

Progettazione concettuale

corso di basi di dati e laboratorio

Prof. Alfio Ferrara

Anno Accademico 2017/2018

prima fase di progettazione, passa dal mondo reale ad una prima astrazione

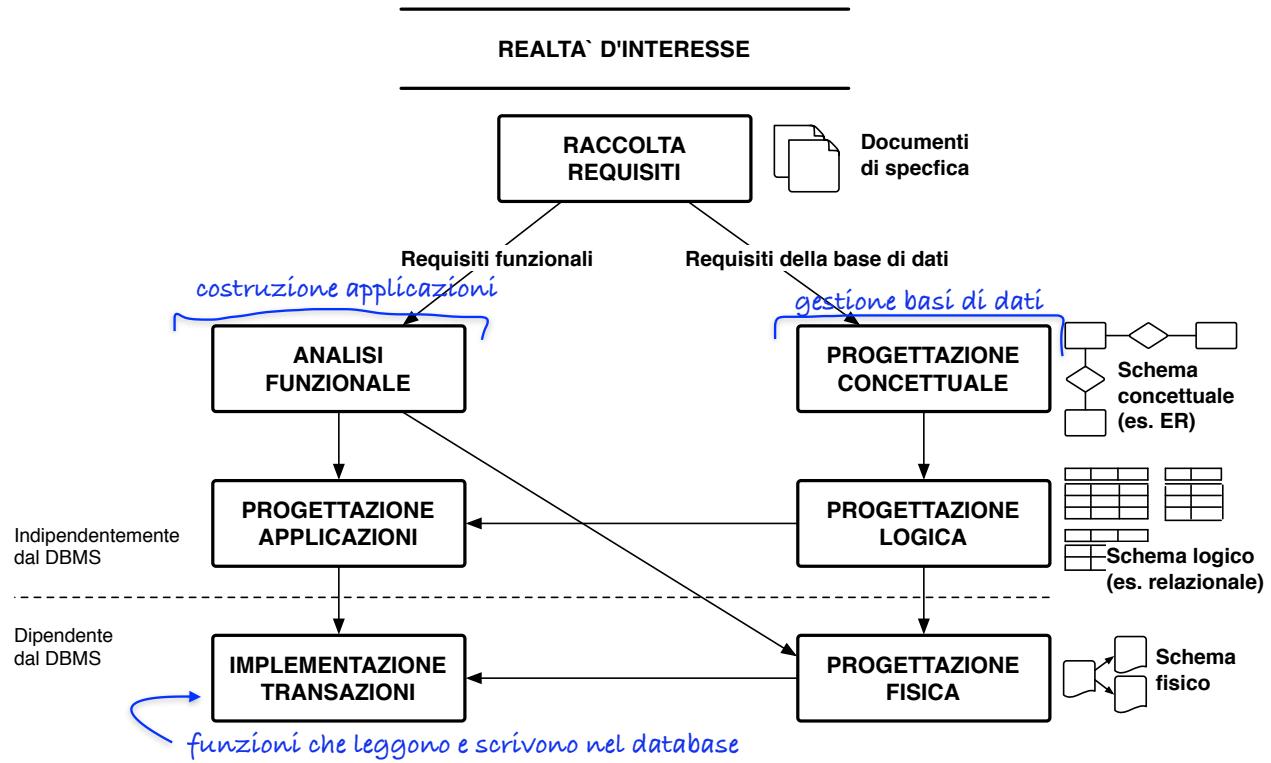
Indice

1	Sistemi di basi di dati	1
1.1	Criteri di progettazione	1
2	Progettazione concettuale	4
2.1	Il modello ER	4
2.2	Entità, relazioni e attributi	5
2.3	Cardinalità	7
2.4	Identifieri	10
2.5	Gerarchie di generalizzazione	11

1 Sistemi di basi di dati

1.1 Criteri di progettazione

L'attività di progettazione



Raccolta e analisi dei requisiti

- Comprensione degli obiettivi della base di dati.
- Interazione con gli utenti che descrivono le necessità, i problemi, gli obiettivi facendo uso del linguaggio naturale.
- Formulazione di descrizioni informali dei dati (requisiti sui dati) e delle operazioni sui dati (requisiti funzionali).

Progettazione della base di dati

si introducono degli errori se si passa direttamente dalle specifiche allo schema

- **Progettazione concettuale** Rappresentare i requisiti informali sulla realtà di interesse in termini di una descrizione concettuale formale e completa (schema concettuale), indipendente dallo specifico DBMS utilizzato per la gestione della BD. non ha un corrispettivo software
- **Progettazione logica** Traduzione dello schema concettuale nel modello dei dati adottato dal DBMS scelto per la gestione della BD. Questa attività è dipendente dallo specifico DBMS utilizzato. costruiamo schema relazionale
- **Progettazione fisica** Lo schema logico viene completato con la specifica dei parametri fisici di memorizzazione dei dati (organizzazione file e indici).

Modelli dei dati → per la progettazione concettuale e logica

- **Modelli Concettuali:** utilizzati nella fase di progettazione concettuale per costruire schemi concettuali, ovvero descrizioni dei dati ad alto livello di astrazione, indipendente dallo specifico DBMS. Es., modello Entity-Relationship (ER). *noi vedremo il modello Er standard*
- **Modelli Logici:** utilizzati nella fase di progettazione logica per ottenere schemi logici, ovvero descrizioni dei dati processabili dal DBMS. Es., modello relazionale.

La nozione di astrazione

- Il criterio fondamentale per la progettazione è l'**astrazione**.
- Si intende per astrazione il processo mentale eseguito quando ci si concentra sulle proprietà / caratteristiche essenziali di un insieme di oggetti, ignorando le differenze (selezione delle proprietà rilevanti di un insieme di oggetti).
- L'obiettivo è la definizione di descrizioni generali di classi di oggetti simili, prescindendo dalle caratteristiche specifiche di ognuno.

Criteri di astrazione → serve per capire le caratteristiche generali degli oggetti

- **Classificazione:** si individua una classe o insieme che possa caratterizzare un gruppo omogeneo di individui specifici. Es., *Italy, Germany, France, ... → Country*.
- **Aggregazione:** si individua un concetto che possa essere definito dall'insieme delle sue parti. Es., *Italy, Germany, France, ... → Europe*. *il volante e l'automobile, perché il volante non è un automobile ma ne fa parte*
- **Generalizzazione:** si individua una classe che possa essere intesa come concetto più generale (e esteso) rispetto a altre classi. Es., *Country, Continent, Region, ... → Territory*. *studente e persona, la persona è una classe più ampia di cui fa parte anche studente*

Interpretazione logica dei meccanismi di astrazione

Tra i meccanismi di astrazione, l'aggregazione corrisponde alla relazione “essere parte di” che non ha un immediato corrispettivo in termini insiemistici, contrariamente a classificazione e generalizzazione:

- **Classificazione** → dato un oggetto o , o è classificato in una classe C se $o \in C$.
- **Generalizzazione** → date due classi C e D , D generalizza C se $C \subseteq D$, ovvero se $\forall x | x \in C \rightarrow x \in D$.

Aggregazione è molto poco usata

Si noti che la definizione stessa di generalizzazione implica che tutte le caratteristiche proprie degli oggetto classificati in D saranno necessariamente proprie anche degli oggetti classificati in C . Si dice pertanto che C “eredita” le caratteristiche di D .

2 Progettazione concettuale

2.1 Il modello ER

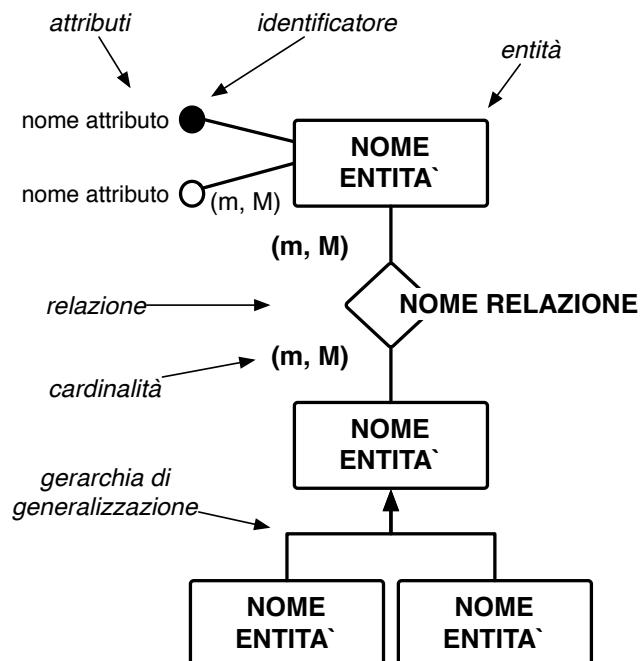
Modello Entity-Relationship (ER) NB “relazione” non vuol dire quello che abbiamo detto fino ad ora

- Il principale modello dei dati usato per la progettazione concettuale è il modello ER (Entity-Relationship)
- Chen, Peter Pin-Shan. “The entity-relationship model—toward a unified view of data.” ACM Transactions on Database Systems (TODS) 1.1 (1976): 9-36.
- Ne esistono anche versioni estese o diverse per notazione, così come esistono anche altre alternative per la progettazione concettuale come UML.

Costrutti

entità = classi di oggetti
relazione = collegamento tra classi di oggetti

- Costrutti principali**
- Entità (Entity)
- Relazione (Relationship) (detta a volte associazione)
- Attributo
- Altri costrutti**
- Cardinalità (relazioni e attributi)
- Identificatori
- Attributi composti
- Gerarchie di generalizzazione



compaiono solo le entità (=classi) e non gli oggetti, la classificazione è automatica
appare “persone” e non Giacomo e Mario

2.2 Entità, relazioni e attributi

Entità

- Rappresenta una classe di oggetti aventi proprietà comuni ed esistenza ‘autonoma’ ai fini dell’applicazione di interesse.
- Oggetti con esistenza fisica (es., Territory)
- Oggetti che esistono a livello concettuale (es., Name)
- Ogni entità all’interno di uno schema ha un nome che la identifica univocamente ed è rappresentato graficamente con un rettangolo

Relazione (o associazione)

- Rappresenta un legame logico tra due o più entità, significativo per l’applicazione di interesse.
- Es., l’associazione *has_name* fra *country* e *name* che rappresenta il fatto che una nazione può avere diversi nomi in lingue diverse.
- Un’associazione R all’interno di uno schema ha un nome che la identifica univocamente.
- L’associazione è rappresentata da un rombo collegato alle entità interessate.

Esempi di relazione

gli oggetti che metto in relazione appartengono
alla stessa classe di oggetti

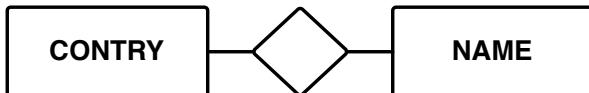
NB non sono lo stesso oggetto, sono oggetti della stessa classe (quello sarebbe RIFLESSIVO)

2.2 Entità, relazioni e attributi

RELAZIONE BINARIA

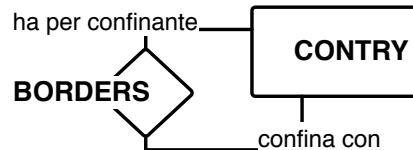
rappresenta un legame che dipende da COUNTRY e NAME

HAS_NAME



RELAZIONI RICORSIVE

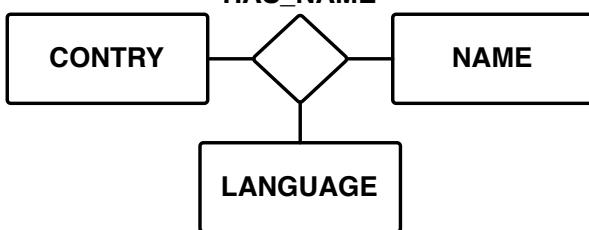
rappresentano legami fra oggetti classificati nella stessa entità



RELAZIONE TERNARIA

rappresenta un legame che dipende dalla tripla
COUNTRY, NAME e LANGUAGE

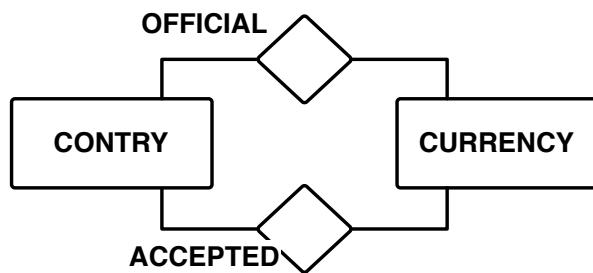
HAS_NAME



es CORSO-ORARIO-AULA

RELAZIONI MULTIPLE

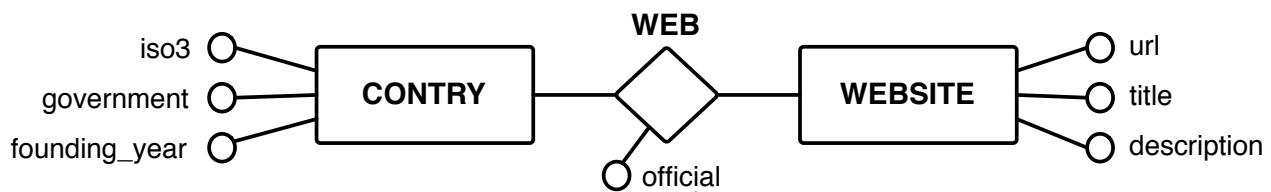
rappresentano legami indipendenti e con diversa
semantica fra le stesse due entità



Attributi se l'oggetto che osservo ha anche altre caratteristiche uso gli attributi

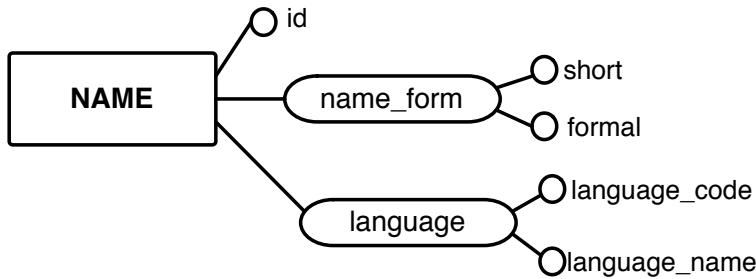
- Descrive una proprietà elementare delle entità e delle associazioni di interesse ai fini dell'applicazione.
- Es., iso3, government, founding_year per l'entità Country.
- Ogni istanza di entità o associazione possiede un valore per ciascuno dei suoi attributi
- Il valore di un attributo appartiene a un dominio dell'attributo che contiene i valori ammessibili.
- Es., government e iso3 hanno dominio stringa, mentre founding_year ha per dominio i numeri interi di 4 cifre.

Esempio di uso degli attributi



Attributi composti

- E' possibile raggruppare attributi aventi affinità d'uso o di significato.
- L'insieme degli attributi ottenuto in questo modo prende il nome di attributo composto.



2.3 Cardinalità

Vincoli di cardinalità

quanti elementi di E_2 corrispondono a E_1

- I vincoli di cardinalità esprimono le regole che devono regolare la partecipazione delle istanze di un'entità a una relazione con altre entità.
- Date due entità E_1 e E_2 e una relazione fra esse R , vi sono due vincoli di cardinalità, uno dal lato di E_1 e l'altro dal lato di E_2
- Il primo vincolo risponde alla domanda: “Data un’istanza (elemento) di E_1 , con quante istanze di E_2 può essere in relazione R come minimo e come massimo?”; il secondo vincolo risponde alla domanda: “Data un’istanza (elemento) di E_2 , con quante istanze di E_1 può essere in relazione R come minimo e come massimo?”
- Ogni vincolo di cardinalità è denotato dall'espressione (mc, MC) , dove mc rappresenta la cardinalità minima, mentre MC la cardinalità massima.

Cardinalità minima

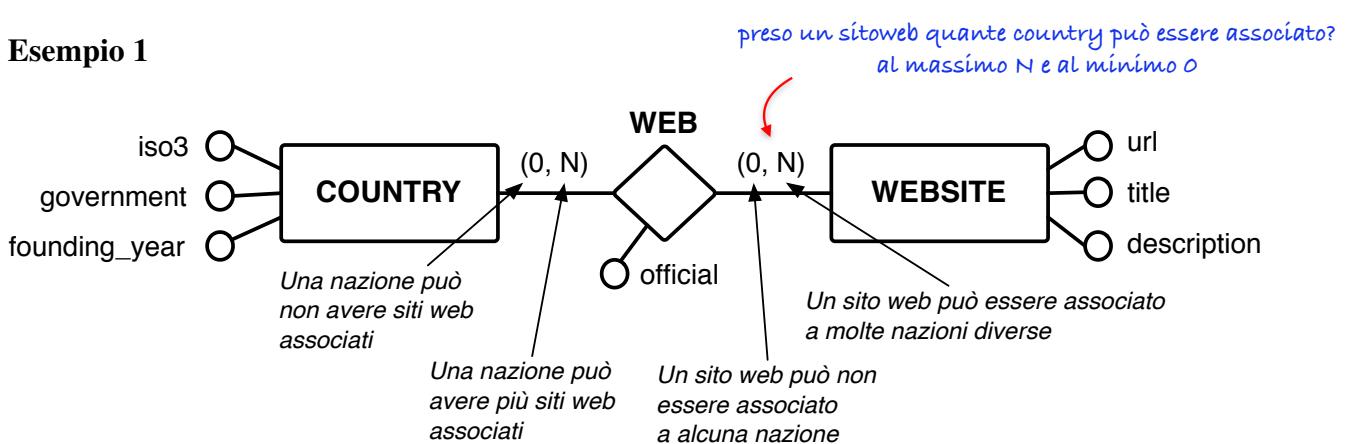
- Un vincolo di cardinalità minima fra un'entità E e una relazione R esprime:
- Il numero minimo di istanze di associazione R a cui un'istanza di E può partecipare. E' detto anche vincolo di partecipazione perché specifica se l'esistenza di un'istanza di entità dipende dal suo essere correlata a un'altra istanza di entità attraverso un'associazione.
- In particolare:

- $mc = 0 \rightarrow$ partecipazione opzionale (o parziale)
- $mc \geq 1 \rightarrow$ partecipazione obbligatoria (o totale), detta anche dipendenza di esistenza

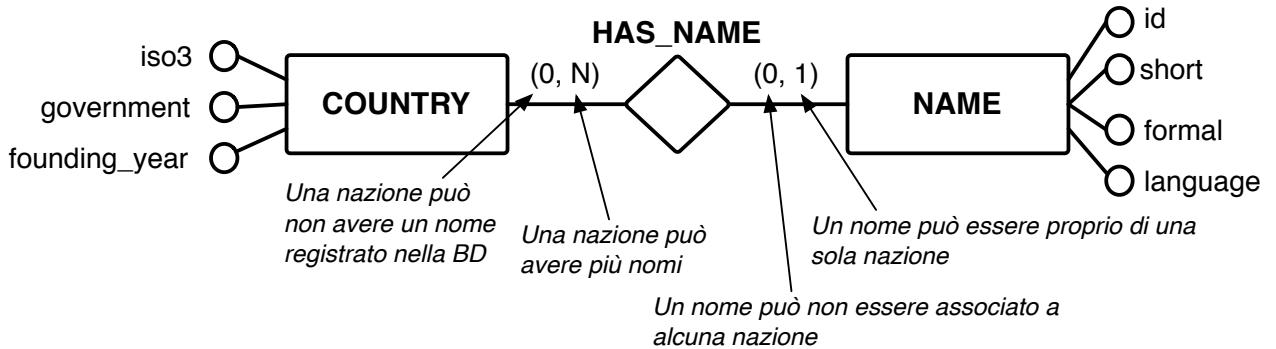
Cardinalità massima

- Numero massimo di istanze dell'associazione R a cui una istanza dell'entità E può partecipare.
- $MC = n \rightarrow$ in generale (nr. qualunque)

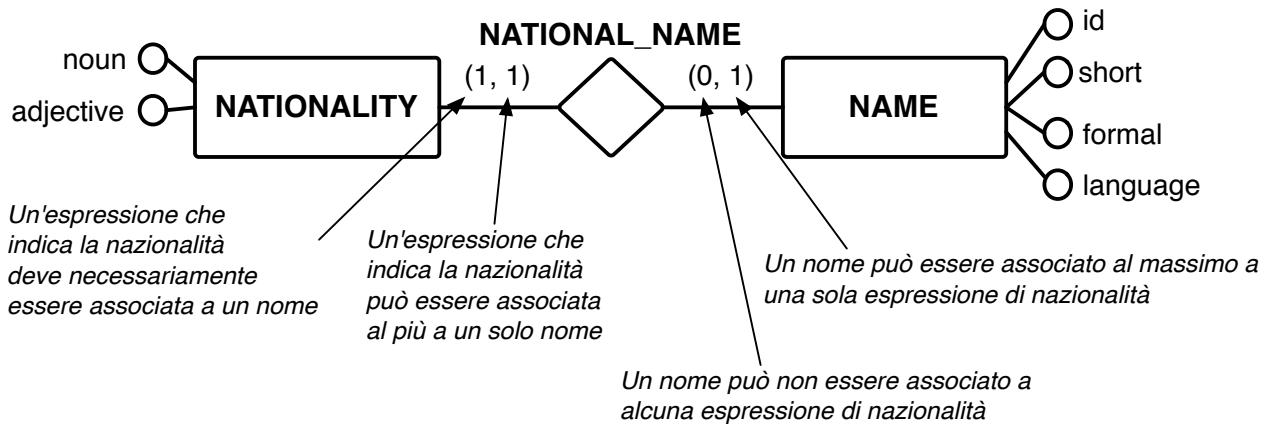
Esempio 1



Esempio 2



Esempio 3



Considerazioni sulle cardinalità minime

- Nella definizione delle cardinalità minime di un'associazione, si noti che la partecipazione obbligatoria di tutte le entità è piuttosto rara.
- Infatti, quando si inserisce una nuova istanza di entità, spesso non sono note (o non esistono) le corrispondenti istanze di entità collegate attraverso l'associazione a cui l'entità partecipa.

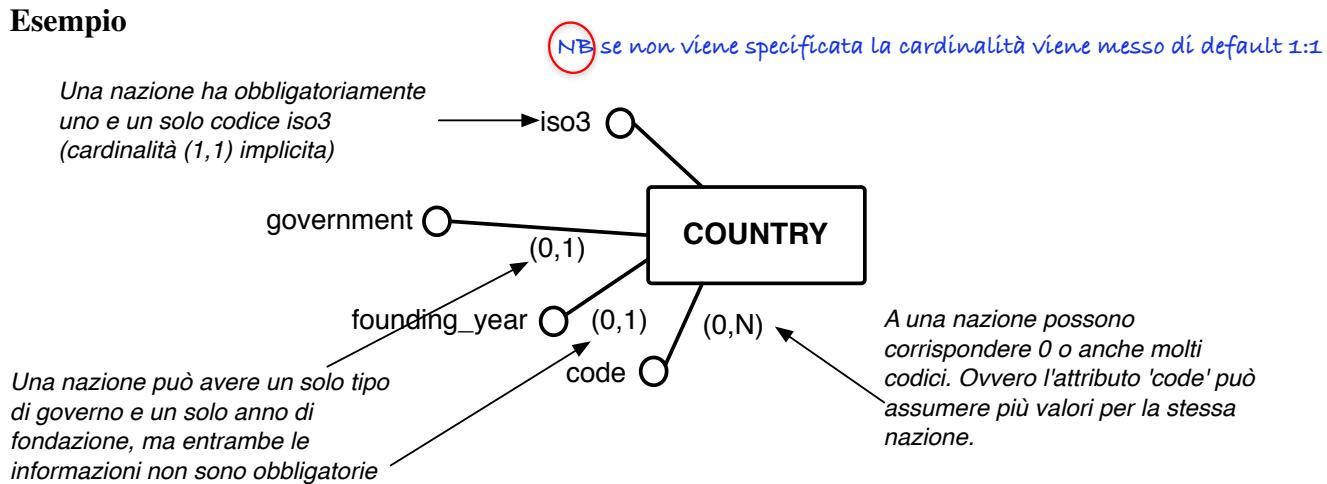
Tipologie di associazioni binarie guardando solo le cardinalità massime

- In base ai valori di cardinalità massima MC si hanno le seguenti tipologie di relazioni binarie:
- 1:1 (Uno-a-uno): $MC = 1$ per entrambe le entità $E1$ e $E2$
- 1:N (Uno-a-molti): $MC = 1$ per un'entità e $MC = N$ per l'altra
- N:M (Molti-a-molti): $MC = N$ per entrambe le entità $E1$ e $E2$

Cardinalità degli attributi

- Descrivono il numero minimo e massimo di valori dell'attributo associati ad ogni istanza di entità o associazione.
- Il valore (1,1) si verifica nella maggioranza dei casi e viene omesso (default).
- Il valore di un attributo può però essere nullo (sconosciuto, non applicabile) cardinalità minima = 0 (attributo opzionale)
- Esistono diversi valori di un attributo associati alla stessa istanza di entità: cardinalità massima = N (attributo multivaleure)

Esempio



2.4 Identificatori

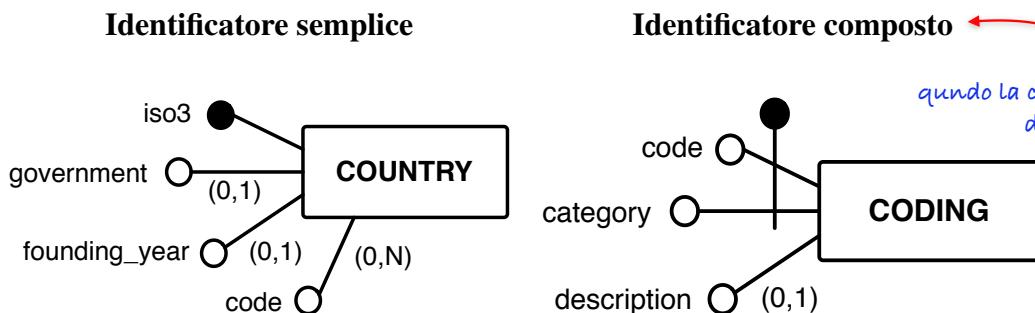
Identificatori

essendo univoco per ogni entità mi consente di identificare quella entità

- Un identificatore di un'entità E è una collezione di attributi e/o entità connesse ad E che permettono di identificare univocamente le istanze di E .
- Vi sono due tipi di identificatori: interno e esterno

Gli identificatori interni

Un identificatore interno è costituito esclusivamente da attributi dell'entità identificata. Può essere semplice (costituito da un solo attributo) o composto (costituito da più attributi).



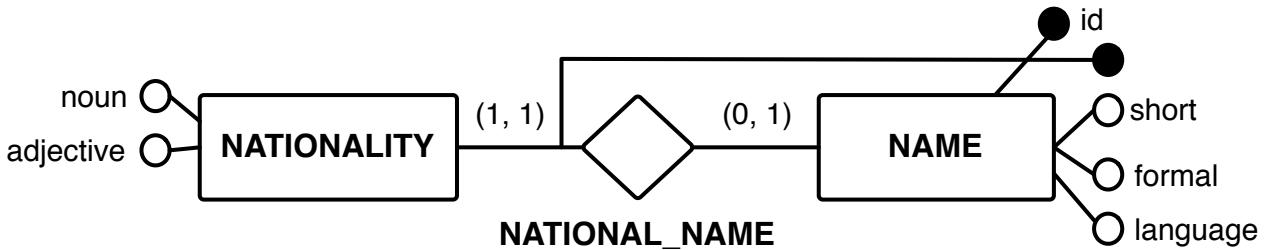
Gli identificatori esterni

- Uno o più attributi di E non sono sufficienti ad individuare un identificatore per E ; in tal caso E è detta entità debole

NB un buon identificatore non deve essere solo univoco, deve essere anche sempre disponibile

posso mettere un identificatore artificiale generato dalla base di dati: ID

- Identificatore Esterno: si utilizzano entità con cui E ha un vincolo di dipendenza (i.e., si considerano tipi di associazioni binarie a cui E partecipa con cardinalità (1,1)).



2.5 Gerarchie di generalizzazione

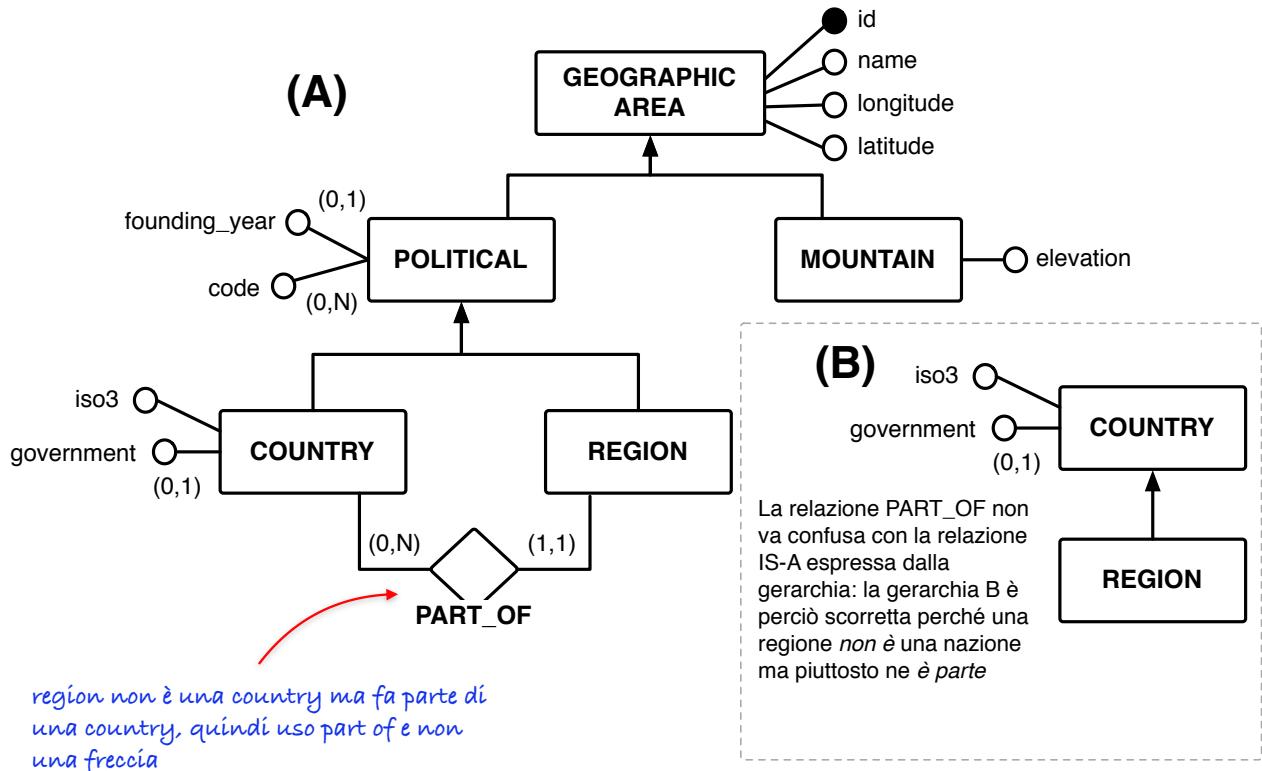
Gerarchia di generalizzazione è un particolare tipo di relazione tra più entità

- Una gerarchia di generalizzazione è un particolare tipo di relazione che intercorre fra un'entità E e altre entità E_1, E_2, \dots, E_n .
- E è una superclasse di E_1, E_2, \dots, E_n , se ogni istanza di E_1, E_2, \dots, E_n è anche istanza di E .
- Si dice che E_1, E_2, \dots, E_n sono sottoclassi di E .
- Si dice che E è *padre* di E_1, E_2, \dots, E_n e che E_1, E_2, \dots, E_n sono *figlie* di E .

Ereditarietà

- Dalla definizione di gerarchia di generalizzazione consegue il meccanismo di ereditarietà da superclasse a sottoclasse:
- Ogni proprietà di una superclasse E (attributi, identificatori, associazioni) è ereditata da tutte le sottoclassi di E , ovvero esse sono caratterizzate da tutte le caratteristiche che appartengono a E .
- Le proprietà eritate non vanno rappresentate esplicitamente.

Esempio



Vincoli sulle specializzazioni

- **Totalità:** una gerarchia è totale se ogni entità della superclasse è istanza di almeno una sottoclassificazione; altrimenti è parziale.
- **Disgiunzione:** una gerarchia è esclusiva (o disgiunta) se un'istanza di entità può essere istanza al più di una delle sottoclassificazioni della specializzazione; altrimenti è sovrapposta.
- Si definiscono perciò le seguenti tipologie di gerarchia:

 - Totale-Esclusiva (TE)
 - Totale-Sovrapposta (TO)
 - Parziale-Esclusiva (PE)
 - Parziale-Sovrapposta (PO)

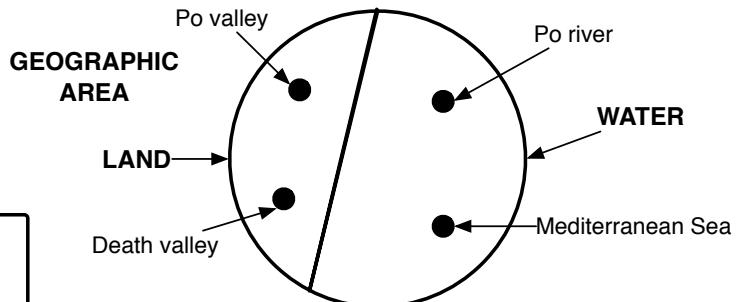
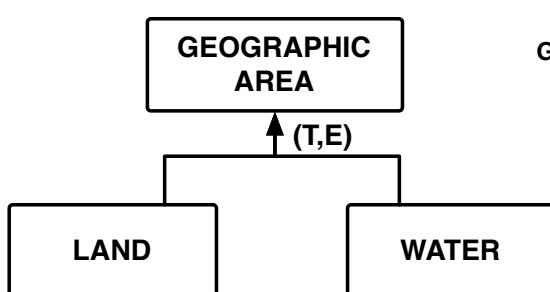
Interpretazione insiemistica

Data una gerarchia di generalizzazione fra un'entità padre E e le entità figlie E_1, E_2, \dots, E_n :

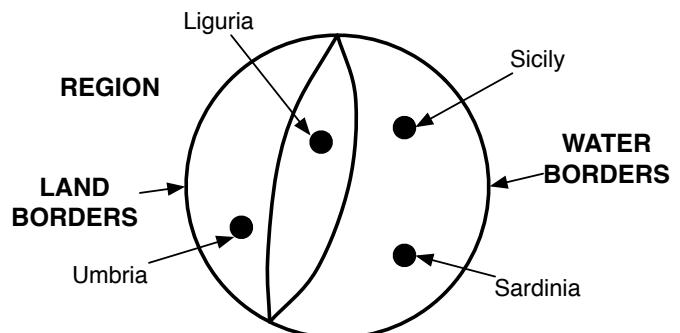
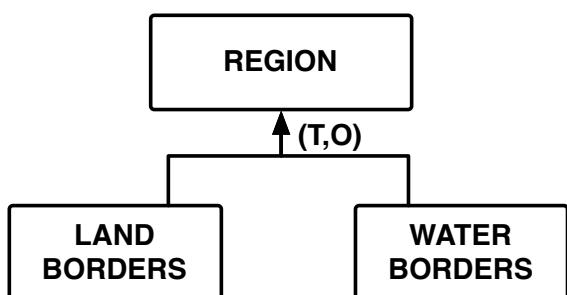
$$\text{La gerarchia è totale sse } \bigcup_{i=1}^n E_i \equiv E$$

La gerarchia è esclusiva sse $\bigcap_{i=1}^n E_i \equiv \emptyset$

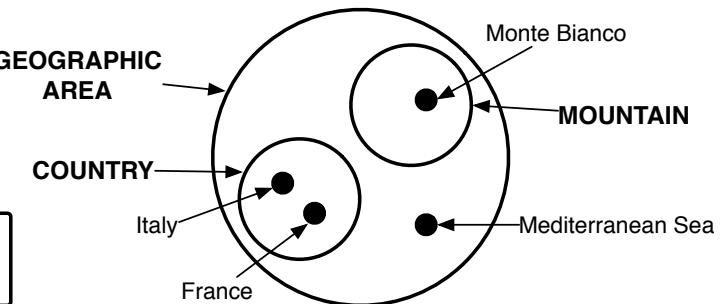
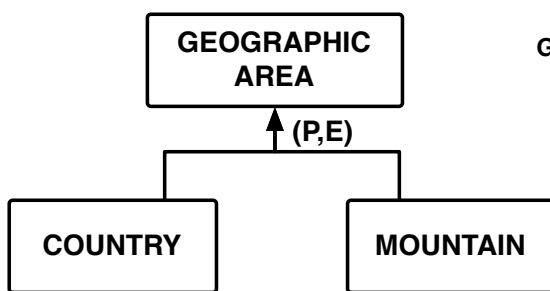
Esempio: totale e esclusiva



Esempio: totale e sovrapposta



Esempio: parziale e esclusiva



Esempio: parziale e sovrapposta

2.5 Gerarchie di generalizzazione

