



UNIVERSITÀ
DEGLI STUDI DI TRIESTE



Corso di Laurea in Tecniche di Radiologia Medica per immagini e Radioterapia Informatica Medica

2CFU – 20 ore

DATABASE E DATABASE MANAGEMENT SYSTEMS

Prof. Sara Renata Francesca Marceglio

Definizione

DATABASE (base di dati) = collezione di dati correlati logicamente tra loro con un significato implicito:

- Ha una sua **coerenza logica** (non è un aggregato casuale di dati)
- È progettato, costruito e popolato con un **obiettivo specifico**
- Rappresenta alcuni aspetti del mondo reale (miniworld)

Terminologia

- **DATO** → **singola informazione**
- **CAMPO** (/FIELD) → descrive **l'aspetto fisico del dato**. Pezzo di memoria in cui mettere il dato: il dato occupa una certa parte di memoria.
- **RECORD** → raccolta **finalizzata** di **DATI**
- **FILE** → raccolta **finalizzata** di **RECORD**

Record: esempio

- Un esempio di **RECORD** è il RECORD DI IDENTIFICAZIONE di un Paziente presente nella cartella clinica

- Nome

- Cognome

- Indirizzo

- Mappa della zona

- Codice Fiscale

-

- Nel RECORD DI IDENTIFICAZIONE voglio solo i DATI necessari alla identificazione del paziente

Record in un linguaggio di programmazione

- Un esempio di **RECORD** usando la notazione linguaggio C

```
• struct Anagrafica {  
    varchar Nome[30];  
    varchar Cognome[30];  
    indirizzi Indirizzo;  
    maps Mappa della zona;  
    char Codice Fiscale[21];  
    ....  
};
```

Record complessi: esempio

- Un altro esempio di RECORD è il RECORD ANAMNESI FAMILIARE di un Paziente.
- In esso inserisco i dati relativi ai parenti più stretti del paziente: considero fino ad un certo grado di parentela discendenti ed ascendenti. In esso trovo dei DATI che hanno pertinenza se il paziente soffre di malattie di tipo ereditario,...
- Nel RECORD ANAMNESI FAMILIARE voglio solo i DATI relativi a

Record complessi: esempio

ANAMNESI FAMILIARE

03

	A	B	C	D	E	F	G	H
	Padre	Madre	Fratello	Sorella	Coniuge	Figli	Figlie	Altri
01	Vivente							
02	Deceduto							
03	Sano							
04	Longevo oltre 70 anni							
05	Coronaropatie							
06	Altre cardiopatie							
07	Ipertensione arteriosa							
08	Vasculopatie cerebrali							
09	Diabete							
10	Obesità							
11	Tumori							
12	Malattie del fegato							
13	Malattie dello stomaco							
14	Malattie dei reni							
15	Malattie del sangue							
16	Malattie reumatiche							
17	Malattie allergiche							
18	Malattie polmonari							
19	Tubercolosi							
20	Nevrosi							
21	Psicosi							
22	Malattie ereditarie							
23	Non si sa							

Già ricoverato in questo Ospedale?

si 24
no

In questa Divisione

si 25
no

Numero di precedenti
posizioni di Archivio

26

Osservazioni: _____

FILE – CARTELLA CLINICA

RECORD – ANAGRAFICA / Identificazione del paziente

RECORD – RAGIONI DEL RICOVERO

RECORD – ANAMNESI FAMILIARE

RECORD – ANAMNESI FISIOLOGICA

RECORD – ANAMNESI PATOLOGICA PROSSIMA

RECORD – ANAMNESI PATOLOGICA REMOTA

RECORD – ESAME OBIETTIVO PARTICOLARE

RECORD – ESAMI DI LABORATORIO

RECORD – DECISIONI

RECORD – FOGLIO DI DIAGNOSI

.....

FILE – CARTELLA CLINICA

Questo file è una collezione
di dati collegati tra di loro che
rappresentano il paziente:

È un database?

NO...

RECORD – ANAGRAFICA / Identificazione del paziente

RECORD – RAGIONI DEL RICOVERO

RECORD – ANAMNESI FAMILIARE

RECORD – ANAMNESI FISIOLOGICA

RECORD – ANAMNESI PATOLOGICA PROSSIMA

RECORD – ANAMNESI PATOLOGICA REMOTA

RECORD – ESAME OBIETTIVO PARTICOLARE

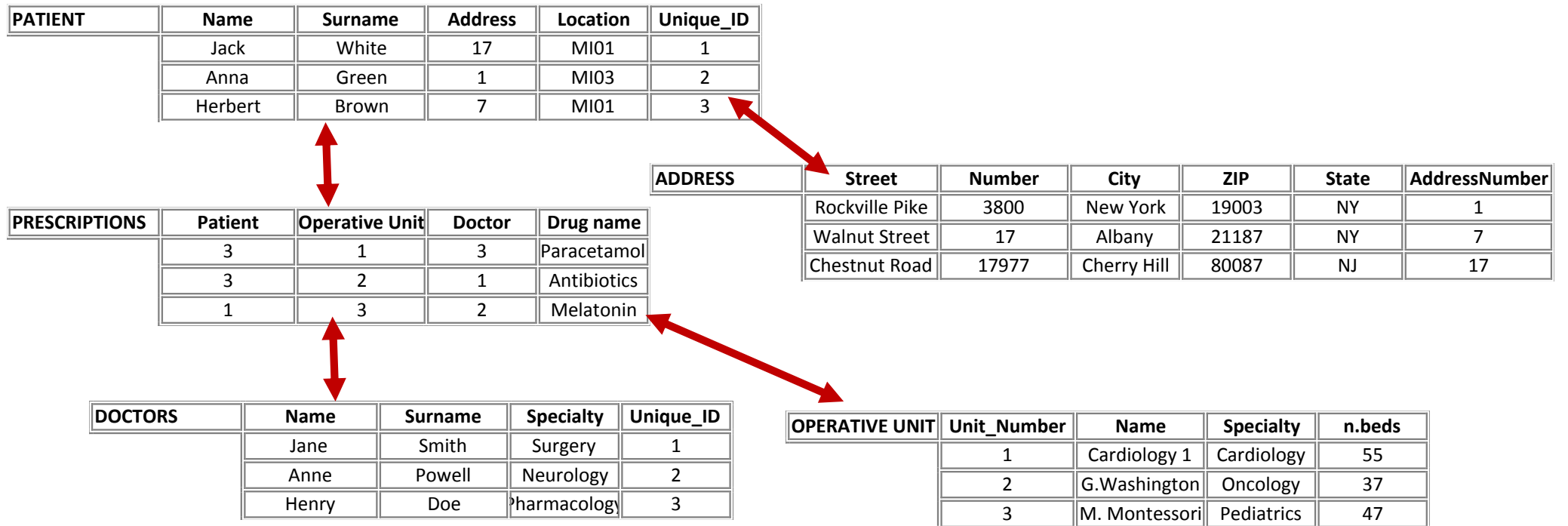
RECORD – ESAMI DI LABORATORIO

RECORD – DECISIONI

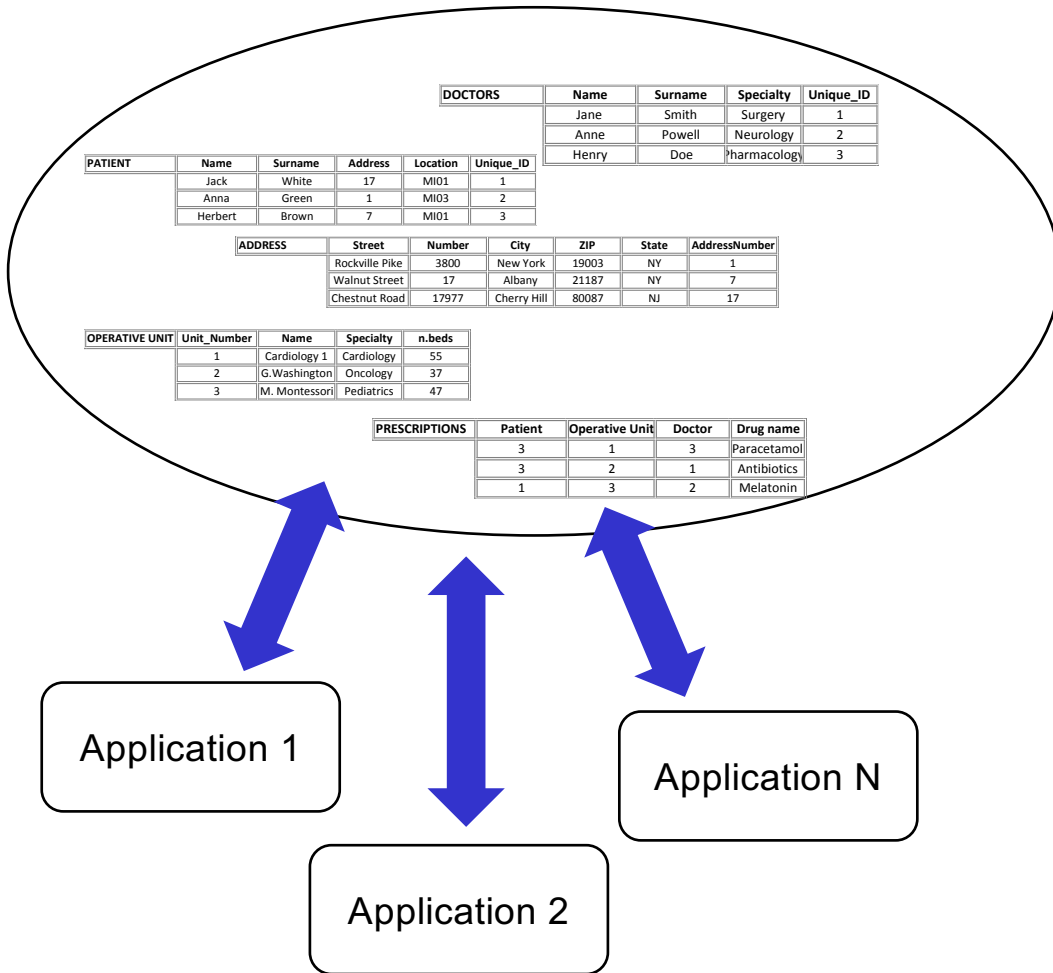
RECORD – FOGLIO DI DIAGNOSI

.....

Database



Programmazione con database e programmazione con file



Application 1

```
struct Patient {
    char Name[30];
    char Surname[30];
    address Address;
    char location[5];
    char uniqueID [21];
    ....
};
```

➔ Patient FindPatient (char Name[30]);

Application N

```
struct Patient {
    char Name[30];
    char Surname[30];
    address Address;
    char location[5];
    char uniqueID [21];
    ....
};
```

➔ CreateNewPrescription (Patient pat);

Database e file

- **UN DATABASE FILES** è una raccolta **ordinata** e **finalizzata** di
- Bisogna definire cosa si intende per base di dati e quali dati raccolgo in essa.
- Ci sono due aspetti contrastanti con cui convivere nella costruzione di un database:
 1. Raccogliere **il più** possibile
 2. Raccogliere **solo** quello che è utile

Data Base Management System

DBMS → collezione di programmi che consentono agli utenti di creare e gestire più basi di dati

Software che fornisce dei «semilavorati» che supportano nelle operazioni di gestione del database:

- **DEFINIRE** (definizione dei tipi di dati,...),
- **COSTRUIRE** (memorizzazione dei dati su qualche supporto,...)
- **MANIPOLARE** (interrogazione, aggiornamento, generazione di report...)

... e la **CANCELLAZIONE?**

Data Base Management System: PROPRIETA'

1. Gestione di grandi moli di dati
2. Gestione della persistenza
3. Condivisione dei dati (e controllo della concorrenza)
4. Controllo della ridondanza
5. Operazioni di salvataggio e ripristino (backup & recovery)
6. Privatezza dei dati

Gestioni di grandi moli di dati

- Un DBMS è in grado di **gestire grandi moli di dati** (ordine dei Gbyte o TByte).
- Esempi: base di dati relativa ai pazienti di una divisione ospedaliera (sono mantenuti anche i dati dei pazienti non più correntemente ricoverati); base di dati e immagini di una divisione radiologica.
- Il DBMS deve essere in grado di **gestire i dati in memoria secondaria**.

Gestione della Persistenza

- I dati appartenenti ad una base di **dati** sono **persistenti**: il loro tempo di vita va oltre il tempo previsto dell'esecuzione delle applicazioni che li utilizzano.
- NB: c'è differenza con i dati delle applicazioni (es. linguaggio C) residenti in memoria centrale.

Condivisione dei Dati

- **Molti e differenti utenti** caratterizzano una base di dati, anche clinica.
- Esempi:
 - Medici,
 - Personale paramedico
 - Personale tecnico
 - Responsabile della divisione
 - Responsabile amministrativo
 - Responsabile del Sistema Informativo Sanitario
 - ...

Condivisione dei Dati

- Ogni utente accederà alla base di dati con finalità e modalità diverse a seconda della categoria a cui appartiene:
- il DBMS deve essere in grado di gestire il fatto che più utenti accedono contemporaneamente alla base di dati;
- il DBMS deve permettere agli utenti di compiere solo azioni consistenti sui dati (controllo di concorrenza)

Controllo della ridondanza

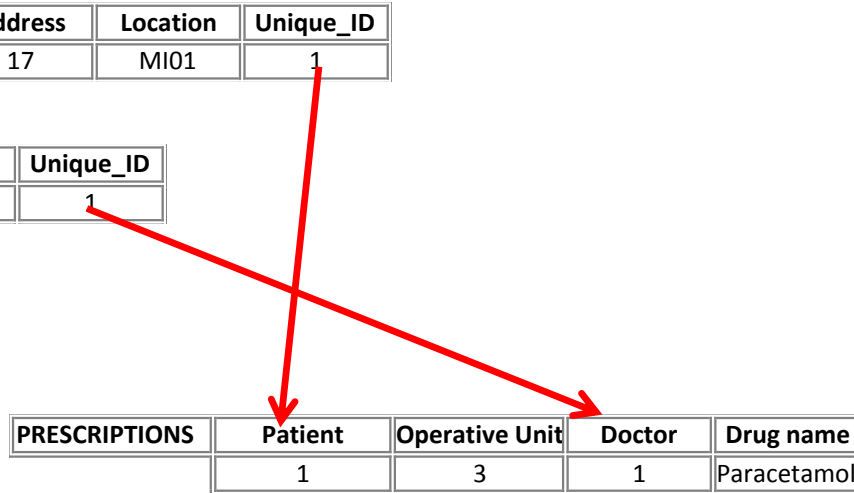
- Essendo consentito a differenti utenti di accedere ai dati, il DBMS permette di **evitare la ridondanza** dei dati; vengono evitate anche le inconsistenze dei dati: la **correzione** di errori andrebbe effettuata **una sola volta**.

PATIENT	Name	Surname	Address	Location	Unique_ID
	Jack	White	17	MI01	1

DOCTORS	Name	Surname	Specialty	Unique_ID
	Jane	Smith	Surgery	1

Le informazioni sul paziente e sul medico non sono ripetute nella prescrizione

PRESCRIPTIONS	Patient	Operative Unit	Doctor	Drug name
	1	3	1	Paracetamol



Operazioni di salvataggio e ripristino (back up and recovery)

- Il DBMS deve essere in grado di conservare il contenuto della base di dati di fronte a guasti e/o **malfunzionamenti**;
- Le operazioni di **salvataggio e recupero** permettono di riportare la base di dati all'ultimo **stato consistente**, nel caso un malfunzionamento capiti durante un'operazione complessa
- Viene garantita cioè **la affidabilità del sistema**

Privatezza dei dati

- Tramite la gestione di accessi autorizzati, il DBMS garantisce che differenti utenti o categorie di utenti –con compiti e responsabilità diverse – possano compiere sui **dati** solo le **operazioni** ad essi consentite

Architettura di un DBMS: Livelli

- **LIVELLO INTERNO**

Descrive la struttura fisica della base di dati. Include tutti i dettagli per la memorizzazione dei dati ed i percorsi di accesso alla base di dati.

- **LIVELLO CONCETTUALE**

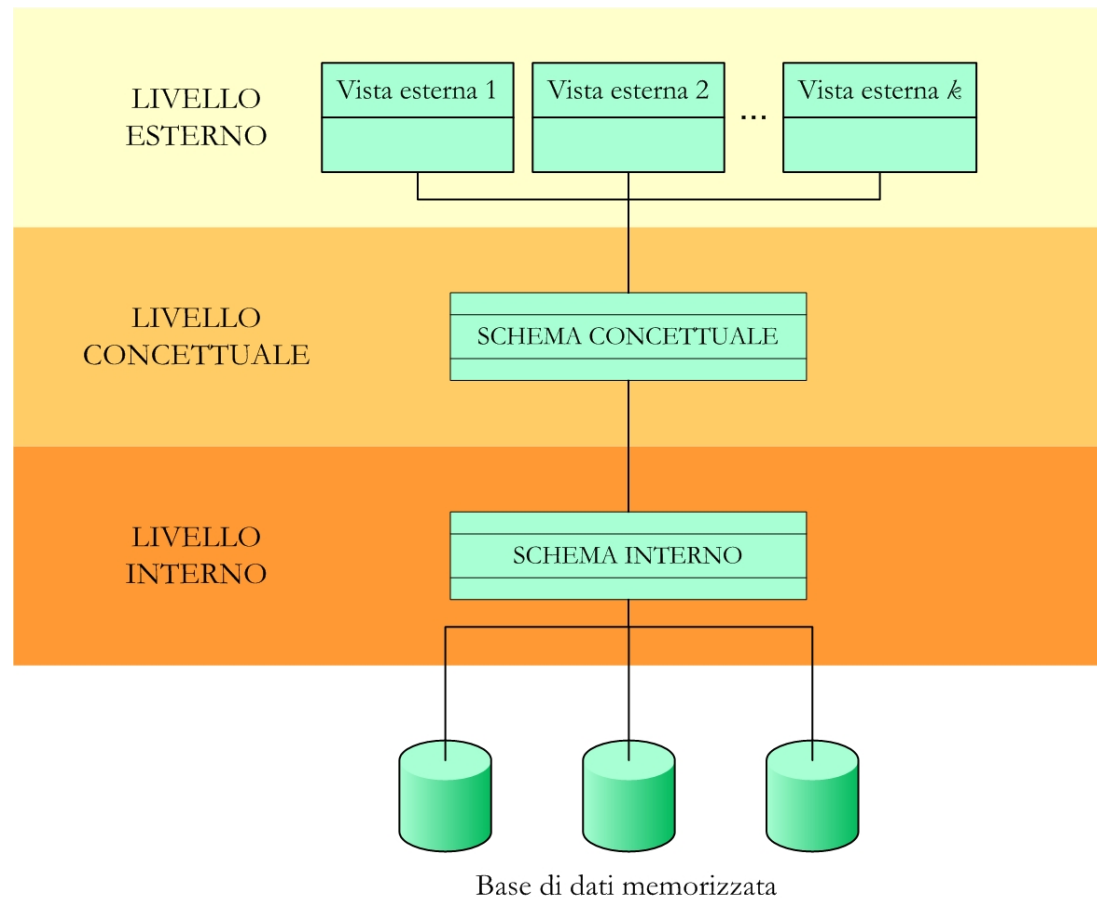
Descrive la struttura della base di dati per un insieme di utenti. Descrizione globale, che nasconde i dettagli della struttura della memorizzazione fisica, di entità, tipi di dati, relazioni tra entità, vincoli sui dati.

- **LIVELLO ESTERNO, o LIVELLO DELLE VISTE**

Descrive la vista della base di dati consentita ad un gruppo di utenti.

Architettura a 3 livelli di un DBMS

Utenti finali (soggetti o applicativi)



Vantaggi dell'architettura di un DBMS

1. Indipendenza tra dati e programmi
2. Possibilità di utilizzo di viste multiple sui dati a seconda dell'utente
3. Uso di un catalogo per memorizzare la struttura (schema) della base di dati
4. Possibilità di effettuare modifiche su uno qualunque dei livelli senza necessariamente andare a modificare gli altri

Linguaggi in un DBMS

- **Data Definition Language (DDL).** Il linguaggio di **definizione dei dati** consente di definire la struttura della base di dati, individuando per ogni attributo il nome e il tipo relativi
- **Storage Definition Language (SDL).** Il linguaggio di **costruzione dei dati** consente di definire lo schema interno di una base di dati, ovvero come le informazioni vengono memorizzate fisicamente

Linguaggi in un DBMS

• **View Definition Language (VDL).** Il linguaggio per la **definizione delle viste** consente di definire le viste per ciascun utente, cioè i modi secondo i quali vari utenti hanno la possibilità di vedere la parte di base di dati a cui è loro consentito l'accesso.

• **Data Manipulation Language (DML).** Il linguaggio di **manipolazione dei dati** consente di manipolare i dati, cioè di inserirli, modificarli, cancellarli, ricercarli. Il DML può essere inserito in un vero e proprio linguaggio di programmazione fornito dal DBMS oppure può essere usato interattivamente come linguaggio a sé stante, chiamato Query Language QL.

Utenti in un DBMS

E' possibile raggruppare gli utenti di un DBMS in base alla modalità di interazione:

1. Gestore della base di dati
2. Programmatori delle applicazioni
3. Utenti finali

Modelli dei dati: il modello fisico

- Fornisce i concetti che descrivono i dettagli di come i dati sono memorizzati nel calcolatore. Tali concetti sono rivolti agli sviluppatori, non agli utenti finali della base di dati.
- In una base di dati, supponete di avere una tabella che contiene 70 milioni di righe. Come la organizzereste in memoria? In un solo file? In più file? Con quale logica?

Modelli dei dati: il modello concettuale

- I modelli concettuali dei dati sono usati per descrivere i dati in maniera indipendente dal modello logico scelto.
- I modelli concettuali descrivono i concetti del mondo reale piuttosto che i dati utili a rappresentarli. Sono impiegati nella fase di progettazione concettuale di base di dati, per analizzare il dominio di interesse, senza sottostare ad aspetti di tipo implementativo.
- Vedremo il modello Entità-Relazione (Entity-Relationships Model)

Modello dei dati: il modello logico

- Sono i modelli dei dati effettivamente presenti nei Sistemi di Gestione di Basi di Dati (DBMS), cioè che possono essere implementati in calcolatori (modello relazionale, modello gerarchico, modello reticolare, modello a oggetti)
- Sono chiamati “logici” perché le strutture usate da questi modelli, pur astratte, seguono una particolare organizzazione:
 - modello relazionale → tabelle
 - modello gerarchico → alberi
 - modello reticolare → grafi
 - modello a oggetti → oggetti

Schemi e Istanze di una Base di Dati

- Lo **SCHEMA** di una base di dati è la descrizione, attraverso i concetti forniti dal modello dei dati, della struttura, in accordo con la quale sono memorizzati i dati (componente **intensionale**)
- L'**ISTANZA** di una base di dati fa riferimento al contenuto della base di dati in uno specifico momento (componente **estensionale**)

Cognome	Nome	Data_di_nascita	Sesso
Aldobrandi	Enrico	29/02/1960	M
Bianchi	Luca	08/05/1962	M
Mascheroni	Marinella	02/12/1965	F
Strozzi	Giulia	11/02/1964	F

Classificazioni dei DBMS

In base:

1. Alla distinzione del tipo di modello dei dati implementato
2. Al numero di utenti supportati contemporaneamente
3. Al numero di luoghi fisici ove risiedono base di dati e i programmi
 - DBMS centralizzati (tutti i dati immagazzinati in un unico elaboratore)
 - DBMS distribuiti (i dati sono distribuiti fisicamente su più elaboratori indipendentemente dal modo con il quale siano collegati e da quanto sia distanti tra loro)
 - DBMS federati (insieme eterogeneo di DBMS)
4. Costo

Vantaggi dei DBMS

1. I DBMS permettono di considerare i dati come una risorsa a disposizione dell'intera organizzazione (ad es., una base di dati clinica-ospedaliera può essere utilizzata per differenti scopi da differenti attori);
2. La base di dati fornisce un modello unificato della parte di mondo reale di volta in volta considerato
3. I DBMS permettono ai gestori della base di dati di controllare i dati in modo centralizzato
4. I DBMS permettono di ridurre inconsistenze e ridondanze
5. Indipendenza dei dati

Limiti dei DBMS

- Costi (dell'installazione, dell'istruzione del personale, della manutenzione HW/SW,...)
- Effettiva necessità di tutti i servizi offerti da un DBMS (DBMS vs applicazione *ad hoc*)