

Basi di dati

UniVR - Dipartimento di Informatica

Fabio Irimie

1° Semestre 2025/2026

Indice

1	Introduzione	2
2	Sistema informativo	2
2.1	Base di dati	2
2.1.1	Modello dei dati	3
3	Progettazione di una base di dati	4
3.1	Progettazione dei dati	4
3.2	Requisiti	5
3.2.1	Progettazione concettuale	5
3.2.2	Progettazione logica	5
3.2.3	Progettazione fisica	5
4	Modello Entità-Relazione	5
4.1	Entità	6
4.1.1	Istanza	6
4.2	Relazione	6
4.2.1	Istanza	7
4.2.2	Relazione ricorsiva	7
4.3	Attributo	8

1 Introduzione

Le basi di dati sono raccolte di dati strutturati, organizzati in modo tale da permettere un facile accesso. Questi dati sono persistenti, ovvero rimangono memorizzati anche dopo la chiusura del programma che li ha creati.

2 Sistema informativo

Un sistema informativo è l'insieme delle attività umane e dei dispositivi di memorizzazione ed elaborazione che organizza e gestisce l'informazione di interesse per un'organizzazione di dimensioni qualsiasi. Non contiene necessariamente dati memorizzati in un computer.

Un sistema informativo è composto da:

- **Dato:** è l'elemento di conoscenza di base costituito da simboli che devono essere elaborati
- **Informazione:** è l'interpretazione dei dati che permette di ottenere una conoscenza più o meno esatta di fatti e situazioni

2.1 Base di dati

Definizione 2.1. Una **base di dati** è una **collezione di dati persistenti** utilizzati per rappresentare **con tecnologia informatica** le informazioni di interesse per un **sistema informativo**

La soluzione convenzionale per la gestione dei dati è l'uso di file, ma questa presenta alcuni problemi:

- Scarsa efficienza nell'accesso ai dati (accesso sequenziale)
- Ridondanza nei dati
- Inconsistenza nei dati (aggiornamenti parziali)
- Progettazione dei dati replicata per ogni applicazione

Per risolvere questi problemi si è creato un livello di astrazione maggiore tra le applicazioni e il filesystem, ovvero il **Data Base Management System (DBMS)**.

Definizione 2.2. Un **DBMS** è un sistema che gestisce su **memoria secondaria** collezioni di dati (chiamate "basi di dati"). Le caratteristiche principali sono:

- Grandi
- Condivise, cioè accessibili da più utenti
- Persistenti

Un DBMS assicura:

- Affidabilità, cioè nessuna perdita di dati
- Privacy
- Accesso efficiente

2.1.1 Modello dei dati

Un **modello dei dati** è un insieme di strutture che permettono di descrivere una base di dati. Per accedere a questi dati si usano delle **interrogazioni**, cioè delle richieste, in un linguaggio dichiarativo specifico, che permettono di ottenere i dati desiderati.

Ci sono diversi linguaggi per interagire con un DBMS:

- Linguaggio per la definizione dei dati (DDL), consente di definire la struttura della base di dati
- Linguaggio per l'interrogazione e aggiornamento dei dati (DML), consente di interrogare e aggiornare i dati
 - Linguaggio di interrogazione: estrae informazioni da una base di dati, ad esempio SQL, algebre relazionale, calcolo relazionale
 - Linguaggio di manipolazione: popola la base di dati, modifica il suo contenuto con aggiunge, cancellazioni e variazioni sui dati, ad esempio SQL

Il modello di dati è un insieme di **costrutti** forniti dal DBMS per descrivere la struttura e le proprietà dell'informazione contenute in una base di dati.

Ci sono diversi tipi di modelli di dati:

- **Modelli di dati del passato:**
 - Modello reticolare
 - modello gerarchico
- **Modelli di dati attuali:**
 - Modello relazionale
 - Modello ad oggetti
 - Modello a oggetti-relazionale
 - Modello basato su documenti (JSON)
 - Modelli NoSQL

I modelli vengono utilizzati per creare:

- **Schema di una base di dati:** è la descrizione della struttura e delle proprietà di una specifica base di dati fatta utilizzando i costrutti del modello dei dati (lo schema di una base di dati è invariante nel tempo)
- **Istanza di una base di dati:** è costituita dai **valori effettivi** che in un certo istante popolano le strutture dati (l'istanza di una base di dati varia nel tempo)

Lo schema di una base di dati è diviso in tre livelli:

- **Schema esterno:** è la visione dell'utente della base di dati, cioè la parte di base di dati che interessa a un particolare utente o gruppo di utenti
- **Schema logico:** è la visione globale della base di dati, cioè la struttura logica della base di dati che descrive tutti i dati e le relazioni tra essi
- **Schema interno:** è la rappresentazione fisica della base di dati, cioè il modo in cui i dati sono effettivamente memorizzati nella memoria secondaria

Le proprietà dello schema sono:

- **Indipendenza fisica:** lo schema logico della base di dati è completamente indipendente dallo schema interno
- **Indipendenza logica:** gli schemi esterni della base di dati sono indipendenti dallo schema logico

3 Progettazione di una base di dati

Il ciclo di vita di un processo di automazione di un sistema informativo è diviso in diverse fasi:

- **Studio di fattibilità:** si valuta se l'automazione del sistema informativo è possibile e conveniente
- **Raccolta e analisi dei requisiti:** si individuano proprietà e funzionalità del sistema (dati e applicazioni) producendo una descrizione completa ma informale
- **Progettazione:** si produce una descrizione formale del sistema informativo

La progettazione si divide in due parti principali che vanno di pari passo:

- **Progettazione dei dati:** si produce una descrizione formale dei dati (schema). Una volta progettati i dati vengono implementati in un DBMS
- **Progettazione delle applicazioni:** si produce una descrizione formale delle applicazioni (specifica)

Una volta implementati i dati e le applicazioni si passa alla fase di **validazione e collaudo**

3.1 Progettazione dei dati

Una metodologia di progettazione dei dati è costituita da:

- **Decomposizione:** dividere in passi le attività di progetto
- **Strategie:** individuare un insieme di strategie e criteri di scelta da seguire
- **Modelli di riferimento:** utilizzare modelli di dati e tecniche di progettazione consolidate

Una buona metodologia deve essere:

- Generale
- Facile da usare
- Deve produrre un risultato di qualità

3.2 Requisiti

3.2.1 Progettazione concettuale

La progettazione concettuale è la prima fase della progettazione dei dati. Lo scopo è quello di produrre una descrizione formale dei dati (schema concettuale). Lo schema deve essere **indipendente dall'implementazione**.

Non è solo un progetto intermedio, ma costituisce anche una porzione del risultato finale perchè rappresenta una descrizione di **alto livello** del contenuto della base di dati, comprensibile anche per utenti poco esperti.

3.2.2 Progettazione logica

La progettazione logica è la seconda fase della progettazione dei dati. Lo scopo è quello di tradurre lo schema concettuale in uno schema logico in modo da poterlo utilizzare su un sistema specifico. Lo schema logico infatti è dipendente dalle tecnologie utilizzate. Bisogna tenere anche in considerazione le operazioni più frequenti che le applicazioni effettueranno sulla base di dati.

3.2.3 Progettazione fisica

La progettazione fisica è la terza fase della progettazione dei dati. L'obiettivo è quello di ottimizzare l'accesso ai dati completando lo schema logico con i parametri relativi alla memorizzazione fisica dei dati e con gli opportuni metodi d'accesso (**indici**).

4 Modello Entità-Relazione

È un modello, formale e non ambiguo, utilizzato per la progettazione concettuale di una base di dati. Fornisce strumenti formali (costrutti), con sintassi grafica, per specificare la struttura e le proprietà dei dati da rappresentare indipendentemente dalla tecnologia.

Ogni costrutto viene definito specificando:

- Il suo significato (o semantica)
- La sua sintassi grafica
- La rappresentazione delle sue istanze (o occorrenze)

Progettare indipendentemente dalle tecnologie significa:

- **Non considerare** eventuali ottimizzazioni

- **Considerare** tutti i requisiti senza semplificazioni o convenzioni
- **Considerare** sempre i processi di generazione e modifica dei dati per verificare che ogni situazione sia rappresentabile da un'istanza "pulita" della base di dati

4.1 Entità

Un'entità E rappresenta una **classe di oggetti** che hanno le seguenti caratteristiche:

- **Proprietà comuni**
- **Esistenza autonoma** rispetto ad altre classi di oggetti
- **Identificazione univoca**, cioè esiste una chiara corrispondenza tra gli oggetti istanze di entità e concetti istanziati nel sistema informativo

Un'entità si rappresenta con un rettangolo che contiene il nome dell'entità:



Figura 1: Rappresentazione grafica di un'entità

4.1.1 Istanza

Un'istanza dell'entità E è un **oggetto** appartenente alla classe rappresentata da E. Si indica con $I(E)$ l'insieme delle istanze di E che esistono nella base di dati in un certo istante e alla creazione della base di dati è vuota: $I(E) = \emptyset$.

Esempio 4.1. Rappresentiamo con il costrutto entità il concetto di **persona**. Bisogna gestire nella base di dati le informazioni che descrivono un gruppo di persone.

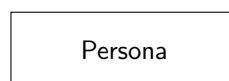


Figura 2: Rappresentazione grafica dell'entità Persona

L'insieme delle istanze dell'entità Persona è il seguente:

$$I(\text{Persona}) = \{p_1, p_2, p_3, \dots\}$$

4.2 Relazione

Una relazione R rappresenta un **legame logico** tra **due o più** entità. Può esserci anche una relazione all'entità stessa (relazione ricorsiva).

Una relazione si rappresenta nello schema con un rombo a cui si collegano attraverso delle linee le entità coinvolte nella relazione. Il nome della relazione viene scritto a fianco al rombo:



Figura 3: Rappresentazione grafica di una relazione tra due entità

4.2.1 Istanza

Data una relazione R tra n entità E_1, \dots, E_n un'istanza della relazione R è una **ennupla di istanze di entità**:

$$(e_1, \dots, e_n) \text{ dove } e_i \in I(E_i) \text{ per } 1 \leq i \leq n$$

La popolazione di R rappresenta l'insieme delle coppie di istanze delle entità E e F che sono in relazione in un certo istante:

$$I(R) = \{(e_i, f_j) \mid e_i \in I(E), f_j \in I(F)\}$$

Esempio 4.2. Supponiamo che nello schema ci siano le entità **Persona** e **Comune**, bisogna gestire la **Residenza** delle persone nei comuni italiani.

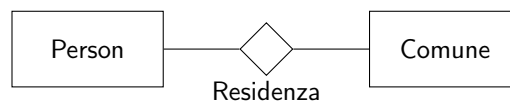


Figura 4: Rappresentazione grafica della relazione Residenza

Ciò implica che per esistere un'istanza di residenza devono esistere un'istanza di persona e un'istanza di comune.

Data una relazione R tra n entità $\{E_1, E_2, \dots, E_n\}$ vale **sempre** la seguente proprietà sull'insieme delle istanze $I(R)$:

$$I(R) \subseteq I(E_1) \times I(E_2) \times \dots \times I(E_n)$$

La conseguenza di questa proprietà è che non è possibile rappresentare la stessa ennupla più volte.

4.2.2 Relazione ricorsiva

È una relazione binaria sulla stessa entità:

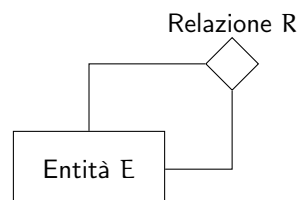


Figura 5: Rappresentazione grafica di una relazione ricorsiva

4.3 Attributo

Rappresenta una proprietà elementare di un'entità o di una relazione. Ogni attributo di un'entità o di una relazione associa ad ogni istanza **un solo** valore appartenente ad un dominio di valori ammissibili. Può essere visto come una funzione che ha come dominio le istanze dell'entità (o relazione) e come codominio l'insieme dei valori ammissibili:

$$f_A : I(E) \mapsto D$$

dove a è un attributo dell'entità E , mentre $I(E)$ l'insieme delle istanze di E e D è l'insieme dei valori ammissibili.

La sintassi grafica di un attributo è un cerchio **vuoto** collegato con una linea all'entità con accanto il nome dell'attributo:

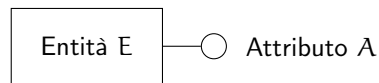


Figura 6: Rappresentazione grafica di un attributo di un'entità

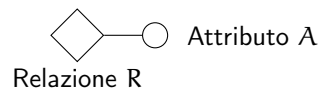


Figura 7: Rappresentazione grafica di un attributo di una relazione

Esempio 4.3. Rappresentiamo il concetto di persona tramite un'entità, bisogna gestire nella base di dati il nome e il cognome di un gruppo di persone.

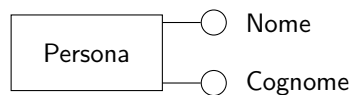


Figura 8: Rappresentazione grafica dell'entità Persona con gli attributi Nome e Cognome