

Basi di Dati

Federico Matteoni

A.A. 2019/20

Indice

1	Introduzione	5
2	Base di Dati	7
2.1	Costruzione di una Base di Dati	7
2.1.1	Figure Coinvolte	7
2.1.2	Sistemi Informativi	7
2.1.3	Sistemi Informatici	8
2.1.4	Classificazione dei sistemi informatici	9
2.1.5	Requisiti per l'Analisi dei Dati	10
2.1.6	Big Data	10
2.2	DBMS	10
2.2.1	Dati	11
2.2.2	DDL	12
2.2.3	DML	12
2.2.4	Schemi e Istanze	13
2.2.5	Meccanismi per il controllo dei dati	13
2.2.6	Transazioni	13
2.3	Progettazione	13
2.3.1	Modellazione	13
2.3.2	Aspetti del problema	15
2.3.3	Conoscenza concreta	15
2.3.4	Modellazione ad oggetti	16
2.3.5	Un esempio elaborato	18

Capitolo 1

Introduzione

Obiettivi del corso Modelli dei dati, linguaggi e sistemi per lo sviluppo di applicazioni che prevedono l'uso di grandi quantità di dati permanenti organizzati in **basi di dati**.

Testo di Riferimento *Fondamenti di Basi di Dati*, A. Albano, G. Ghelli e R. Orsini, Zanichelli. Scaricabile liberamente da fondamentidibasididati.it

Terminologia

Base di dati: tecnologia di base, gestione delle attività quotidiane dell'organizzazione e **tema di questo corso**

Data Warehouse, Data Lake, Big Data, Data Science: termini che hanno a che vedere con l'**analisi dei dati** e che non rientrano nei temi trattati nel corso.

Capitolo 2

Base di Dati

Cos'è una base di dati? Una **base di dati** è un **insieme organizzato di dati** usati per il supporto allo svolgimento di un'attività (di un ente, azienda, ufficio, persona...)

Qualche esempio

Titolo	Codice	Materie	Syllabus
Basi di Dati	AA024	Progettazione e interrogazione...	
Reti di Calcolatori	AA019	Realizzazione e uso di reti, protocollo TCP...	

Materia	AA	Corsi	Titolare
		Semestre	
AA024	2007	1	Albano
AA024	2007	1	Ghelli
AA019	2007	1	Brogi

2.1 Costruzione di una Base di Dati

2.1.1 Figure Coinvolte

Committente

Dirigente
Operatore

Fornitore

Direttore del progetto
Analista
Progettista del DB
Programmatore di applicazioni che usano il DB

Manutenzione e messa a punto del DB – **Gestione del DBMS**

Amministratore del DBMS

2.1.2 Sistemi Informativi

Definizione Un **sistema informativo** di un'organizzazione è una **combinazione di risorse, umane e materiali, e di procedure** organizzate per raccolta, archiviazione, elaborazione e scambio **delle informazioni** necessarie alle attività:

Operative (informazioni di servizio)

Programmazione e **controllo** (informazioni di gestione)

Pianificazione strategica (informazioni di governo)

Esempi di sistemi informativi Un comune

Gestione servizi demografici (anagrafe, stato civile, servizio elettorale e vaccinale) e della rete viaria

Gestione attività finanziaria secondo la normativa vigente

Gestione del personale per il calcolo della retribuzione in base al tipo di normativa contrattuale

Gestione dei servizi amministrativi e sanitari delle USL

Gestione della cartografia generale e tematica del territorio

Sistema informativo nelle organizzazioni



2.1.3 Sistemi Informatici

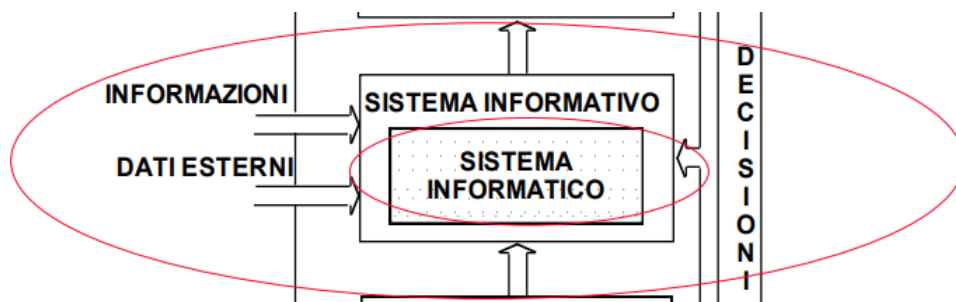
Sistema Informativo Automatizzato Quella parte del sistema informativo in cui le informazioni sono raccolte, elaborate, archiviate e scambiate usando un **sistema informatico**.

Sistema Informatico Insieme delle tecnologie informatiche e della comunicazione (**ICT**, Information and Communication Technologies) a supporto delle attività di un'organizzazione.

Terminologia

Sistema informativo → Sistema informativo automatizzato

Sistema informativo automatizzato → Sistema informatico



2.1.4 Classificazione dei sistemi informatici

Sistemi Informatici Operativi → Sistemi Informatici Direzionali

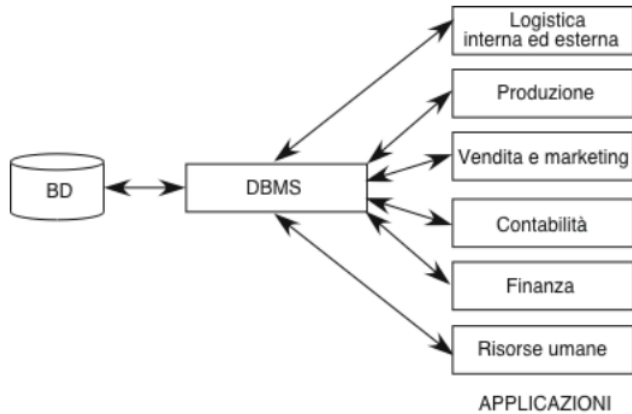
Sistemi Informatici Operativi I dati sono organizzati in DB. Le applicazioni si usano per svolgere le classiche attività strutturate e ripetitive dell'azione nelle aree amministrativa e finanziaria: vendite, risorse umane, produzione. . .

Alcune sigle:

DP Data Processing

EDP Electronic Data Processing

TPS Transaction Processing Systems



DBMS Le caratteristiche del DB sono **garantite da un sistema per la gestione della base di dati** (DBMS, Data Base Management System) che ha il controllo dei dati e li rende accessibili agli utenti autorizzati.

OLTP On-Line Transaction Processing, modo d'uso principale dei DBMS. Tradizionale elaborazione di transazioni, che realizzano processi operativi per il funzionamento di organizzazioni:

Operazioni predefinite e relativamente semplici

Ogni operazione coinvolge *pochi* dati

Dati di dettaglio, aggiornati

Sistemi Informatici Direzionali I dati sono organizzati in data warehouse (DW) e gestiti ad un opportuno sistema. Le applicazioni, dette di **business intelligence**, sono strumenti di supporto ai processi di controllo delle prestazioni aziendali e di decisione manageriale. Terminologia:

MIS Management Information Systems

DSS Decision Support Systems, data-based o model-based

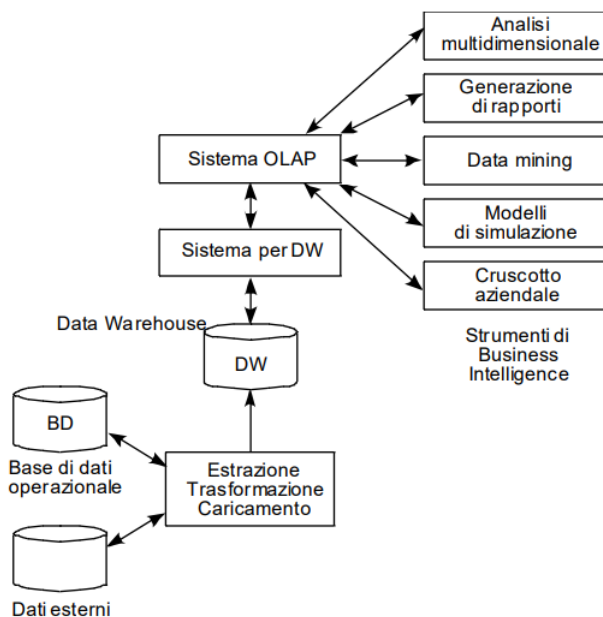
EIS Executive Information System

OLAP On-Line Analytical Processing modo d'uso principale dei DW. Analisi dei dati di supporto alle decisioni:

Operazioni complesse e casuali

Ogni operazione può coinvolgere *molte* dati

Dati aggregati, storici, anche non attualissimi



Differenze tra OLTP e OLAP

	OLTP	OLAP
Scopi	Supporto operatività	Supporto decisioni
Utenti	Molti, esecutivi	Pochi, dirigenti e analisti
Dati	Analitici, relazionali	Sintetici, multidimensionali
Usi	Noti a priori	Poco prevedibili
Quantità di dati per attività	Bassa (decine)	Alta (milioni)
Orientamento	Applicazione	Soggetto
Aggiornamenti	Frequenti	Rari
Visione dei dati	Corrente	Storica
Ottimizzati per	Transazioni	Analisi

2.1.5 Requisiti per l'Analisi dei Dati

Aggregati Non interessa **un** dato, ma la **somma**, la **media**, il **minimo**/**massimo** di una misura. . .

Multidimensionale Interessa **incrociare le informazioni**, per analizzarle da punti di vista diversi e valutare i risultati del business per intervenire sui problemi critici o per cogliere nuove opportunità

Diversi livelli di dettaglio Per esempio, una volta scoperto un calo delle vendite in un determinato periodo in una specifica regione, si passa ad un'analisi dettagliata nell'area di interesse per cercare di scoprirne le cause (dimensioni con **gerarchie**)

2.1.6 Big Data

Ampio Big data è un termine ampio riferito a situazioni in cui l'approccio "schema-first" tipico di DB e DW risulta troppo restrittivo o troppo lento.

3 V Volume, Varietà, Velocità

I Big Data sono in genere associati a sistemi NoSQL, machine learning e approcci Data Lake.

2.2 DBMS

Un **DBMS** è un sistema (**software**) in grado di **gestire collezioni di dati** che siano, tra le altre cose:

Grandi

Persistenti, con un periodo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano

Condivise, usate da applicazioni diverse

garantendo **affidabilità** (resistenza a malfunzionamenti hardware e software-recovery) e **privacy** (con una disciplina e un controllo degli accessi).

Come ogni altro software, un DBMS deve essere **efficiente** (usare al meglio le risorse di spazio e tempo del sistema) ed **efficace** (rendere produttive le attività degli utilizzatori).

Un DBMS offre opportuni linguaggi per:

Definire lo schema di un DB, che va definito prima di creare dati

Scegliere le strutture dati per la memorizzazione

Memorizzare i dati **rispettando i vincoli** definiti nello schema

Recuperare e modificare i dati, interattivamente (**query language**, linguaggio di interrogazione) o da programmi

2.2.1 Dati

I dati permanenti contenuti in un DB sono divisi in due categorie:

Metadati

Descrivono dati sullo schema dei dati, utenti autorizzati, applicazioni, parametri quantitativi. . .

I metadati sono descritti da uno schema usando il modello dei dati usato dal DBMS e sono interrogabili con le stesse modalità previste dai dati

Dati

Rappresentazioni di certi fatti conformi alle definizioni dello schema. Hanno le seguenti caratteristiche:

Organizzati in **insiemi strutturati e omogenei**, fra i quali sono definite delle **relazioni**. La struttura dei dati e le relazioni sono **descritte nello schema** usando i meccanismi di astrazione del modello dei dati del DBMS.

Sono **molti**, sia in assoluto che rispetto ai metadati, e non possono essere gestiti in memoria temporanea

Sono **accessibili mediante transazioni, unità di lavoro atomiche** che **non possono avere effetti parziali**

Sono **protetti** sia **da accesso da parte di utenti non autorizzati**, sia **da corruzione dovuta a malfunzionamenti** hardware o software

Sono **utilizzabili contemporaneamente da utenti diversi**

Il **modello relazionale dei dati** è il più diffuso fra i DBMS commerciali. Il **meccanismo di astrazione** fondamentale è la **relazione (tabella)**, sostanzialmente un insieme di record dai campi elementari.

Lo schema di una relazione ne definisce il nome e ne descrive la struttura dei possibili elementi della relazione (insieme di attributi con il loro tipo)

Esempio

Definizione del DB

```
create database EsempioEsame
```

Definizione schema

```
create table Esami(Materia char(5), Candidato char(8), Voto int, Lode char(1),
Data char(6))
```

Inserzione dati

```
insert into Esami values ('BDSI1', '080709', 30 'S', '070900')
```

Interrogazione

```
select Candidato from Esami where Materia = "BDSI1" and Voto = 30
> Candidato
> 080709
```

2.2.2 DDL

Data Definition Language Linguaggio per la definizione della base di dati.
Utile distinguere tre diversi livelli di descrizione dei dati (**schemi**):

Livello di **vista logica**

Livello **logico**

Livello **fisico**



Livello Logico Descrive la struttura degli insiemi di dati e delle relazioni fra loro, secondo un certo modello dei dati, senza nessun riferimento alla loro organizzazione fisica nella memoria permanente.

Esempi:

```
Studenti(Matricola char(8), Nome char(20), Login char(8), Anno int, Reddito float)
Corsi(IdeC char(8), Titolo char(20), Credito int)
Esami(Matricola char(8), IdeC char(8), Voto int)
```

Livello Fisico Descrive come vanno organizzati fisicamente i dati nelle memorie permanenti e quali strutture dati ausiliarie prevedere per facilitarne l'uso (schema fisico o interno).

Esempi: relazioni Studenti e Esami organizzate in modo seriale, Corsi organizzata sequenziale con indice, indice su Matricola.

Vista Logica Descrive come deve apparire la struttura del DB ad una certa applicazione (**schema esterno** o **vista**). Esempio:

```
InfCorsi (IdeC char(8), Titolo char(20), NumEsami int)
```

Nell'organizzazione di una banca, lo **schema logico** conterrà tutte le tabelle e i dati relativi ai conti correnti, ma anche al personale. Lo schema logico conserva **tutte le informazioni** della banca. Nello **schema esterno** ogni correntista potrà **accedere solo ad alcune informazioni** di suo interesse: quelle del proprio conto corrente.

Indipendenza L'approccio con tre livelli è stato proposto per garantire le proprietà di indipendenza logica e fisica dei dati, fra gli obiettivi più importanti dei DBMS.

Indipendenza fisica: i programmi applicativi non devono essere modificati in seguito a modifiche dell'organizzazione fisica dei dati

Indipendenza logica: i programmi applicativi non devono essere modificati in seguito a modifiche dello schema logico

2.2.3 DML

Data Manipulation Language Linguaggio per l'uso dei dati.

Un DBMS deve prevedere più modalità d'uso per soddisfare esigenze di diverse categorie d'utenti: GUI per accedere ai dati, linguaggio di interrogazione per i non programmatori, linguaggio di programmazione per chi sviluppa le applicazioni, linguaggio di sviluppo per le interfacce delle applicazioni.

Linguaggi vari e interfacce diverse:

Linguaggi testuali interattivi, SQL

Comandi (come quelli del linguaggio interattivo) immersi in un linguaggio ospite, come il C

Comandi (come quelli del linguaggi interattivo) immersi in un linguaggio ad hoc (come PL/SQL) con anche altre funzionalità (come grafici e stampe strutturate)

Interfacce amichevoli

2.2.4 Schemi e Istanze

Schema Descrive la **struttura dei dati**, sostanzialmente invariante nel tempo: le "classi", intestazione delle tabelle

Istanza **Valori attuali** dei dati che possono cambiare anche molto rapidamente: gli "oggetti", il corpo di ciascuna tabella

2.2.5 Meccanismi per il controllo dei dati

Caratteristica molto importante dei DBMS è il tipo di meccanismi usati per garantire le seguenti proprietà

Integrità: mantenimento delle proprietà specificate nello schema

Sicurezza: protezione da usi non autorizzati

Affidabilità: protezione da malfunzionamenti e interferenze dovute all'accesso concorrente di più utenti

2.2.6 Transazioni

Definizione Una **transazione** è una **sequenza di azioni di lettura/scrittura in memoria permanente e di elaborazione dati in memoria temporanea**, con le seguenti proprietà:

Atomicità: le transazioni che terminano prematuramente (**aborted transactions**) sono **trattate dal sistema come se non fossero mai iniziate**. Eventuali effetti sul DB sono **annullati**.

Serializzabilità: esecuzioni concorrenti di più transazioni danno come effetto quello di una esecuzione seriale

Persistenza: le **modifiche** sul DB di una transazione terminata normalmente sono **permanenti**, cioè **non alterabili da malfunzionamenti**

2.3 Progettazione

Progettare Progettare un DB significa **progettare la struttura dei dati e delle applicazioni**. La progettazione dei dati è l'attività più importante e per progettare al meglio i dati è necessario che essi siano un **modello fedele del dominio** in esame. Per questo ora parleremo della **modellazione**.

2.3.1 Modellazione

Definizione Un **modello astratto** è la **rappresentazione formale di idee e conoscenze relative ad un fenomeno**.

Aspetti di un modello:

Il **modello** è la **rappresentazione di certi fatti**.

La **rappresentazione** è **data con un linguaggio formale**.

Il **modello** è il **risultato di un processo di interpretazione**, guidato dalle idee e conoscenze possedute dal soggetto che interpreta.

La stessa realtà può utilmente essere modellata in modi diversi e a diversi livelli di astrazione.



Metodologia di progetto Per garantire prodotti di buona qualità è opportuno seguire una metodologia di progetto, con:

Articolazione delle attività in fasi (**decomposizione**)

Criteri di scelta (**strategie**)

Modelli da rappresentare

Generalità rispetto al problema in esame e agli strumenti a disposizione

Qualità del prodotto

Facilità d'uso

Progettazione della base di dati Suddivisa nelle seguenti fasi:

1. **Analisi** dei requisiti
2. Progettazione **concettuale**
3. Progettazione **logica**
4. Progettazione **fisica**

Ciascuna fase è incentrata sulla modellazione, che discuteremo quindi con riferimento alla problematica della progettazione del DB.

Modello dei dati Insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica. Il componente fondamentale è l'insieme dei **meccanismi di strutturazione** (o **costruttori di tipo**). Come nei linguaggi di programmazione, esistono meccanismi che permettono di definire nuovi tipi, così **ogni modello dei dati prevede alcuni costruttori**: per esempio, il **modello relazionale prevede il costruttore *relazione***, che permette di definire insiemi di record omogenei.

2.3.2 Aspetti del problema

Aspetto ontologico

Quale conoscenza del dominio del discorso si rappresenta? Ontologico cioè studio di ciò che si suppone esista nell'universo del discorso e che sia quindi necessario modellare. Cosa si modella:

Conoscenza concreta: i fatti

Conoscenza astratta: la struttura e i vincoli sulla conoscenza concreta

Conoscenza procedurale, comunicazioni: le operazioni base, le operazioni degli utenti, come si comunicherà con il sistema informatico

Ci concentreremo sulla conoscenza concreta e astratta.

Aspetto logico

Con quali meccanismi di astrazione si modella? Il modello dei dati a oggetti.

Modello dei dati Insieme dei meccanismi di astrazione per descrivere la struttura della conoscenza concreta.

Schema: descrizione della **struttura della conoscenza concreta** e dei **vincoli di integrità** usando un particolare modello dei dati.

Useremo come notazione grafica una **variante** dei diagrammi a oggetti (o diagrammi Entità-Relazione, diagrammi ER). Nozioni fondamentali: oggetto, tipo di oggetto, classe, ereditarietà, gerarchia fra tipi e gerarchia fra classi.

Aspetto linguistico

Con quale linguaggio formale si definisce un modello?

Aspetto pragmatico

Come si procede per costruire un modello? Metodologia da seguire nel processo di modellazione, cioè l'insieme di regole finalizzate alla costruzione del modello informatico.

2.3.3 Conoscenza concreta

La conoscenza concreta riguarda i fatti specifici che si vogliono rappresentare:

Entità, sono **ciò di cui interessa rappresentare alcuni fatti o proprietà**. Ad esempio: una descrizione bibliografica di un libro, un libro, un documento, un prestito, un utente della biblioteca...

Le **proprietà** sono **fatti che interessano solo in quanto descrivono caratteristiche di determinate entità**. Ad esempio un indirizzo interessa perché è l'indirizzo di un utente. Hanno delle classificazioni:

Primitiva/strutturata

Obbligatoria/opzionale

Univoca/multivalore

Costante/variabile

Calcolata/non calcolata

Una proprietà è una coppia attributo-valore di un certo tipo. Ogni entità appartiene ad un **tipo** che ne specifica la natura. Ogni proprietà ha associato un **dominio**, l'insieme dei possibili valori.

Una proprietà è **atomica** se il suo valore non è scomponibile, altrimenti è **strutturata**. Inoltre è **univoca** se ha valore unico, altrimenti è **multivalore**, e **totale** (obbligatoria) se ogni entità dell'universo in esame ha per essa un valore specificato, altrimenti è detta **parziale** (opzionale)

Certi fatti possono essere interpretati come proprietà in certi contesti e come entità in altri. Ad esempio, una *DescrizioniBibliografiche* con attributi *autori*, *titolo*, *editore*..., **oppure** un *Autori* con attributi *nome*, *nazionalità*... e *Editori* con *nome*, *indirizzo*...

Collezioni variabili nel tempo di entità omogenee. Ad esempio, la collezione di tutti gli utenti della biblioteca.

Associazioni fra entità

2.3.4 Modellazione ad oggetti

Oggetti Ad ogni entità del dominio corrisponde un oggetto del modello. Un **oggetto** è un'entità **software con stato, comportamento ed identità** che modella un'entità dell'universo.

Stato modellato da un insieme di costanti o variabili con valori di qualsiasi complessità

Comportamento dato da un insieme di procedure locali chiamate **metodi**, che modellano le operazioni di base che riguardano l'oggetto e le proprietà derivabili da altre.

Un oggetto può rispondere a dai **messaggi**, restituendo valori memorizzati nello stato o calcolati con una procedura locale.

Classe **Insieme di oggetti dello stesso tipo**, modificabile con operatori per includere o estrarre elementi dall'insieme. Può essere specificata a diversi livelli.



Tipo oggetto Il primo passo nella costruzione di un modello consiste nella classificazione delle entità del dominio con la definizione dei tipi degli oggetti che la rappresentano.

Un **tipo oggetto definisce l'insieme dei messaggi (interfaccia) a cui può rispondere un insieme di possibili oggetti**. I nomi dei messaggi sono detti anche attributi degli oggetti.

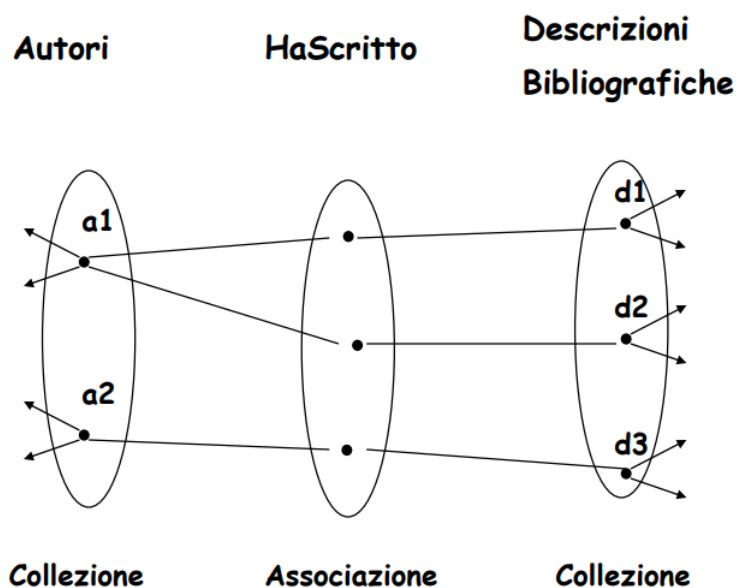
Nei diagrammi ER i tipi oggetti non si rappresentano, perché l'attenzione è sulle collezioni e sulle associazioni. Tuttavia, la rappresentazione grafica di una collezione indica anche gli attributi del tipo oggetto associato.

Associazioni Un'istanza di associazione è un **fatto che correla due o più entità**, stabilendo un legame logico fra loro. Ad esempio, l'utente Tizio ha in prestito una copia della Divina Commedia.

Un'associazione $R(X, Y)$ fra due collezioni di entità X e Y è un **insieme di istanze di associazione** tra elementi di X e di Y che **varia in generale nel tempo**.

Il prodotto cartesiano $X \times Y$ è il dominio dell'associazione.

Un esempio:



Un associazione è **caratterizzata da due proprietà strutturali: molteplicità e totalità**.

Vincolo di univocità Un'associazione $R(X, Y)$ è **univoca rispetto a X** se $\forall x \in X \exists$ al più un elemento di Y associato ad x .

Se non vale questo vincolo, l'associazione è **multivalore rispetto ad X**.

Cardinalità dell'associazione:

$R(X, Y)$ è 1:N se è multivalore su X ed univoca su Y

$R(X, Y)$ è N:1 se è univoca su X e multivalore su Y

$R(X, Y)$ è N:M se è multivalore su X e multivalore su Y

$R(X, Y)$ è 1:1 se è univoca su X ed univoca su Y

Qualche esempio:

Frequenta(Studenti, Corsi) ha cardinalità N:M

Insegna(Professori, Corsi) ha cardinalità 1:N

SuperatoDa(Esami, Studenti) ha cardinalità N:1

Dirige(Professori, Dipartimenti) ha cardinalità 1:1

Vincolo di totalità Un'associazione $R(X, Y)$ è **totale** (o surgettiva) su X se $\forall x \in X \exists$ almeno un elemento di Y associato ad x .

Se non vale questo vincolo, l'associazione è **parziale rispetto a X**.

Ad esempio, Insegna(Professori, Corsi) è totale su Corsi perché non può esistere un corso senza il corrispondente docente.

Rappresentazione Un'associazione si rappresenta con una linea che collega le classi che rappresentano le due collezioni. La linea è etichettata con il nome dell'associazione, di solito scelto utilizzando un predicato.

L'univocità dell'associazione rispetto ad una classe si rappresenta disegnando una freccia singola sulla linea che esce dalla classe ed entra nella destinazione. L'assenza di tale vincolo è indicata da una freccia doppia.

Similmente, la parzialità è rappresentata da un taglio vicino alla freccia, mentre la totalità è rappresentata dall'assenza del taglio.



Ogni esame riguarda uno ed un solo studente
Parzialità e assenza di univocità sugli esami
superati da uno studente.



Possono avere proprietà ed essere ricorsive.

2.3.5 Un esempio elaborato

Gli utenti della biblioteca vengono sospesi dal servizio se non rispettano le regole del prestito. Gli utenti regolari possono essere studenti o docenti. Di uno studente interessa anche la matricola e di un docente anche il telefono dell'ufficio.

Alcune opere sono per la sola consultazione e possono essere presi in prestito solo da docenti.

