

- ▶ **Database e GeoDatabase**
- ▶ **Tipologie di Database**
- ▶ **Le tabelle e gli attributi**
- ▶ **Relazioni e campi chiave**
- ▶ **Le relazioni Join e Relate**
- ▶ **Architetture di Geodatabase**

# Il database

Con il progredire della tecnologia GIS sono divenuti sempre più marcati i problemi relativi

- all'accesso multiutente dello stesso dataset (sol. duplicazione dati ⇒ accessi separati ⇒ problemi di allineamento degli aggiornamenti su tutte le copie)
- dimensione territoriale dell'area da analizzare e densità di informazioni tali da richiedere l'utilizzo di file di grandi dimensioni e numerosità (sol. suddivisione in file più piccoli ricoprenti aree più ristrette ⇒ navigazione difficoltosa, rallentamento ricerca file richiesti ⇒ funzioni analisi spaziale limitate dal frazionamento dell'area, ecc.)

**Soluzione:** strutturare i dati in un **database** gestito da appositi software che ne garantiscano l'efficienza in termini di sicurezza, di accessi e di prestazioni.

Anni '60 : necessità di gestire grandi volumi di dati, in rete, con molti accessi in contemporanea, con applicazioni differenti.

Alcune aziende iniziano ad investire per sviluppare la tecnologia dei database e dei relativi sistemi di gestione (**DBMS – Data Base Management System**).

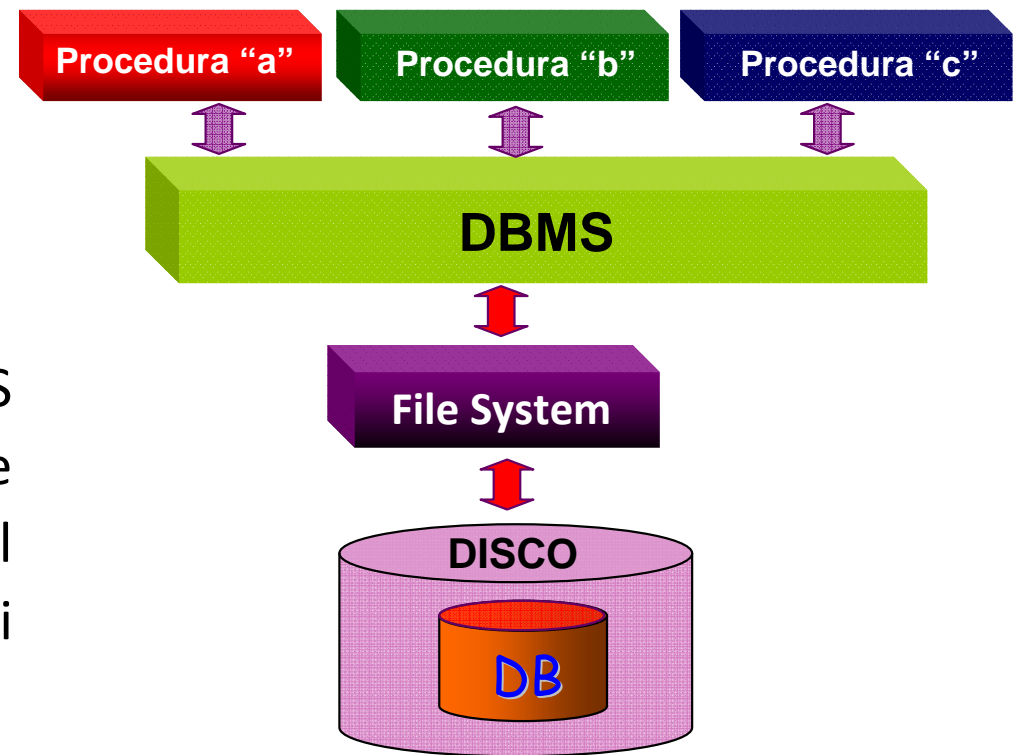
Database:

- **collezione di dati** di diversa tipologia,
- strettamente correlati tra loro,
- archiviati su una memoria di massa (Hard Disk),
- secondo una precisa organizzazione logica,
- manipolati da specifici programmi applicativi (**DBMS**).

Le funzioni principali di un DBMS sono **quattro**:

- inserimento/aggiornamento dati,
- consultazione/estrazione,
- modifica,
- cancellazione.

Dal punto di vista informatico il DBMS si frappone fra l'utente ed i dati veri e propri ed utilizza il file system del Sistema Operativo per manipolare i files di dati.



In ambito GIS i dati spaziali e aspaziali sono inseriti in un **database geografico** o **GeoDatabase**, definibile come un **archivio di entità territoriali e delle loro relazioni**, strutturato in file organizzati da un sistema che ne garantisce la gestione efficiente e l'accesso da molte applicazioni ed utenti.

# Vantaggi di un DBMS

---

- **Controllo centralizzato:** controllo delle restrizioni, sicurezza, accessi, conflitti, integrità dei dati e aggiornamenti.
- **Indipendenza dalla struttura dei dati:** utenti e applicativi non necessitano di conoscere come sono fisicamente strutturati ed archiviati i dati perché accedono al database attraverso il DBMS; ne “vedono” solamente una rappresentazione logica.
- **Condivisione dei dati:** controllo e permesso di accesso al database da parte di più utenti (ed applicazioni) in contemporanea, anche sulle medesime aree geografiche e quindi sugli stessi dati.
- **Riduzione delle ridondanze:** la condivisione di un unico database evita inutili duplicazioni dei dati e quindi la possibilità di effettuare modifiche e aggiornamenti diversi sugli stessi dati (disallineamento dei dati).

- **Accesso diretto ai dati:** vari tipi di utenti (non informatici, informatici, specialisti applicativi) possono accedere ai dati ed eseguire qualsiasi tipo di analisi senza compromettere l'integrità dei dati, con interfacce semplici ed intuitive, senza dover conoscere la struttura logica degli archivi.
- **Efficienza:** accesso multiutente a grandi volumi di dati.
- **Sicurezza:** possibilità di definire i profili degli utenti e le porzioni del database a cui possono accedere (accesso controllato per estrarre o inserire nuovi dati, apportare aggiornamenti, modificare la struttura per migliorare le prestazioni, inserire nuove entità e relazioni).
- **Linguaggi standard:** le applicazioni si interfacciano con un DBMS tramite linguaggi standard di interrogazione del database tipo **SQL** (*Structured Query Language*) che garantiscono una modalità di dialogo standard fra il sistema ed i programmi, indipendentemente dal software GIS utilizzato.

# Tipologie di database

---

In generale nei database i dati sono archiviati secondo dei precisi **criteri** che definiscono

- l'**organizzazione logica dei dati**, cioè le modalità secondo cui le diverse tipologie di informazioni archiviate vengono poste in relazione tra loro,
- la **modalità di gestione**, ovvero il grado di complessità delle operazioni di consultazione, estrazione, modifica, cancellazione e inserimento dei dati.

In base all'organizzazione logica dei dati, si possono distinguere tre diversi modelli di database:

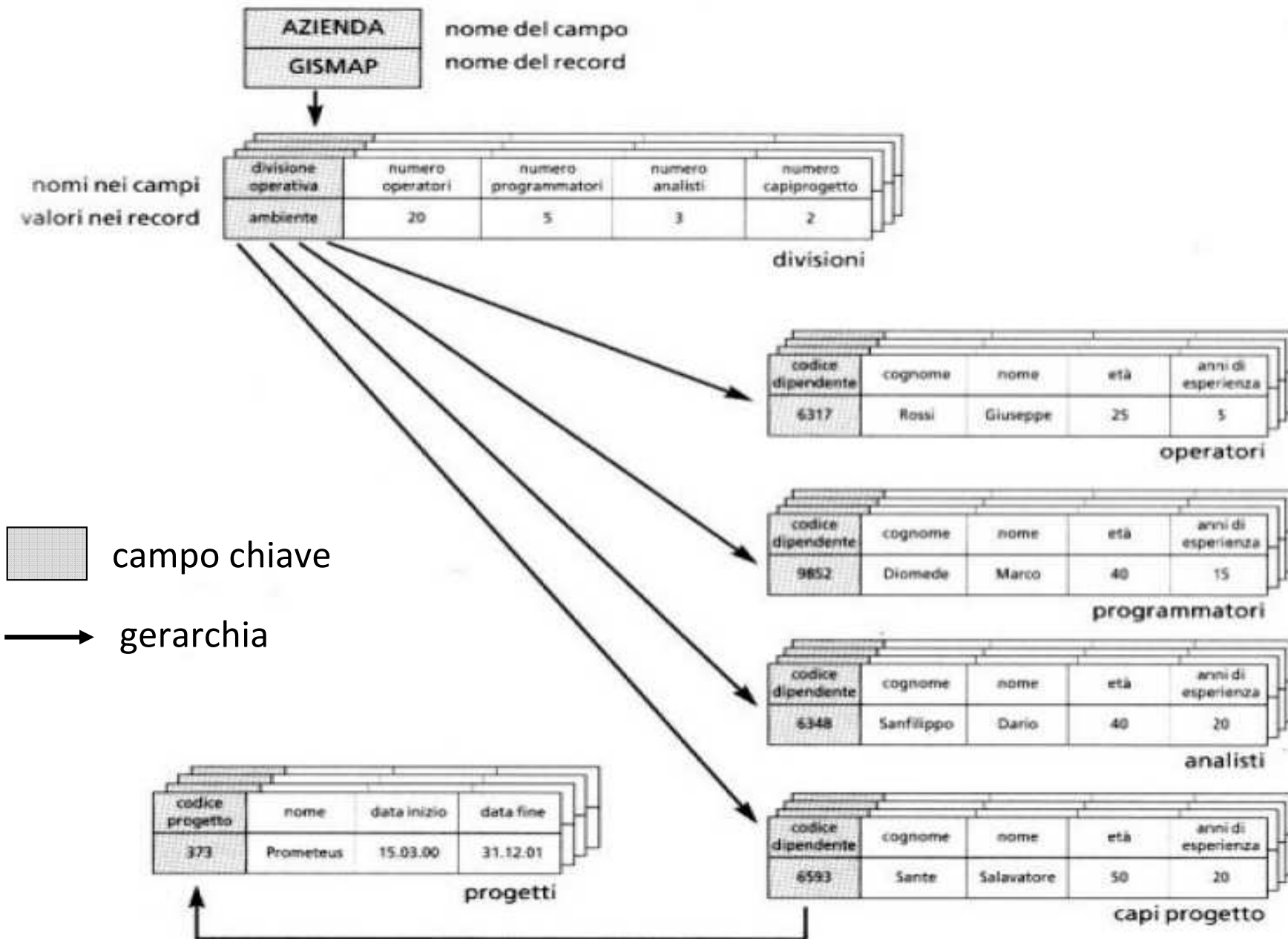
- **Gerarchico**
- **Reticolare**
- **Relazionale**

# Il database gerarchico

- Dati organizzati secondo uno schema ad **albero rovesciato** basato sulla **relazione *padre-figlio***.
- Livello capostipite (*radice*): rappresenta il record principale (*nodo padre*) al quale sono collegati tutti gli altri.
- Relazioni di tipo **gerarchico**: ogni record è relazionato ad un solo elemento immediatamente superiore (*nodo padre*), mentre può essere collegato con più record posti a livello inferiore (*nodi figli*).
  - Possibili solo relazioni 1:N (**uno a molti**): un padre ha più figli, ma ogni figlio ha un solo padre.
  - Non ci sono collegamenti tra record dello stesso livello (cioè tra *nodi figli* di *nodi padre* diversi).



# Esempio di database gerarchico



- Ogni record ha un **campo chiave** che serve ad organizzare la gerarchia, cioè a connettere un nodo *padre* ad un nodo *figlio*.
- L'interrogazione del database avviene lungo i rami della gerarchia, tramite i campi chiave, solamente nella **direzione che va dall'alto in basso** (cioè si può interrogare il database andando da un nodo *padre* ad un nodo *figlio* e non viceversa).

### Esempio azienda GISMAP

- a) Ricerca dei capi progetto, analisti, programmatori ed operatori di una specifica divisione molto efficiente  $\Rightarrow$  esiste una relazione diretta tra le divisioni ed il personale afferente.*
- b) Non è possibile conoscere a quale progetto lavora un'analista, perché nello schema proposto il collegamento tra i record degli analisti e quello dei progetti non è stato previsto !*

*Bisogna crearlo esplicitamente, ricostruendo ex-novo tutto il database.*

*c) Ricerca dei progetti che afferiscono ad una specifica divisione*

*⇒ procedura a due fasi:*

- Selezione di tutti i capi progetto che afferiscono alla divisione,*
- Selezione di tutti i progetti che afferiscono ai capi progetto precedentemente selezionati.*

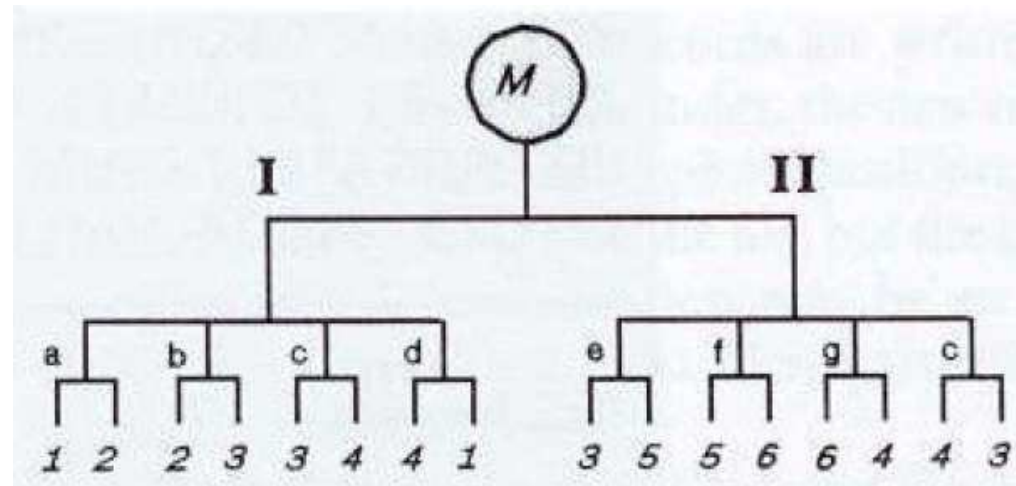
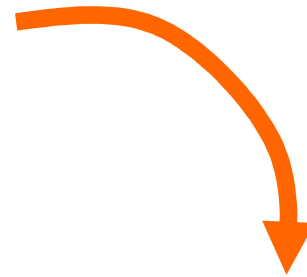
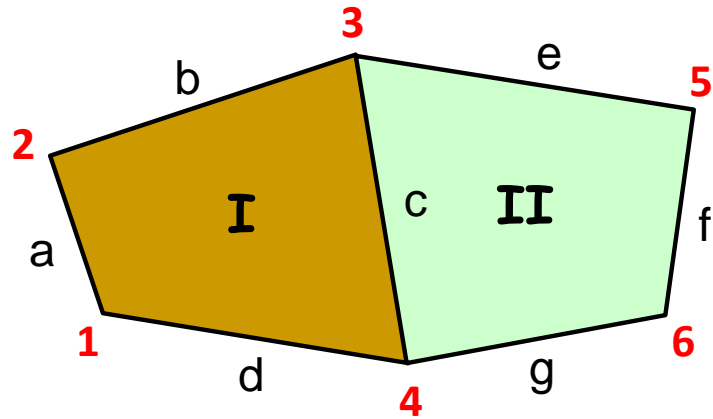
*Processo meno efficiente ⇒ è necessario operare una selezione su un'entità intermedia, l'operazione può diventare ancora più onerosa se le entità intermedie sono più di una.*

## Svantaggi:

- **Scarsa flessibilità** della struttura gerarchica: bisogna modificare le relazioni e quindi ricostruire il database quando si vuole effettuare ricerche che
  - coinvolgono collegamenti, tra campi chiave, diversi da quelli per cui è stata creata la struttura stessa.
  - coinvolgono collegamenti tra campi NON chiave.
- **Ridondanze:** la struttura gerarchica comporta la duplicazione di alcuni record (es. programmatore che lavora su progetti di settori diversi dell'azienda GISMAP  $\Rightarrow$  il suo record va duplicato).
- Struttura ottimizzata per rispondere a determinate richieste che devono essere conosciute **PRIMA** della progettazione del database.

Un GIS richiede una grande flessibilità, perché è praticamente impossibile stabilire a priori tutte le richieste  $\Rightarrow$  **il modello gerarchico non è applicato.**

# Esempio di database gerarchico per geometrie



mappa

poligoni

linee

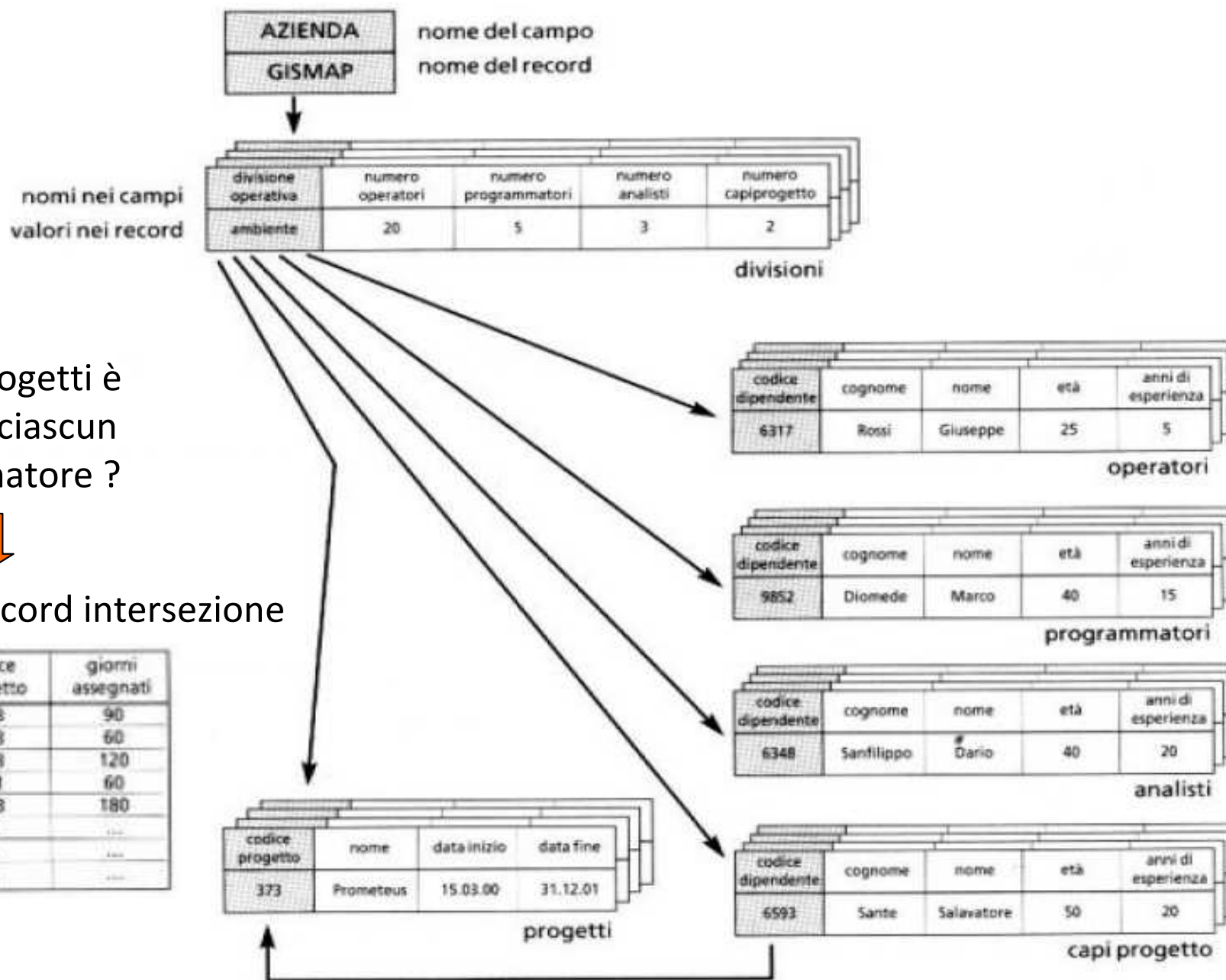
punti

Struttura logica di tipo gerarchico delle due forme poligonali

# Il database reticolare

- Un *nodo figlio* può essere collegato anche a più elementi di livello superiore (*nodi padre*).
- Eliminazione degli elementi duplicati: non è più necessario ripetere le informazioni (record aventi gli stessi valori su nodi diversi).
- Modifica dei dati semplice in quanto ogni dato è presente una volta sola.
- Ricerca dati (**query**) più veloci (più collegamenti tra i nodi).
- E' possibile stabilire relazioni **1:N** ed anche **1:1**.
- **Complessità dei collegamenti tra i nodi:** cancellazione e inserimento dati sono operazioni faticose poichè richiedono la modifica di tutti i nodi collegati a quello da cancellare o inserire.
  - ⇒ Struttura poco flessibile: **non è usata nei GIS.**

# Esempio di database reticolare

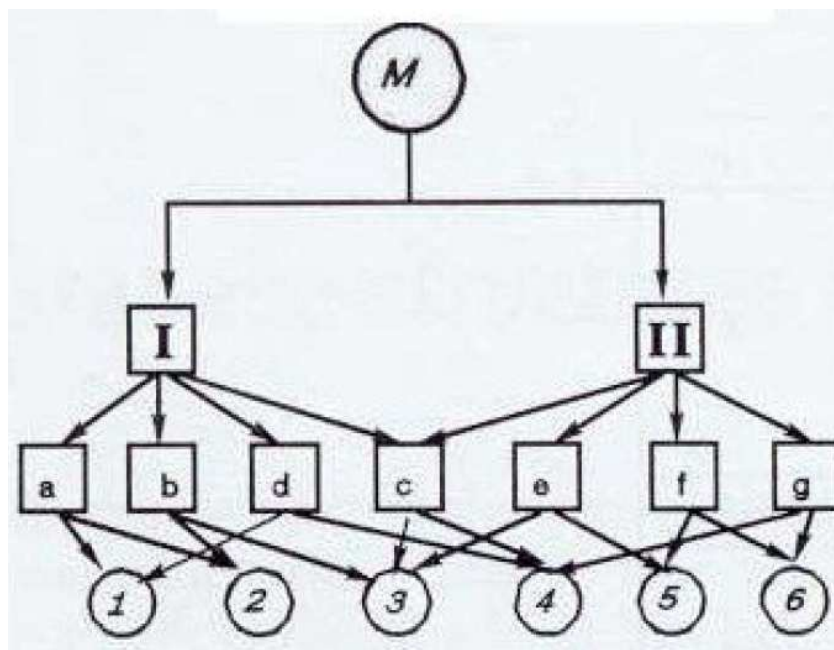
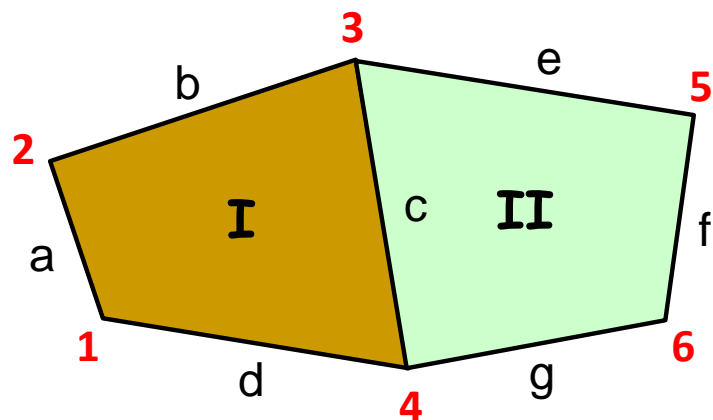


In quali progetti è coinvolto ciascun programmatore ?



Creare un record intersezione

## Esempio di database reticolare per geometrie



mappa

poligoni

linee

punti

Struttura logica di tipo reticolare delle due forme poligonali



# Il database relazionale

Sviluppato negli anni '70 dal matematico della IBM *Edgard Frank Codd*, raggruppa in sé le seguenti caratteristiche:

- si basa sul concetto matematico di **relazione tra insiemi**;
- usa una terminologia che si ispira alla matematica;
- i dati sono organizzati in una **struttura logica rigorosa ma flessibile**;
- permette di eseguire in modo semplice operazioni di inserimento e ricerca dati e di stabilire vincoli di integrità (vincoli su modifica/validità del dato).

Questo tipo di database si fonda sui concetti di

- **Tabella**
- **Record**
- **Campo**
- **Chiave**

## Definizioni

### **Tabella** (Relazione):

- organizzazione “rettangolare” dei dati in righe e colonne.
- ogni tabella, in genere, rappresenta una raccolta di informazioni su uno specifico argomento o tematica.

### **Record** (tupla):

- singola **riga** della tabella.
- consente di identificare un determinato insieme di dati relativi ad una data entità, all’interno di tutti quelli che sono contenuti nella tabella.

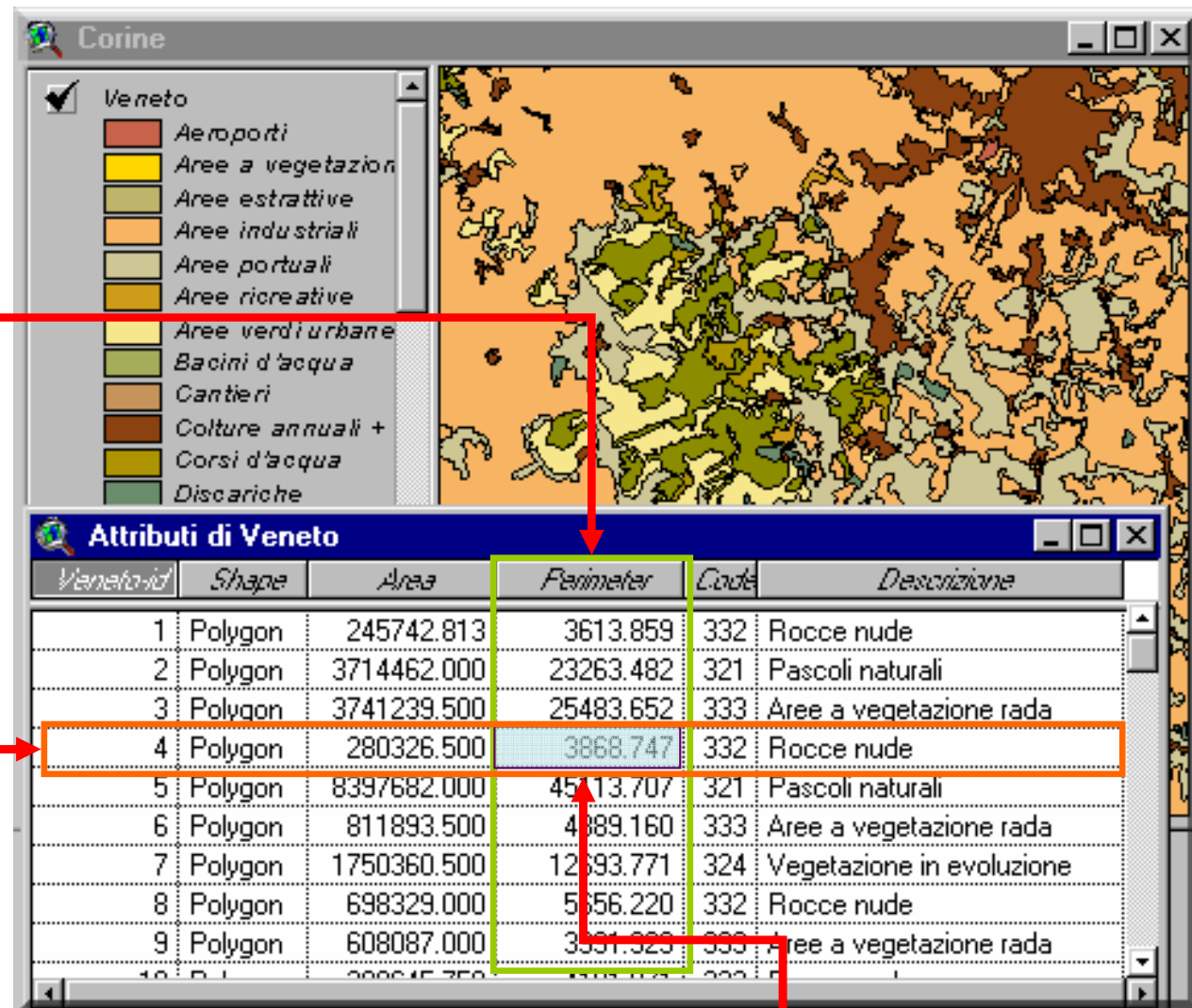
### **Campo**:

- **colonna** della tabella, specifica un attributo delle entità rappresentate.
- si può specificare il tipo di ciascun campo (testo, numeri, interi e decimali ecc.) e si può imporre una certa serie di regole.

### **Chiave** (chiave primaria):

- attributo (**campo**) o combinazione di attributi che identificano in modo univoco ed inequivocabile un singolo record all’interno della tabella. 18

# La Tabella



**campo**

(contiene i valori di un determinato attributo per tutte le entità)

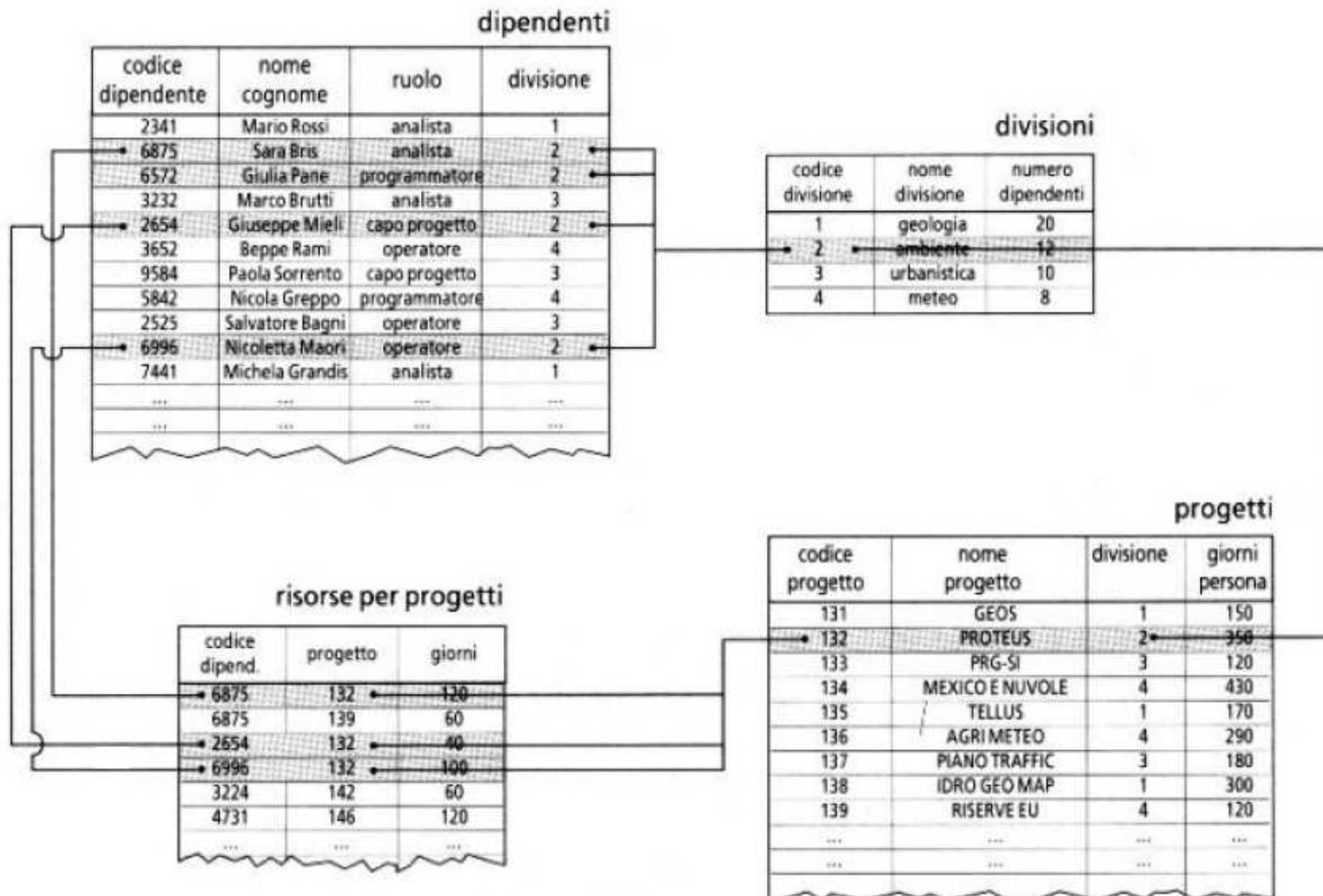
**record**

(Contiene i valori di tutti gli attributi riferiti ad una singola entità geografica)

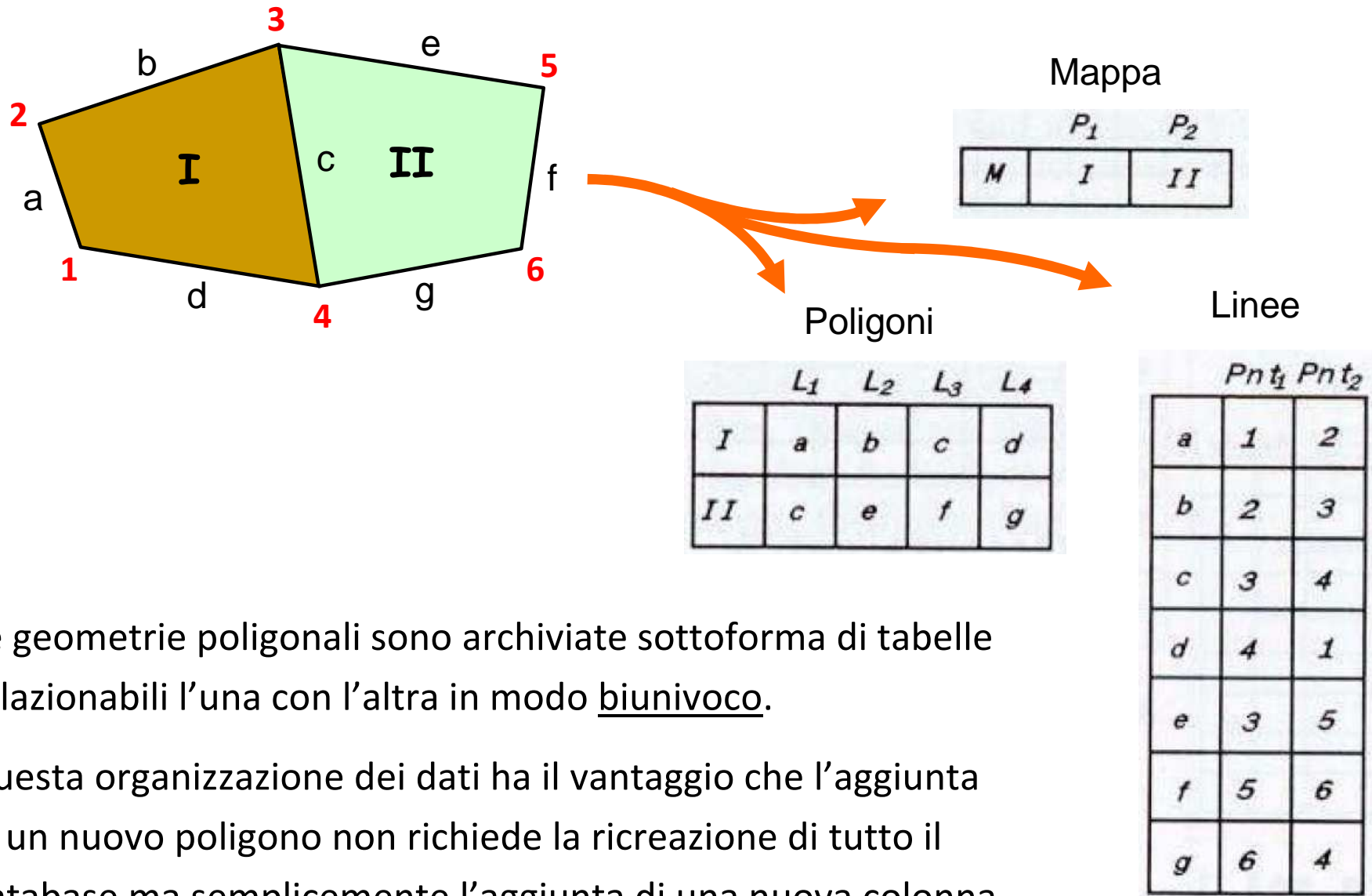
(Contiene il valore di un attributo di un'entità, quella associata al record)

**cella**

# Esempio di database relazionale



## Esempio di database relazionale per geometrie



Le geometrie poligonali sono archiviate sottoforma di tabelle relazionabili l'una con l'altra in modo biunivoco.

Questa organizzazione dei dati ha il vantaggio che l'aggiunta di un nuovo poligono non richiede la ricreazione di tutto il database ma semplicemente l'aggiunta di una nuova colonna o record ove necessario.

## Proprietà della tabella

- **No record duplicati:** non esistono due righe identiche tra loro, in questo modo è possibile identificare univocamente ogni record.

**NO !**

ID	Nome	Cognome	Telefono
101	Mario	Draghi	2544
102	Paolo	Tarolli	2576
103	Giorgio	De Sanctis	2688
104	Paolo	Tarolli	2576

- **Struttura dati “rettangolare”:** ogni riga (record) della tabella ha lo stesso numero di colonne (campi), corrispondente agli attributi.

**NO !**

ID	Nome	Cognome	Telefono	Dipartimento
101	Mario	Draghi	2544	DAFNAE
102	Paolo	Selleri	2576	TESAF
103	Giorgio	De Sanctis	2688	TESAF

CIRGEO
--------

- **Campi omogenei:** le colonne devono contenere per ciascun attributo valori dello stesso tipo (es. testo, numeri interi, numeri decimali, booleani, ecc.).

**NO !**

ID	Nome	Cognome	Telefono
101	Mario	Draghi	049-272644
102	Paolo	Tarolli	2576
103	Giorgio	De Sanctis	Ignoto
102	Paolo	Tarolli	2576

- **Un valore per cella:** ogni cella della tabella può contenere uno ed un solo valore dell'attributo.

**NO !**

ID	Nome	Cognome	Telefono
101	Mario	Draghi	2544
102	Paolo	Tarolli	2576 332887753
103	Giorgio	De Sanctis	2688

- **ID univoco dei campi:** ogni colonna ha un nome unico che la distingue dalle altre.

**NO !**

<b>ID</b>	<b>Nome</b>	<b>Cognome</b>	<b>ID</b>
101	Mario	Draghi	2544
102	Paolo	Tarolli	2576
103	Giorgio	De Sanctis	2688
104	Carlo	Selleri	2688

- **Molteplicità delle chiavi:** una tabella può avere uno o più attributi utilizzabili come campi chiave per stabilire delle relazioni con altre tabelle.
- **Univocità della chiave primaria:** due record distinti non possono avere lo stesso valore dell'attributo (campo) chiave primaria.
- **No campi chiave con valore indefinito** (= NULL, è associato a celle prive di un qualsiasi valore dell'attributo).



- **Ordine di record e campi modificabile:** è possibile cambiare l'ordine delle righe e delle colonne senza modificare il significato della tabella.

### Ordinamento dei record

ID	Nome	Cognome	Telefono
101	Mario	Draghi	2544
102	Paolo	Tarolli	2576
103	Giorgio	De Sanctis	2688
104	Carlo	Selleri	2424

=

ID	Nome	Cognome	Telefono
101	Mario	Draghi	2544
104	Carlo	Selleri	2424
103	Giorgio	De Sanctis	2688
102	Paolo	Tarolli	2576

### Ordinamento dei campi

ID	Nome	Cognome	Telefono
101	Mario	Draghi	2544
102	Paolo	Tarolli	2576
103	Giorgio	De Sanctis	2688
104	Carlo	Selleri	2424

=

ID	Cognome	Nome	Telefono
101	Draghi	Mario	2544
102	Tarolli	Paolo	2576
103	De Sanctis	Giorgio	2688
104	Selleri	Carlo	2424



- **Atomicità dei valori:** ogni cella deve contenere un valore dell'attributo non ulteriormente suddivisibile (*valore elementare*).

## Esempi

Come organizzare in una struttura relazionale l'insieme di attributi contenuti nelle seguenti affermazioni ?

1. *La particella catastale N. 12089, di proprietà del sig. Guido Rossi, ha una superficie di 3500 m<sup>2</sup>.*



ID particella	Nome proprietario	Cognome proprietario	Area particella
12089	Guido	Rossi	3500

Tipo dato      *intero*                      *testo*                      *testo*                      *decimale*

2. *Il sig. Guido Rossi, età 65 anni, pensionato, abita a Venezia al terzo piano di uno stabile sito in via Pellestrina 14, in un appartamento di 90 m<sup>2</sup>.*



Nome	Cognome	Età	Stato civile	Comune	Tipo strada	Indirizzo	N° civico	Sup. app.	N° piano
Guido	Rossi	65	Pens.	Venezia	via	Pellestrina	14	90	3

*testo      testo      intero      testo      testo      testo      testo      intero      decimale      intero*

3. *Il terreno con particella catastale 894 sito in località Vigodarzere, provincia di Padova, di proprietà dei sig.ri Luciano e Paolo Carraro è coltivato a ceduo e fustaia.*



?

## Relazioni e Campi chiave

---

Ad ogni entità del mondo reale è possibile associare delle caratteristiche o “**proprietà**” che ne descrivono determinati aspetti (qualitativi e/o quantitativi).

Esempio:

*In una biblioteca ogni libro può essere definito da*

- titolo,*
- nome dell'autore,*
- codice identificativo di archiviazione (ID libro).*

ID libro	Titolo libro	Nome Autore	Cognome Autore
----------	--------------	-------------	----------------

Alcune di tali "proprietà" possono essere **condivise** anche tra oggetti di tipo diverso.

Esempio:

*In una biblioteca ad ogni utente può essere assegnato l'ID progressivo di prestito di un libro e tale ID può essere poi replicato nella scheda associata al libro medesimo.*

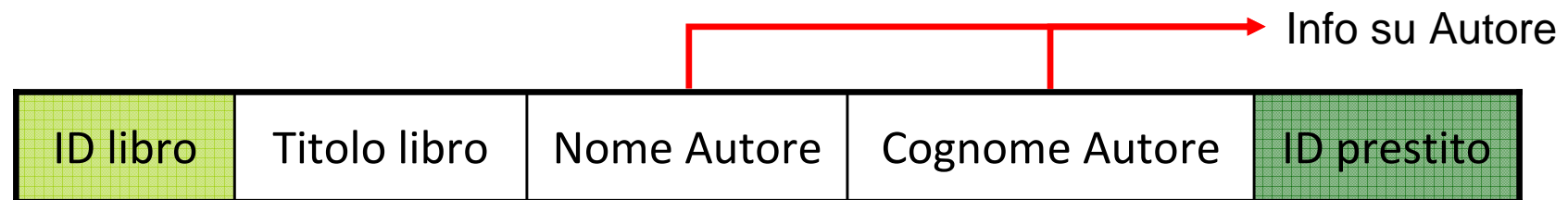
Nome Cliente	Cognome Cliente	C.F. cliente	ID prestito
--------------	-----------------	--------------	-------------

ID libro	Titolo libro	Nome Autore	Cognome Autore	ID prestito
----------	--------------	-------------	----------------	-------------

Tale **condivisione** genera una **relazione**, ovvero un legame tra due oggetti distinti, tramite la quale è possibile accrescere il livello di informazione iniziale.

Esempio:

*l'ID di prestito consente di reperire informazioni associate all'utente e/o all'autore del libro, interrogando differenti database ove siano presenti tali dati correlati tra loro da ulteriori relazioni.*



Info su Cliente

In un database relazionale una relazione si stabilisce

- **sempre e solo tra due tabelle,**
- tramite **valori corrispondenti** di un campo comune (attributo con gli stessi valori nelle due tabelle)  $\Rightarrow$  **campo chiave** (*Key field*).

In particolare, in una relazione tra due tabelle, A (*principale o destinazione*) e B (*esterna o sorgente*), si distinguono due tipi di chiavi:

- **chiave primaria**, definita da una o più colonne di A che ne identificano **univocamente** i record. I valori in esse contenute NON possono essere duplicati e di tipo *Null*.
- **chiave esterna**, colonna di B i cui valori corrispondono esclusivamente ai valori della chiave primaria di A. I valori che compaiono in tale campo chiave possono essere duplicati.

## Esempio di chiave primaria

### Professori

ID	Nome	Cognome	Telefono	Dipartimento
101	Mario	Draghi	2544	Biologia
102	Paolo	Selleri	2576	TESAF
103	Giorgio	De Sanctis	2688	TESAF
104	Carlo	Selleri	2424	Selvicoltura

I campi ID e Telefono sono chiavi primarie in quanto entrambi identificano **in modo univoco** i singoli record.

I campi Nome e Cognome **non sono chiavi primarie** in quanto potrebbero contenere, in futuro, un valore già esistente.

I campi Nome, Cognome e Dipartimento possono essere usati come **chiave esterna**.



# Esempio di relazione

**Professori**

ID	Nome	Cognome	Telefono	Dipartimento
101	Mario	Draghi	2544	Agronomia
102	Paolo	Tarolli	2576	Biotecnologie Agr.
103	Giorgio	De Sanctis	2688	TESAF
104	Carlo	Selleri	2424	Scienze Animali

*Chiave esterna*



**Tabella esterna**  
(sorgente)

**Dipartimenti**

ID_Dip	Facoltà	Personale	Dipartimento
24	Agraria	56	Agronomia
12	Agraria	86	TESAF
8	Veterinaria	36	Scienze Animali
11	Agraria	34	Biotecnologie Agr.

*Chiave primaria della  
tabella esterna*

**Tabella principale**  
(destinazione)

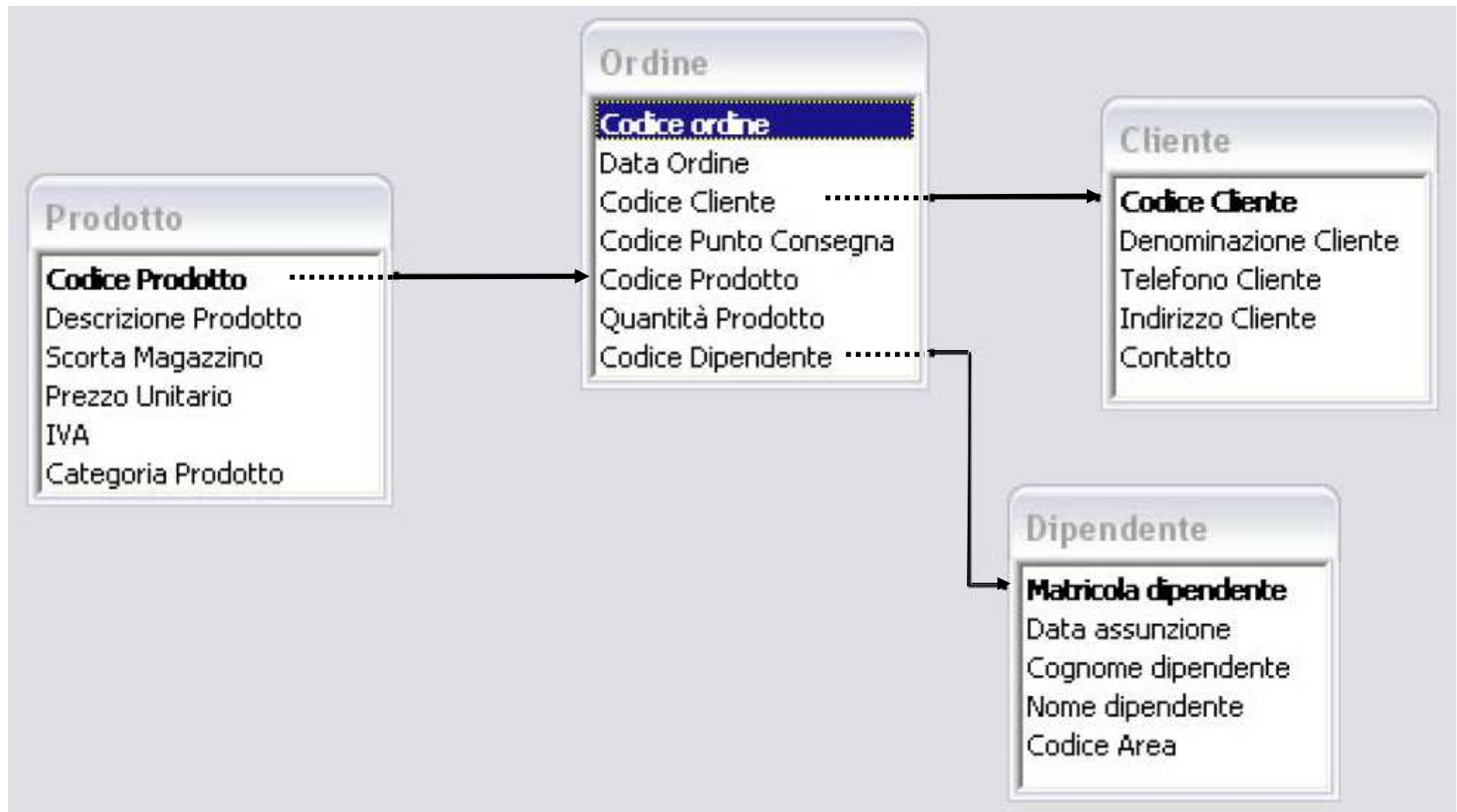
*Chiave primaria della tabella principale*

## Proprietà dei campi chiave

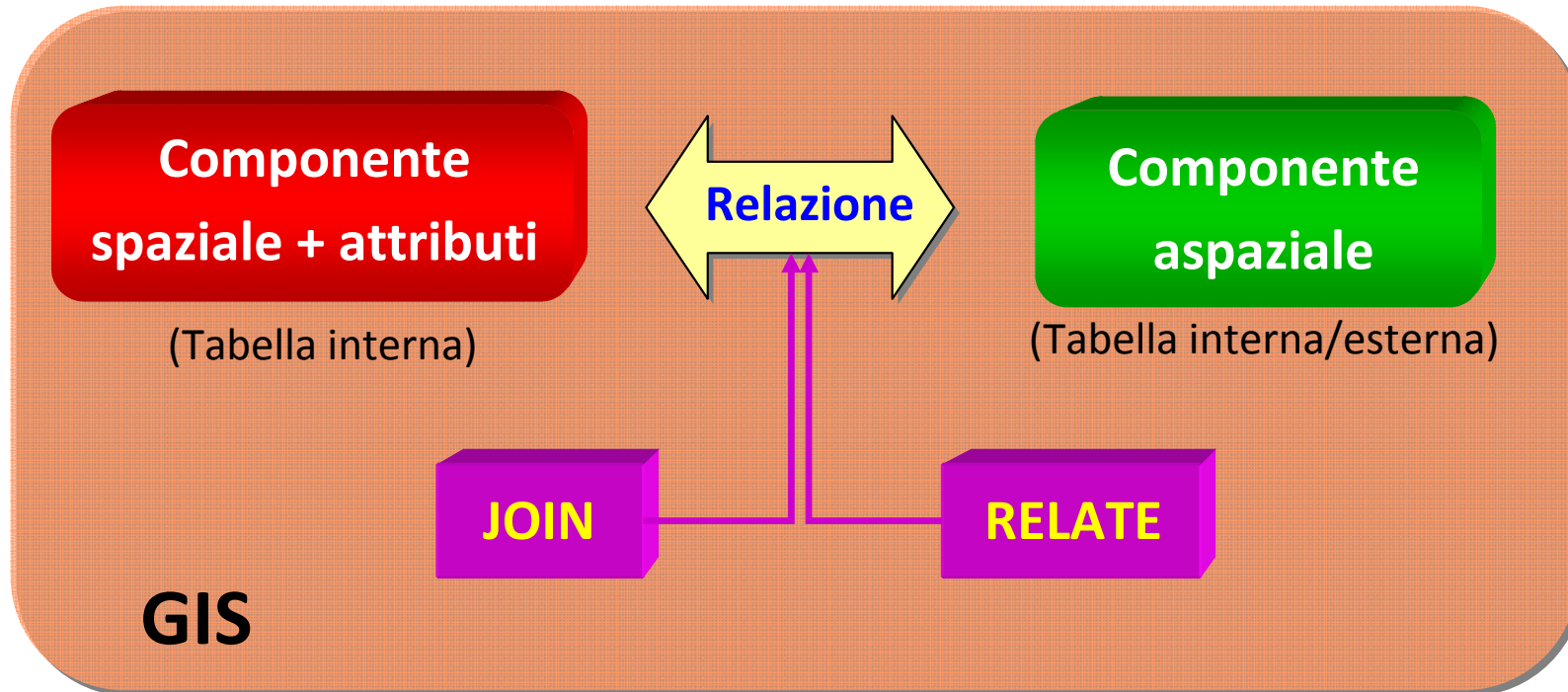
---

- Una chiave primaria può essere utilizzata come chiave esterna in una relazione ma **non viceversa**, poiché una chiave esterna non identifica necessariamente in modo univoco i record della tabella (ovvero alcuni valori possono ripetersi).
- Le informazioni contenute in due relazioni diverse vengono correlate attraverso i **valori comuni dei campi chiave**.
- Il nome dei campi chiave primaria ed esterna possono essere diversi, l'importante è che essi contengano lo stesso tipo di valori, ovvero che siano **coerenti** tra loro (es. testo-testo, decimali-decimali, ecc.).

## Esempio di database relazionale



# Join & Relate



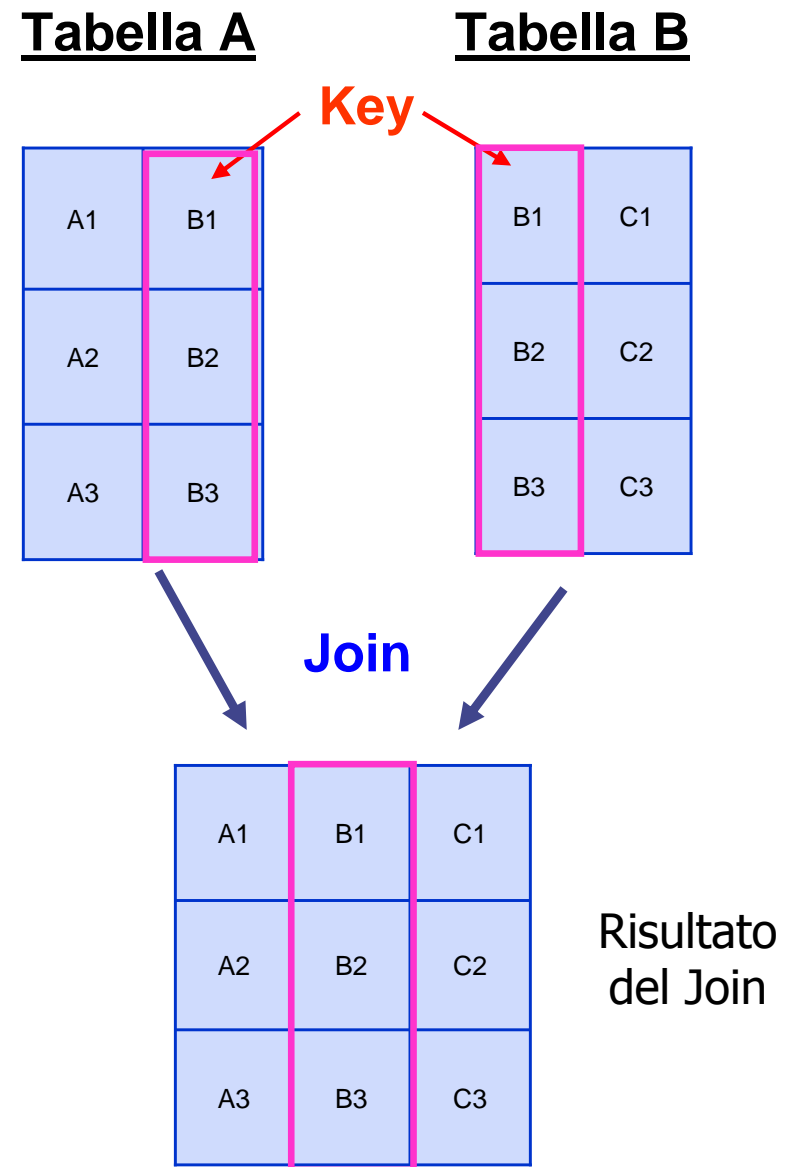
Sfruttando la possibilità di relazionare attributi provenienti da tabelle differenti, l'utente è in grado di eseguire interrogazioni del database (**queries**) di varia complessità, aumentando la quantità di informazioni rispetto a quelle inizialmente disponibili.

## Join:

L'operazione di join collega la tabella degli attributi del dataset A ad una tabella esterna B.

Il collegamento è solo **virtuale**: i files originali non vengono alterati e rimangono separati.

Il risultato è una **nuova** tabella composta da tutti i possibili records che rappresentano una combinazione di due records, uno da ognuna delle tabelle specificate.



## Confronto tra Join e Relate

- Quando si effettua una operazione di **JOIN** tra due tabelle, si “**agganciano**” gli attributi della prima agli attributi della seconda basandosi su un campo comune alle due tabelle.
  - ⇒ relazioni più utili perché riescono a **espandere** il contenuto informativo originale del dataset e lo rendono disponibile per l’analisi e la visualizzazione.
- Quando si effettua una operazione di **RELATE** tra due tabelle si definisce una relazione tra di esse sempre basata su un campo comune, ma **non si agganciano** gli attributi dell’una all’altra.
  - ⇒ relazioni meno potenti di quelle di Join, sono utilizzate soprattutto come metodo alternativo di interrogazione; è infatti sempre possibile visualizzare i dati relazionati quando necessario.

# Tipologie di relazioni tra tabelle

In un database relazionale, tra due tabelle A (**sorgente**) e B (**destinazione**), si possono stabilire quattro diverse tipologie di relazioni:

*Sorgente*  $\Rightarrow$  destinazione

**UNO - A - UNO**

**UNO - A - MOLTI**

Relazioni di  
tipo **JOIN**

**MOLTI - A - UNO**

**MOLTI - A - MOLTI**

Relazioni di  
tipo **RELATE**

## Relazione Uno-a-Uno

Ad un record della tabella sorgente A corrisponde **uno ed un solo** record della tabella destinazione B e **viceversa**.

Possono esistere record di A oppure di B che non hanno corrispondenti nella relazione (le due tabelle non devono avere necessariamente lo stesso numero di righe e colonne).

Non può esistere alcun record, né di A né di B, che compaia due volte, abbinato ad entità diverse.



**Tabella sorgente  
(A)**

A1
A2
A3
A4
A5

↑  
(Solo attributi)

**Tabella destinazione  
(B)**

B1
B2
B3
B4
B5

↑  
(Geometrie & attributi)



**Tabella risultante  
(C)**

B1	A1
B2	A2
B3	A3
B4	A5
B5	A4

*NB: il colore delle celle sta ad indicare quali sono i record delle due tabelle che si corrispondono in modo uno-a-uno.*

**Esempio:**

A = Tabella dei Capoluoghi ⇔ B = Tabella delle Province

(Ad ogni provincia corrisponde UNO E UN SOLO capoluogo e viceversa.

Tra A e B si può pertanto instaurare solamente una relazione Uno-a-Uno)

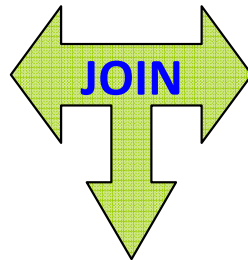
# Esempio di JOIN *Uno-a-Uno*

Tabella **destinazione**  
(geometrie & attributi)

Tabella **sorgente** (solo attributi)

Shape	A_nome	C_supha	D_pop
Polygon	VERONA	309713.00	838000
Polygon	VICENZA	272429.43	807000
Polygon	BELLUNO	367412.83	210500
Polygon	TREVISO	247977.24	808000
Polygon	VENEZIA	247061.92	813000
Polygon	PADOVA	214306.55	858000
Polygon	ROVIGO	181814.55	243500

Provincia	00-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59
VERONA	15475	39151	47006	62499	69871	71324	60304	54661	55494	49698
VICENZA	17738	38089	45933	61480	69992	70491	58397	51440	52400	47772
BELLUNO	25930	9165	10845	14719	16507	16643	14935	14287	15247	13330
TREVISO	11663	37662	45183	61642	69296	70534	59998	53499	52629	47613
VENEZIA	97886	33835	43019	61487	67707	69890	60607	54602	57291	54115
PADOVA	14149	38812	47880	65282	73633	76281	64423	57086	57020	51623
ROVIGO	27008	10969	13854	17582	18452	19689	18548	17685	17348	14000



Chiave primaria

Chiave esterna

Tabella risultante

Dati "accodati"

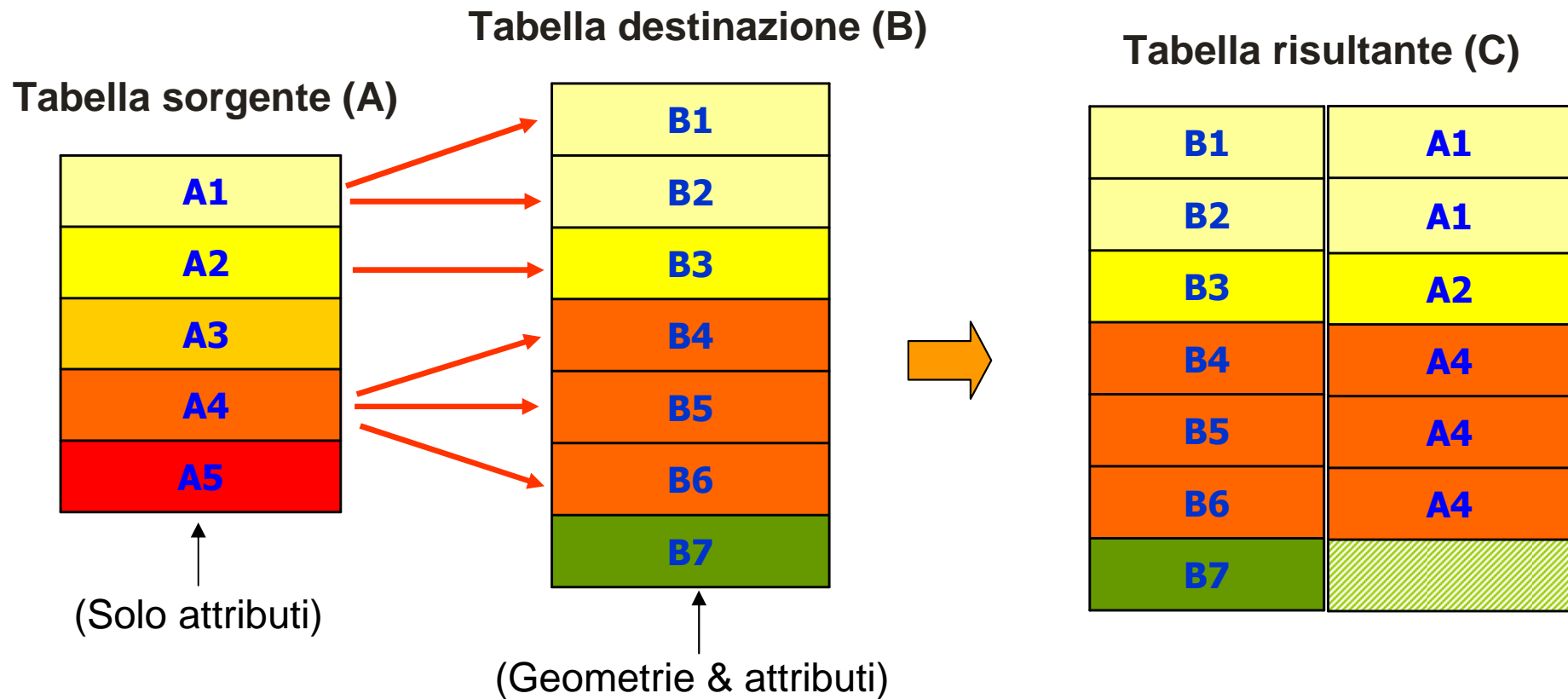
Shape	A_nome	C_supha	D_pop	00-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59	60-64	65-69	Totale
Polygon	VERONA	309713.00	838000	115475	39151	47006	62499	69871	71324	60304	54661	55494	49698	50105	150994	826582
Polygon	VICENZA	272429.43	807000	117738	38089	45933	61480	69992	70491	58397	51440	52400	47772	47148	133437	794317
Polygon	BELLUNO	367412.83	210500	25930	9165	10845	14719	16507	16643	14935	14287	15247	13330	13564	44378	209550
Polygon	TREVISO	247977.24	808000	111663	37662	45183	61642	69296	70534	59998	53499	52629	47613	46842	138703	795264
Polygon	VENEZIA	247061.92	813000	97886	33835	43019	61487	67707	69890	60607	54602	57291	54115	54140	155007	809586
Polygon	PADOVA	214306.55	858000	114149	38812	47880	65282	73633	76281	64423	57086	57020	51623	51419	152249	849857
Polygon	ROVIGO	181814.55	243500	27008	10969	13854	17582	18452	19689	18548	17685	17348	14000	14739	52664	242538

## Relazione Uno-a-Molti

---

Ad **un record** della tabella sorgente A possono corrispondere **più record** della tabella destinazione B.

Il **viceversa non è possibile**: uno stesso record della tabella destinazione B non può corrispondere a record distinti della tabella sorgente A.



**Esempio:**

A = Tabella delle Province ⇔ B = Tabella dei Comuni

(Ogni comune appartiene ad UNA E UNA SOLA provincia; una MEDESIMA provincia può però apparire in più comuni. Tra A e B si instaura quindi una relazione Uno-a-Molti)

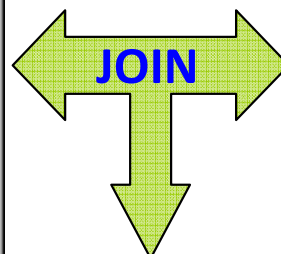
# Esempio di JOIN *Uno-a-Molti*

Tabella **destinazione**  
(geometrie & attributi)

Tabella **sorgente** (solo attributi)

Shape	Id	Gr_idrolog	Area	Perimeter
Polygon	5	A	10323.980	1082.824
Polygon	6	A	2162357.010	12435.100
Polygon	7	A	77076.618	2032.559
Polygon	8	A	148008.351	4696.349
Polygon	18	B	144232.288	1838.674
Polygon	19	B	1865253.097	6849.891
Polygon	20	B	368608.131	3333.722
Polygon	21	B	7443410.535	28409.833
Polygon	22	B	158533.587	2318.518
Polygon	24	B	97876.179	3101.663
Polygon	26	C	368.428	86.755
Polygon	27	C	890.325	156.534
Polygon	28	C	580.911	105.638
Polygon	29	C	10707.314	684.264
Polygon	30	C	1610157.338	12804.005

Cn	Cod
Alpi e pascoli	A
Arbusteti	B
Aree urbanizzate	C
Ceduo	D
Ceduo scadente	E
Fustaia	F
Fustaia rada	G
Improduttivi	H
Laghi paludi e zone umide	L
Prati e colture agrarie	M
Rupi boscate	N



Chiave primaria

Chiave esterna

Tabella risultante

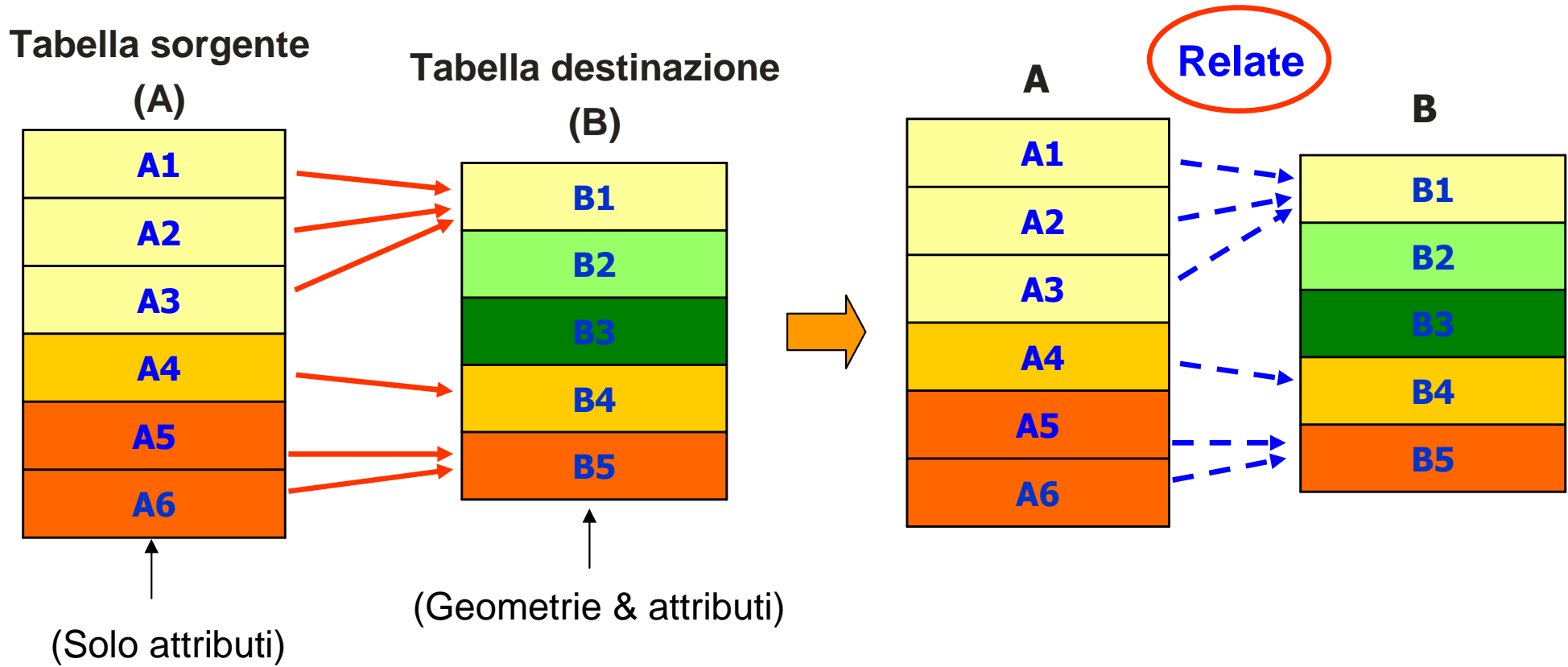
Shape	Id	Gr_idrolog	Area	Perimeter	Cn
Polygon	5	A	10323.980	1082.824	Alpi e pascoli
Polygon	6	A	2162357.010	12435.100	Alpi e pascoli
Polygon	7	A	77076.618	2032.559	Alpi e pascoli
Polygon	8	A	148008.351	4696.349	Alpi e pascoli
Polygon	18	B	144232.288	1838.674	Arbusteti
Polygon	19	B	1865253.097	6849.891	Arbusteti
Polygon	20	B	368608.131	3333.722	Arbusteti
Polygon	21	B	7443410.535	28409.833	Arbusteti
Polygon	22	B	158533.587	2318.518	Arbusteti
Polygon	24	B	97876.179	3101.663	Arbusteti
Polygon	26	C	368.428	86.755	Aree urbanizzate
Polygon	27	C	890.325	156.534	Aree urbanizzate
Polygon	28	C	580.911	105.638	Aree urbanizzate
Polygon	29	C	10707.314	684.264	Aree urbanizzate
Polygon	30	C	1610157.338	12804.005	Aree urbanizzate

Dati "accodati"

## Relazione Multi-a-Uno

**Più record** della tabella sorgente A possono corrispondere ad **un'unico record** della tabella destinazione B.

Il viceversa non è invece possibile.



**Esempio:**

A = tabella dei Comuni ⇔ B = tabella delle Province

(Ogni provincia ospita PIÙ DI UN comune, tra A e B si instaura una relazione di tipo Multi-a-Uno)

# Esempio di RELATE Multi-a -Uno

Tabella **destinazione**  
(geometrie & attributi)

Shape	A_name	Food
Polygon	VERONA	23
Polygon	VICENZA	24
Polygon	BELLUNO	25
Polygon	TREVISIO	26
Polygon	VENEZIA	27
Polygon	PADOVA	28
Polygon	ROVIGO	29

Chiave primaria

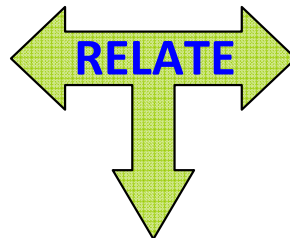


Tabella **sorgente** (solo attributi)

A_name	B_codice	D_supha	Food
Veronella	23092	2169.55	23
Vestenanova	23093	2420.05	23
Vigasio	23094	3083.83	23
Villa Bartolomea	23095	5309.30	23
Villafranca di V.	23096	5724.62	23
Zevio	23097	5469.61	23
Zimella	23098	1726.24	23
Agugliaro	24001	1461.34	24
Albetone	24002	2017.48	24
Alonte	24003	1113.06	24
Altavilla Vic.	24004	1661.49	24
Altissimo	24005	1507.01	24
Arcugnano	24006	4181.40	24

Chiave esterna

## Risultato:

Non si verifica l'accodamento.

Selezionando un record della tabella destinazione vengono selezionati automaticamente TUTTI i corrispondenti record della tabella sorgente.

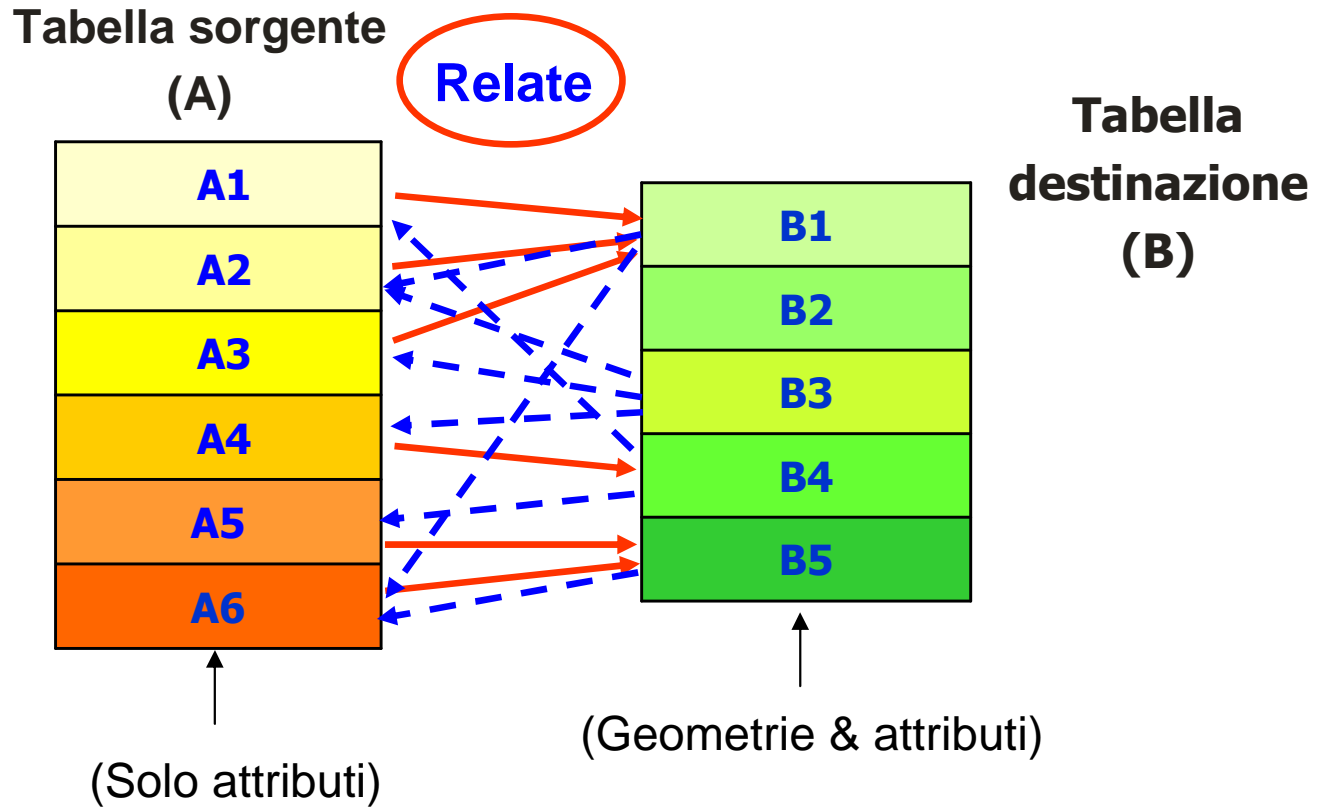
Shape	A_name	C_supha	D_pop	Food
Polygon	VERONA	309713.00	838000	23
Polygon	VICENZA	272429.43	807000	24
Polygon	BELLUNO	367412.83	210500	25
Polygon	TREVISIO	247977.24	808000	26
Polygon	VENEZIA	247061.92	813000	27
Polygon	PADOVA	214306.55	858000	28
Polygon	ROVIGO	181814.55	243500	29

A_name	B_codice	D_supha	Food
Cassola	24026	1276.02	24
Castegnaro	24027	1164.82	24
Castelgomberto	24028	1744.37	24
Chiampo	24029	2264.50	24
Chiuppano	24030	471.91	24
Cison del G.	24031	3490.23	24
Cogollo del C.	24032	3597.88	24
Conco	24033	2665.74	24
Cornedo Vic.	24034	2353.15	24
Costabissara	24035	1308.34	24
Creazzo	24036	1069.22	24
Crespadoro	24037	3024.72	24
Dueville	24038	2016.94	24
Enego	24039	5259.46	24
Fara Vic.	24040	1523.65	24
Foza	24041	3506.93	24
Gallo	24042	4790.25	24
Gambellara	24043	1312.52	24
Gambugliano	24044	797.21	24
...	...	...	...



## Relazione Multi-a-Multi

**Più record** della tabella sorgente A possono essere messi in relazione con **più record** della tabella destinazione B e viceversa.



**Esempio:**

Autisti ⇔ Linee Autobus

(Una data linea può essere percorsa da più autisti e viceversa un autista può percorrere più linee.)

# Vantaggi delle strutture RELAZIONALI - 1

- Struttura dati **più semplice** in quanto
  - tutti i dati sono rappresentati in tabelle, cioè sono organizzati secondo righe e colonne;
  - tutte le operazioni vengono effettuate su tabelle;
  - tutte le operazioni restituiscono come risultato delle tabelle;
  - qualsiasi campo può essere usato come chiave di ricerca o di relazione.
- Notevole **riduzione della ridondanza** dei dati.
- Possibilità di **definire nuove relazioni** in funzione di specifiche richieste senza dover ricostruire ex novo il database.

## Vantaggi delle strutture RELAZIONALI - 2

---

- Sono strutture più **flessibili** perché:
  - consentono di fare collegamenti fra le tabelle a seconda delle proprie necessità (al contrario, le strutture ad albero e reticolare sono di tipo rigido, i collegamenti fra i dati sono stabiliti a priori).
  - consentono l'aggiornamento/modifica dei dati, mantenendo inalterate le capacità di relazionamento, e la visualizzazione immediata sulla mappa degli effetti di queste modifiche.
- Attraverso una struttura relazionale è possibile considerare lo stesso oggetto contemporaneamente sotto **punti di vista diversi**, cronologico, tipologico normativo ecc. mentre con una struttura gerarchica è possibile solo un tipo di interdipendenza tra gli oggetti.

# Vantaggi delle strutture RELAZIONALI - 3

- Strutture **semplici** lato utente: le *query* sono formulate in termini di **COSA** occorre trovare, e non **COME** lo si deve trovare.

Viceversa, con le strutture ad albero e reticolare occorre sapere come sono collegate fra loro le tabelle che contengono i dati.

The screenshot displays a database application window titled 'Attributes of fSite'. It contains a table with the following columns: OBJECTID, Id, Name, SiteAdded, IdCompiler, Country, Region, Province, Type Site, MaxDistance, Timeall, Lat, Lon, and IdDatum. The table lists 23 records, with records 2 through 13 highlighted in blue. A 'Select by Attributes' dialog box is open over the table, showing a list of attributes on the left and a list of values on the right. The 'Where' clause in the dialog is set to '[Region] = Sicily'. The dialog also includes a 'Method' dropdown set to 'Create a new selection', a 'Get Unique Values' button, and a 'Go To' field.

OBJECTID	Id	Name	SiteAdded	IdCompiler	Country	Region	Province	Type Site	MaxDistance	Timeall	Lat	Lon	IdDatum
1	1	Siponto	<Null>		Italia	Puglia	Foggia						
2	2	Torre Degli Inglesi	<Null>		Italy	Sicily	Mesima						
3	3	Lesina Est	<Null>		Italy	Puglia	Foggia						
4	4	Augusta	<Null>		Italy	Sicily	Siracusa						
5	5	Priolo	<Null>		Italy	Sicily	Siracusa						
6	6	Torre Squilace	<Null>		Italy	Puglia	Lecce						
7	7	Torre Santa Sabina	<Null>		Italy	Puglia	Brindisi						
8	8	Magnisi Peninsula	<Null>		Italy	Sicily	Siracusa						
9	9	Maddalena Peninsula	<Null>		Italy	Sicily	Siracusa						
10	10	Ognina	<Null>		Italy	Sicily	Siracusa						
11	12	Sarkoy	<Null>		Turkey	Marmara Region	Tekirdag						
12	13	Stromboli	<Null>		Italy	Sicily	Messina						
13	14	Didim	<Null>		Turkey	Southwestern Turkey	Aydin						
14	15	Dalaman	<Null>		Turkey	Southwestern Turkey	Mugla						
15	16	Fethiye	<Null>		Turkey	Southwestern Turkey	Mugla						
16	17	Taskopru	<Null>		Turkey	Eastern Marmara	Yalova						
17	18	Kucukcekmece	<Null>		Turkey	Marmara	Istanbul						
18	19	Aiki	<Null>		Greece	southern Corinth Gulf	Egion						
19	20	Cap De Ses Salnes	<Null>		Spain	Balearic Islands	SE Mallorca						
20	21	Artá Peninsula	<Null>		Spain	Balearic Islands	NE Mallorca						
21	22	Torre S. Emiliano	<Null>		Italy	Puglia	Otranto						
22	23	Kirra	<Null>		Greece	northern Corinth Gulf	Delphi						

Esempio di query sugli attributi tramite SQL.

# Strutturazione di un database relazionale

---

L'organizzazione dei dati e la scelta dei campi non è un banale fatto tecnico.

Poiché la base dati è la rappresentazione della realtà che si vuole conoscere, la validità delle conclusioni di una ricerca dipende in eguale misura dalla **qualità dei dati** e dalla bontà della loro **organizzazione**.

**Regole principali** per strutturare un database relazionale:

- identificare e definire le varie entità,
- stabilire le relazioni tra gli elementi,
- rappresentare il modello secondo un diagramma,
- assegnare attributi e costruire una tabella per ogni entità.

## Cosa contengono i campi ?

---

Gli attributi (dati spaziali) possono essere classificati in modo diverso a seconda dei seguenti tre criteri:

- in base alla **sorgente** dati da cui sono estratti;
- in base alla **natura** del dato;
- in base alla **tipologia** del dato rappresentato.

Classificando gli attributi in base alla **tipologia di sorgente** da cui sono estratti è possibile distinguere le due classi seguenti:

- **dati socio-economici;**
- **dati ambientali.**

Svariati studi hanno messo in evidenza che i dati derivanti dalla maggior parte dei campi scientifici possono essere analizzati da un punto di vista geografico.

In particolare si ritiene che all'**80%** dei dati disponibili nel mondo possa essere associata (direttamente o indirettamente) una componente spaziale.



I dati **socio-economici** sono per lo più già disponibili presso vari enti pubblici o privati a livello locale, regionale o nazionale, come risultato di rilevazioni demografiche (es. ISTAT, banche, Camere di Commercio, ecc.)

In ambito GIS i dati socio-economici sono impiegati ad esempio per:

- individuare nuove aree di mercato (ad es. per individuare la posizione migliore per un nuovo centro commerciale);
- realizzare sistemi di prevenzione sanitaria (ad es. analisi della distribuzione geografica di un virus);
- ottimizzare la gestione dei servizi sociali (ad es. per stabilire la congruenza tra dislocazione dei centri per anziani e l'effettivo bacino d'utenza).

## Esempio di dati socio-economici - 1

Popolazione residente nella Provincia di Padova per Comune

Comune	Età												Totale
	00-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59	60-64	65-hi	
028001 - Abano Terme	2283	797	999	1364	1420	1538	1436	1348	1360	1223	1168	3270	18206
028002 - Agna	392	149	191	226	277	298	219	208	206	177	177	638	3158
028003 - Albignasego	2596	945	1077	1507	1636	1771	1520	1487	1374	1311	1203	2720	19147
028004 - Anguillara Veneta	542	249	327	311	356	385	352	329	329	281	280	998	4739
028005 - Arquà Petrarca	223	109	128	147	139	140	121	152	153	119	107	338	1876
028006 - Arre	289	88	112	138	167	173	156	133	118	94	116	445	2029
028007 - Arzergrande	585	244	252	310	303	395	342	273	270	222	222	695	4113
028008 - Bagnoli di Sopra	464	165	214	343	364	330	273	239	271	221	246	752	3882
028009 - Baone	401	190	193	227	200	260	247	259	256	180	165	560	3138
028010 - Barbona	105	36	36	52	70	74	62	42	48	40	45	170	780
028011 - Battaglia Terme	508	154	215	303	353	355	288	264	269	218	297	928	4152
028012 - Boara Pisani	285	120	148	187	195	211	215	170	180	137	147	512	2507
028013 - Borgoricco	1098	319	414	600	679	688	525	424	405	347	387	1053	6939
028014 - Bovolenta	403	180	204	245	284	235	241	236	222	180	161	553	3144
028015 - Brugine	947	321	406	487	606	625	491	456	384	297	285	802	6107
028016 - Cadoneghe	2021	698	783	1079	1235	1367	1265	1024	1008	992	1009	2274	14755
028017 - Campodarsego	1689	551	707	912	1099	1116	851	711	704	671	729	1734	11474
028018 - Campodoro	368	129	127	145	187	223	210	155	122	93	106	305	2170
028019 - Camposampiero	1735	599	620	837	1012	979	818	718	669	531	493	1669	10680
028020 - Campo San Martino	779	297	354	451	464	464	399	354	353	282	278	896	5371
028021 - Candiana	354	130	133	185	176	238	205	155	147	110	113	509	2455
028022 - Carceri	238	66	80	102	132	154	107	98	78	65	106	298	1524

## Esempio di dati socio-economici - 2

### Censimento Agricoltura 2000

Utilizzazione del terreno a seminativi e coltivazioni legnose. Regione del Veneto

	descrizione	Province							Totale Veneto
		Verona	Vicenza	Belluno	Treviso	Venezia	Padova	Rovigo	
<b>SAU</b>	aziende	26206	33375	7352	43697	24760	41400	10705	187495
	superficie	177520.32	114170.31	52893.28	138493.72	119995.31	135668.13	114002.81	852743.88
<b>SEMINATIVI</b>	aziende	12329	16671	3496	30560	22726	36864	10123	132769
	superficie	97809.84	56158.92	5161.03	85618	108676.06	116865.42	109985.02	580274.29
<b>CEREALI</b>	aziende	8651	13434	2327	25918	17240	32485	8959	109014
	superficie	48427.46	35467.85	2715.03	55808.3	56589	83519.39	68786.16	351313.19
frumento tenero	aziende	1948	1699	4	1751	1480	4036	2691	13609
	superficie	6436.09	3572.44	24.66	3140.09	5878.44	8279.62	10918.81	38250.15
frumento duro	aziende	217	161	7	667	266	527	268	2113
	superficie	673.41	264.29	16.1	1124.68	853.79	1017.5	1222.76	5172.53
segala	aziende	5	10	1	10	5	23	4	58
	superficie	8.92	24.36	13.02	17.29	8.1	39.6	7.48	118.77
orzo	aziende	975	864	18	1998	303	607	119	4884
	superficie	2669.87	1227.28	30.85	2561.75	715.43	907.17	379.27	8491.62
avena	aziende	37	47	7	72	29	103	14	309
	superficie	124.68	52.59	2.74	81.53	69.33	393.27	56.12	780.26
granoturco escluse foraggere	aziende	7262	12304	2302	23994	16343	30292	8364	100861
	superficie	36655.37	29871.17	2600.55	48477.88	48629.07	72471.38	54757	293462.42
riso	aziende	59	17	3	20	10	22	49	180
	superficie	1737.2	169.51	1.39	36.92	265.47	129.1	1392.97	3732.56
sorgo	aziende	8	159	5	22	16	38	2	250
	superficie	12.62	219.78	1	45.97	75.87	112.81	3.77	471.82
altri cereali	aziende	28	40	7	79	23	54	10	241
	superficie	109.3	66.43	24.72	322.19	93.5	168.94	47.98	833.06
<b>LEGUMI</b>	aziende	44	200	947	55	100	59	58	1463
	superficie	186.52	30.01	38.73	64.82	87.14	50.16	31.47	488.85
pisello proteico	aziende	11	25	13	7	18	21	6	101
	superficie	101.31	6.22	0.68	1.02	7.84	19.62	11.05	147.74
pisello secco	aziende	12	31	25	1	26	8	7	110
	superficie	12.66	1.96	0.55	0.01	2.56	5.06	6.39	29.19

I dati **ambientali**, invece, sono acquisiti principalmente da

- mappe topografiche già esistenti,
- da strumenti di *remote sensing*,
- rilievi topografici
- cartografia tematica (cioè mappe relative a specifici aspetti geologici e territoriali).



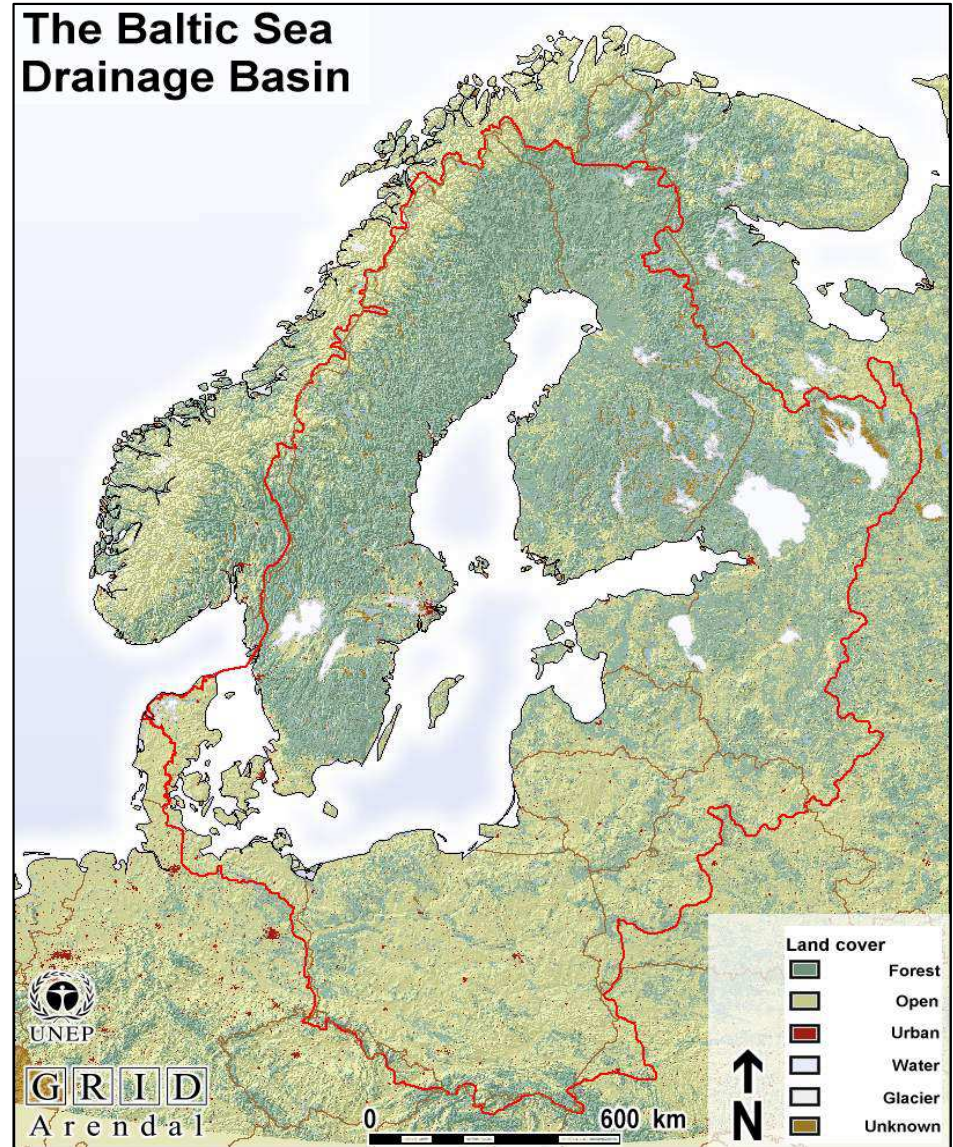
## Immagine satellitare

Da un'immagine satellitare è possibile estrarre le aree coperte da differenti tipologie di vegetazione o coltivazione.

## Ortofoto

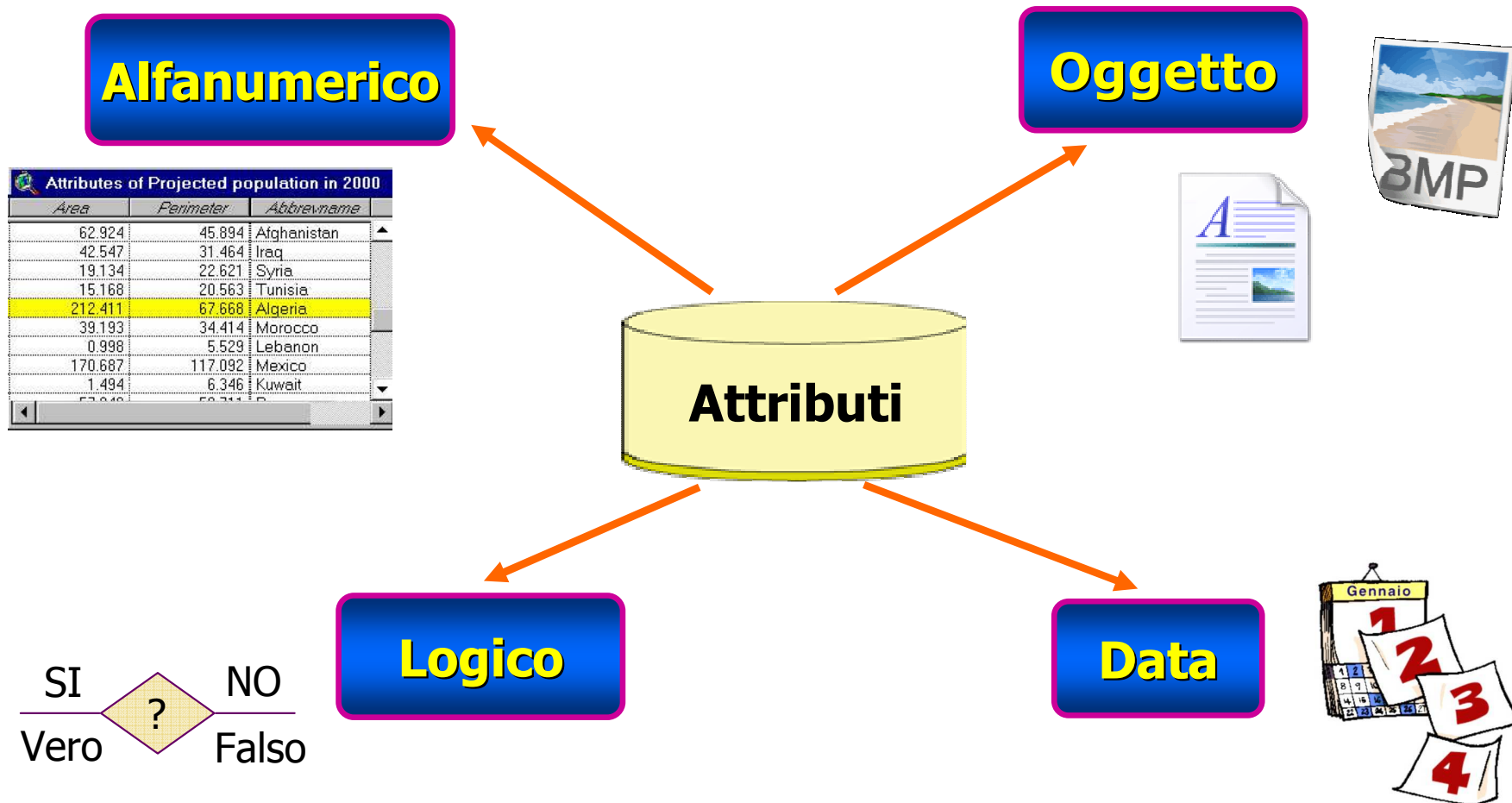


Da un'ortofoto è possibile estrarre per vettorializzazione i dati relativi all'uso del suolo.



## Carta tematica

Gli attributi possono essere classificati anche in base al **contenuto**, distinguendo così le seguenti tipologie:



Gli attributi possono essere infine classificati in base alla **tipologia** del dato che rappresentano, distinguendo così le seguenti quattro categorie:

- **Nominali**: valori alfabetici (es. Tevere, Rossi, medio, agricolo, ecc.);
- **Ordinali**: valori che stabiliscono una gerarchia (1,2,3...a,b,c..), possono essere di tipo numerico o testo;
- **Intervallo**: intervalli di ampiezza costante o variabile (1-10, forte-debole, ecc.), sono rappresentabili in formato numerico o testo;
- **Numeri reali**: (1, -100, 2.4599,..), valori numerici (interi o decimali).

L'insieme dei valori che un attributo può assumere prende il nome di **dominio**.

Il dominio (se noto a priori) può essere utilizzato per istituire dei controlli automatici nella fase di inserimento/modifica dei valori dell'attributo.

Ad esempio per il dato Temperatura il dominio potrebbe essere rappresentato dall'intervallo numerico  $[-40^{\circ}, +40^{\circ}]$ .

Nel caso di un campo che rappresenti un'attività erosiva, il dominio potrebbe essere definito da un intervallo di valori di tipo testo, cioè: [bassa, medio- bassa, media, medio-alta, elevata].

Per un attributo di tipo **logico** il dominio è costituito da soli due valori, *vero* o *falso*.

Per un attributo di tipo **data** i valori del dominio dipendono dal tipo di calendario utilizzato (es. gregoriano, giuliano, ecc.).



# Architettura di un GIS: il Geodatabase

---

In un GIS i dati spaziali e aspaziali vengono memorizzati e gestiti attraverso un **Geodatabase**, cioè un archivio integrato contenente in formato digitale

- dati geografici,
- attributi delle entità geografiche.

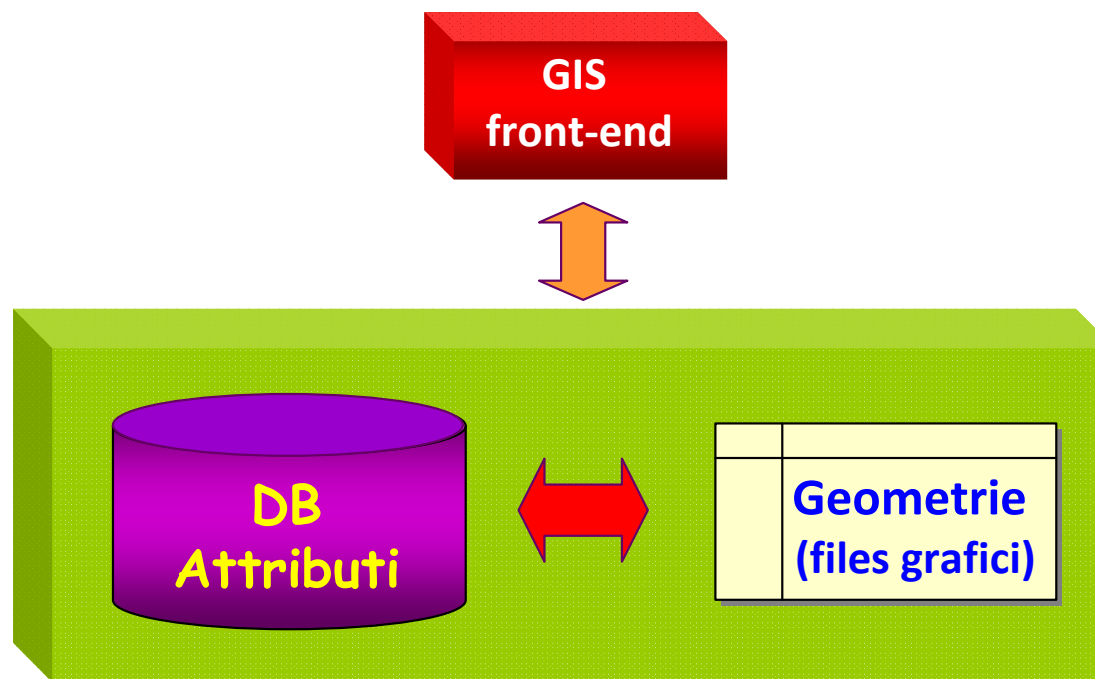
Le possibili soluzioni architettoniche di un GeoDB si possono raggruppare in tre categorie:

- **duplice** (con primitive geometriche o topologiche),
- **a layer**,
- **integrata**.

## Architettura duplice

L'architettura duplice prevede **due archivi separati**, uno per gli attributi ed uno per la componente spaziale, gestiti separatamente dai relativi software.

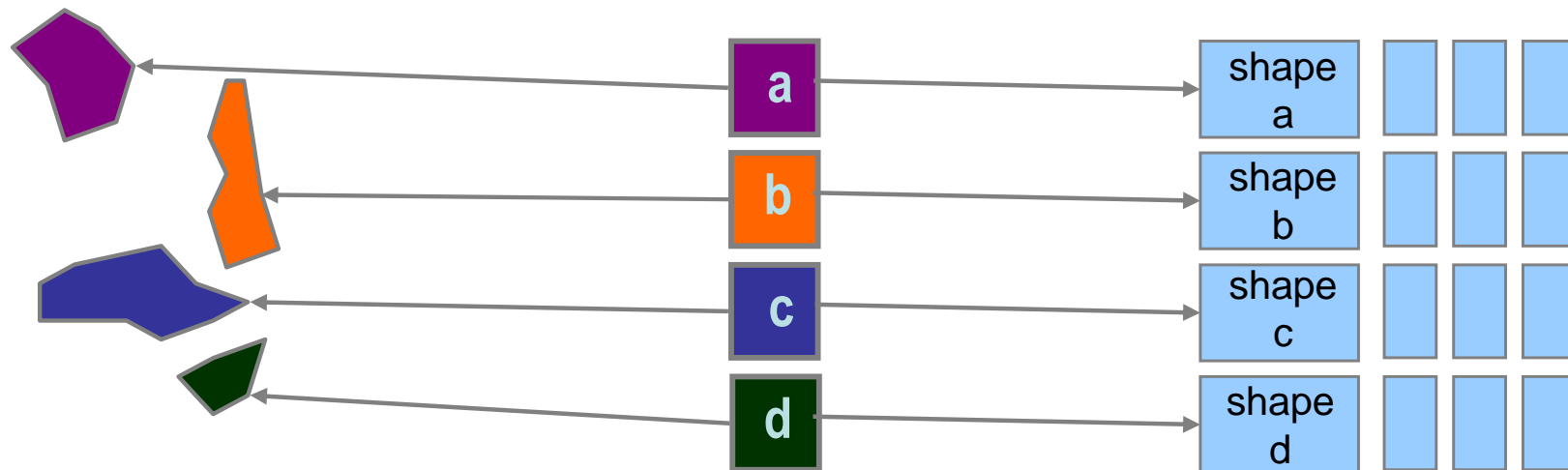
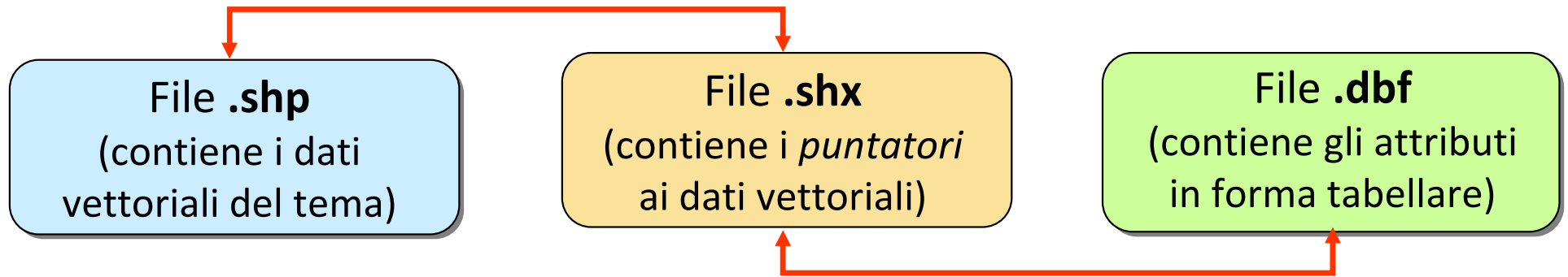
La connessione logica tra le due componenti è assicurata da **puntatori** fra i corrispondenti record dei due archivi.



## Es. struttura duplice con sole geometrie: Shapefile di ESRI

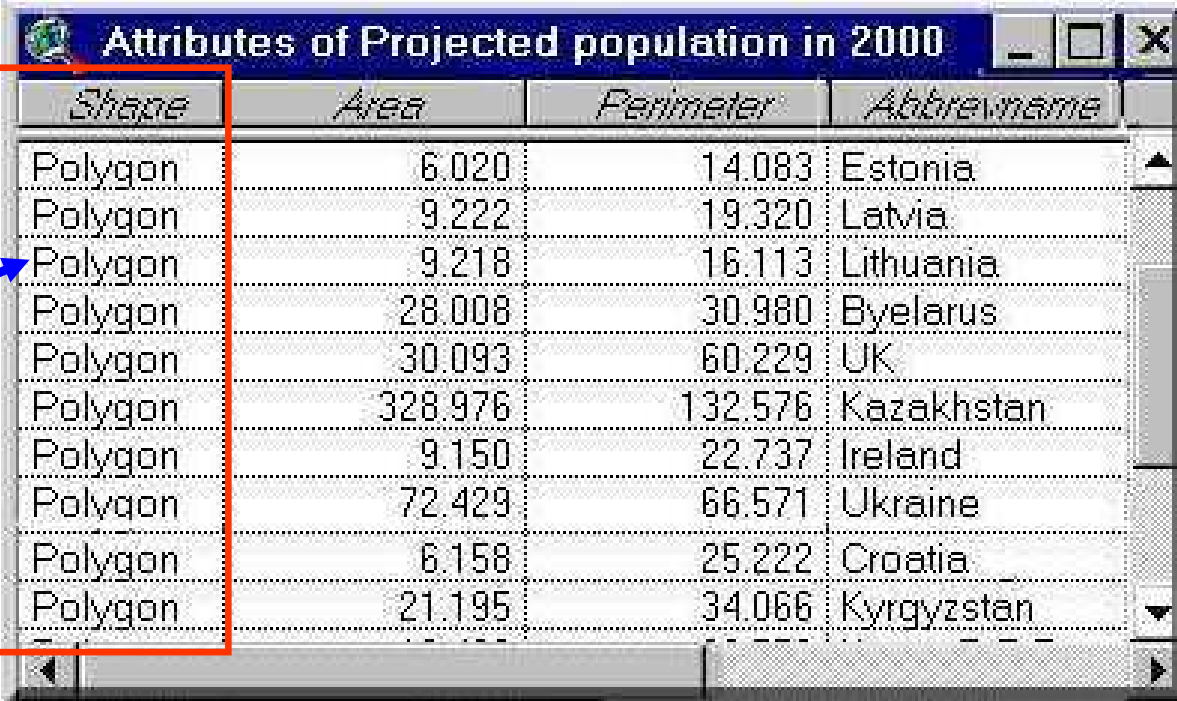
Il legame fisico tra dati geografici e gli attributi si realizza tramite una struttura particolare, diversa da software a software.

Ad esempio, **ArcGIS** utilizza lo schema seguente:



Questo schema si manifesta all'utente tramite il campo **Shape**, presente in ogni tabella degli attributi. Qui ciascun valore del campo è un **puntatore** al relativo elemento vettoriale nel file .shp.

Il nome del valore del campo ne identifica anche la forma geometrica: **Point, Polyline, Polygon**.



<i>Shape</i>	<i>Area</i>	<i>Perimeter</i>	<i>Abbrevname</i>
Polygon	6.020	14.083	Estonia
Polygon	9.222	19.320	Latvia
Polygon	9.218	16.113	Lithuania
Polygon	28.008	30.980	Byelarus
Polygon	30.093	60.229	UK
Polygon	328.976	132.576	Kazakhstan
Polygon	9.150	22.737	Ireland
Polygon	72.429	66.571	Ukraine
Polygon	6.158	25.222	Croatia
Polygon	21.195	34.066	Kyrgyzstan

Tabella (file .dbf)

## Architettura a layer

Prevede un **unico DB relazionale** composto da

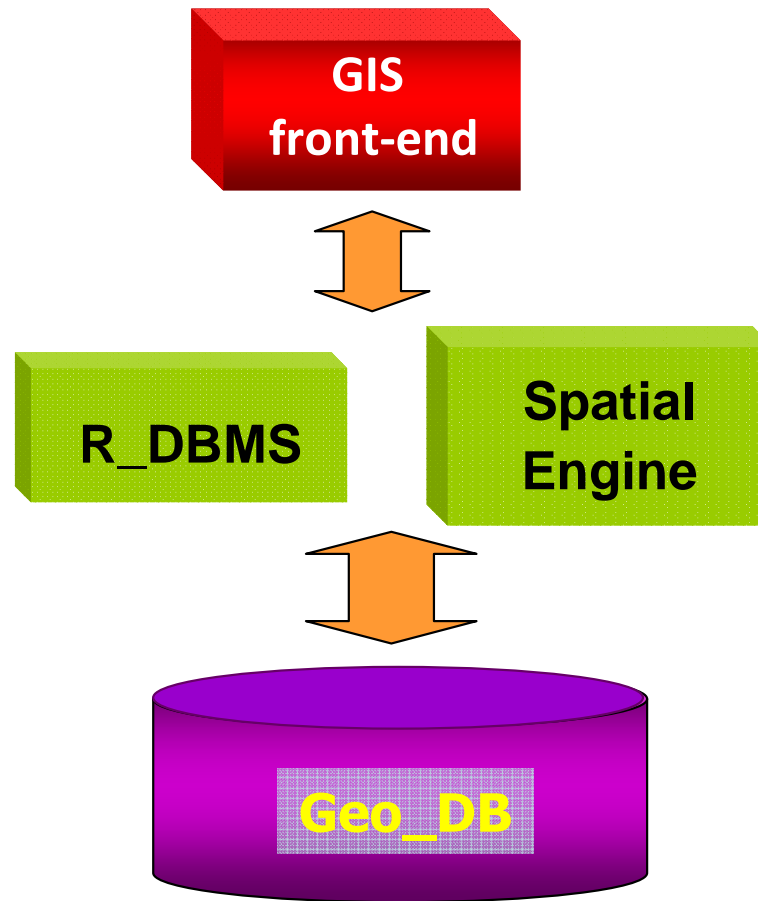
- campi in cui memorizzare gli attributi alfanumerici,
- campi speciali (*BLOB* o *ADT*) in cui memorizzare le primitive geometriche o topologiche.

I *BLOB* contengono primitive geometriche semplici (punto, linea, poligono), mentre gli *ADT* sono definiti dall'utente e possono contenere entrambi i tipi di primitive.

A livello software, il Geo\_DB è gestito da

- un DBMS per gli attributi,
- un Geo\_DBMS (**spatial engine**) per i campi contenenti la componente spaziale.

## Esempio di architettura a layer



*Esempio:* Access e Oracle per gli attributi e SDE (Spatial Data Engine) di ESRI per la componente spaziale.

## Architettura integrata

Un solo archivio per attributi alfanumerici e componenti spaziali ed un unico Geo\_DB per la loro gestione integrata.

*Esempio:* My SQL, PostGIS (PostgreSQL), Oracle 10, ecc.

