

# 1 Sistemi informativi e sistemi informatici

L'idea di una scuola pubblica viene generalmente identificata con la sua funzione primaria, ovvero quella di favorire lo sviluppo delle competenze dei suoi studenti. Perché questo possa avvenire, una organizzazione scolastica deve gestire numerosi dati come quelli necessari alle attività didattiche:

- studenti iscritti
- docenti assegnati
- corsi attivati
- orario annuale delle lezioni
- spazi disponibili
- gestione dei risultati intermedi e finali degli studenti

Affinché tali attività possano essere condotte correttamente, però, una scuola deve gestire anche altri dati a cui non è immediato pensare, per esempio quelle relativi:

- al personale di servizio
- al personale amministrativo che opera nei vari uffici
- al magazzino dei beni di consumo

Un'adeguata gestione del processo generale dei servizi impone che l'organizzazione scolastica sia strutturata in modo tale che dirigenti, organi consultivi e decisionali, uffici, docenti, studenti, genitori, etc. **debbano scambiarsi continuamente informazioni funzionali ai processi operativi e di governo della scuola stessa.**

Situazioni come questa e molte altre più o meno complesse **prevedono inevitabilmente la gestione di notevoli quantità di dati.**

## 1.1 Dati e informazione

Per **dato** intendiamo la misura di un fenomeno che siamo interessati ad osservare, ad esempio possiamo utilizzare un metro per misurare l'altezza di una persona, oppure un termometro per misurare la temperatura corporea. Nel primo caso otteniamo 160 cm, nell'altro 37 °C, così abbiamo ottenuto 2 dati.

Per **informazione** è ciò che si ottiene dall'elaborazione di un insieme di dati e che **accresce lo stato di conoscenza relativo a un fenomeno**, ad esempio nella misurazione della temperatura corporea precedente elaborando il dato possiamo dedurre che l'utente abbia una leggera febbre.

Quindi si può affermare che **maggiore** è la **quantità di dati** di cui disponiamo rispetto ad un fenomeno, **migliore** sarà il **livello di conoscenza rispetto a quel fenomeno, ma non sempre tutti i dati risultano utili alla sintesi dell'informazione**.

## 1.2 Sistemi informativi e informatici

Raccogliere, archiviare ed elaborare dati per gestire e comunicare informazioni è una necessità che da sempre ha caratterizzato le attività umane.

Un **sistema informativo** è un insieme strutturato di procedure e di risorse umane e materiali finalizzate alla raccolta, all'archiviazione, all'elaborazione e alla comunicazione di dati, allo scopo di ottenere le informazioni necessarie a un'organizzazione per gestire le attività operative sia quelle di governo. *Il concetto di sistema informativo è indipendente dagli strumenti utilizzati.*

Un sistema informativo viene utilizzato per il conseguimento di specifici obiettivi, sostanzialmente due:

- **Scopo operativo:** tutte le organizzazioni hanno la necessità di gestire dati funzionali relativi alle loro attività operative
- **Scopo decisionale:** per poter prendere decisioni relative alla attività di programmazione, controllo e valutazione

La maggiore diffusione di tecnologie informatiche hanno reso vantaggioso la gestione delle informazioni, per cui si definisce **sistema informatico** il sottoinsieme di un sistema informativo dedicato al trattamento *automatico* di informazioni derivanti dalla gestione di dati archiviati in formato digitale.

La presenza di tecnologie informatiche per la gestione dell'informazione **non implica una completa automatizzazione del sistema informativo**, infatti esistono ancora aspetti di un sistema informativo per cui non conviene automatizzare.

## 1.3 Ciclo di vita di un sistema informatico

La progettazione di un sistema informatico generalmente passa attraverso un processo piuttosto complesso che deve essere condotto da personale professionalmente qualificato. Le conseguenze di un progetto gestito male possono portare a inefficienza, perdita di dati, alti costi di manutenzione e blocco delle attività produttive.

Tale processo di progettazione si articola su un insieme di attività raggruppabili in *3 macrofasi* (4, inclusa la manutenzione/aggiornamento del sistema informatico):

- **raccolta richieste degli utenti**
  - Indagine preliminare relativa all'introduzione del sistema informatico nell'organizzazione (svolta dall'organizzazione stessa) per **individuare i settori potenzialmente interessati all'introduzione delle tecnologie informatiche** e valutarne, per ognuno di essi, l'effettiva convenienza; **valutare gli impatti dovuti all'introduzione di tali tecnologie; valutare il budget che l'organizzazione ha a disposizione** per l'introduzione di tecnologie informatiche
  - Analisi del sistema informativo esistente (svolta dell'**analista**)
  - Definizione dei requisiti del nuovo sistema, che, in base alle osservazioni e valutazioni svolte dall'analista il quale produce una documentazione

dettagliata descrivendo **dati utilizzati, vincoli di integrità, descrizione delle procedure da automatizzare, volume iniziale e previsione dei dati nel tempo, grado di privatezza dei dati**

- **progettazione concettuale**

- Questa fase consiste, grazie alla documentazione precedentemente prodotta, nella definizione di un modello astratto del sistema informatico destinato a definire le linee guida di riferimento e di lavoro per i progettisti della successiva fase di realizzazione.
- Nella realtà non esiste un accordo preciso su cosa debba essere un progetto concettuale, ma la maggior parte degli attuali approcci considera il progetto concettuale come un insieme di **documenti, schemi e diagrammi** che *descrivono in modo organico e coerente la struttura dei dati, i vincoli di ammissibilità dei dati, procedure di elaborazione dati, vincoli sui tempi di risposta, integrità e riservatezza delle informazioni e costi in termini di risorse hardware per realizzare il progetto.*

- **Realizzazione**

- Consiste nella realizzazione effettiva del sistema informatico nelle varie componenti (**infrastrutturali**, ovvero piattaforme hw/sw e di comunicazione, **applicative**, ovvero acquisizione di prodotti software già disponibili sul mercato)

Le 3 fasi producono, rispettivamente, come output:

- definizione dei requisiti a cui il sistema informatico dovrà essere conforme
- progetto concettuale
- progetto logico e fisico che rappresenta l'insieme delle componenti software che implementano il sistema informatico

Il processo di progettazione è ciclico perché con l'uso del sistema gli utenti e i committenti avanzano richieste **correttive** o **evolutive** del software.

La **centralità dei dati** ha sempre caratterizzato le applicazioni informatiche per l'automazione dei sistemi informativi, ma solo a partire dalla fine degli anni '60 sono stati sviluppati ambienti software specificatamente dedicati alla loro gestione. Questo tipo di approccio viene generalmente indicato come approccio su **file system**, in considerazione del fatto che l'archiviazione dei dati avviene mediante file memorizzati nella memoria di massa.

Dagli anni '80, l'approccio basato su file system sarà pian piano sostituito dai **DBMS**.

## 1.4 Aspetto estensionale e intenzionale dei dati

Per **aspetto estensionale** dei dati si intende l'**istanza del dato**, quindi il suo valore specifico.

Un insieme di valori (dati) però non ha molto significato. Per comprenderli abbiamo bisogno di una *chiave interpretativa* per capirne la *semantica*. Quindi per **aspetto intenzionale** dei dati si intende lo **schema** o **interpretazione dei dati**.

Un insieme di dati aventi la stessa interpretazione è detta **categoria**.

## 1.5 File

Un **file** è un archivio contenente un insieme di dati identificato da un **nome**, **memorizzato permanentemente**.

Un file ha *vita indipendente dal programma* che ne fa uso, e questo fa sì che qualunque programma potrebbe usarlo per usarne i dati all'interno.

### 1.5.1 Operazioni su file

Su un file possiamo fare varie operazioni:

- **Creazione**, ovvero si crea su una memoria di massa

- **Lettura**, possiamo aprirlo per guardare il contenuto
- **Modifica**, possiamo modificarne il contenuto
- **Eliminazione**, possiamo eliminare il contenuto oppure l'intero file

## 1.5.2 Organizzazione file

Per **organizzazione di un file** si intende sia il modo in cui esso è memorizzato sia il modo in cui viene elaborato.

Quindi è distinta su due livelli:

- **Fisico**
- **Logico**, relativo alle modalità di accesso al file

## 1.5.3 Modalità di accesso al file

Fondamentalmente esistono tre modalità di accesso al file, ovvero:

- **Sequenziale**, dove ogni **record**<sup>1</sup> è salvato uno dopo l'altro, per cui, nel caso peggiore ovvero quando il record non esiste, dovremmo aspettare di aver scorso tutto il file prima di scoprirlo, per cui è una **modalità di ricerca lenta**.
- **Accesso diretto** o random, dove c'è un *numero*, incrementato ad ogni valore inserito, che rappresenta la posizione del record nel file.
- **Indicizzata**, dove nel record viene identificato un **campo chiave** che *identifica univocamente quel record*. In questo tipo di organizzazione c'è un secondo file (**file indice**) contenente una tabella di chiavi: questo file viene ordinato dal software che, quando dovrà cercare per un record consulerà prima questo file indice (*ricerca veloce*), prenderà la chiave che gli serve e accederà direttamente al record che gli interessa nel file primario dove ci sono i record.

---

<sup>1</sup> **Record**: In un file i dati sono raggruppati in record, e un record rappresenta l'*aspetto estenzionale*, mentre la tabella di un DB rappresenta lo *schema intenzionale*. Ogni singolo elemento di questo schema è chiamato **field**.

## 1.6 Basi di dati e sistemi di gestione delle basi di dati

Per **DBMS** si intende un *sistema software in grado di gestire grandi collezioni di dati integrati, condivisi e persistenti assicurando loro affidabilità e privacy*.

Per **dati integrati** si intende un insieme di dati strutturati e permanenti memorizzati senza ridondanze e organizzati in modo da poter essere usati da più programmi.

Per **condiviso** si intende appunto più programmi possono usufruire di questi dati.

Per **persistenti** si intende che i dati vengano salvati su supporti di memoria secondaria.

Per **affidabilità** si intende che questi dati non vengano persi, ad esempio facendo più backup su dischi di memoria diversi.

Per **basi di dati** invece si intende una *collezione di dati gestita da un DBMS*.

La differenza tra la gestione dei dati tramite DBMS e file system è il fatto che il significato dei dati e delle loro relazioni che “legano” i vari file tra di loro sono immersi nel codice del programma, e questo rende difficile l’interpretazione della struttura informativa. Con un DBMS invece la struttura è chiara e non dobbiamo ricercare le relazioni sul programma che ne fa uso, e questo alleggerisce il software.

### 1.6.1 Utenti del DBMS

E’ possibile classificare l’utenza di un sistema di gestione delle basi di dati nel seguente modo:

- **Programmatori di app**, interagiscono con le basi di dati per sviluppare l’app
- **Utenti finali**, non interagiscono con le basi di dati, solo utilizzando l’app

- **Utenti avanzati**, conoscono la struttura del DB e possono interrogarlo
- **Amministratori (DBA, DataBase Administrator)**, si occupano della manutenzione del DB

## 1.6.2 Linguaggi

L'utente interagisce con il sistema di gestione delle basi di dati tramite specifici linguaggi:

- **DDL (Data Definition Language)**: linguaggi per la definizione della **struttura dei dati**
- **DML (Data Manipulation Language)**: linguaggi per la **gestione e l'utilizzazione dei dati** contenuti in una base di dati

## 1.6.3 Realizzazione

L'approccio fondato su DBMS prevede che la fase di realizzazione possa essere scomposta in due sottofasce distinte:

- **Progettazione logica**: conversione progetto concettuale in un progetto logico
- **Progettazione fisica**: descrizione organizzazione fisica dei file

## 1.7 Architettura logica di un DBMS

L'architettura logica di un DBMS è strutturata su tre livelli:

- **Livello logico utente**, in cui ogni utente ha un proprio spazio di lavoro. A questo livello sono definite dall'amministratore le *viste utente*, ovvero dei sottoinsieme del modello logico globale dell'intera base dati. In poche parole possono visualizzare solo alcuni DB
- **Livello logico globale**, relativo alla definizione della struttura logica generale della basi dati
- **Livello fisico di memorizzazione**, relativo alla memorizzazione dei dati



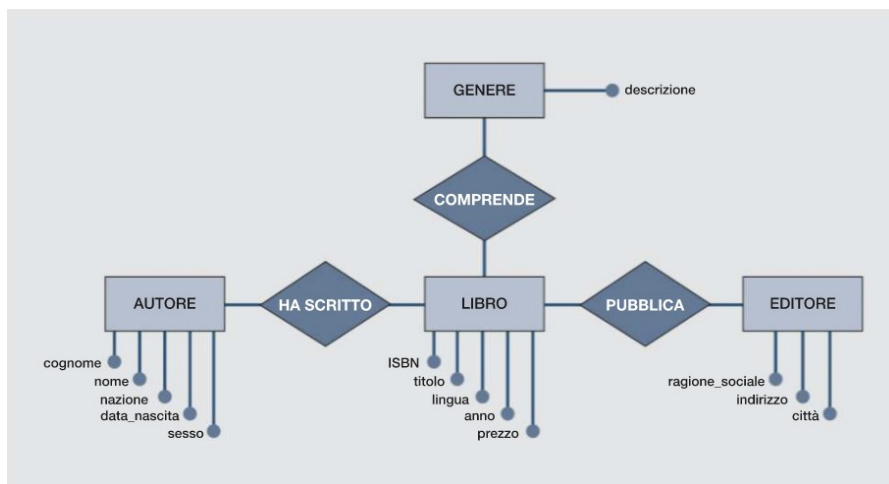
## 2 Le basi di dati relazionali

Si deve realizzare un sistema informatico per gestire i dati di una piccola biblioteca. In questo scenario si possono individuare alcune **classi di dati fondamentali** insieme ai loro **attributi**:

- **Attributi libri:** ISBN, titolo, anno di pub., lingua, prezzo
- **Attributi autori:** cognome, nome, nazionalità, data nascita, sex
- **Attributi editori:** ragione sociale, indirizzo, città
- **Attributi generi:** categorie libri, descrizione categoria

Inoltre:

- Un autore può aver scritto più di un libro, anche se ogni libro è stato scritto da un solo autore
- Un editore ha pubblicato numerosi libri, mentre normalmente un libro è pubblicato da un determinato autore
- Ogni genere comprende un intero insieme di libri



Un modo per rappresentare questo scenario è quello di fornire una rappresentazione grafica mediante un **diagramma Entità/Associazione**<sup>2</sup> (diagramma E/R).

<sup>2</sup> **Diagramma E/R:** vengono introdotti alcuni termini, quali **PK** (Primary Key, elemento identificativo) e **FK** (Foreign Key, elemento associativo)

In particolare il formalismo grafico E/R viene utilizzato per la realizzazione di modelli logici di scenari reali che prescindono da come i dati saranno effettivamente memorizzati e organizzati nei computer di un sistema informatico. Questo modello verrà ereditato dal linguaggio grafico UML.

## 2.1 Diagramma E/R

Il diagramma E/R **descrive a livello concettuale le caratteristiche del database da implementare.**

### 2.1.1 Entità

Rappresentano **classi di oggetti** che hanno **proprietà comuni** ed **esistenza autonoma** nel contesto dello scenario analizzato (analogo alle classi dell'UML).

In un diagramma E/R ogni entità ha un nome (come le classi dell'UML) che la identifica univocamente ed è rappresentata graficamente con un rettangolo e l'elenco delle sue proprietà, nell'esempio di prima sono: *Autori*, *Libri*, *Editori*, *Genere*.

### 2.1.2 Associazioni

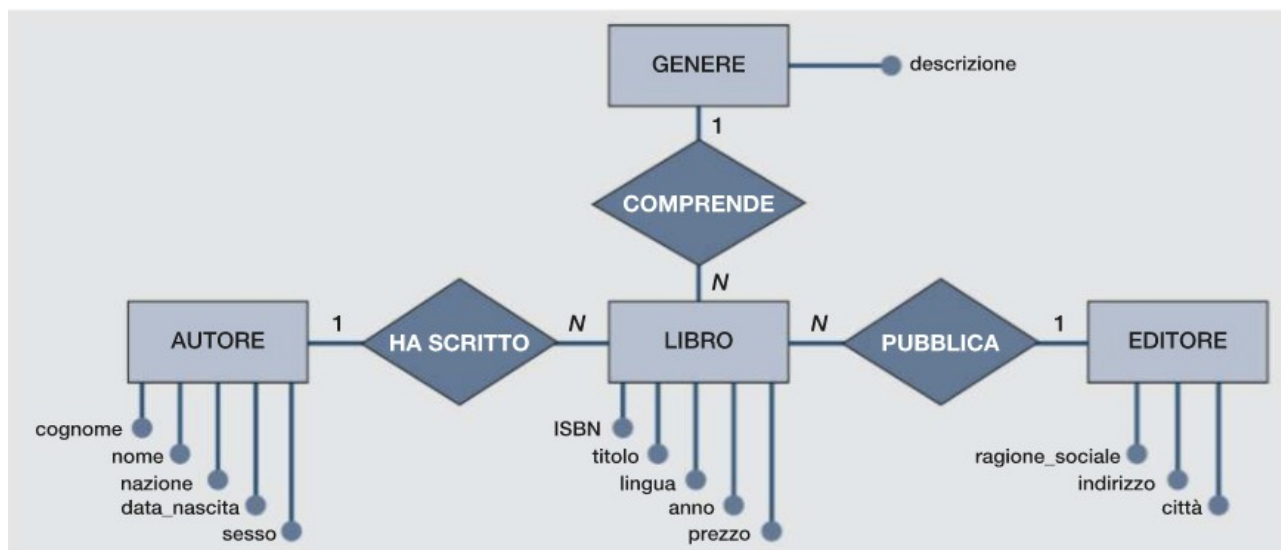
Le **associazioni**, o relazioni, costituiscono il mezzo tramite il quale sono modellate le corrispondenze tra le istanze di due o più entità.

Il **grado dell'associazione** indica il numero delle entità coinvolte. Nel caso particolare di associazioni di grado 2 o binarie, si distinguono **corrispondenze di cardinalità**:

- **Associazione 1:1** => Corrispondenze biunivoche, a ogni istanza della classe A ne corrisponde una e una sola della classe B e viceversa
- **Associazione 1:N** => legano **un solo elemento** appartenente alla prima entità con **più elementi** della seconda entità; viceversa, invece, legano più elementi seconda con un solo elemento della prima.

- **Associazione M:N** => A ogni istanza dell'entità A corrispondono M istanze dell'entità B. Viceversa, a ogni istanza di B corrispondono N istanze dell'entità A.
  - **N.B.:** Un'associazione dalla classe A alla classe B (o viceversa) è **totale** se è definita per ogni istanza della classe A o B, altrimenti è **parziale**.

Nel nostro caso queste sono le associazioni:



Simbolo	Descrizione
	Associazione 1:1 parziale
	Associazione 1:1 totale
 	Associazione 1:M parziale (l'uso del simbolo con la linea continua verrà chiarito a pag. A38)
	Associazione 1:M totale

Nei diagrammi utilizzati dai progettisti di db e dagli sviluppatori software, vengono usati questi simboli per rappresentare le associazioni:

La simbologia illustrata non permette di rappresentare un'associazione M:N in quanto essa non è direttamente implementabile mediante un db di tipo

relazionale. Questa relazione è sempre trasformabile in due associazioni di tipo 1:N.

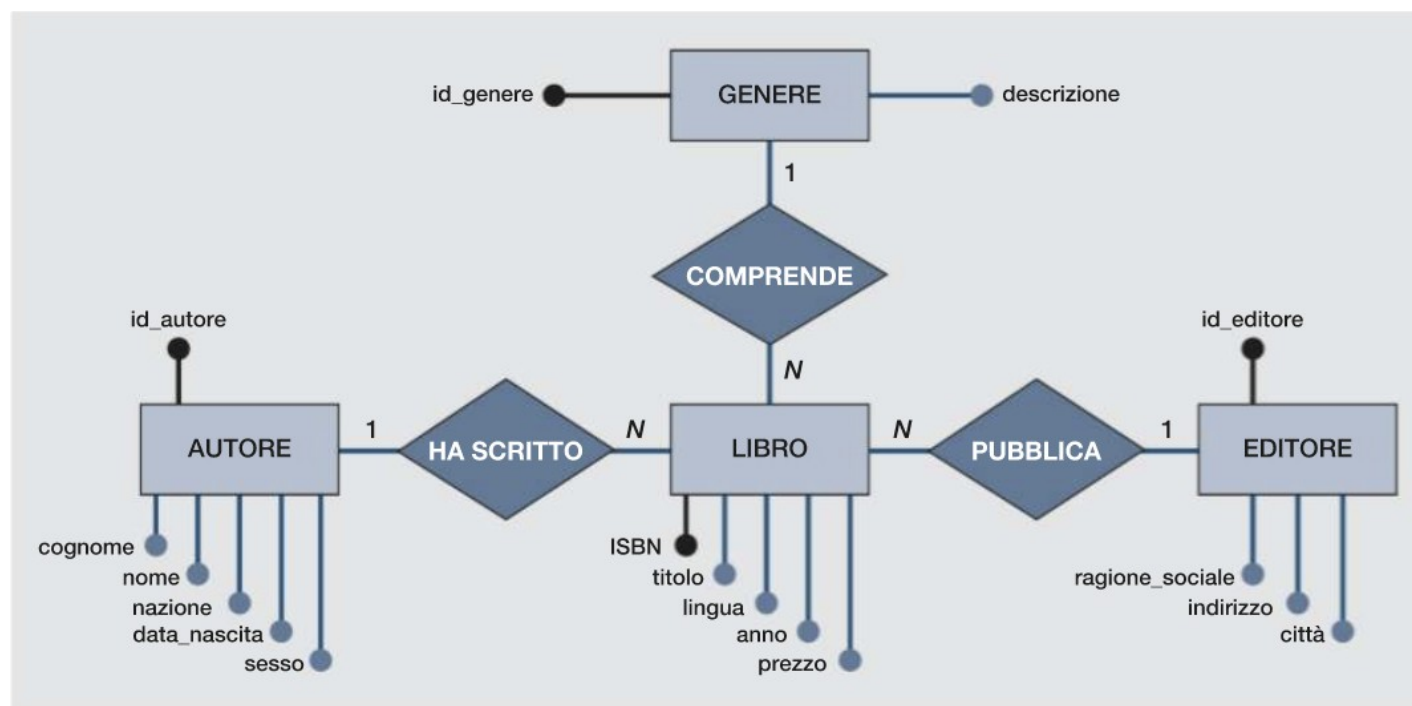
### 2.1.3 Attributi

Tutte le istanze di una stessa entità hanno gli stessi attributi.

L'individuazione degli attributi riflette il livello di dettaglio con il quale si intendono rappresentare le informazioni alle entità.

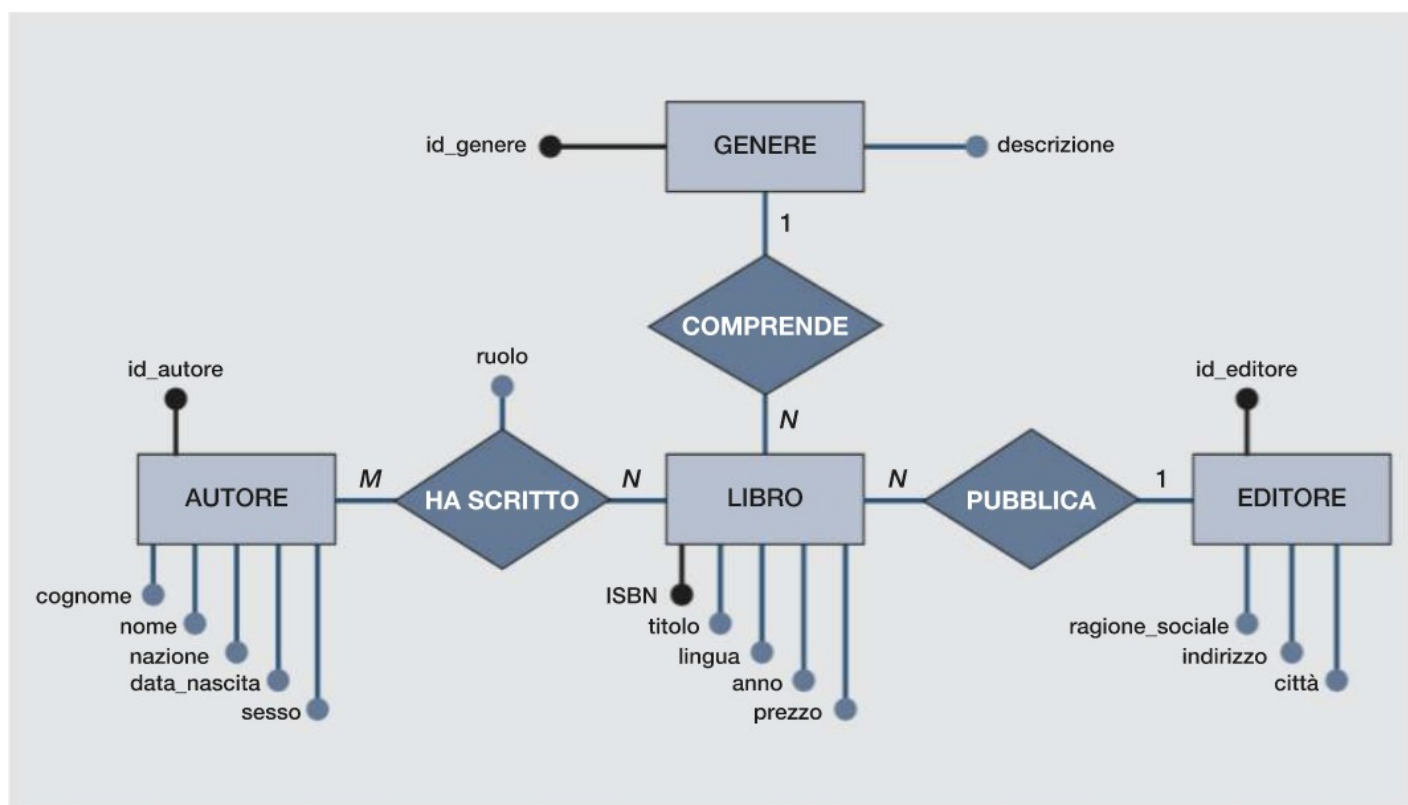
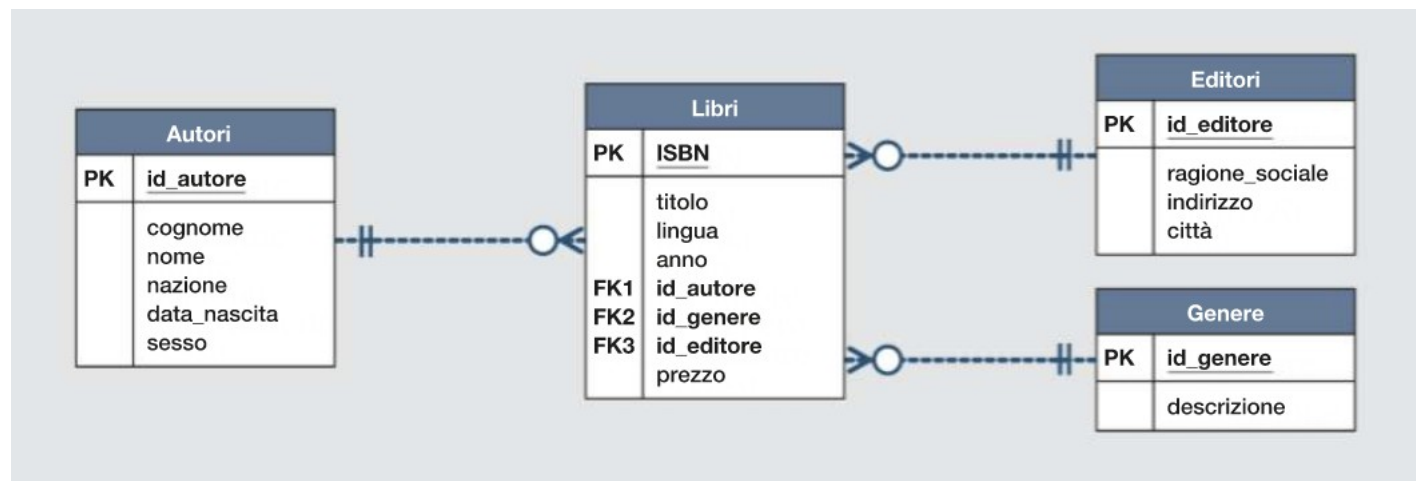
## 2.2 Chiavi primarie ed esterne

Per ciascuna entità si definisce **chiave primaria (PK)** un insieme minimale di **attributi**, cioè costituito dal numero minimo di attributi, **che identifica univocamente ciascuna istanza dell'entità**.



Per modellare le associazioni di cardinalità 1:N si ricorre alle cosiddette **chiavi esterne (FK)**: l'associazione viene realizzata aggiungendo l'attributo o gli attributi che formano la chiave primaria dell'**entità dominio all'entità codominio** (nel nostro esempio, l'entità dominio è *Autore* mentre l'entità codominio è *Libro*).

Generalizzando il diagramma E/R che rappresenta lo scenario della biblioteca è più corretto **considerare** l'associazione tra Libri e Autori di cardinalità M:N, in quanto un libro può essere stato scritto da più autori.

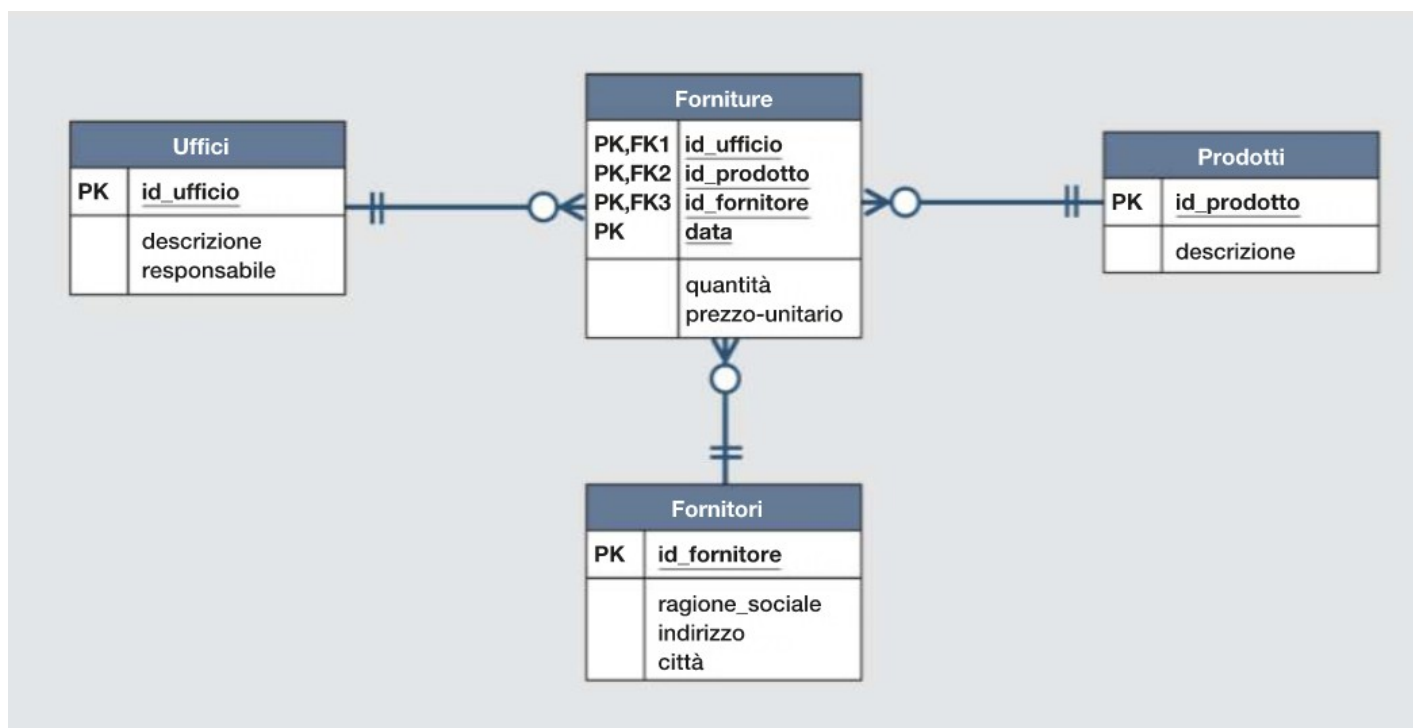


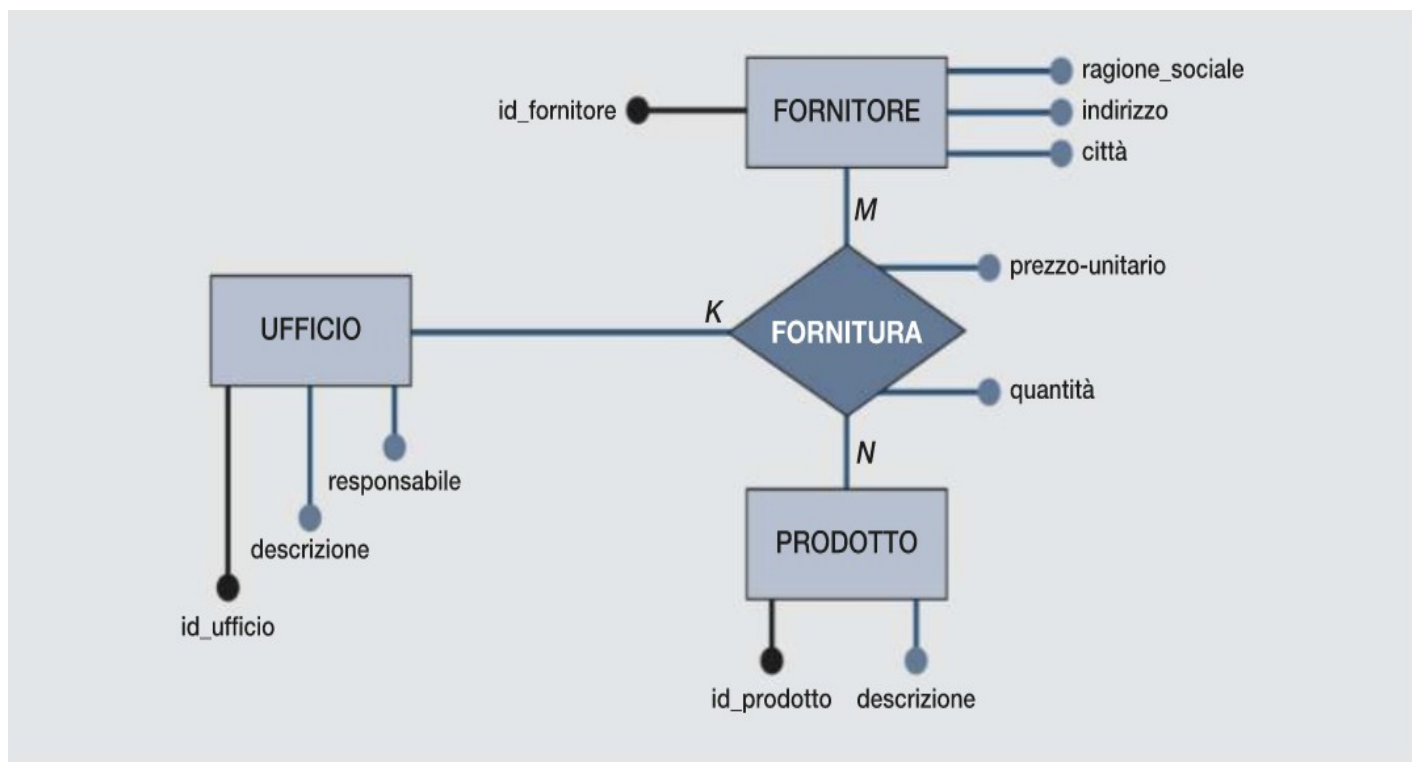
Aiuta all'implementazione di basi di dati scomporre la cardinalità M:N in 2 associazioni di tipo 1:N introducendo un'entità **intermedia** i cui attributi sono

chiavi esterne che si riferiscono alle chiavi primarie delle entità originalmente associate.

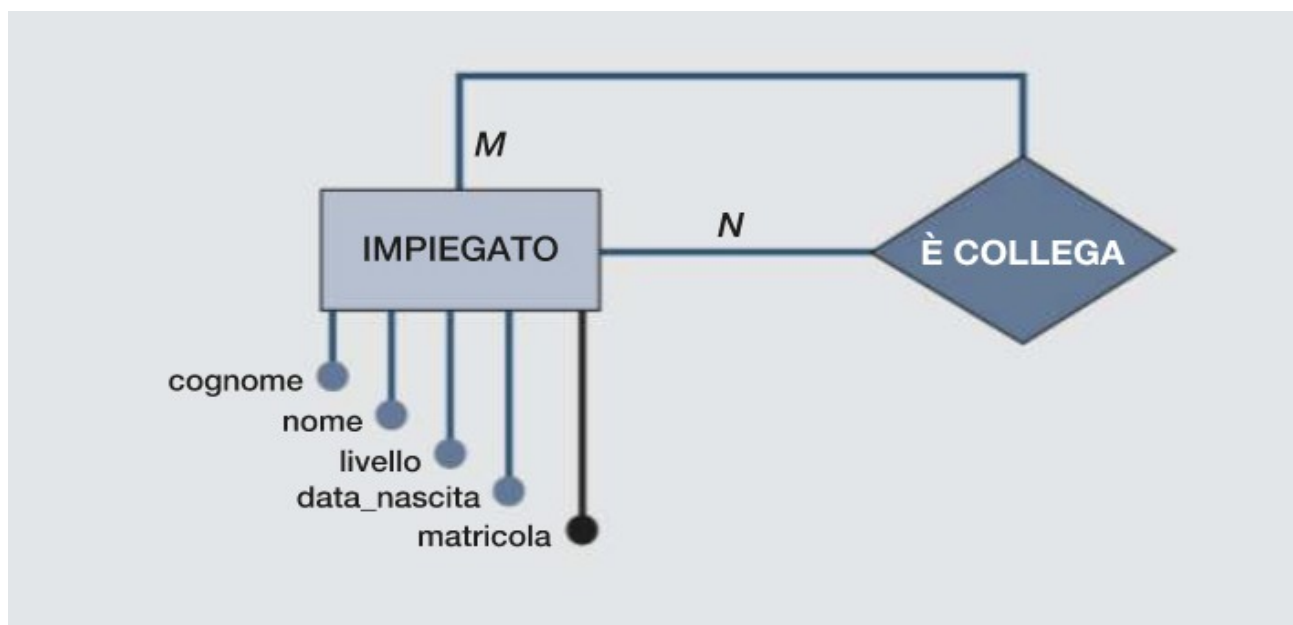
Nell'esempio vengono indicate 2 primary key (tabella *Ruolo*) che stanno a significare che nella tabella non ci possono essere 2 record che hanno lo stesso "id\_autore" e "ISBN". In realtà non sono due primary key, ma *entrambi formano la primary key*.

Anche se è buona norma ricondursi, quando possibile, a situazioni che comprendono esclusivamente associazioni binarie, esistono casi complessi in cui le associazioni sono multiple: un caso non troppo raro è quello relativo alle relazioni ternarie. In questo caso, le 3 entità (*Ufficio*, *Fornitore*, *Prodotto*) sono tutte associate con un'associazione di cardinalità M:N, quindi, per semplificare, viene introdotta una 4 entità (*Fornitura*) avente una primary key composta da 3 FK e 4 PK (che formano **1 sola PK**).

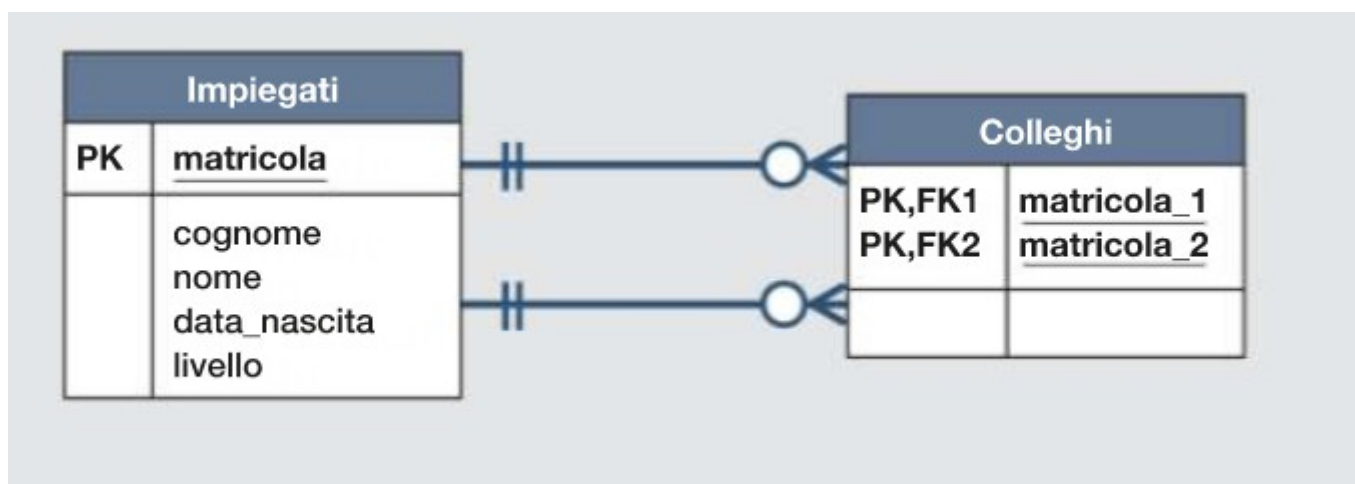




È possibile associare un'entità con se stessa (*associazione ricorsiva*). Il passaggio alla rappresentazione grafica della base di dati prevede l'inserimento di una nuova entità che implementi l'associazione di cardinalità M:N.







## 2.3 Il modello dei dati relazionali

Il modello dei dati relazionali fu proposto da **Edgar Codd** nel 1970: con esso la gestione dei dati viene ricondotta all'unico concetto di *relazione*, intesa in senso algebrico.

Il modello dei dati relazionale è un modello formale per la rappresentazione della struttura logica di una base di dati finalizzato al conseguimento di una *buona indipendenza dei dati dalle procedure di elaborazione* e alla semplificazione dei linguaggi impiegati per la loro definizione e manipolazione, basato su un approccio matematico rigoroso che ha come fondamenti teorici il concetto di relazione matematica.

Dati gli insiemi  $D_1, D_2, \dots, D_N$  una **relazione R** su questi N insiemi è costituita da un insieme di **ennuple**<sup>3</sup>  $(d_1, d_2, \dots, d_N)$  tali che l'elemento  $d_i$  appartiene all'insieme  $D_i$  per ogni indice  $(1, 2, \dots, N)$ . In altre parole R è un **sottoinsieme del prodotto cartesiano** tra  $D_1 * D_2 * \dots * D_N$ . Gli insiemi  $D_1, D_2, \dots, D_N$ , il numero N e il numero delle ennuple che compongono la relazione R costituiscono rispettivamente i **domini**, il **grado** e la **cardinalità** di R.

Una relazione di fatto è un insieme di ennuple e quindi:

- non è definito alcun ordinamento fra le sue ennuple

<sup>3</sup> **Ennupla:** Dati tot. insiemi, prendo un valore per ogni di questi insiemi, si definisce un *insieme di ennuple* la lista di questi valori



- le ennuple di una relazione sono tutte distinte una dall'altra perché in un insieme non possono esistere elementi uguali
- ciascuna ennupla al proprio interno è **ordinata** in quanto l' $i$ -esimo valore appartiene all' $i$ -esimo dominio