. A differenza dei modelli precedenti (reticolare e gerarchico), che includevano riferimenti espliciti a puntatori fisici, il modello relazionale è basato su valori e non richiede la conoscenza delle strutture fisiche per accedere ai **dati**

Linguaggi e utenti delle basi di dati

I DBMS sono caratterizzati da un lato dalla presenza di molteplici linguaggi per la gestione dei dati, dall'altro dalla presenza di molteplici tipologie di utenti.
Per i **linguaggi della base di dati** si distinguono in:

- **Linguaggi di definizione dei dati (Data Definition Language)**: utilizzati per definire gli schemi logici, esterni e fisici e le autorizzazioni di accesso
- **Linguaggi di manipolazione dei dati (Data Manipulation Language)**: utilizzati per l'interrogazione e aggiornamento delle istanze di basi di dati

Il termine query language viene spesso usato come sinonimo di DML.
Le istruzioni di un DML definite **iterrogazioni** sono quelle che non modificano il database.
Originariamente la distinzione fra DDL, DML e query language era netta ma successivamente son stati proposti linguaggi che integrano le funzionalità suddette, come SQL.

Classificazione dei DBMS

Il criterio principale utilizzato per classificare è il modello dei dati sul quale si fonda il DBMS, distinguendo le basi di dati gerarchiche da quelle reticolari (network), relazionali, orientate a oggetti, ecc.

Altri criteri sono:

- **Il numero di utenti**: Sistemi single-user supportano solo un utente per volta e sono per lo più usati con personal computer, maggior parte dei DBMS sono multi-user.
- **Il numero di centri (site) in cui è distribuito il DB**:
  - Nei DBMS **centralizzati** i dati sono memorizzati in un unico centro e può supportare più utenti, eventualmente remoti.
  - Nei DBMS **distribuito** si possono avere dati e software distribuiti in più centri connessi da una rete locale o geografica, tipicamente i database distribuiti hanno un’architettura client-server.
    Questi ultimi si suddividono in
    - **Omogeneei**: usano lo stesso software
    - **Eterogenei**: utilizzano software per accedere a diversi DB autonomi pre-esistenti

Vantaggi e svantaggi dei DBMS

Oltre ai vantaggi di **integrazione dei dati** e di **flessibilità** del DB, la tecnologia delle basi di dati consente anche:

- stabilire degli **standard** circa la strutturazione e nomenclatura dei dati in una organizzazione
- **ridurre i tempi di sviluppo** delle applicazioni rispetto a quelli richiesti usando la tecnologia degli archivi

Tuttavia si hanno anche degli **svantaggi**:
- I DBMS richiedono impegno hardware e software per messa a punto e funzionamento, di conseguenza anche costi più elevati e personale con competenze per lo specifico DBMS

- Impatto sulla struttura organizzativa
- Progetto di DB ed applicazioni richiedono personale qualificato e strumenti opportuni, l’impiego di DB richiede ristrutturazione dei dati in archivi esistenti e riscrittura degli applicativi
- Non consigliati per applicazioni con uno o pochi utenti, senza necessità di accessi concorrenti e relativamente stabili nel tempo