



## Introduzione al corso

Joseph Giovanelli  
Università di Bologna

# Sistemi informativi e Basi di Dati

- **Sistema Informativo:** complesso sistema di procedure, informatizzate e non, che permettono di gestire le informazioni utili ai processi aziendali. La definizione e la gestione di un Sistema Informativo richiede quindi la perfetta conoscenza dei:
  - ▮ Processi aziendali
  - ▮ Informazioni necessarie ai processi
  - ▮ Struttura aziendale
- Progettare un Sistema Informativo richiede non solo competenze informatiche, ma anche in ambito di organizzazione aziendale, economia, psicologia, ecc.
- La **base di dati** è una componente del Sistema Informativo atta alla memorizzazione strutturata delle informazioni.
  - ▮ La sua funzione è quella di fornire un **supporto informatico** per la memorizzazione dei dati;
  - ▮ essendo questi il nucleo di tutti i sistemi informativi le basi di dati rivestono un **ruolo centrale**.

# Database e DBMS



- Un **database** in termini generali rappresenta una collezione di dati d'interesse per una o più applicazioni; nel contesto del corso è intesa come una collezione di dati gestita tramite un DBMS. I dati sono strutturati e collegati tra loro, a **livello logico**, nel rispetto del **modello di rappresentazione** (es. relazionale) adottato dal DBMS e, a **livello fisico**, risiedono su dispositivi di memoria organizzati in particolari strutture. Gli utenti si interfacciano con la base di dati attraverso un opportuno **linguaggio** (es. SQL).
- Un **DBMS** (**D**ata **B**ase **M**anagement **S**ystem) è sistema software in grado di gestire **efficientemente** le informazioni necessarie a un SI, rappresentandone i dati in **forma integrata**, secondo un **modello logico**, e garantendone la **persistenza**, la **condivisione**, l'**affidabilità** e la **riservatezza**.

# Applicazioni (1)

## □ Applicazioni "tradizionali"

▮ gran parte dell'informazione memorizzata è testuale o numerica

- OLTP (On-Line Transactional Processing)
- OLAP (On-Line Analytical Processing)

OLTP		OLAP	
Dati operazionali recenti		Dati storici	
Quantità di dati "ridotte"		Grandi quantità di dati	
Obiettivo: svolgimento delle operazioni ordinarie per un determinato contesto		Obiettivo: supporto alle decisioni strategiche aziendali	
Necessità di elevata velocità di esecuzione		Operazioni tipicamente eseguite offline	
Operazioni di tipo lettura/scrittura		Prevalentemente operazioni di lettura	
DBMS relazionali		Datawarehouse	

*Oggetto di  
questo corso*

- Applicazioni legate alla gestione di **big data** o sistemi **NOSQL**
  - ▮ Necessità di archiviare e gestire **grandi quantità di dati eterogenei** (testuali, immagini, video).
  - ▮ Principali caratteristiche:
    - **Scalabilità**: dati memorizzati in sistemi distribuiti, aggiunta di ulteriori nodi all'aumentare del volume di dati;
    - **Disponibilità**, replicazione: possibilità di accedere ai dati in modo continuativo, anche a fronte di guasti di singoli nodi;
    - **Sharding** di file: distribuzione del carico di lavoro su più nodi;
    - Accesso ai dati ad **alte prestazioni**: utilizzo di tecniche di hashing o partizionamento per migliorare l'efficienza;
  - ▮ Rispetto ai sistemi relazionali:
    - Memorizzazione di dati **semi-strutturati** che si autodescrivono (JSON o XML);
    - Linguaggi di **interrogazione meno potenti**.
    - **Versioning** per la storicizzazione dei dati.



# Gli attori principali

- **Data Base Administrator (DBA)**: installa, configura e gestisce il DMBS:
  - ▮ crea gli oggetti logici necessari (es. tabelle, viste, indici, ecc.) per le applicazioni;
  - ▮ crea gli utenti e concede loro i dovuti privilegi;
  - ▮ garantisce la sicurezza e l'integrità dei DB;
  - ▮ effettua controllo e monitoraggio degli accessi ai DB;
  - ▮ monitora e ottimizza le performance dei DB delle applicazioni che li utilizzano;
  - ▮ pianifica strategie di backup e recovery.
- **Data Base Designer**: cura la progettazione di un modello dettagliato del DB da implementare; il modello esplicita tutte le scelte progettuali a livello concettuale, logico e fisico e può essere usato per l'implementazione del database.
- **Software Engineer**: analisti di sistema e programmatori di applicazioni; i primi determinano le esigenze degli utenti finali e dettano specifiche per le transazioni che saranno realizzate a cura dei programmatori.
- **End User**: sono persone che interagiscono, a vari livelli, con una o più basi di dati per lo svolgimento delle proprie attività lavorative o per esigenze di altra natura.

# End users

- Gli utenti finali (end users), che possono beneficiare dell'uso di un DB, in sintesi sono inquadrabili in due classi.

## ▮ Naïve End User

- accede al DB tramite query preconfezionate all'interno di un'applicazione (es. prenotazione di un viaggio, prelievo di denaro con bancomat, inserimento di movimenti contabili, ...);
- non ha di norma nessuna conoscenza né del DBMS né della struttura della base dati.

## ▮ Sophisticated End User

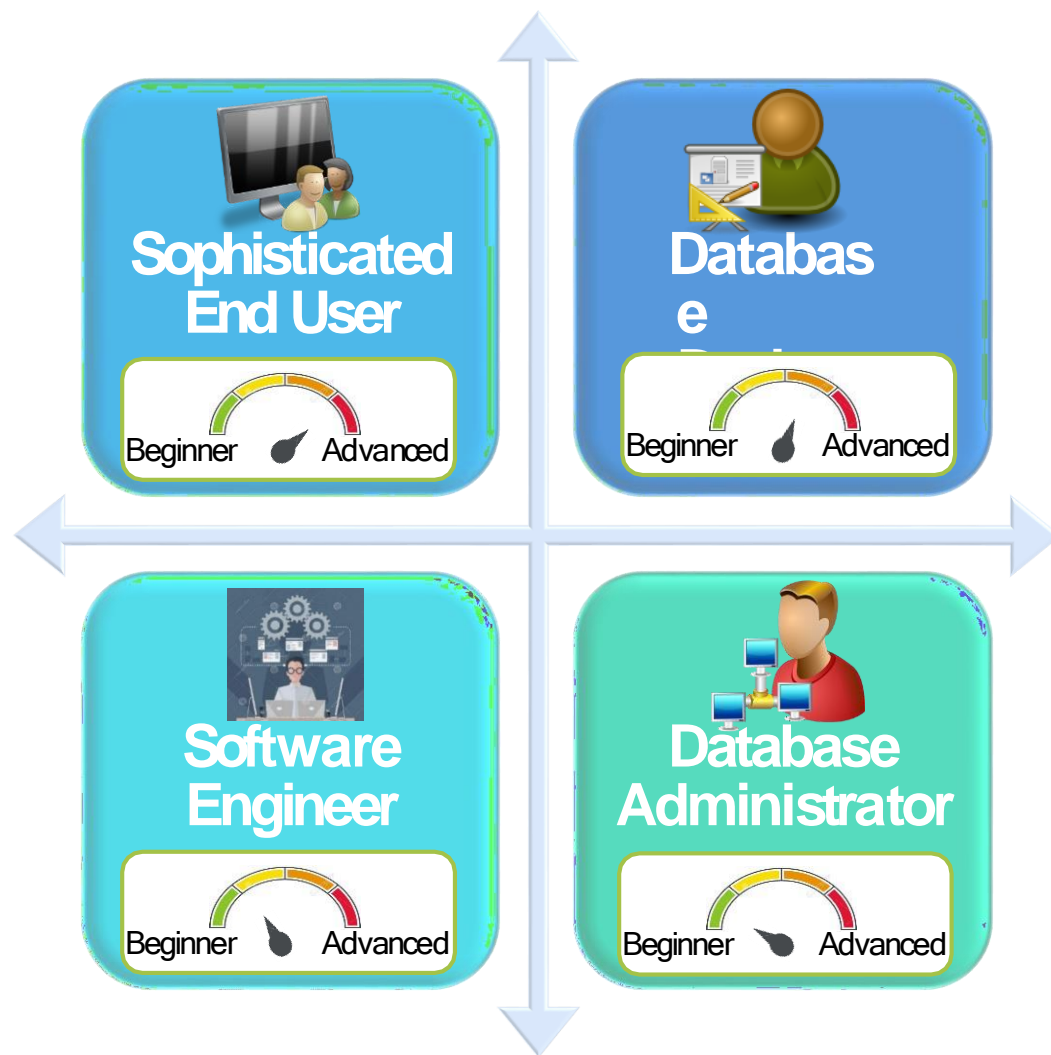
- ha un certo grado di conoscenza della struttura del DB e delle potenzialità del DBMS;
- è in grado di interagire direttamente con la base dati, attraverso l'uso di un linguaggio d'interrogazione, o indirettamente attraverso l'uso d'interfacce e/o di strumenti avanzati di reportistica e di analisi dei dati.

# Obiettivi del corso

- Illustrare gli aspetti fondamentali delle tecnologie delle **basi di dati relazionali**.
- Fornire linee guida, metodi e strumenti per:
  - ▮ **progettare** e **realizzare** basi di dati **relazionali**
  - ▮ **interagire** con basi di dati relazionali
  - ▮ progettare e realizzare **applicazioni database** (e più in generale moduli di un sistema informativo).
- Inquadrare il **trend** di sviluppo del settore delle basi di dati e delle relative tecnologie



# Competenze acquisite al termine del corso



# Contenuti del corso (1)



## Progettazione di database

- Introduzione a sistemi informativi, basi di dati e DBMS
- Introduzione alla progettazione di DB
- **Progettazione concettuale**
  - ▮ Analisi dei requisiti
  - ▮ Modello concettuale **Entity-Relationship**
- Modelli logici dei dati, il **modello relazionale**
  - ▮ Proprietà, schemi e istanze, vincoli
  - ▮ Forme normali e normalizzazione di schemi
- **Progettazione logica**
  - ▮ Ristrutturazione dello schema concettuale
  - ▮ Traduzione di entità e associazioni

# Contenuti del corso (2)



## Lavorare coi DBMS

- **Algebra relazionale**
- **Il linguaggio SQL**
  - ▮ Data Definition Language (creazione tabelle e vincoli), Data Manipulation Language (inserimento, aggiornamento, interrogazione e cancellazione di dati), Data Control Language
- **Funzionalità e architetture dei DBMS**
- **La gestione delle transazioni**
  - ▮ Proprietà ACID
  - ▮ Gestione della concorrenza, protocolli
- **Organizzazioni dei dati** e relativi metodi di gestione
  - ▮ Organizzazioni sequenziale e ad accesso diretto
  - ▮ Indici primari e secondari
  - ▮ B-tree, B\*- tree e B+-tree
- **Funzionalità avanzate DBMS (e SQL)**

# Materiale didattico

- Dispense disponibili sul sito web del corso (sufficienti per la preparazione dell'esame)
- Libro di esercizi:
  - ▮ D. Maio, S. Rizzi, A. Franco. Esercizi di Progettazione di basi di dati, Esculapio.
- Risorse aggiuntive
  - ▮ P. Atzeni, S. Ceri, P. Fraternali, S. Paraboschi, R. Torlone. Basi di dati, McGraw-Hill Italia, IV edizione.
  - ▮ R. A. Elmasri, S. B. Navathe. SISTEMI DI BASI DI DATI, Pearson, VII edizione.



# Domande?

---

