

# Basi di Dati

Introduzione ai sistemi di basi di dati

# Introduzione ai sistemi di basi di dati

---

- ▶ Gestione dei Dati
- ▶ Una prospettiva storica
- ▶ File system verso DBSM
- ▶ Vantaggi di un DBMS
- ▶ Modelli dei dati
- ▶ Utenti di un sistema di DBMS
- ▶ Linguaggi di “interazione” con la base di dati
- ▶ Architettura di un DBMS

# Introduzione ai sistemi di basi di dati

---

## **La gestione dei dati**

Nello svolgimento di ogni attività sono essenziali la disponibilità di informazioni e la capacità di gestirle in modo efficace.

Ogni organizzazione è dotata di un sistema informativo, che organizza e gestisce le informazioni necessarie per perseguire gli scopi dell'organizzazione stessa.



# Introduzione ai sistemi di basi di dati

---

## La gestione dei dati

Un sistema informatico deve gestire due aspetti fondamentali:

- i **dati** che rappresentano le informazioni di interesse per l'organizzazione
  - sono una risorsa preziosa per un'organizzazione
  - vanno conservati anche a fronte di cambiamenti tecnologici
- le **operazioni su tali dati** (acquisizione, archiviazione, elaborazione, ecc.)
  - In molti contesti applicativi i dati rappresentano l'aspetto essenziale di un'organizzazione.

# Introduzione ai sistemi di basi di dati

---

## **Base di dati (Database o Data Base):**

- Insieme (collezione) di dati utilizzati in uno specifico sistema Informativo, di tipo aziendale, scientifico, amministrativo o altro
- “codifica” l’informazione di una particolare realtà aziendale
- Deve rappresentare i diversi aspetti della realtà
  - oltre ai dati veri e propri anche le relazioni tra i dati

# Introduzione ai sistemi di basi di dati

---

## **Sistema di gestione di basi di dati (Database Management System o DBMS):**

- software progettato per aiutare a gestire e utilizzare grandi insiemi di dati
  - Come alternativa alla memorizzazione dei dati in file
- si interpone fra le applicazioni e la memoria di massa in cui si trovano collezioni di dati, per consentirne la gestione, in modo **indipendente dalle applicazioni stesse.**
- I dati non appartengono ad una specifica applicazione, ma le diverse applicazioni vi accedono attraverso il DBMS.



# Introduzione ai sistemi di basi di dati

---

## Database Management System o DBMS:

Normalmente le applicazioni accedono a dati *locali gestendoli attraverso file che appartengono alle applicazioni stesse.*

La finalità dei DBMS è di estendere le funzionalità del file system offrendo:

- nuove modalità di accesso ai dati
- condivisione dei dati
- gestione più sofisticata dei file



# Una prospettiva storica

---

Memorizzare e manipolare dati è stato uno degli obiettivi principali dai primi giorni dei computer.

- ▶ Il primo DBMS per uso generico (*Integrated Data Store*), fu progettato da Charles Bachman alla General Electric nei primi anni '60
  - ▶ costituì la base per il **modello di dati reticolare**
- ▶ Sul finire degli anni '60, l'IBM sviluppò il DBMS Information Management System (IMS)
  - ▶ fornì la base per una struttura alternativa di rappresentazione dei dati chiamata **modello di dati gerarchico**
- ▶ Nel 1970 Edgar Codd propose una nuova struttura di rappresentazione dei dati chiamata **modello di dati relazionale**
- ▶ Negli anni '80 il modello relazionale consolidò la sua posizione di dominanza.
  - ▶ Il linguaggio di interrogazione **SQL** per basi di dati relazionali, sviluppato come parte del progetto System R dell'IBM, è divenuto il linguaggio di interrogazione standard.



# Una prospettiva storica

---

I modelli logici, seppure astratti, riflettono la struttura con cui i dati sono organizzati.

➤ **Relazionale**

Il più diffuso, basato su un modello tabellare dei dati

➤ **Gerarchico**

Nei primi DBMS (anni 60), tuttora utilizzato, basato su strutture ad albero

➤ **Reticolare**

Estensione del modello gerarchico, basato su grafi

➤ **A oggetti**

Estensione del modello relazionale basato sui paradigmi OOP  
(*Object Oriented Programming*)

# Dagli Archivi ai DBMS

Prima dello sviluppo dei DBMS l'approccio tradizionale al problema dell'archiviazione dei dati:

- Utilizzo diretto delle strutture del **file system**

## Differenza tra approccio file system e DBMS:

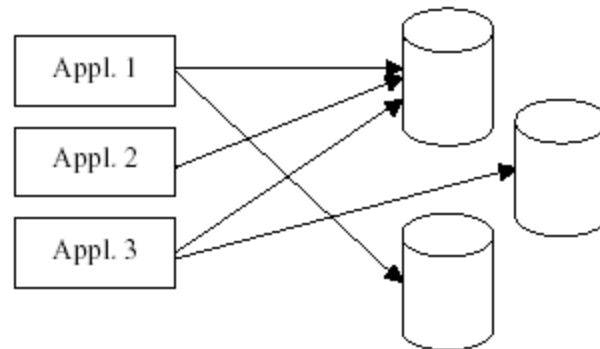
- Presenza di un **componente specializzato** in tale ruolo



# Approccio file system

---

Le applicazioni accedono direttamente agli archivi, quindi ognuna deve conoscere la struttura interna degli archivi e le relazioni tra i dati e deve evitare la duplicazione degli stessi. Inoltre la non volatilità dei dati e la gestione degli accessi contemporanei di più applicazioni agli archivi viene relegata a strati software non specializzati per tali compiti, quali il sistema operativo.



Approccio file system

# Approccio file system

---

- ▶ Nell' approccio "classico" (**file system**):
  - ▶ si utilizza il concetto di archivio e registrazione (file e record)
  - ▶ si programmano le varie procedure con un linguaggio di programmazione standard (es. COBOL, C). Esse accederanno agli archivi ed alle registrazioni; la struttura di questi ultimi deve essere nota a tutti gli applicativi.
- ▶ Archivio (o file):
  - ▶ è un insieme di registrazioni (record)
- ▶ Registrazioni (o record):
  - ▶ insieme di informazioni elementari dette attributi o campi o field

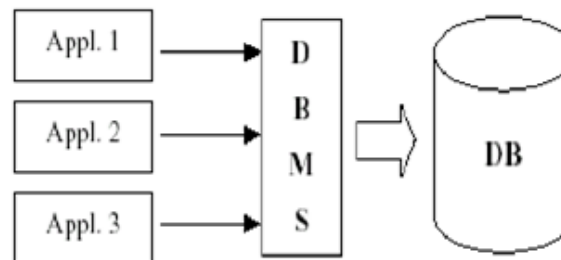
# Approccio DBMS

---

Le applicazioni rivolgono al **DBMS** le proprie richieste di accesso alla base di dati, il quale gestisce i dati svincolando le applicazioni da tale onere.

- le funzionalità di gestione del database sono raggruppate in un unico insieme
- le applicazioni risultano alleggerite e quindi più veloci da realizzare e, soprattutto, nessuna potrà effettuare operazioni scorrette sul database

**DBMS** ➡ **modulo specializzato** nella gestione del DB a cui tutte le applicazioni si rivolgono per accedere ai dati



Approccio DBMS

# Limiti approccio file system

---

Una tecnica di progettazione basata su archivi e procedure di accesso ai dati è insufficiente

▶ non risolve i problemi di:

▶ **Ridondanza e inconsistenza**

- ▶ Poiché i file ed i programmi applicativi sono creati da differenti programmatori e su intervalli di tempo lunghi, i file possono avere differente formato ed i programmi essere scritti in linguaggi diversi
- ▶ I dati possono essere replicati in più file, con consumo in termini di tempo di accesso e di risorse di memorizzazione
- ▶ La ridondanza può portare ad inconsistenza
  - esempio: il cambiamento dell'indirizzo deve essere riportato in tutti i file che contengono tale informazione

# Limiti approccio file system

---

- ▶ **Difficoltà di accesso ai dati**

- ▶ manca un meccanismo efficiente di ricerca delle informazioni (necessità di un “linguaggio di interrogazione”)

- ▶ **Concorrenza**

- ▶ solitamente viene concessa la possibilità a più utenti di accedere contemporaneamente al sistema per svolgere operazioni di aggiornamento sui file (accesso concorrente)

- ▶ **Sicurezza**

- ▶ Non tutti gli utenti del sistema devono essere autorizzati ad accedere a tutti i dati

- ▶ **Integrità**

- ▶ i dati contenuti nel sistema informativo devono soddisfare alcuni vincoli di consistenza.

# DBMS

---

Occorre un approccio “**integrato**” alla definizione ed alla gestione dei dati e delle procedure di accesso ad essi

L'utente non deve definire le procedure e le tecniche di controllo, ma accedere e manipolare i dati con l'assistenza del “sistema”

- **DataBase Management System** (DBMS) sono sistemi per la gestione e l'elaborazione “automatizzata” di informazioni di grandi dimensioni
  - tecnicamente, sono un insieme di dati correlati e di procedure per accedere ai dati
- **Obiettivo primario** di un DBMS è fornire un ambiente che sia “conveniente” ed “efficiente” nella memorizzazione e recupero delle informazioni contenute nel DB



# Obiettivi nella realizzazione di un DBMS

---

- **Integrazione dei dati**

un unico insieme di dati e tanti programmi che accedono solo sui dati di loro pertinenza e interesse (eventualmente vedendoli in modo disaggregato e poi riaggregato in altra maniera). L'integrazione limita la ridondanza e annulla la probabilità di inconsistenza.

- **Trasportabilità**

Il sistema deve essere capace di adattarsi, ad esempio, ad innovazioni tecnologiche dell'hardware o del software di sistema senza necessità di interventi massicci.

- **Astrazione dei dati**

Gli utenti del sistema hanno visioni dei dati "ad alto livello", ossia vengono nascoste all'utente le modalità con cui vengono implementati e memorizzati i dati.



# Vantaggi di DBMS rispetto file system

---

## Coerenza dei dati

- I **file system** non prevedono verifiche
  - diversi file possono essere creati, cancellati e modificati indipendentemente l'uno dall'altro
  - questo può creare incoerenze rispetto alle informazioni che il contenuto dei file rappresentano
- Nei **DBMS**
  - sono presenti meccanismi che permettono di garantire che venga mantenuta la coerenza reciproca dei dati

# Vantaggi di DBMS rispetto file system

---

## Condivisione dei dati

- I **file system** prevedono forme di condivisione
  - accesso condiviso in lettura
  - accesso esclusivo in scrittura: se è in corso un'operazione di scrittura su un file, altri non possono accedere al file
- Nei **DBMS**
  - c'è maggiore flessibilità e granularità
  - si può accedere contemporaneamente a record diversi di uno stesso file o addirittura allo stesso record (in lettura)

# Vantaggi di DBMS rispetto file system

---

## Integrazione dei dati

Nell' accesso ai file:

- ogni programma contiene una descrizione della struttura del file stesso
- nascono rischi di incoerenza fra le descrizioni (ripetute in ciascun programma) e i file stessi

Nei **DBMS**:

- esiste una porzione della base di dati (il catalogo o dizionario) che contiene una descrizione centralizzata dei dati, che può essere utilizzata dai vari programmi.

# Perché condividere i dati?

---

- ▶ I diversi settori in cui si articola una (grande) organizzazione e/o le diverse applicazioni hanno in comune vari dati di interesse. Pertanto la gestione integrata e la condivisione di tali dati permettono di evitare ripetizioni
  - ▶ **ridondanza** dovuta a copie multiple dello stesso dato
  - ▶ inutile **spreco di risorse** (memoria)
- ▶ Inoltre, la ridondanza può dar luogo a problemi di inconsistenza delle copie e, in ogni caso, comporta la necessità di propagare le modifiche, con ulteriore spreco di risorse (CPU e rete)
  - ▶ Il settore Ordini di un'azienda manifatturiera memorizza i propri dati in un file, non condiviso con gli altri settori aziendali. Ogni volta che arriva un ordine, i dati relativi devono essere trasmessi al settore Spedizioni, affinché l'ordine possa essere evaso. A spedizione eseguita, i dati relativi devono essere ritrasmessi al settore Ordini

# Astrazioni sui Dati

---

- ▶ Le basi di dati supportano ***astrazioni sui dati***:
  - ▶ *livello fisico*
  - ▶ *livello logico/concettuale*
  - ▶ *livello delle viste*

Gli utenti del DBMS non devono essere necessariamente utenti esperti. Il sistema maschera i dettagli implementativi attraverso differenti livelli di astrazione, per semplificare la loro interazione con il sistema.

# Livelli di astrazione

---

## Livelli di astrazione dei dati:

- **Livello Fisico**
  - ✦ è il più basso livello di astrazione. Descrive come i dati sono memorizzati nel database (file, blocchi, etc.)
  - ✦ La gestione del DB fisico è a carico di chi amministra il DBMS e non degli utenti, che quindi possono concentrarsi su aspetti di più alto livello per lo sviluppo delle loro applicazioni
- **Livello Concettuale/Logico**
  - ✦ è il livello successivo. Descrive quali dati sono memorizzati nel database e le relazioni che intercorrono fra di essi, tralasciando i dettagli implementativi
- **Livello “View” (delle viste)**
  - ✦ sono visioni parziali al di sopra del Livello Concettuale. Vengono definite in quanto non tutti gli utenti hanno la necessità di conoscere tutta la struttura concettuale del database, ma solo le parti di effettivo interesse. Possono esistere più viste dello stesso database

# A cosa servono le viste?

---

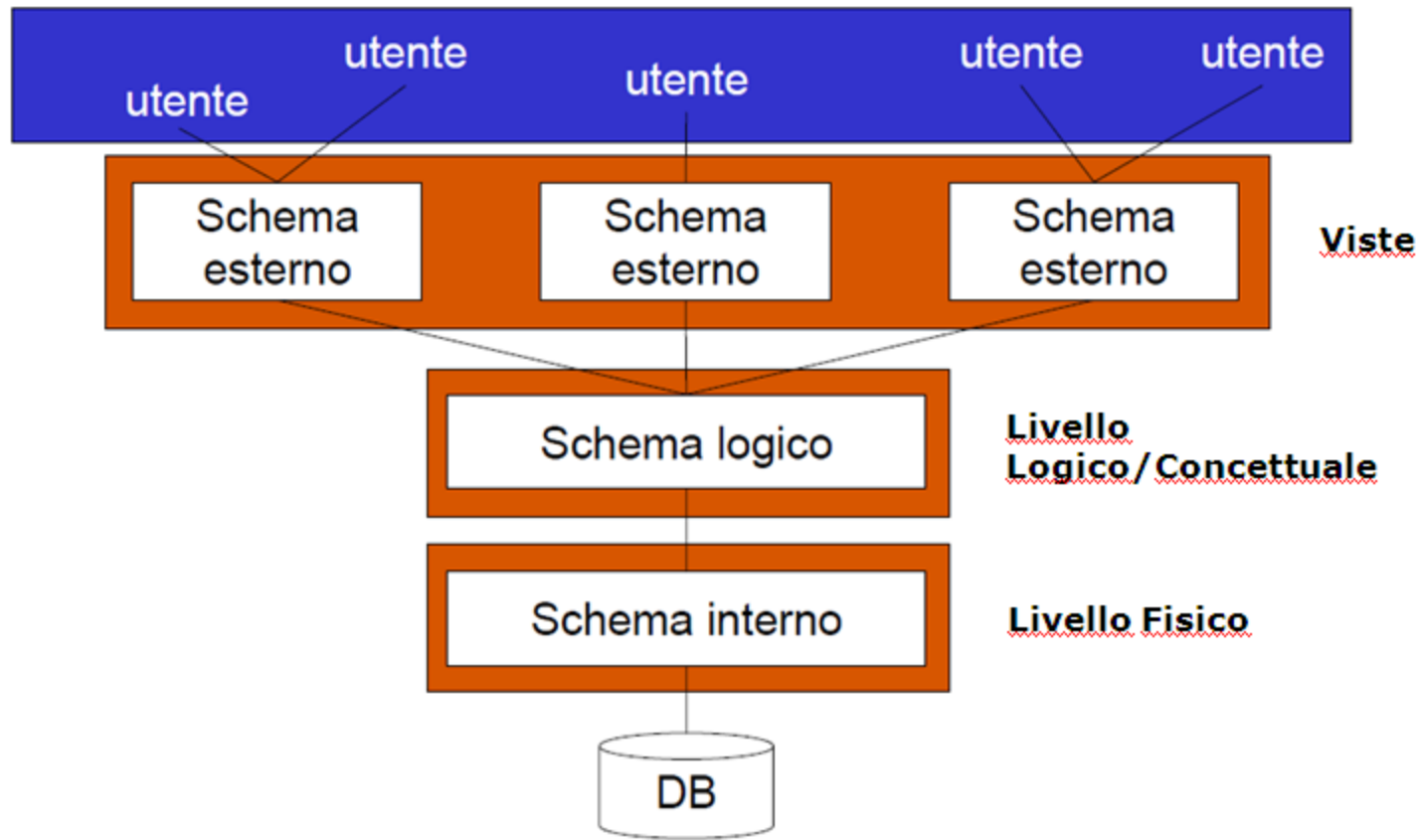
Oltre a fornire una visione “personalizzata” del DB, le **viste** possono svolgere un ruolo importante anche per diversi altri motivi:

- ▶ Una ristrutturazione dello schema logico integrato può, in alcuni casi, essere opportunamente “mascherata” facendo uso di viste
- ▶ Mediante le viste è possibile regolare meglio il **controllo degli accessi** al DB
- ▶ Le viste possono essere usate per “calcolare” dinamicamente nuovi dati a partire da quelli memorizzati nel DB, senza per questo introdurre ridondanza



# Livelli di astrazione

I tre livelli di astrazione dei dati e le loro relazioni:



# Modelli dei dati

---

- ▶ Un **modello dei dati** e' una collezione di “tool” concettuali per descrivere i dati, le loro relazioni e i vincoli di consistenza sui dati.
- ▶ I livelli di astrazione forniti da un DBMS vengono descritti in termini di un opportuno modello dei dati (il DBMS supporta uno specifico modello dei dati)

**Tramite il modello dei dati viene fornita una “interpretazione” della realtà che occorre “implementare” nella base di dati**

# Modelli dei dati

---

## Due tipi (principali) di modelli:

- **Modelli logici:** utilizzati nei DBMS esistenti per l'organizzazione dei dati
  - ad essi fanno riferimento i programmi
  - sono indipendenti dalle strutture fisiche
  - **relazionale, reticolare, gerarchico, a oggetti**
- **Modelli concettuali:** permettono di rappresentare i dati in modo indipendente da ogni sistema, cercando di descrivere i concetti del mondo reale
  - sono utilizzati nelle fasi preliminari di progettazione
  - il più noto è il modello **Entità-Relazione**

# Progettazione Basi di Dati

---

Grossa parte del lavoro di progettazione di una base di dati consiste nella definizione di un opportuno **schema dei dati** in relazione al problema da risolvere.

A tale attività, seguirà lo sviluppo delle procedure di creazione, manipolazione e accesso ai dati.



# Istanze e Schemi di una base di dati

---

- L' **Istanza** di un database è l'insieme delle informazioni contenute nel database ad un certo istante
  - cambiano di frequente nel tempo
  - aggiornamenti e modifiche dei dati, ma non della loro struttura
- La struttura complessiva di un database, descritta in termini di un opportuno modello dei dati, è lo **Schema** del database
  - cambia di rado nella “vita” della base di dati
    - un cambiamento dello schema è frutto o di una errata progettazione, o di un cambiamento della “realtà”

# Indipendenza dei dati

---

La possibilità di modificare la definizione di uno schema ad un livello di astrazione senza dover modificare la definizione del livello di astrazione successivo è chiamata **indipendenza dei dati**.

- Due livelli di indipendenza dei dati dovrebbero essere garantiti da un database:
  - **Indipendenza fisica**
    - ✦ la capacità di modificare lo schema fisico senza dover riscrivere i programmi applicativi . Ciò avviene solitamente per ottimizzare le prestazioni
  - **Indipendenza logica/concettuale**
    - ✦ la capacità di modificare lo schema concettuale senza dover riscrivere i programmi applicativi

# Esempi di interazione con Basi di Dati

---

Biblioteche (Libri, Prestiti, Utenti,...)

Banche (Clienti, Conti Correnti,...)

Gestione Magazzino (Giacenze, acquisti, vendite,...)

Università (studenti, docenti, corsi,...)

Obiettivo: organizzazione di grandi quantità di dati e loro condivisione



# Linguaggi di “interazione” con la base di dati

---

L'interazione fra utenti del DB e DB avviene utilizzando differenti linguaggi

- ▶ In un DBMS il livello di astrazione di tali linguaggi dipende fortemente dal modello dei dati cui ci si riferisce

## Data Definition Languages (DDL):

- consente di definire lo schema di un database e le autorizzazioni per l'accesso
- i risultati della esecuzione degli statement DDL sono i vari schemi, memorizzati nel *data-dictionary*
- il data-dictionary è un file che viene consultato prima che i dati effettivi del database siano letti e/o scritti

## Data Storage and Definition Languages (SDL)

- vengono utilizzati per specificare le strutture di memorizzazione e i metodi di accesso utilizzati nella base di dati
- il risultato dell'esecuzione di tali statement sono un insieme di istruzioni che specificano i dettagli implementativi dello schema della base di dati





# Linguaggi di “interazione” con la base di dati

---

## Data Manipulation Languages (DML):

- consente agli utenti la manipolazione dei dati organizzati nel relativo livello di astrazione
  - il reperimento dell'informazione memorizzata nel database
  - l'inserimento o la cancellazione di informazioni dal database
  - la modifica dei dati memorizzati nel database
- la parte di DML che riguarda il recupero dell'informazione prende il nome di **“Linguaggio di Interrogazione “ (QL)**

## Due tipologie di DML:

- **procedurali**: l'utente specifica *quale* dato è richiesto e *come* ottenerlo
- **non procedurali**: l'utente specifica *quale* dato è richiesto, senza occuparsi del *come* ottenerlo

# Linguaggi di “interazione” con la base di dati

---

## Considerazioni:

- ▶ più che di differenti linguaggi si tratta di differenti funzionalità presenti in un unico linguaggio
  - ▶ SQL contiene tutti i ruoli
    - per cui si parla del DDL di SQL, del DML di SQL ...
- ▶ l'utilizzo delle diverse funzionalità avviene in momenti diversi, e con privilegi e ruoli diversi
- ▶ una volta definito lo schema logico della base di dati, la progettazione fisica della base di dati avviene ricorrendo ai DDL
  - ▶ il caricamento dei dati avviene tramite i DML
  - ▶ la ricerca e la manipolazione delle informazioni avviene tramite le query (ossia i linguaggi di interrogazione)
  - ▶ L'ottimizzazione avviene tramite i Data Storage Language

# Gli utenti di una BD

---

Gli utenti di un DB sono naturalmente suddivisibili in diverse tipologie, a cui vanno pertanto associate autorizzazioni distinte.

- ▶ La gestione delle autorizzazioni può essere oltremodo complessa, per questo motivo sono previste specifiche figure che conferiscono agli utenti i “giusti” privilegi
- ▶ Utenti che hanno il controllo completo del sistema
  - ▶ **DBA (Data Base Administrator)**
- ▶ Utenti che possono essere classificati sulla base dell’esperienza d’uso e sulle funzionalità
  - ▶ **Sviluppatori di applicazioni**
  - ▶ **Utenti “sofisticati”**
  - ▶ **Utenti “specializzati”**
  - ▶ **Utenti “inesperti”**

# Gli utenti di una BD

---

## **DBA (Data Base Administrator)**

- ▶ Tra i compiti di un **DBA** ci sono:
  - ▶ definizione degli schemi
  - ▶ definizione delle strutture dati e dei metodi di accesso
  - ▶ modifica degli schemi e dell'organizzazione fisica
  - ▶ definizione delle politiche di accesso ai vari utenti e concessione dei permessi
  - ▶ specifica dei vincoli di integrità

# Gli utenti di una BD

---

## **Utenti “Sofisticati”**

- interagiscono con il sistema direttamente tramite il DML, ad esempio per funzioni di supporto alle decisioni (DSS) on-line

## **Sviluppatori di Applicazioni**

- sviluppano applicativi, utilizzando linguaggi di programmazione ad alto livello (es. C, Java, Cobol, PL/I) al cui interno sono contenuti statement di DML



# Gli utenti di una BD

---

## Utenti “Specializzati”

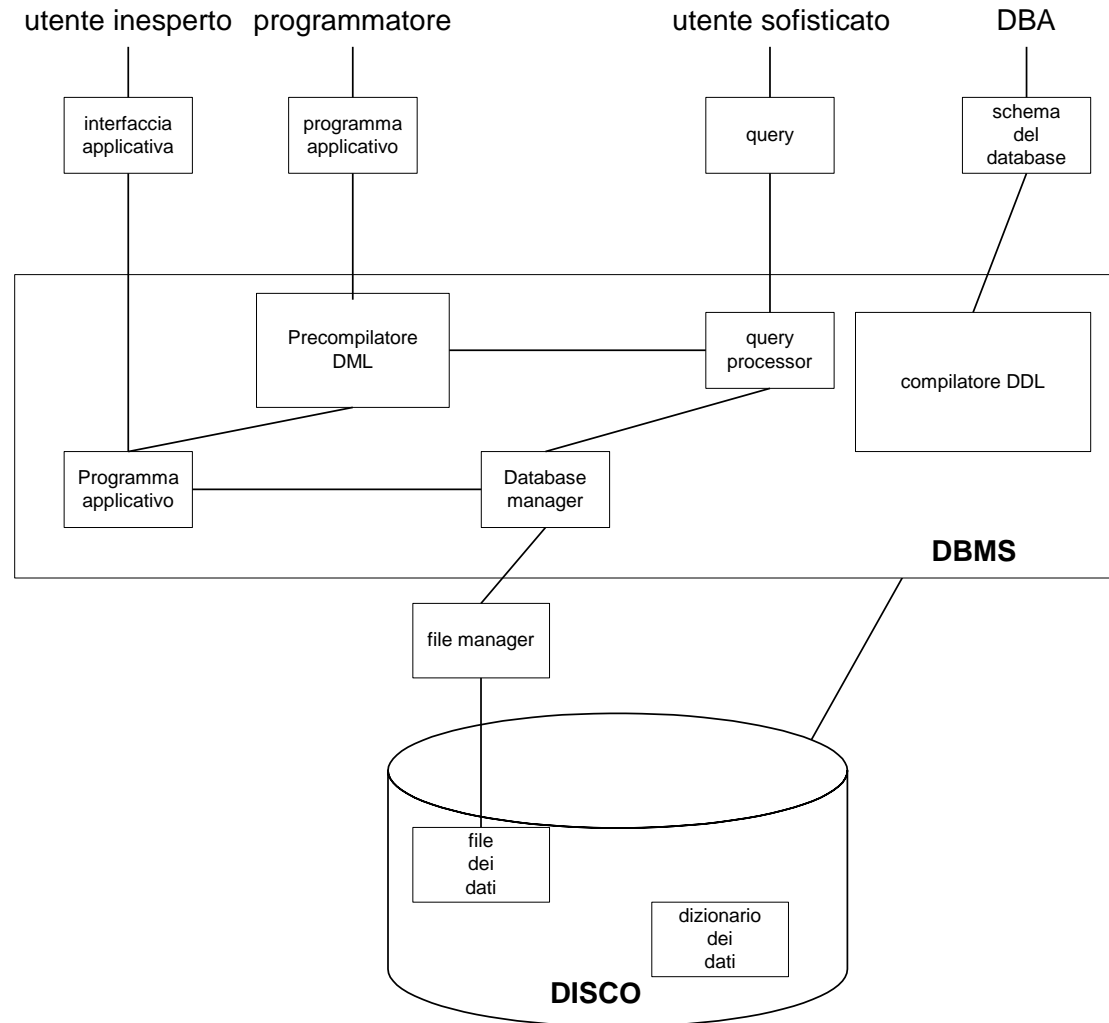
- sono utenti “s sofisticati” che interagiscono con il sistema per svolgere accessi “non convenzionali” solitamente hanno conoscenza dello schema fisico della base di dati
  - *debug di applicazioni, test sulle prestazioni e sul consumo di risorse*

## Utenti “Inesperti”

- utenti finali dell’applicativo, accedono al sistema solo tramite le maschere fornite dalle applicazioni



# Architettura di un DBMS



# Architettura di un DBMS

---

## Database Manager

- ▶ è il core del sistema
- ▶ fornisce l'interfaccia fra i dati (di livello fisico) memorizzati nel database e i programmi applicativi e le interrogazioni (query) sottoposte al sistema
- ▶ svolge i seguenti compiti:
  - interazione con il file manager
    - ▶ i dati sono memorizzati su disco utilizzando il file system fornito dal sistema operativo. Il database manager traduce gli statement DML in comandi validi per il file system
  - Controllo ed imposizione dei vincoli di integrità
  - Gestione della sicurezza
  - Gestioni delle funzionalità di back-up e recovery
    - ▶ è di competenza del DB Manager provvedere al recupero da situazioni di errore, dovute a crash di sistema o ad errori del software. Tale attività deve essere il più possibile automatizzata
  - Controllo della concorrenza



# Architettura di un DBMS

---

## File Manager

- gestisce l'allocazione dello spazio su disco e le strutture dati usate per rappresentare l'informazione memorizzata sul disco

## Query Processor

- traduce gli statements del DML in istruzioni a basso livello che il Database Manager è in grado di capire
- si occupa delle strategie di ottimizzazione, ossia traduce le query dell'utente in query equivalenti ma di più rapida esecuzione

## Precompilatore DML

- converte gli statement DML contenuti in un programma applicativo in normali chiamate di procedura nel linguaggio in cui è scritta l'applicazione. Il precompilatore deve interagire con il query processor per generare il codice appropriato

## Compilatore DDL

- converte gli statement DDL in un insieme di tabelle contenenti meta-dati ossia "dati sui dati"



# Architettura di un DBMS

---

**Strutture dati che completano la parte fisica del sistema**  
(i precedenti elementi sono processi o funzionalità nell'ambito di un processo)

## **File di Dati**

in essi sono contenute le informazioni del database

## **Data Dictionary**

- conserva informazioni sulla struttura del database
- Accesso da parte delle query per verificare la correttezza delle interrogazioni
- è utilizzato in modo pesante dal sistema

## **Indici**

- sono strutture dati che accelerano il reperimento dell'informazione



# Conclusioni

---

- ▶ Un **DBMS** è un sistema software complesso
- ▶ I sistemi DBMS servono per la gestione (memorizzazione, aggiornamento, recupero e manipolazione) di informazioni di grosse dimensioni
- ▶ la dimensione del problema e la necessità di prestazioni fanno sì che non si possa affrontare il problema in modo “euristico”
- ▶ i DBMS forniscono un approccio “integrato” alla gestione dei dati
  - ▶ forniscono livelli di astrazione sui dati
    - ▶ Mediante un’organizzazione a 3 livelli, un DBMS permette di ottenere gradi di **indipendenza fisica** e **logica** dei dati
  - ▶ l’accesso ai dati avviene tramite DML e SDL
  - ▶ si occupano dei problemi di protezione, concorrenza e backup

# Conclusioni

---

- ▶ i sistemi DBMS liberano l'utente da aspetti tecnici di gestione dei dati, lasciandogli la possibilità di concentrarsi sugli aspetti informativi
  - nessun dettaglio fisico è lasciato all'utente
  - nessun aspetto della gestione della sicurezza (che deve essere definita dall'utente)
  - nessun aspetto della gestione della concorrenza
  - l'utente ha visione dei dati ad alto livello ed accede ad essi tramite DDL, DML ed SDL