

Introduzione alle basi di dati e ai DBMS (riferimento: [SA15, Cap. 13])

EUGENIO OMODEO
Università degli Studi di Trieste.

Trieste, 11.11.2020

Citazione del giorno

Una base di dati (database) è una collezione di dati correlati creata con lo scopo specifico di rappresentare adeguatamente e memorizzare le informazioni relative a una realtà di interesse.

La base di dati del sito cinematografico conterrà dati su attori, registi, film e così via;

quella relativa agli esami conterrà dati su studenti, corsi di laurea, esami, docenti e così via.

Un'altra caratteristica fondamentale delle basi di dati è che i dati memorizzati vengono mantenuti costantemente aggiornati nel tempo, per riflettere le modifiche che man mano si verificano nella realtà ...

[CFM09, pag. 2]

Scaletta

DBMS

Mansioni

Gestione dell'informazione su piú livelli

La progettazione concettuale

Il modello ER in generale

I costrutti del modello ER

Sistemi di gestione di basi di dati

*Un **DBMS** (Database Management System) è un sistema software specificamente realizzato per supportare la definizione e la manipolazione della base di dati da parte degli utenti.*

[CFM09, pag. 2]

A che serve un DBMS ?

- ▶ Fornisce agli utenti una **visione di alto livello** dei dati nel DB, **nascondendo i dettagli relativi alla loro memorizzazione fisica.**
- ▶ Sostiene la **condivisione** dati da parte di più utenti e applicaz., garantendo corretta gestione di accessi contemporanei.
- ▶ Garantisce la **sicurezza** dei dati, permettendo l'accesso ai dati e la loro modifica solo agli utenti autorizzati (offrendo a ciascuno solo le operazioni 'lecite').
- ▶ Fornisce **viste** personalizzate dei dati alle diverse tipologie di utenti e applicazioni.
- ▶ Offre funzionalità di **salvataggio** e **ripristino** dei dati, affinché non possano perdersi dati neppure in caso di malfunzionam. o guasti.
- ▶ Assicura la coerenza fra i dati e i **vincoli d'integrità**.

Intensione ed estensione di un DB

- ▶ Un DBMS realizza in termini concreti un **modello dei dati**.
- ▶ La descrizione delle caratteristiche dei dati di un DB utilizzando un modello dei dati costituisce lo **schema del DB**.
- ▶ I dati veri e propri contenuti in un certo momento nella base di dati costituiscono lo **stato corrente**, o **istanza** del DB.

Schema e contenuto

Modello: Il m. oggi prevalente è quello **relazionale**.¹

Schema: Lo s. fornisce una descrizione del contenuto del DB, sostanzialm. **invariante** nel tempo.

Stato: Lo s. **varia** di frequente, per riflettere gli aggiornamenti apportati ai dati.

¹Storicam., sono apparsi prima lo schema detto *gerarchico* e quello *reticolare*.

Origine del modello relazionale

Information Retrieval

P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. Codd

IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data organization, hierarchies of data, networks of t redundancy, consistency, composition, join, retr calculus, security, data integrity

CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22,

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

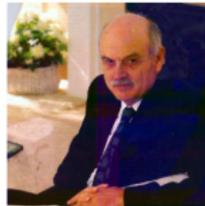
A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the “connection trap”).

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

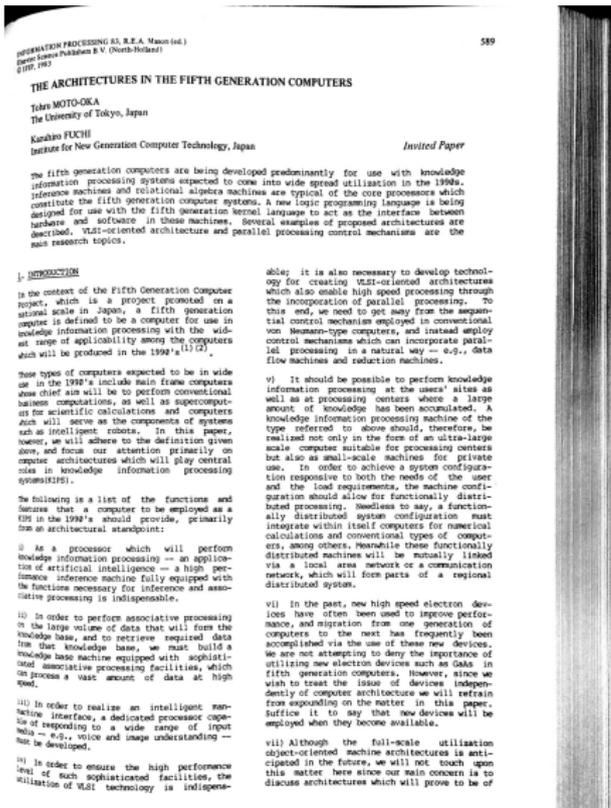
1.2. DATA DEPENDENCIES IN PRESENT SYSTEMS

The provision of data description tables in recently de-

N.B.: Risale agli anni 1970
l'*algebra relazionale di Codd*
per i sistemi DataBase



Evoluzione del metodo relazionale: la sfida giapponese '80



INFORMATION PROCESSING 83, R.E.A. Mason (ed.)
North-Holland Publishers B.V. (North-Holland)
© 1983, 1982

589

THE ARCHITECTURES IN THE FIFTH GENERATION COMPUTERS

Takao MOTO-OKA
The University of Tokyo, Japan
Kazuhito FUCHI
Institute for New Generation Computer Technology, Japan

Invited Paper

The fifth generation computers are being developed predominantly for use with knowledge information processing systems expected to come into wide spread utilization in the 1990s. Inference machines and relational algebra machines are typical of the core processors which constitute the fifth generation computer systems. A new logic programming language is being designed for use with the fifth generation kernel language to act as the interface between hardware and software in these machines. Several examples of proposed architectures are described. VLSI-oriented architecture and parallel processing control mechanisms are the main research topics.

1. INTRODUCTION

In the context of the Fifth Generation Computer Project, which is a project promoted on a national scale in Japan, a fifth generation computer is defined to be a computer for use in knowledge information processing with the widest range of applicability among the computers which will be produced in the 1990's [1] [2].

These types of computers expected to be in wide use in the 1990's include main frame computers whose chief aim will be to perform conventional business computations, as well as supercomputers for scientific calculations and computers which will serve as the components of systems such as intelligent robots. In this paper, however, we will adhere to the definition given above, and focus our attention primarily on computer architectures which will play central roles in knowledge information processing systems [3] [4].

The following is a list of the functions and features that a computer to be employed as a FGC in the 1990's should provide, primarily from an architectural standpoint:

1) As a processor which will perform knowledge information processing — an application of artificial intelligence — a high performance inference machine fully equipped with the functions necessary for inference and associative processing is indispensable.

2) In order to perform associative processing on the large volume of data that will form the knowledge base, and to retrieve required data from that knowledge base, we must build a knowledge base machine equipped with sophisticated associative processing facilities, which can process a vast amount of data at high speed.

3) In order to realize an intelligent man-machine interface, a dedicated processor capable of responding to a wide range of input media — e.g., voice and image understanding — must be developed.

4) In order to ensure the high performance level of such sophisticated facilities, the utilization of VLSI technology is indispen-

sable; it is also necessary to develop technology for creating VLSI-oriented architectures which also enable high speed processing through the incorporation of parallel processing. To this end, we need to get away from the sequential control mechanism employed in conventional von Neuman-type computers, and instead employ control mechanisms which can incorporate parallel processing in a natural way — e.g., data flow machines and reduction machines.

5) It should be possible to perform knowledge information processing at the users' sites as well as at processing centers where a large amount of knowledge has been accumulated. A knowledge information processing machine of the type referred to above should, therefore, be realized not only in the form of an ultra-large scale computer suitable for processing centers but also as small-scale machines for private use. In order to achieve a system configuration responsive to both the needs of the user and the load requirements, the machine configuration should allow for functionally distributed processing. Needless to say, a functionally distributed system configuration must integrate within itself computers for numerical calculations and conventional types of computers, among others, meanwhile these functionally distributed machines will be mutually linked via a local area network or a communication network, which will form parts of a regional distributed system.

6) In the past, new high speed electron devices have often been used to improve performance, and migration from one generation of computers to the next has frequently been accomplished via the use of these new devices; we are not attempting to deny the importance of utilizing new electron devices such as GaAs in fifth generation computers, however, since we wish to treat the issue of devices independently of computer architecture we will refrain from expounding on the matter. In this paper, suffice it to say that new devices will be employed when they become available.

7) Although the full-scale utilization of object-oriented machine architectures is anticipated in the future, we will not touch upon this matter here since our main concern is to discuss architectures which will prove to be of

Es. di: oggetti / insiemi di oggetti / attributi / associaz.

Informazione	Descrizione	Istanza
Oggetti	Elem. individuali e particolari della realtà, cui riconosciamo un'esistenza	L'attrice <i>Uma Thurman</i> , il film <i>Pulp Fiction</i>
Insiemi	formati di oggetti accomunati da certe caratteristiche	L'insieme degli <i>attori</i> , l'insieme dei <i>film</i>
Attributi	Caratteristiche secondarie (proprietà) degli oggetti	Il <i>nome</i> degli attori, il <i>titolo</i> dei film
Associazioni	Legami logici fra gli oggetti	<i>Uma Thurman ha recitato in Pulp Fiction</i>

Il modello relazionale

Nel *m. relazionale*:

- ▶ i dati sono memorizzati in **tabelle**, dette appunto *relazioni*,
- ▶ composte da un certo numero di **colonne**, dette *attributi*,
- ▶ e di un certo numero di **righe**, dette *tuple*.

Quest'organizzaz. tabellare risulta intuitiva e fa sí che i DB relazionali possano essere interrogati con relativa facilità anche da utenti non esperti della tecnologia relazionale.

Rappresentazione tabellare dei dati

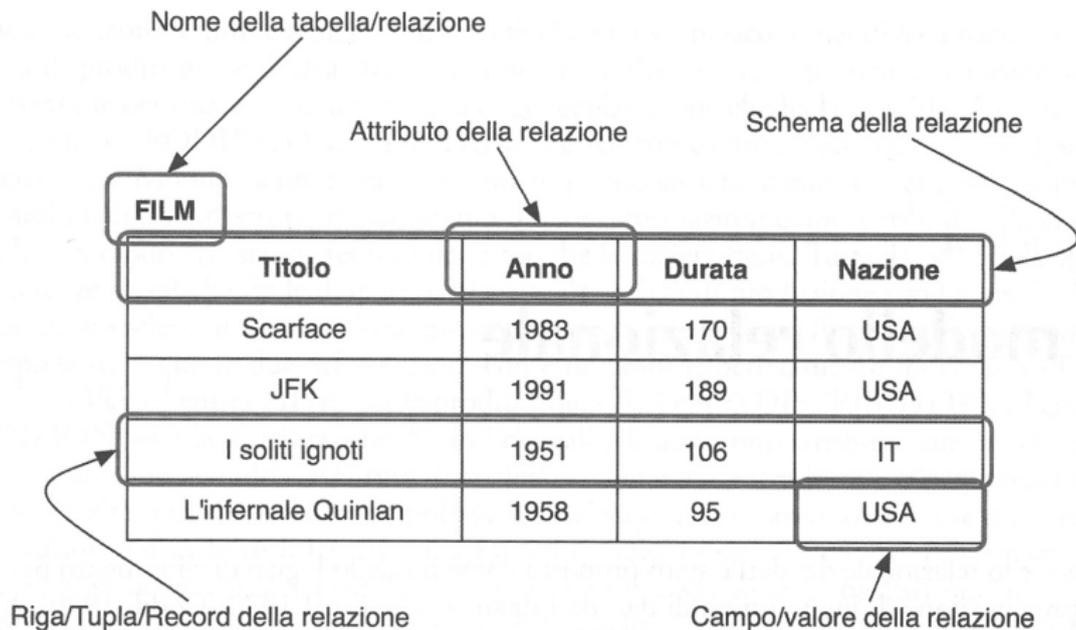


Figura 4.1 Esempio di rappresentazione tabellare dei dati.

Livelli di astrazione

Un DB può essere modellato a tre livelli di astrazione:

Schema logico: fornisce una descrizione dell'intero DB vicina alla *concettualizzazione* della realtà per mezzo del modello dei dati utilizzato nel DBMS.

Schema fisico: troppo tecnico e complesso per interessare l'utente *finale*, cui viene *nascosto* (viene gestito internamente dallo stesso DBMS).

Viste esterne: a varie *tipologie di utenti* i dati possono essere presentati secondo schemi diversi, che fanno riferimento solo ai dati che sono d'interesse specifico per quella particolare utenza.

Modello Entità-Relazione (ER)

Chen, 1976

- ▶ La *realizzazione*, come sistema *software*, di un DB relativo a una **realtà di interesse** dev'essere preceduta da una fase di *progettazione*.
- ▶ Tale **progettazione**, detta **concettuale**, culmina in un diagramma che, utilizzando uno specifico formalismo, schematizza la realtà d'interesse.
- ▶ Uno dei formalismi piú usati per tali scopi è il **modello Entità-Relazione**.

Una regola pragmatica

“lo schema non rappresenta la realtà di riferimento in termini assolutamente veri, ma solo funzionali all'applicazione che necessita della base di dati.

... nella progettazione delle basi di dati l'obiettivo è la definizione della realtà di riferimento dell'applicazione e non la mimesi della realtà in quanto tale”.

[CFM09, pag. 18]

Entità, relazioni, attributi

I principali costrutti del modello ER sono le:

entità

relazioni (dette anche '*associazioni*') e gli

attributi

A questi costrutti si aggiungono notazioni per indicare *vincoli* e *proprietà*, fra cui:

cardinalità delle relazioni

gerarchie di generalizzazione

Entità e attributi

Un' **entità**, nel modello ER, rappresenta un insieme di oggetti della realtà che hanno proprietà comuni ed *esistenza autonoma* ai fini dell'applicazione d'interesse.

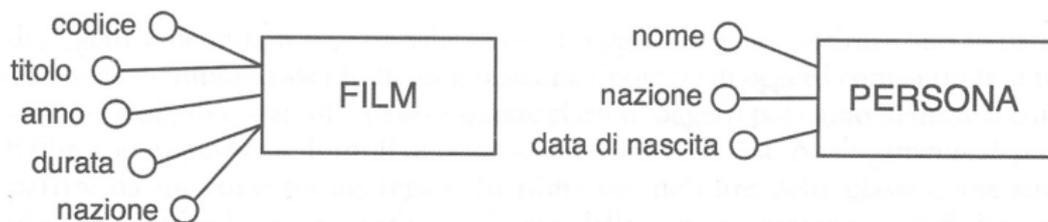


Figura 3.1 Esempio di entità con attributi.

Entità e attributi



Ogni entità, in uno schema ER,

- ▶ ha un **nome** che univocamente la identifica;
- ▶ è raffigurata da un **rettangolo**;
- ▶ è caratterizzata da una collezione di *proprietà elementari*, chiamate **attributi**,² comuni agli oggetti della realtà rappresentati dall'entità stessa.

²Gli a. dipendono dall'e.; non viene riconosciuta 'esistenza autonoma'

Entità e attributi



Gli attributi sono raffigurati come **cerchietti**, ciascuno associato al nome dell'attr. e collegato con una linea all'entità cui s'intende riferirlo.

Relazioni fra entità

Una **relazione**, nel modello ER, rappresenta un legame logico fra due o piú entità.

- ▶ Ha un **nome** che univocamente la identifica;
- ▶ è raffigurata da un **rombo**, collegato con linee alle entità che partecipano alla relazione;
- ▶ può possedere *attributi*.

Relazioni fra entità

Tre esempi

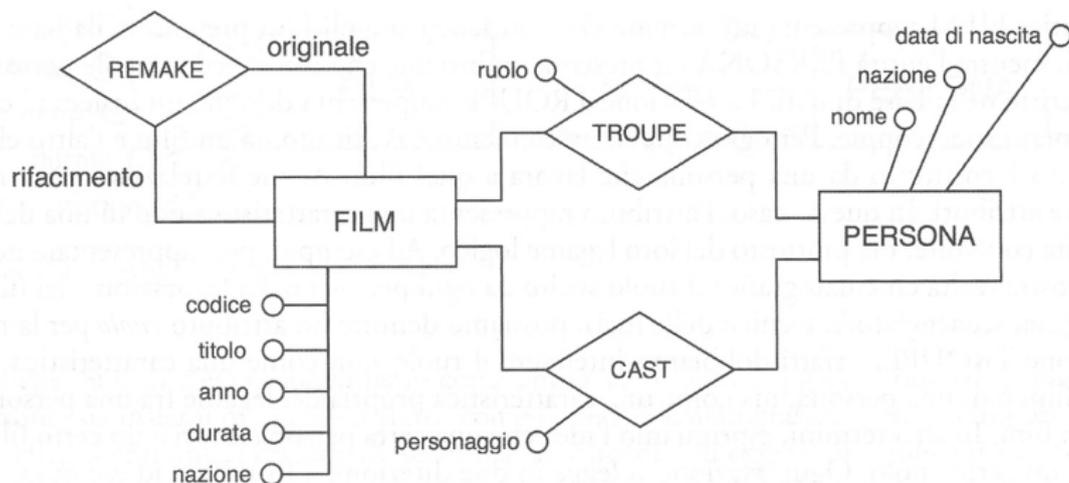


Figura 3.2 Esempi di relazioni fra entità.

(Anche la REMAKE, che è *ricorsiva*, ha due direzioni;
il 'ruolo', in TROUPE, può essere: regista, . . . , tecnico delle luci)

Indicatori di cardinalità

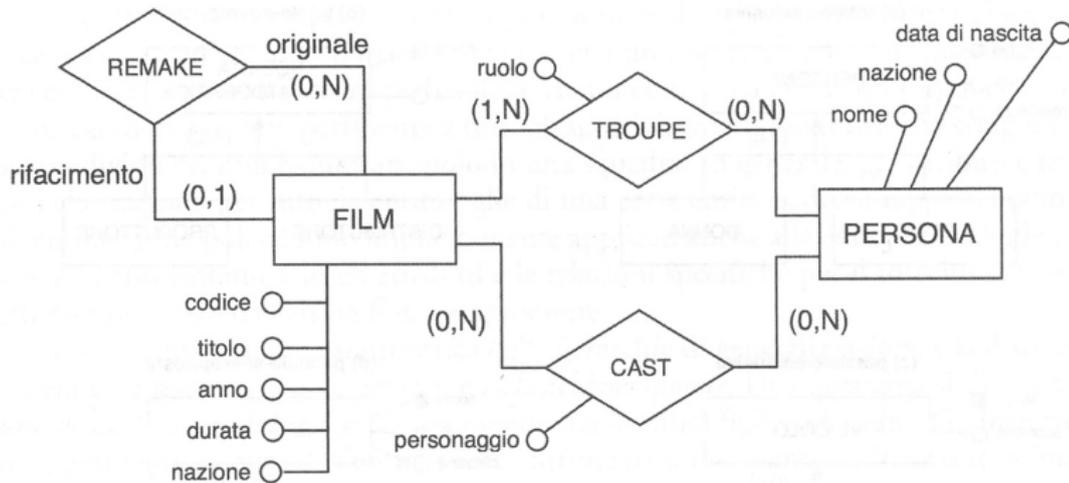


Figura 3.3 Esempi di relazioni fra entità con l'indicazione della cardinalità.

(Un CAST con 0 attori sarà un cartone animato;
una PERSONA può non lavorare ad alcun film)

Indicatori di cardinalità

Spiegazione

Sono importanti il *numero minimo* e *massimo* di corrispondenze che si possono avere fra gli oggetti delle *due* entità nella realtà di riferimento.

Dal momento che ogni relazione ha due sensi di lettura, è importante indicare tali numeri di corrispondenze per entrambe le direzioni.

Conveniamo che la lettera **N** indichi l'assenza di un limite prefissato.

Identificatori

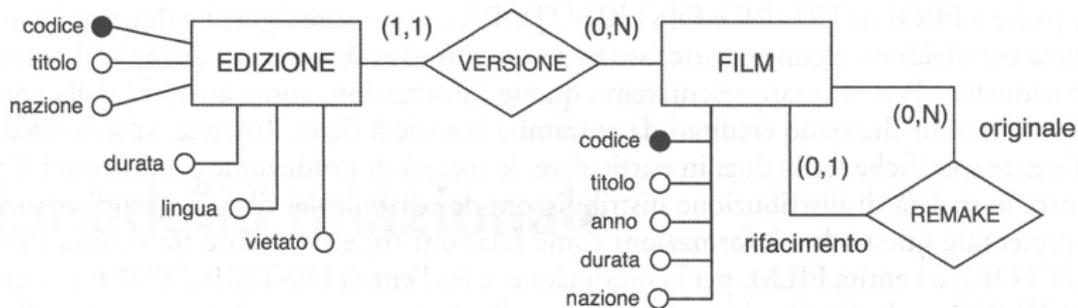


Figura 3.6 Prima porzione di schema ER.

Un **identificatore** è una collezione di attributi di E che consente di identificare univocam. gli elementi di E .

Gerarchie di generalizzazione

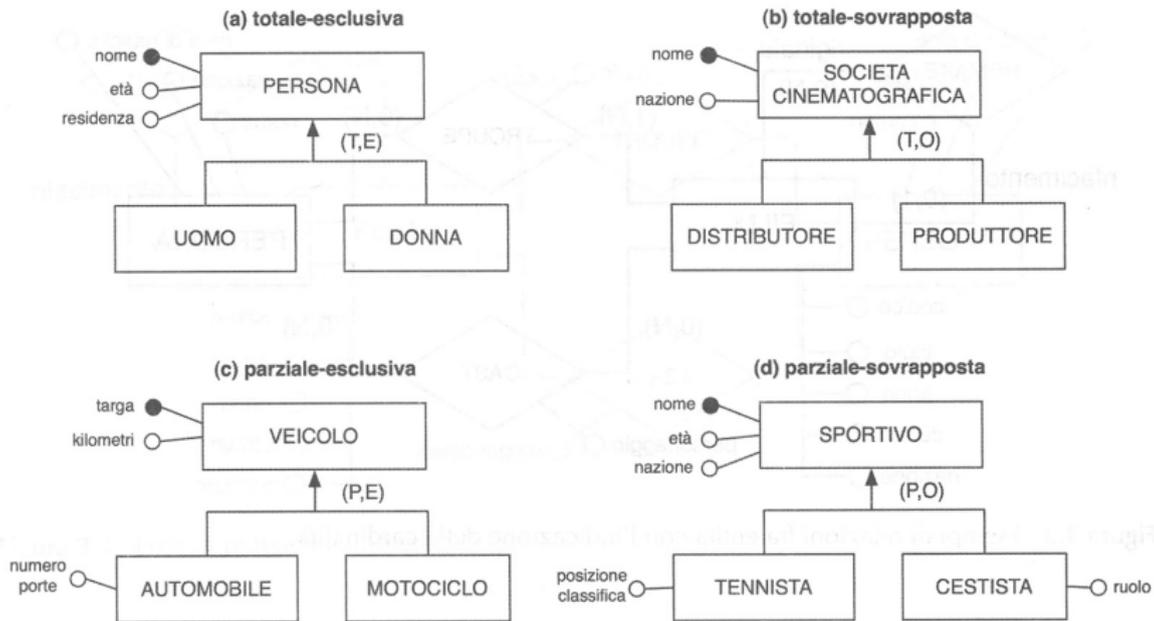


Figura 3.4 Esempi di diverse tipologie di gerarchie di generalizzazione.

Gerarchie di generalizzazione

Spiegazione

- ▶ Le **relazioni** viste sopra rappresentano un *generico* legame fra entità;
- ▶ ma le **gerarchie** rappresentano un legame di *specializzazione* di un'entità rispetto a un'altra.

L'entità piú specifica, detta **figlia**, 'eredita' tutti gli attributi dell'entità **madre** e partecipa a tutte le relazioni cui essa partecipa; il vice-versa non è vero.

Tipologie di gerarchia di generalizzazione

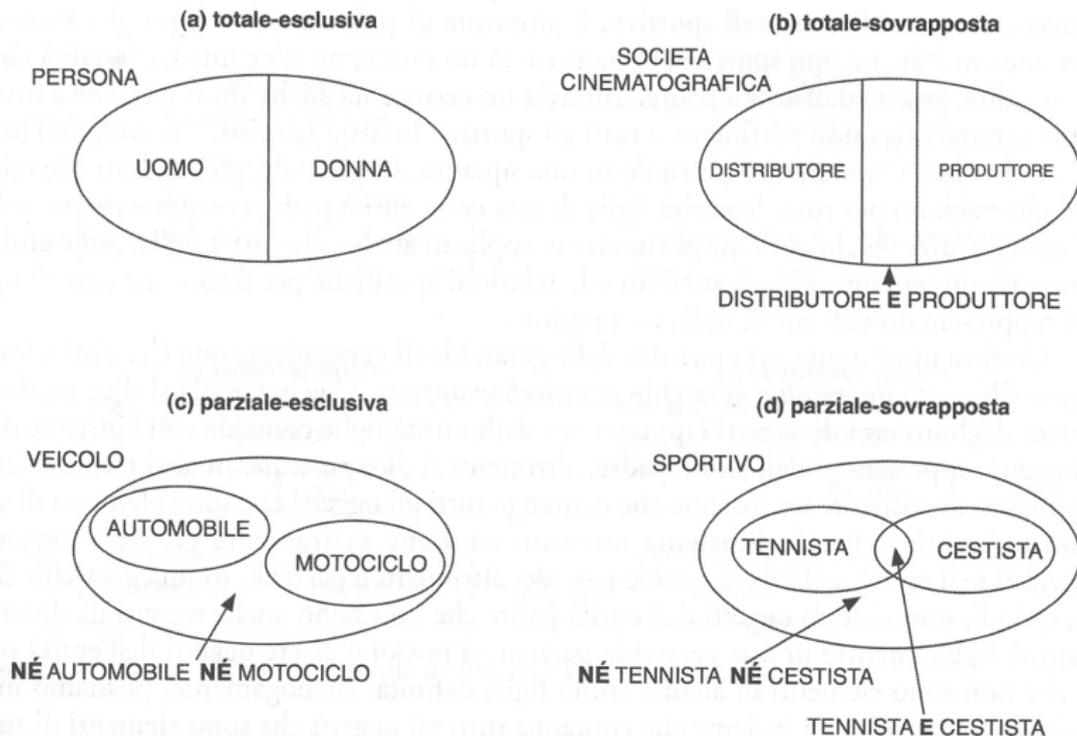


Figura 3.5 Rappresentazione insiemistica delle tipologie di gerarchie di generalizzazione

Esempio di schema ER

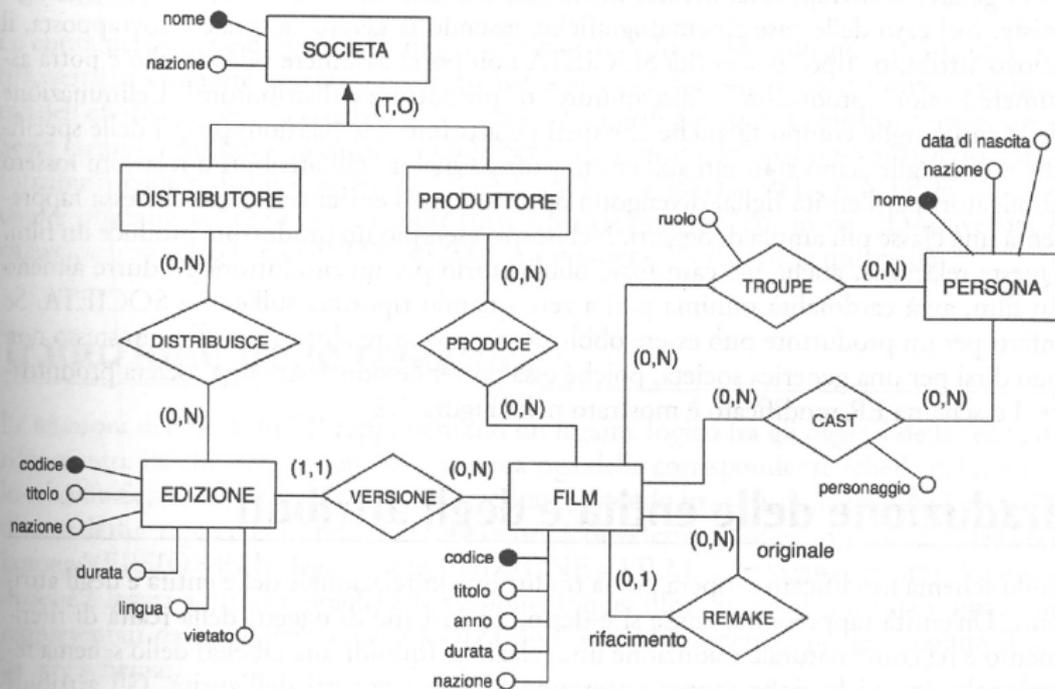


Figura 5.1 Esempio di schema ER.

Riferimenti bibliografici

-  Silvana Castano, Alfio Ferrara, and Stefano Montanelli.
Informazione, conoscenza e web — per le scienze umanistiche.
Pearson / Addison Wesley, 2009.
-  Lawrence Snyder and Alessandro Amoroso.
FLUENCY –Conoscere e usare l'informatica.
Pearson Italia, Milano-Torino, 5^a edition, 2015.