

Entity-Relationship

Vincenzo Calabrò

Data modeling

È una disciplina che aiuta a definire:

- i dati che ci interessa rappresentare
- le relazioni tra questi dati

- per organizzare logicamente le informazioni che ci servono
- (e) per definire archivi (basi dati) che ci consentano di gestirle nel tempo

Dati (in formato tabellare)

	PARTITE				M.I.	RETI		RIGORI		PUNTI
	G	V	N	P		F	S	FAV.	CON.	
Inter	16	10	4	2	2	32	16	6	1	34
Roma	16	9	6	1	1	24	9	3	0	33
Chievo	16	9	3	4	-2	28	19	4	4	30
Juventus	16	7	7	2	-4	27	13	2	0	28
Milan	16	7	6	3	-5	25	18	3	2	27
Lazio	16	6	7	3	-7	21	12	1	1	25
Bologna	16	7	3	6	-8	13	16	2	0	24
Verona	16	6	4	6	-10	22	24	4	4	22
Atalanta	16	6	3	7	-11	21	27	1	4	21
Udinese	16	6	3	7	-13	24	25	6	2	21
Perugia	16	5	4	7	-13	17	20	2	4	19
Brescia	16	4	6	6	-14	19	28	3	3	18
Piacenza	16	5	3	8	-14	23	24	3	3	18
Lecce	16	4	5	7	-15	18	24	3	4	17
Torino	16	4	5	7	-15	18	22	4	4	17
Fiorentina	16	4	2	10	-18	17	32	1	5	14
Parma	16	3	5	8	-18	17	24	0	4	14
Venezia	16	2	4	10	-20	12	25	0	3	10

Tabella

- è un formato tipico di rappresentazione di dati (non l'unico possibile!)

	PARTITE				M.I.	RETI		RIGORI		PUNTI
	G	V	N	P		F	S	FAV.	CON.	
Inter	16	10	4	2	2	32	16	6	1	34
Roma	16	9	6	1	1	24	9	3	0	33
Chievo	16	9	3	4	-2	28	19	4	4	30
Juventus	16	7	7	2	-4	27	13	2	0	28
Milan	16	7	6	3	-5	25	18	3	2	27
Lazio	16	6	7	3	-7	21	12	1	1	25
Bologna	16	7	3	6	-8	13	16	2	0	24
Verona	16	6	4	6	-10	22	24	4	4	22
Atalanta	16	6	3	7	-11	21	27	1	4	21
Udinese	16	6	3	7	-13	24	25	6	2	21
Perugia	16	5	4	7	-13	17	20	2	4	19
Brescia	16	4	6	6	-14	19	28	3	3	18
Piacenza	16	5	3	8	-14	23	24	3	3	18
Lecce	16	4	5	7	-15	18	24	3	4	17
Torino	16	4	5	7	-15	18	22	4	4	17
Fiorentina	16	4	2	10	-18	17	32	1	5	14
Parma	16	3	5	8	-18	17	24	0	4	14
Venezia	16	2	4	10	-20	12	25	0	3	10

righe

colonne

Progettazione di basi dati

Obiettivo:

- definire gli archivi ("basi dati"), che verranno gestiti da un DBMS (Data Base Management System) o un File System

Vincoli:

- i dati contenuti negli archivi devono corrispondere alla realtà che si vuole rappresentare
- l'integrità (correttezza) dei dati deve essere preservata
- la base dati deve essere accessibile con prestazioni soddisfacenti per l'utilizzatore

Tecniche per la progettazione delle basi dati

Si basano sulla separazione (temporale, logica) di due aspetti:

- **modellazione dati** - riguarda la scoperta e la definizione dei dati necessari, e delle associazioni di significato che esistono tra di essi:
 - il risultato è un modello "concettuale", indipendente dalle caratteristiche tecnologiche del DBMS o file system
- **progettazione logico-fisica** - riguarda l'implementazione del modello concettuale in una specifica tecnologia DBMS
 - il risultato è la definizione effettiva della base dati nel DBMS

Modello (schema) concettuale

- la sua strutturazione dipende esclusivamente dai legami di significato che esistono tra i dati, non da criteri di efficienza
- è totalmente indipendente dalle caratteristiche di ogni specifico DBMS
- se non esistono necessità di ottimizzazioni particolari, il modello concettuale può originare direttamente le basi dati effettive

Peter Chen

- ideatore della tecnica Entity-Relationship
 - CHEN, Peter: The Entity-Relationship Model - Toward a Unified View of Data - ACM 1976
- Chen propose, insieme alla tecnica, una rappresentazione grafica, il diagramma E/R (ERD)
- il diagramma E/R è oggi quello più ampiamente utilizzato per rappresentare le strutture dati, anche se spesso con formalismi diversi da quelli originali

Definizione di entità

"una qualsiasi cosa che può essere distintamente identificata" (Chen)

cioè un qualsiasi oggetto che:

- abbia una propria individualità (sia distinguibile da oggetti consimili)
- abbia rilevanza per il nostro sistema

ad esempio, in una classe, ogni allievo è un'entità distinta
in un sistema di fatturazione, ogni fattura è un'entità distinta

Tipo entità, occorrenza entità

Il tipo entità è un'astrazione, corrispondente all'insieme di tutte le singole entità che hanno caratteristiche analoghe



Cristina Morra



Mario Rossi



Paola Cavalli

- “allievo” è un tipo entità
- i singoli allievi sono considerabili come occorrenze dell'entità “allievo”
- nel corso del tempo, il numero di occorrenze legato al tipo entità “allievo” può variare

Tipo entità, occorrenza entità

- ma nella pratica linguistica corrente,
entità = tipo entità
- “quali sono le entità del sistema?”
Fattura, Fornitore, Ordine
e non
(le fatture) 1, 2, 3, ...
(i fornitori) F1, F5, F48, ...

Attributo

Gli attributi sono le proprietà che caratterizzano le entità

- gli attributi definiti a livello di (tipo) entità si riferiscono a tutte le singole occorrenze della specifica entità
- in ogni specifica occorrenza, ciascun attributo assumerà un valore particolare



attributi di allievo:

- nome
- cognome
- data nascita
- luogo nascita
- sesso



Mario Rossi

- nome: "Mario"
- cognome: "Rossi"
- data nascita: "4/10/1962"
- luogo nascita: "Venezia"
- sesso: "M"



Cristina Morra

- nome: "Cristina"
- cognome: "Morra"
- data nascita: "25/1/1980"
- luogo nascita: "Firenze"
- sesso: "F"

Rappresentazione tabellare

- gli attributi dell'entità corrispondono alle colonne
- le occorrenze, alle righe
- in ogni cella, il valore assunto dall'attributo relativamente alla specifica occorrenza di entità

nome	cognome	data nascita	luogo nascita	sess
Cristina	Morra	25/1/1980	Firenze	F
Mario	Rossi	4/10/1962	Venezia	M
Paola	Cavalli	16/8/1975	Roma	F

cardinalità =
numero
occorrenze

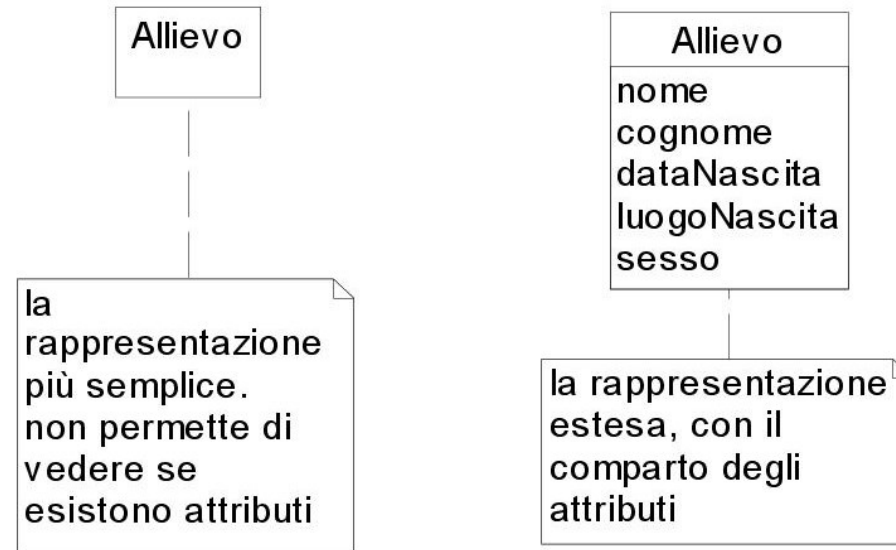
grado = numero attributi

Unified Modeling Language (UML)

- linguaggio (e notazione) universale, per rappresentare qualunque tipo di sistema (software, hardware, organizzativo, ...)
- standard OMG (Object Management Group), dal 1997
- originatori:
 - Grady Booch
 - Ivar Jacobson
 - Jim Rumbaugh

Profilo data modeling UML

- il diagramma delle classi UML può essere utilizzato per rappresentare le entità di un modello Entity-Relationship



Entità e attributi

- un (tipo) entità può essere identificato e definito inizialmente anche senza definire gli attributi
- ma gli attributi sono necessari per indicare le proprietà (i dati) che ci interessa gestire per le occorrenze dell'entità
- ... ed anche per chiarire meglio il significato dell'entità

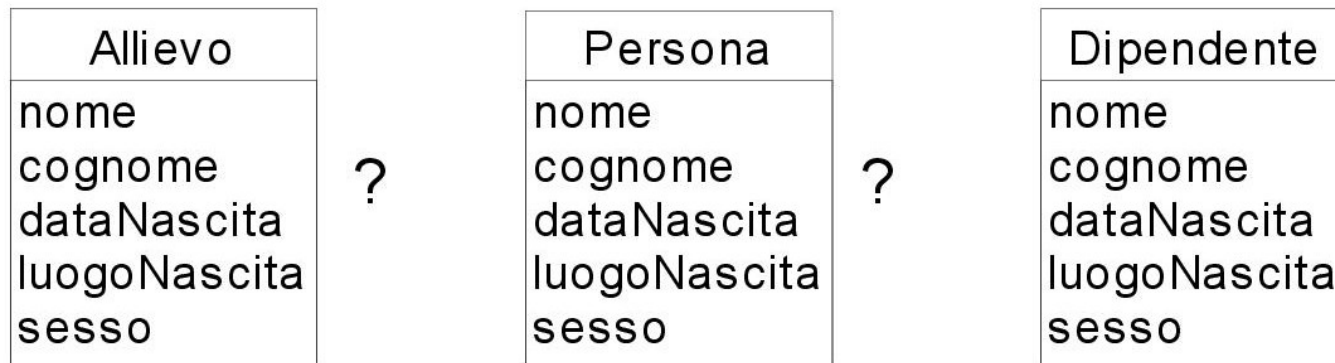
Allievo
nome
cognome
dataNascita
luogoNascita
sex

“Allievo” o “Persona” ?

Nome dell'entità

Il nome del (tipo) entità deve riflettere

- l'insieme di occorrenze che si vuole facciano parte dell'entità
- le proprietà (attributi) che caratterizzano tali occorrenze
- il (i) contesto applicativo in cui verrà utilizzata l'entità

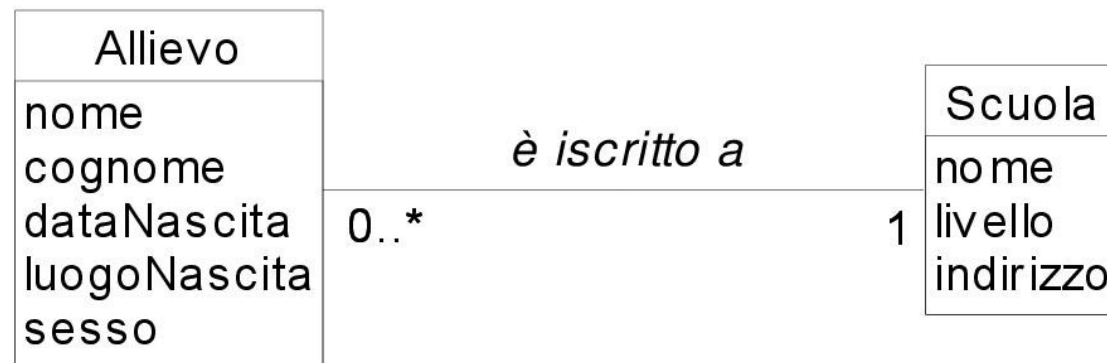


Associazione (relationship)

Le entità sono collegate tra loro da associazioni

L'entità allievo ha una associazione con l'entità scuola:

- ogni (occorrenza di) allievo è iscritto ad una (occorrenza di) scuola
- ogni (occorrenza di) scuola può avere come iscritti molti (occorrenze di) allievi



Molteplicità dell'associazione

Ogni associazione tra due entità definisce delle molteplicità, che specificano a quante occorrenze di una entità è associabile ogni singola occorrenza dell'entità corrispondente



- per ogni allievo esiste come minimo una scuola, e come massimo una sola scuola;
- per ogni scuola esistono come minimo zero allievi, e come massimo N allievi



- per ogni allievo esistono come minimo zero borse di studio, e come massimo una sola borsa di studio;
- per ogni borsa di studio esistono come minimo zero allievi, e come massimo N allievi

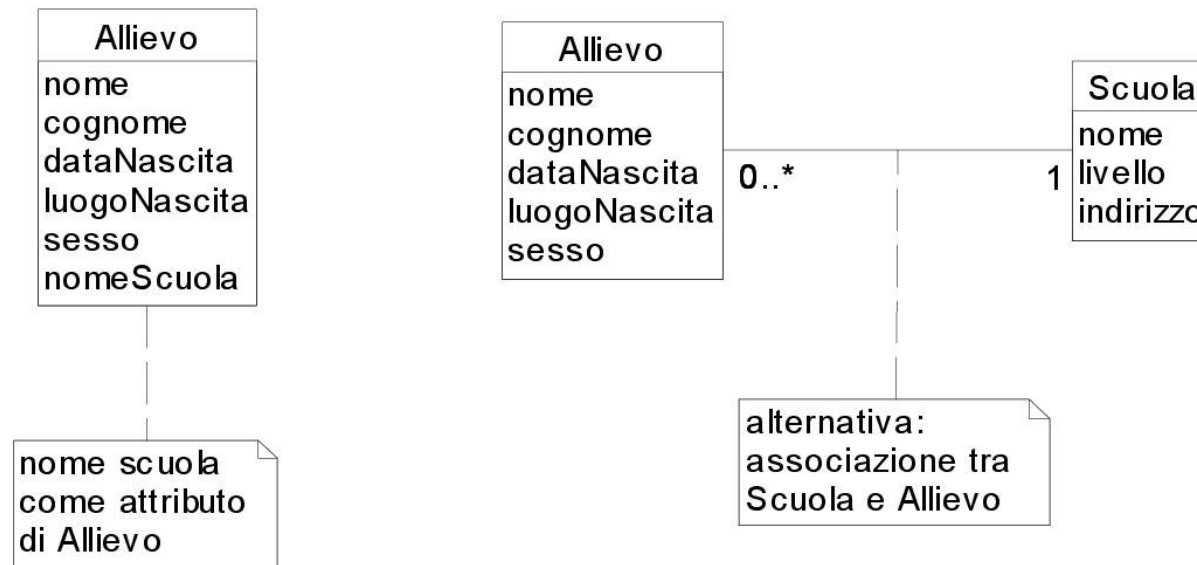
Associazione: ruoli

- in ogni associazione esistono due punti di vista, o **ruoli**, per ciascuno dei quali può essere definito un nome (nome di ruolo)
- il nome esprime il significato delle occorrenze, dal punto di vista delle occorrenze associate nell'altra entità



Attributi e associazioni

- alcune informazioni possono essere considerate, in modo alternativo, o come attributi o come associazioni con altre entità
 - il nome della scuola a cui un allievo è iscritto può essere rappresentato come uno degli attributi dell'allievo, oppure può essere creata un'entità scuola, con la quale viene associata l'entità allievo



Attributi e associazioni (...)

nome scuola come
attributo di allievo: vantaggi

- il modello è più semplice
- riduzione del numero di funzionalità del sistema (non sono necessarie funzionalità specifiche per inserire una nuova scuola, o modificare i dati relativi ad una scuola esistente)

scuola come entità distinta,
associata ad allievo: vantaggi

- possono essere gestite più informazioni relative alla scuola, senza doverle ripetere per ogni allievo iscritto
- anche se in un certo momento non esistono allievi iscritti alla scuola, le informazioni relative alla scuola vengono mantenute

Attributi elementari e aggregati

- gli attributi di un'entità possono essere elementari (non scomponibili) oppure aggregati (scomponibili)
 - la “nazionalità” di un allievo è un attributo elementare, non scomponibile
 - l’“indirizzo” di un allievo è un attributo aggregato, formato dai potenziali attributi elementari “via”, “numero civico”, “CAP”, “comune”, “provincia”, “stato”

Attributi elementari e aggregati

- per ciascun attributo è necessario chiedersi quale sia il livello di "elementarietà" adeguato per il tipo di sistema (e di archivio) da progettare:
 - in ambito postale può essere utile distinguere tra via e numero civico, ma nella maggioranza degli altri sistemi questa distinzione risulta eccessiva
 - in una anagrafe fiscale può essere utile scomporre il codice fiscale nelle sue unità costitutive, ma negli altri tipi di sistemi?
- scomporre fino al livello in cui “ha senso” utilizzare gli attributi come parametri per effettuare ricerche negli archivi, e non oltre

Attributi e valori

- ogni (occorrenza di) allievo "ha" un solo valore per gli attributi nome, cognome, data e luogo di nascita, sesso e stato civile
- il valore di ogni attributo può essere modificato (ad esempio, lo stato civile potrebbe passare da "nubile" a "coniugata"), ma in ciascun istante ogni attributo ha un valore solo (non è possibile essere contemporaneamente "nubile" e "coniugata")
- l'attributo, in questo caso, è detto "atomico"



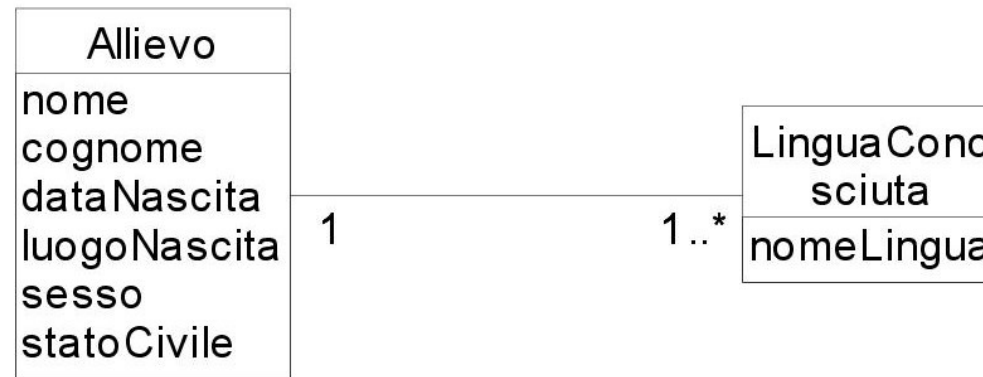
Cristina Morra

- nome: "Cristina"
- cognome: "Morra"
- data nascita: "25/1/1980"
- luogo nascita: "Firenze"
- sesso: "F"
- stato civile: "nubile"

Attributi multivalore

- esistono, però, attributi "multivalore", per i quali, in ogni occorrenza di entità, possono essere validi più valori
- ogni allievo, ad esempio, può conoscere "più" lingue straniere
- "lingue straniere conosciute", in quanto multivalore e non atomico, non può essere rappresentato come attributo di "allievo"

➤ in questi casi si crea una nuova entità, attributiva, che ha l'attributo multivalore ed è associata all'entità



Attributi opzionali

- se un attributo può non essere valorizzato per alcune occorrenze dell'entità è detto opzionale
- es. alcune persone sono sposate, altre no

Persona
nome
cognome
dataNascita
luogo Nascita
sex
nomeConiuge[0..1]
cognomeConiuge[0..1]

Attributi derivati

- un attributo può ricevere i suoi valori dall'applicazione di un algoritmo

Persona
nome
cognome
dataNascita
luogoNascita
sex
nomeConiuge[0..1]
cognomeConiuge[0..1]
/ età

Attributi e data type

- per ogni attributo esiste un tipo di dato ("data type") da cui l'attributo può trarre i propri valori
- il data type può essere di base (system-defined), ossia scelto tra quelli disponibili a livello di DBMS
 - stringa di caratteri alfanumerici
 - numero intero
 - data
 - booleano (vero / falso)
 - registrazione audio
 - filmato
- oppure definito dall'utente (user-defined)

Identificatori delle entità

- per ogni entità deve essere presente almeno un identificatore, che consenta di identificare univocamente ciascuna occorrenza
- l'identificatore può essere costituito da:
 - un attributo
 - un insieme di attributi
 - un insieme di attributi e associazioni
- l'esistenza di un identificatore garantisce che non esistano occorrenze duplicate

Chiave primaria (PK)

- è l'identificatore principale (o unico) dell'entità
- occorrenze diverse devono avere un valore diverso di primary key

Immobile
<<PK>> idImmobile
indirizzo
valore

Identificatori "naturali"

- per alcune entità può essere già disponibile un identificatore "naturale", cioè un attributo (o un insieme di attributi) in grado di distinguere in modo univoco ciascuna occorrenza, e già conosciuto / utilizzato dagli utenti

Contribuente
<<PK>> codFiscale
nome
cognome

Autoveicolo
<<PK>> numTelaio
cilindrata
numRuote

Fattura
<<PK>> anno
<<PK>> numero
data
importo

PK elementare

PK composta

Identificatori “artificiali”

- se non esiste un identificatore naturale, bisogna crearne uno artificiale
 - ad esempio, l'entità allievo non ha un identificatore naturale: non è possibile garantire in modo assoluto l'assenza di omonimie
 - quindi è necessario aggiungere un nuovo attributo che serva da identificatore (tipicamente, un codice numerico progressivo)

Allievo
nome
cognome
dataNascita
luogoNascita
sex
statoCivile
idAllievo

Identificatori alternativi

Può accadere che un'entità abbia più di un identificatore univoco per le proprie occorrenze

Dipendente
nome
cognome
dataNascita
luogoNascita
sex
matricola
codFiscale

matricola e codice fiscale
sono entrambi identificatori
dell'entità dipendente

Ogni possibile identificatore è detto “chiave candidata”

- occorrenze diverse non possono avere lo stesso valore della chiave candidata

Scelta chiave primaria

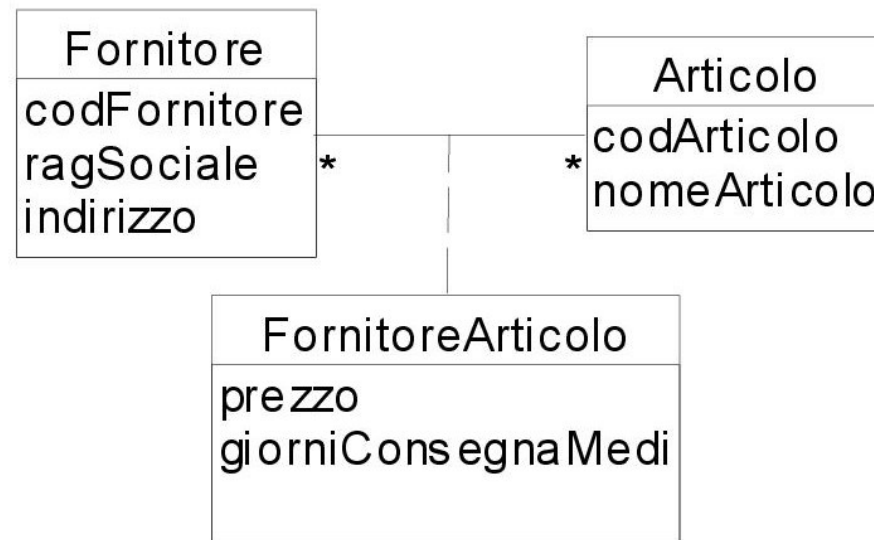
- tra le chiavi candidate, una diventa chiave primaria (PK)
- le altre sono dette “chiavi alternative” (alternate key - AK)

Dipendente
nome
cognome
dataNascita
luogoNascita
sex
<<PK>> matricola
<<AK>> codFiscale

La scelta si basa soprattutto sulla resistenza nel tempo della definizione e dei valori dell'attributo

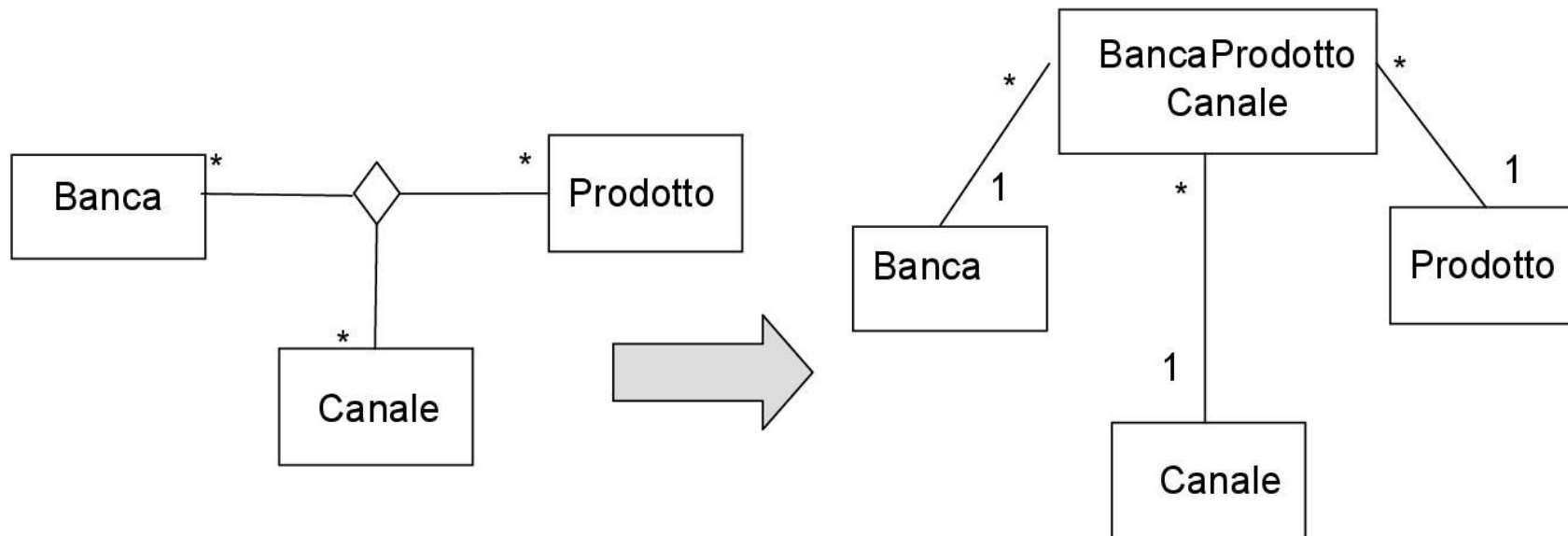
Entità associative

- permettono di gestire attributi relativi all'associazione tra due altre entità
- derivano da associazioni “molti a molti”



Associazioni n-arie

- è possibile definire associazioni tra più di due entità
- la molteplicità è meno chiara (sconsigliato)
- meglio introdurre una nuova entità, collegata alle entità esistenti da associazioni binarie



Domande?

www.vincenzocalabro.it