



# Basi Dati

Basi di dati e utenti di basi di dati

a.a 2020/2021

Prof.ssa G. Tortora

# Introduzione

# Sistemi informativi

- Sistema informativo
- Sistema informatico
- Informazioni
- Dati



# Sistema informativo

- Componente (sottosistema) di una organizzazione che gestisce (acquisisce, elabora, conserva, produce) le informazioni di interesse (cioè utilizzate per il perseguitamento degli scopi dell'organizzazione).



# Sistema informativo

- Ogni organizzazione ha un sistema informativo, eventualmente non esplicitato nella struttura.
- Quasi sempre, il sistema informativo è di supporto ad altri sottosistemi, e va quindi studiato nel contesto in cui è inserito.
- Il sistema informativo è di solito suddiviso in sottosistemi (in modo gerarchico o decentrato), più o meno fortemente integrati.

# Sistema organizzativo

- Insieme di risorse (persone, denaro, materiali, informazioni) e regole per lo svolgimento coordinato delle attività (processi) al fine del perseguimento degli scopi.



# Sistema organizzativo e sistema informativo

- Il sistema informativo è parte del sistema organizzativo.
- Il sistema informativo esegue/gestisce processi informativi (cioè i processi che coinvolgono informazioni).

# Sistemi informativi e automazione

- Il concetto di “sistema informativo” è indipendente da qualsiasi automatizzazione:
  - esistono organizzazioni la cui ragion d’essere è la gestione di informazioni (p. es. servizi anagrafici e banche) e che operano da secoli

# Sistema informatico

- porzione automatizzata del sistema informativo:
  - la parte del sistema informativo che gestisce informazioni con tecnologia informatica.

Sistema azienda

Sistema organizzativo

Sistema informativo

Sistema informatico

# I database

I database sono ormai una componente fondamentale della vita di tutti i giorni: molte delle nostre più banali attività ci portano ad interagire con qualche tipo di database.

Qualche esempio:

- Prenotazioni di alberghi, biglietti aerei,...
- Telepass / Viacard
- Ricerca nel catalogo elettronico di una biblioteca
- Richiesta di documenti
- Spesa al supermercato
- Operazioni bancarie
- ...

# Applicazioni dei database

Le attività appena descritte coinvolgono **applicazioni di database tradizionali**, avendo a che fare principalmente con testi e numeri.

I progressi tecnologici, però, stanno apreendo la strada a nuove interessantissime applicazioni di database:

- I Database Multimediali possono memorizzare immagini, videoclip, suoni, ecc..
- I Sistemi informativi geografici (GIS) possono memorizzare ed analizzare mappe ed immagini satellitari
- I sistemi Data Warehouse permettono di estrarre ed analizzare grandi quantità di dati
  - Forniscono un supporto al processo decisionale
- I motori di ricerca permettono di trovare informazioni sparse sul WWW
- Tecnologie di database real-time
  - Controllo industriale e processi di produzione

# Applicazioni dei database (2)

- Per comprendere le tecnologie elencate , dobbiamo partire dalle basi delle applicazioni di database tradizionali...

# La nozione di DataBase

Un *Database* (*DB* o *Base di Dati*) è una collezione di dati correlati:

- *Esempio*: una rubrica telefonica creata usando Access, Excel, ecc.

Per “dati” si intendono dei fatti noti, con un significato **Implicito**, che possono essere memorizzati.

*Es*: nome, cognome, indirizzo e telefono di un abbonato telefonico.

# La nozione di DataBase (2)

L'uso comune del termine database è più ristretto.  
Un 'database' deve presentare le seguenti proprietà:

- Rappresenta alcuni aspetti del mondo reale, detto **miniworld** o **Universo del Discorso** (UOD). Cambiamenti al miniworld sono riflessi nel database.
- È una collezione di dati logicamente correlati con qualche significato inerente.
  - Un assortimento casuale di dati non può correttamente essere considerato un database.
- È progettato, costruito e riempito di dati per un utilizzo specifico. Ha una tipologia ben definita di utenti ed è realizzato per delle applicazioni a cui tali utenti sono interessati.

# Dimensione di DataBase

Un database può avere **qualsiasi dimensione e complessità**  
**Esempi**

- Una rubrica telefonica personale può avere poche centinaia di voci.
- Il database dei contribuenti americani e delle relative dichiarazioni dei redditi ha delle dimensioni notevoli:
  - 100 milioni di contribuenti
  - mediamente 5 moduli per ciascuna dichiarazione,
  - 200 byte per ogni modulo:

$$100 \times 10^6 \times 200 \times 5 = 100 \times 10^6 \times 100 \times 2 \times 5 = \\ = 10^2 \times 10^6 \times 10^2 \times 10 = 10^{11} \text{ bytes}$$

- Tenendo traccia delle ultime quattro dichiarazioni, risulterebbe una base di dati di  $4 \times 10^{11}$  bytes = 400 Gigabytes
- Questo enorme ammontare di informazioni deve essere organizzato e gestito in modo tale che gli utenti possano interrogare, recuperare ed aggiornare i dati.

# Gestione di DataBase

- Un database può essere gestito manualmente (es. lo schedario di una biblioteca) o attraverso un elaboratore elettronico.
- Un database computerizzato può essere creato e gestito o da programmi realizzati “*ad hoc*” o da un “DBMS”
- DBMS = “Database management system”
  - Collezione di programmi

# DataBase Management System (DBMS)

# Il DBMS

- Un *database management system (DBMS)* è una collezione di programmi che permette di creare e manutenere una base di dati.
- È un software "*general-purpose*" che facilita la creazione, costruzione e gestione di database per differenti applicazioni.
- Fornisce un modo per memorizzare informazioni in strutture dati efficienti, scalabili e flessibili.

# Funzionalità di un DBMS

Un DBMS permette di:

- **Definire** un database
  - specificare per i dati da memorizzare nel database:
    - i tipi di dati
    - le strutture
    - i vincoli
- **Costruire** il database
  - memorizzare i dati stessi su qualche memoria di massa controllata dal DBMS.
- **Manipolare** il database
  - interrogare il database per ritrovare dati specifici aggiornare il database per riflettere i cambiamenti nel miniworld.
  - generare dei reports dai dati.

# Funzionalità di un DBMS (2)

- Gestisce i Meta-data
  - Contengono la definizione del database o le informazioni descrittive.
  - È memorizzato dal DBMS sottoforma di *catalogo* o *dizionario*.
- Condividere un database
  - Permette a differenti utenti e programmi di accedere al database in maniera simultanea.
- Programma applicativo
  - Accede al database inviando le query al DBMS.
- Query
  - Permette di recuperare i dati.

# Funzionalità di un DBMS (3)

- Transazioni
  - Permettono di leggere e scrivere dati nel database.
- Protezione che si differenzia in:
  - System protection,
  - Security protection.
- Manutenzione del sistema database
  - Permette di evolvere il sistema rispettando il cambiamento dei requisiti nel tempo.

# DBMS vs. Database

- Un DBMS è un applicativo per gestire database.
  - *Esempio:* MySQL
- Un database è un insieme di dati.
  - *Esempio:* file con estensione .MDB
- Stessa differenza esistente tra Word (**applicativo**) e file .DOC (**dati**).

# Principali DBMS

- > *Oracle*
- >  (DB2, Universal Server)
- > **Microsoft** (Access, SQL Server)
- > 
- > MySQL

# DBMS special-purpose

- Non è necessario usare un DBMS general-purpose.  
È anche possibile scrivere un proprio insieme di programmi per gestire i dati, creando un DBMS special-purpose.
- Tale approccio può essere vantaggioso nello sviluppo di soluzioni molto piccole.

# Il Database System

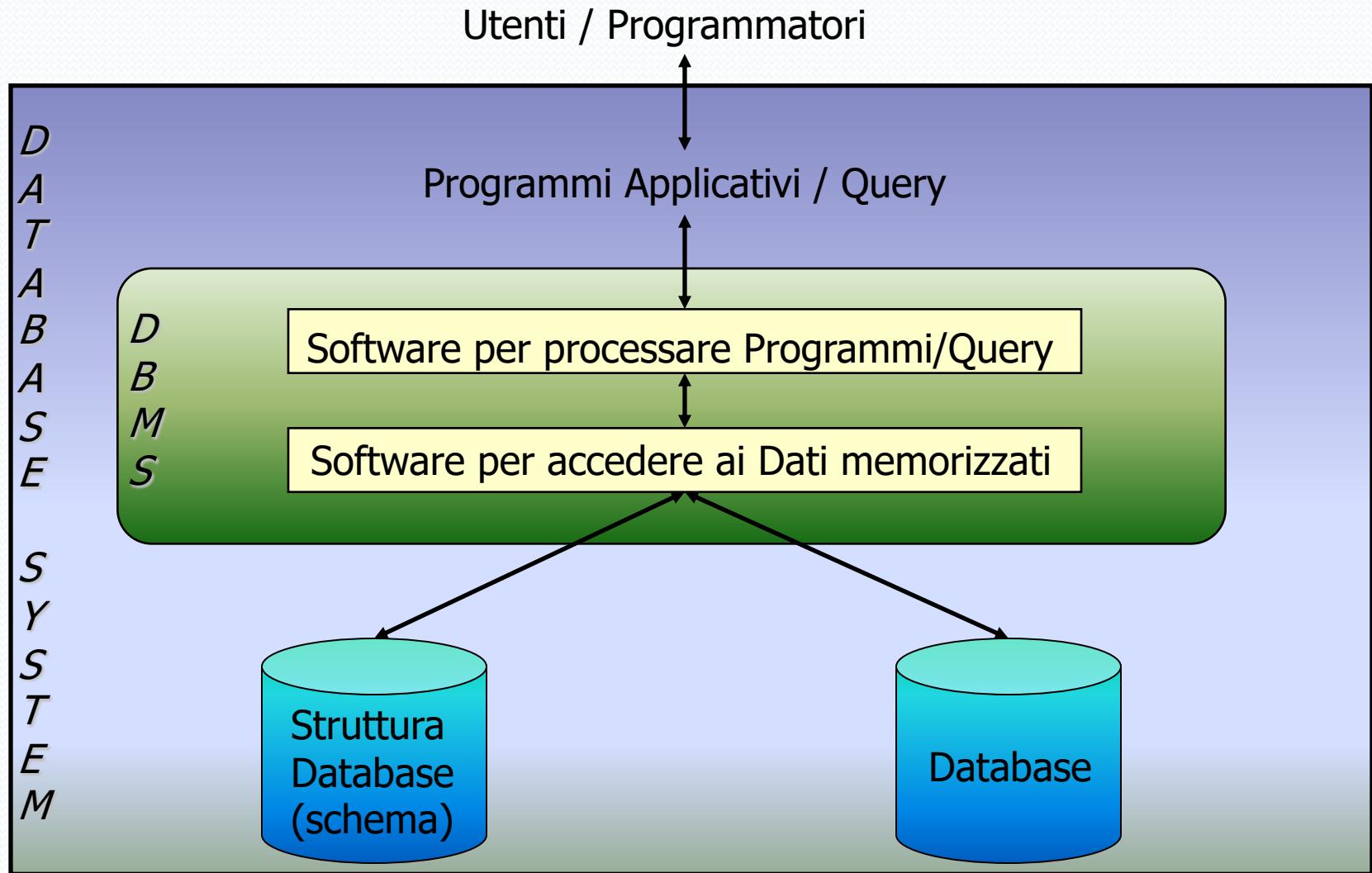
- Il DBMS (general o special purpose), insieme con un database costituisce un “Database System”:

Base di Dati + DBMS (special/general purpose)

=

Database System

# Il Database System (schema)



*Un esempio:* il database Università

# Un esempio di database

Vogliamo realizzare il database **UNIVERSITÀ** per gestire gli studenti, i corsi (con prerequisiti) e gli esami superati.

- Organizzato in quattro file:
  - STUDENTE:
    - Contiene i dati su ciascuno studente iscritto.
  - CORSO:
    - Contiene i dati relativi a ciascun corso.
  - PREREQUISITI:
    - Contiene i prerequisiti di ciascun corso.
  - VOTAZIONE:
    - Contiene i voti riportati dagli studenti nei vari esami.
- Ogni file memorizza dei record di dati dello stesso tipo.

# *Esempio: definizione del DB*

Per definire il database occorre specificare la struttura dei record di ciascun file.

Occorre cioè:

- Specificare i campi (data element) di ogni record.
- Specificare il tipo di ogni data element in ciascun record.

# *Esempio: definizione del DB (2)*

- I data element:
  - Un record del file STUDENTE contiene dati per rappresentare il **nome** dello studente, il numero di **matricola**, e l'**anno** di iscrizione corrente.

| STUDENTE | Nome    | Matricola  | Anno   |
|----------|---------|------------|--------|
|          | Rossi   | 056/000484 | 2 f.c. |
|          | Bianchi | 056/100084 | 5      |
|          | Verdi   | 011/120579 | 3      |
|          | ...     | ...        | ...    |

- Il tipo dei data element:
  - Gli elementi NOME, MATRICOLA ed ANNO sono tutti definiti come stringhe di caratteri.

# *Esempio: definizione del DB (3)*

La definizione del resto del database UNIVERSITÀ:

| CORSO | DENOMIN.         | SEMESTRE | TITOLARE |
|-------|------------------|----------|----------|
|       | Diritto Privato  | 1        | Amato    |
|       | Procedura Civile | 2        | Renzi    |
|       | Procedura Penale | 1        | Esposito |
|       | ...              | ...      | ...      |

| PREREQUISITI | DENOMIN.         | PROPEDEUTICITA' |
|--------------|------------------|-----------------|
|              | Diritto Privato  | nessuna         |
|              | Procedura Civile | Diritto Privato |
|              | Procedura Civile | Diritto Civile  |
|              | ...              | ...             |

| VOTAZIONE | NOME  | DENOMIN.         | VOTO |
|-----------|-------|------------------|------|
|           | Rossi | Diritto Privato  | 24   |
|           | Rossi | Procedura Civile | 28   |
|           | Verdi | Procedura Penale | 30   |
|           | ...   | ...              | ...  |

Per ognuno dei quattro file, viene specificato il tipo di ogni data element contenuto in ciascun record del file.

# *Esempio: costruzione del DB*

- Per *costruire* il database UNIVERSITÀ memorizziamo dati per rappresentare ogni studente, corso, prerequisito e votazione nel file appropriato.
- I record nei vari file possono essere **correlati**:

## *Esempio:*

- Il record per "Rossi" nel file STUDENTE è in relazione con due record nel file VOTAZIONE,
- Il record per "Procedura Civile" nel file CORSO è in relazione con due record nel file PREREQUISITI e con un record nel file VOTAZIONE.

# *Esempio: manipolazione del DB*

- *Manipolare* il database significa interrogare e aggiornare i dati.

## *Esempio di query:*

- Quanti esami ha sostenuto “Rossi”?
- Elencare gli studenti in corso.
- Calcolare la media dei voti di uno studente.
- ...

# *Esempio: manipolazione del DB (2)*

- *Manipolare* il database significa interrogare e aggiornare i dati.

*Esempio di update:*

- Inserire un nuovo studente.
- Registrare un esame.
- ...

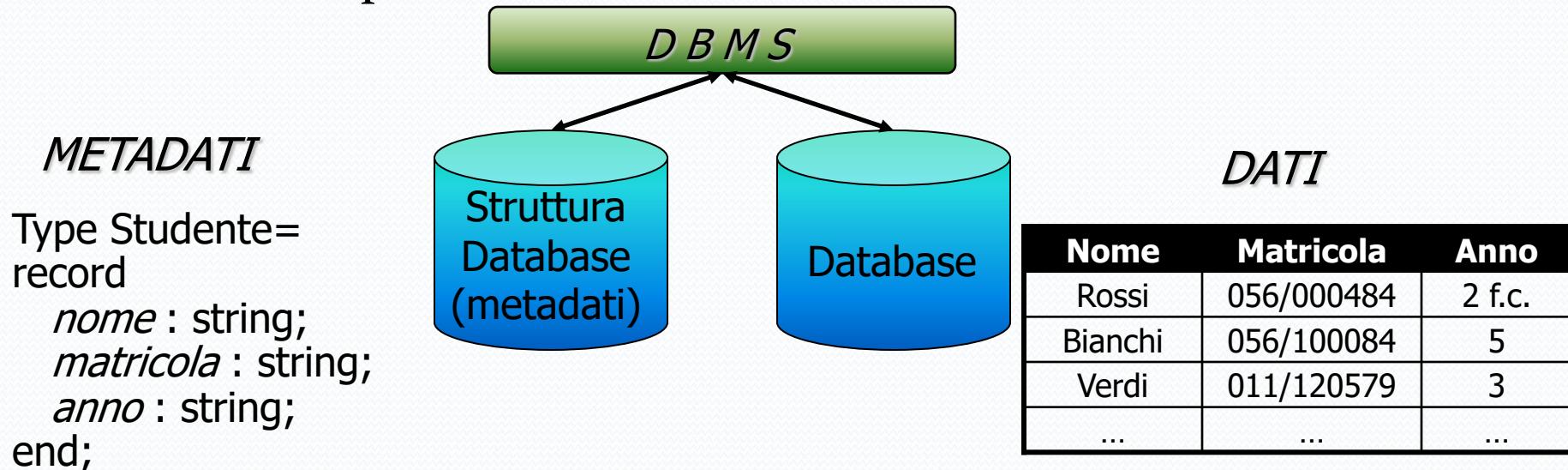
# Fasi per la progettazione di un database

- 1. Specifica ed analisti dei Requisiti**
- 2. Progettazione Concettuale**
- 3. Progettazione Logica**
- 4. Progettazione Fisica**

# Caratteristiche dell'utilizzo di un Database

# Natura Autodescrittiva di un database

- Il database non contiene solo i dati ma anche la **definizione completa** (o descrizione) del database.
- Le informazioni sulla definizione, dette **metadati**, sono memorizzate nel **catalogo di sistema**.
- Il DBMS accede al catalogo per conoscere la struttura dei file dello specifico database.



# FILE PROCESSING vs DATABASE

- **File processing:**

Ogni utente definisce ed implementa i file necessari per una specifica applicazione, con notevole spreco di risorse umane/informatiche e ridondanza dei dati.

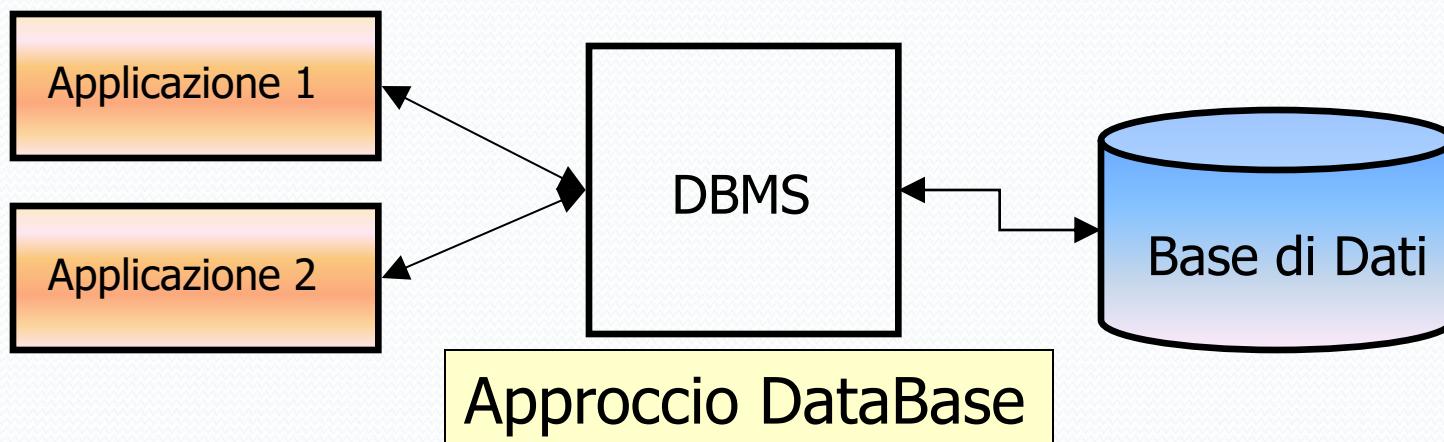
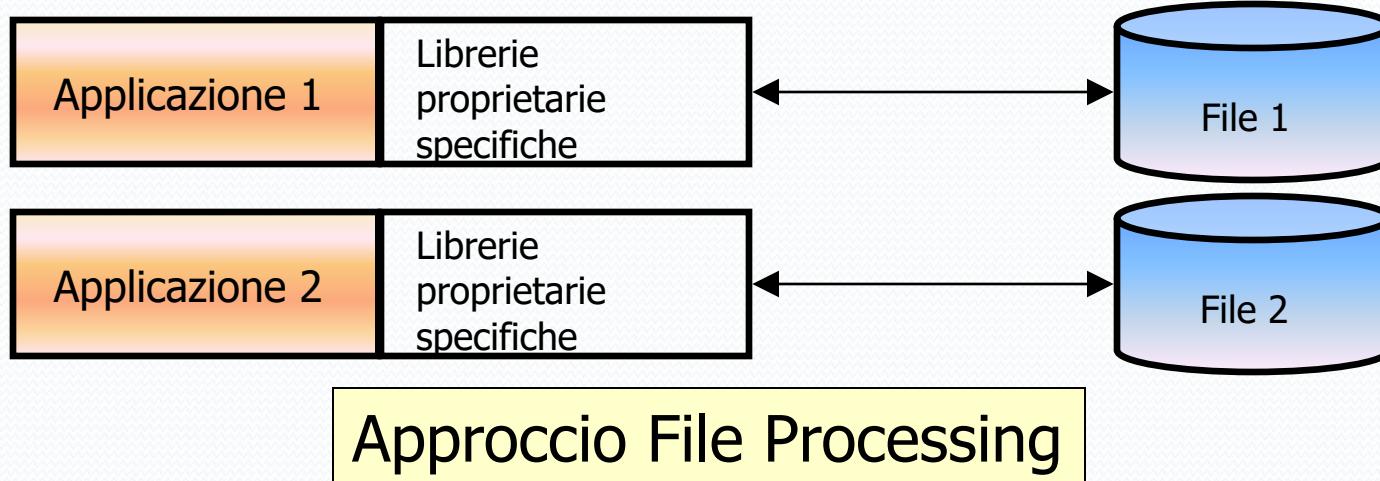
*Esempio:*

- L'ufficio esami mantiene un file che registra gli studenti e le votazioni relative
- L'ufficio iscrizioni mantiene un file che registra gli studenti e i relativi pagamenti delle tasse.

- **Database:**

Si definisce una volta per tutte un **singolo repository** di dati, poi utilizzato dai vari utenti

# FILE PROCESSING vs DATABASE (Schema)



# Indipendenza programma-dati

|                 |   |
|-----------------|---|
| File processing | <ul style="list-style-type: none"><li>◆ La struttura di un file dati è immersa nei programmi che accedono al file:<ul style="list-style-type: none"><li>■ un cambiamento nella struttura del file richiede un cambiamento in tutti i programmi.</li></ul></li></ul> |
| DBMS            | <ul style="list-style-type: none"><li>◆ I programmi di accesso sono scritti indipendentemente dagli specifici file.</li><li>◆ La struttura dei file dati è nel catalogo.</li></ul>  |

# Modelli di dati

- Un DBMS fornisce una **rappresentazione concettuale** dei dati che non include molti dei dettagli su come i dati sono memorizzati.
- Un **modello di dati** (data model) è un tipo di astrazione di dati usato per fornire la rappresentazione concettuale.
- Il modello di dati usa concetti logici quali oggetti, proprietà e loro interrelazioni, nascondendo quindi i dettagli fisici di memorizzazione.

# Proprietà di un Database

- Abilitazione di **viste multiple** dei dati:
  - Un database ha molti utenti e ciascuno può averne una diversa prospettiva (o vista):
    - Una vista può essere un sottoinsieme del database,
    - o può contenere **dati virtuali** (derivati dal database ma non esplicitamente memorizzati).
- Un DBMS deve consentire la definizione di viste multiple.

| ESAMI SUPERATI | Nome  | Matricola  | ESAMI            |      |
|----------------|-------|------------|------------------|------|
|                |       |            | DENOMIN.         | VOTO |
| Rossi          | Rossi | 056/000484 | Diritto Privato  | 24   |
|                |       |            | Procedura Civile | 28   |
| Verdi          | Verdi | 011/120579 | Procedura Penale | 30   |
|                |       |            | Diritto Civile   | 27   |

# Proprietà di un Database

- Condivisione di dati e trattamento di *transazioni* multiutente.
- Un DBMS multiutente deve consentire accesso a più utenti contemporaneamente.
  - **Online transaction processing (OLTP)**
- Deve includere software per il controllo della concorrenza che garantiscano l'aggiornamento corretto.

*Esempio:* il problema della prenotazione di posti per una compagnia aerea (applicazione di transaction processing).

# Proprietà delle transazioni

- Le transazioni dovrebbero possedere alcune proprietà (*dette ACID properties, dalle loro iniziali*):
  - **Atomicità:** una transazione è un'unità atomica di elaborazione da eseguire o completamente o per niente (*responsabilità del recovery subsystem*).
  - **Consistency preserving:** una transazione deve far passare il database da uno stato consistente ad un altro (*responsabilità dei programmatori*).

# Proprietà delle transazioni (2)

- **Isolation:** Una transazione non deve rendere visibili i suoi aggiornamenti ad altre transazioni finché non è committed (*responsabilità del sistema per il controllo della concorrenza*)
- **Durability:** Se una transazione cambia il database, e il cambiamento è committed, queste modifiche non devono essere perse a causa di fallimenti successivi (*responsabilità del sistema di gestione dell'affidabilità*)

# Le figure coinvolte nei DB



# I personaggi dell'ambiente

- DataBase Administrator (DBA):
  - Il DBA è responsabile per autorizzare l'accesso al database, coordinare e monitorare il suo uso, acquisire nuove risorse hardware e software.
  - In grosse organizzazioni è assistito da uno staff.
- Database Designer (Progettista):
  - È responsabile dell'individuazione dei dati da memorizzare nel DB;
  - È responsabile della scelta delle strutture opportune.
  - Deve capire le esigenze dell'utenza del DB e giungere a un progetto che soddisfi i requisiti:
  - Sviluppa viste dei dati;
  - Il DB finale deve essere in grado di supportare i requisiti di tutti i gruppi di utenti.

# I personaggi dell'ambiente (2)

- Utenti finali
  - utenti finali casuali: accedono occasionalmente al DB e lo interrogano attraverso query Languages sofisticati;
  - utenti finali naive o parametrici: rappresentano una parte considerevole di utenza. Usano aggiornamenti e queries standard;
  - utenti finali sofisticati: ingegneri, scienziati e analisti di affari che hanno familiarità con le facilities del DBMS per richieste complesse;
  - utenti stand-alone: gestiscono databases personali usando pacchetti applicativi.
- Analisti di sistema e Programmatori di applicazioni
  - gli analisti di sistema determinano i requisiti degli utenti finali e sviluppano specificazioni per le transazioni;
  - i programmatori di applicazioni implementano le specifiche come programmi.

# I personaggi dell'ambiente (3)

## *Altre figure*

- Progettisti e Implementatori di DBMS
  - Disegnano e implementano moduli e interfacce di DBMS come un pacchetto software. Un DBMS è un complesso sistema software che consiste di molti moduli.
- Sviluppatori di Tools
  - Implementano pacchetti software per il progetto, il monitoraggio, l'interfaccia, la prototipazione, ecc. di databases.
- Operatori e Personale per la Manutenzione
  - Personale che si occupa dell'amministrazione del sistema e della manutenzione hw e sw del sistema di database.

# Vantaggi nell'usare i DBMS

# Vantaggi nell'usare i DBMS

- Controllo della ridondanza
  - Normalizzazione dei dati.
- Limitare l'accesso non autorizzato
  - Sicurezza e sottosistema di autorizzazione.
- Fornisce una memoria *persistente* per gli oggetti dei programmi
  - Oggetti complessi in Java/C++ possono essere permanentemente memorizzati.

# Vantaggi nell'usare i DBMS (2)

- Forniscono strutture di memorizzazione e implementano tecniche di ricerca per eseguire le query in maniera efficiente:
  - **Indici**
  - **Buffering e caching**
  - **Elaborazione ed ottimizzazione delle query**
- Forniscono strumenti di *backup* e *recovery*
- Forniscono interfacce utente multiple
  - **Graphical user interface (GUI)**

# Vantaggi nell'usare i DBMS (3)

- Permettono l'uso dei Trigger:
  - Sono regole attivate dagli aggiornamenti apportati alle tavole.
  - Gestiscono le Stored procedure:
  - Sono procedure di supporto usate per rispettare le regole definite sul database.

# Vantaggi nell'usare i DBMS (4)

- Implicazioni addizionali nell'usare un approccio basato sui database
  - Riduce il tempo di sviluppo dell'applicazione.
  - flessibilità
  - Disponibilità di aggiornamento delle informazioni
  - Economie di scala
    - Vantaggio nel costo medio unitario all'aumentare della scala

# Quando non usare un DBMS

- È desirabile usare l'approccio basato sui *file*, quando:
  - Si sviluppano semplici e ben definite applicazioni di database che non prevedono modifiche.
  - Sono richiesti requisiti rigorosi real-time che non possono essere soddisfatte a causa del sovraccarico eseguito dal DBMS.
  - Si implementano sistemi embedded con una capacità di memorizzazione limitata.
  - L'accesso ai dati non richiede utenti multipli.