

ANNO ACCADEMICO 2004-2005  
SISTEMI INFORMATIVI GEOGRAFICI

SISTEMI INFORMATIVI TERRITORIALI (SIT)  
GEOGRAPHICAL INFORMATION SYSTEMS (GIS)

*Sistemi Informativi Territoriali*

3. I sistemi per la gestione delle basi di dati geografiche: un modello di riferimento per la progettazione logica.

ALBERTO BELUSSI

NOVEMBRE 2004

# Sistemi per la Gestione di Basi di Dati Geografiche

## Premessa

- Consideriamo il modello relazionale come modello di riferimento per i sistemi che gestiscono basi di dati tradizionali.
- La componente geometrica del dato geografico non può essere rappresentata nel modello relazionale utilizzando un insieme di attributi di tipo numerico che contengano coordinate, in quanto:
  - a) i metodi tradizionali per l'accesso ai dati (indici) non sono adeguati per le interrogazioni spaziali,
  - b) l'utente non è in grado di interagire con un'interfaccia che spezza i valori geometrici in coordinate semplici,
  - c) non esiste supporto alle interrogazioni spaziali.
- Un sistema per la gestione di basi di dati basato sul paradigma Object-Oriented non può essere adottato per due motivi:
  - a) le prestazioni di questi sistemi sono ancora insufficienti per questo tipo di applicazioni,
  - b) l'integrazione con basi di dati relazionali è difficoltosa.

## **Soluzione attuale**

Sistemi dedicati al dato geografico (Geo-DBMS) o con moduli aggiuntivi specifici per la gestione del dato geografico.  
Adottano un modello dei dati geo-relazionale.

# Sistemi Per la Gestione di Basi di Dati Geografiche

## Geo-DBMS

Definizione di Geo-DBMS secondo Güting (1994):

1. un sistema per la gestione di dati geografici (Geo-DBMS) e' innanzitutto un sistema per la gestione di dati (DBMS);
2. esso offre tipi predefiniti (Spatial Data Types - SDT) per rappresentare il dato geografico nello schema e per trattare il dato geografico nel linguaggio di interrogazione;
3. esso offre un'implementazione dei dati geografici che include almeno l'accesso ai dati geometrici attraverso un indice spaziale e un algoritmo efficiente per l'operazione di join spaziale.

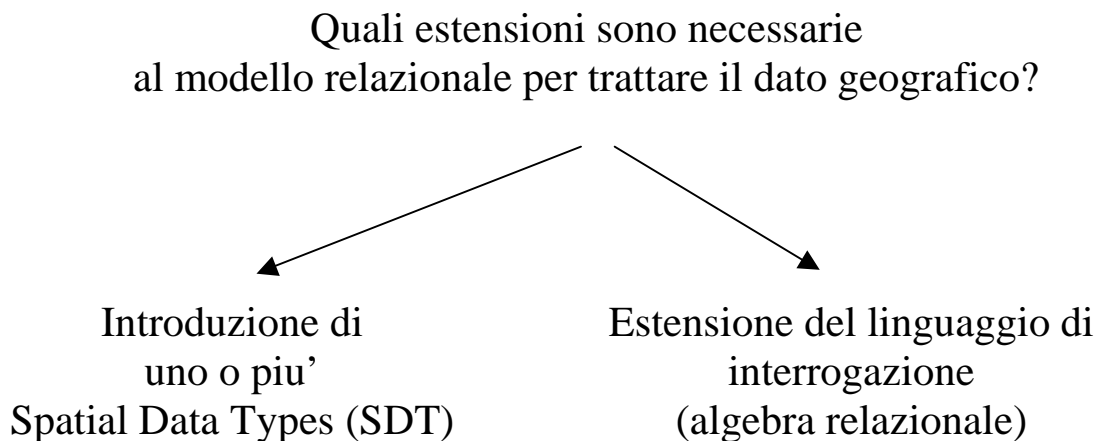
# Sistemi Per la Gestione di Basi di Dati Geografiche

## Il modello dei dati di un Geo-DBMS

La maggior parte degli attuali Geo-DBMS adotta un **modello dei dati georelazionale**.

Tale modello è un'estensione del *modello relazionale che presenta domini e operatori specifici per rappresentare e manipolare il dato geografico*.

NOTA: Un modello simile potrebbe essere ottenuto in un sistema object-relational istanziando domini e metodi specifici per il dato geografico. Tuttavia, rispetto ad un Geo-DBMS, in un sistema object-relational personalizzato per il dato geografico mancherebbero i metodi d'accesso dedicati alle interrogazioni spaziali, gli operatori più complessi e tutta la parte di gestione della grafica dovrebbe essere realizzata dal programmatore dell'applicazione.



# Un modello logico di riferimento per Geo-DBMS

## Spatial Data Types (SDT)

### Caratteristiche principali

- geometria lineare: i componenti di base sono punti, segmenti di retta, spezzate, spezzate chiuse; le equazioni sono quindi lineari, vale a dire del tipo:

$$ax + by + c = 0$$

- valori "mono-dimensionali": si definisce un SDT specifico per ogni dimensione:
  - 0 (punto/point),
  - 1 (spezzata/polyline),
  - 2 (poligono/polygon).

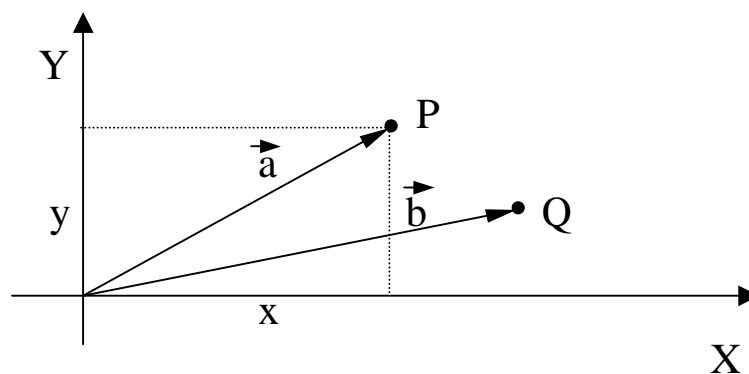
# Un modello logico di riferimento per Geo-DBMS

## Spatial Data Types (SDT)

**Definizioni di base (Worboys, "GIS: a computing perspective")**

Spazio di riferimento: Piano Euclideo ( $\mathcal{R}^2$ )

Punto:  $P = (x,y)$



$\vec{a}$  è il *vettore* che individua il punto P in  $\mathcal{R}^2$ .

Distanza tra due punti P e Q individuati dai vettori  $\vec{a} = (x_a, y_a)$  e  $\vec{b} = (x_b, y_b)$ :

$$\overline{PQ} = \|\vec{a} - \vec{b}\| = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

Operazioni sui vettori:

$$\vec{a} + \vec{b} = \vec{c} \quad \vec{c} = (x_a + x_b, y_a + y_b)$$

$$\vec{a} - \vec{b} = \vec{d} \quad \vec{d} = (x_a - x_b, y_a - y_b)$$

$$\lambda \vec{a} = \vec{e} \quad \vec{e} = (\lambda x_a, \lambda y_a)$$

Segmento di retta per i punti P e Q individuati dai vettori  $\vec{a}$  e  $\vec{b}$ :

$$s = \{\lambda\vec{a} + (1-\lambda)\vec{b} \mid \lambda \in [0,1]\}$$

chiamiamo *estremi del segmento* i punti:  $s.E_1 = Q$  ( $\lambda=1$ ) e  $s.E_0 = P$  ( $\lambda=0$ )

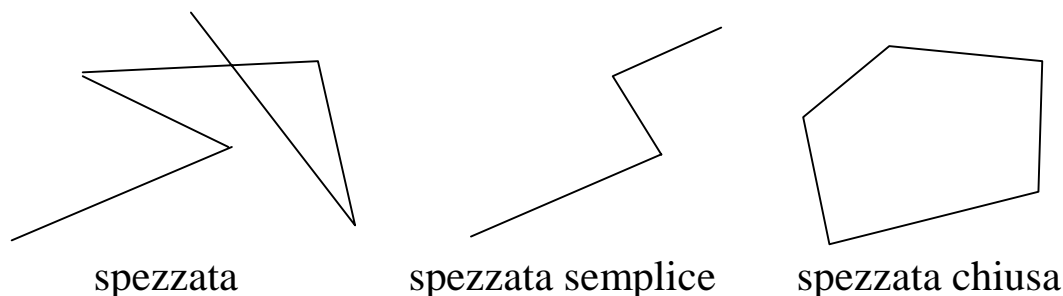
Spezzata (Polyline): è un insieme di segmenti di retta  $sp = \{s_1, \dots, s_n\}$  tali che ogni estremo di segmento è condiviso da esattamente due segmenti, eccetto al massimo due estremi detti *estremi della spezzata*.

Data la spezzata  $sp = \{s_1, \dots, s_n\}$ , se vale la seguente condizione:

$$(\forall s_i, s_j \in sp)(i \neq j \Rightarrow (s_i \cap s_j = \emptyset \vee s_i.E_0 = s_j.E_0 \vee s_i.E_0 = s_j.E_1 \vee s_i.E_1 = s_j.E_0 \vee s_i.E_1 = s_j.E_1))$$

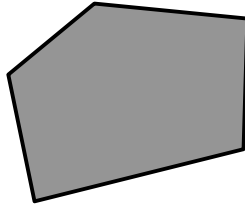
(la spezzata non si autointerseca), allora la spezzata si dice *spezzata semplice*.

Se una spezzata non ha estremi si dice *spezzata chiusa*.



Ogni spezzata  $sp$  è orientata in quanto è definito quale è il suo primo estremo ( $sp.E_0$ ) e quale il secondo ( $sp.E_1$ )

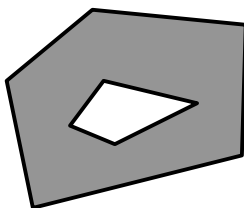
Poligono semplice (Simple Polygon): data una spezzata chiusa  $sp$ , è la porzione di  $\mathcal{R}^2$  delimitata da  $sp$ .  $sp$  fa parte del poligono (il poligono è un insieme di punti chiuso).



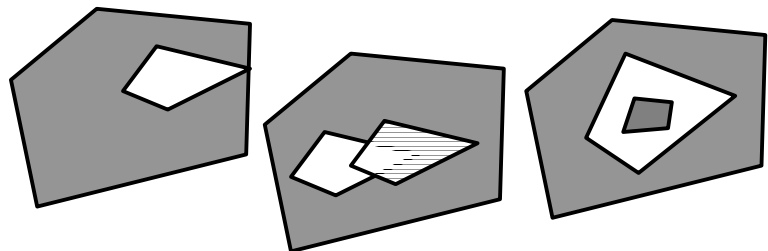
poligono semplice

Poligono bucato (Polygon): data una spezzata  $sp$  e un insieme di spezzate chiuse  $H = \{sp_1, \dots, sp_n\}$ , è la porzione di  $\mathcal{R}^2$  delimitata da  $sp$  alla quale siano tolte le porzioni di piano delimitate da ciascuna  $sp_i \in H$ . La spezzata  $sp$  e tutte le spezzate di  $H$  sono parte del poligono (il poligono è un insieme di punti chiuso), inoltre valgono le seguenti condizioni:

- tutte le spezzate di  $H$  sono contenute nel poligono semplice delimitato da  $sp$ .
- tutte le spezzate di  $H$  non intersecano  $sp$ . ( $sp$  interseca  $sp'$  se esiste almeno un segmento  $s$  di  $sp$  e un segmento  $s'$  di  $sp'$  tali che  $s \cap s' \neq \emptyset$ )
- $\forall (sp_i, sp_j \in H) (i \neq j \Rightarrow ((sp_i \text{ non interseca } sp_j) \wedge (sp_i \text{ non è contenuta nel poligono delimitato da } sp_j)))$



poligono bucato



poligoni non ammessi



# Un modello logico di riferimento per Geo-DBMS

## Spatial Data Types (SDT)

### POINT

*Def:* E' un insieme finito P di punti di  $\mathcal{R}^2$ .

*Boundary of P:* è l'insieme vuoto.

*Interior of P:* è P stesso.

*Proprietà:*

Dato P: POINT:

- *P.X:* REAL;  
coordinata x del baricentro di P.
- *P.Y:* REAL  
coordinata y del baricentro di P.
- *P.Distanza(Q: POINT):* REAL  
distanza euclidea tra il baricentro di P e il baricentro di Q.
- *P.NumPunti:* INTEGER  
numero di punti presenti in P.
- *P.Punto(I: INTEGER):* POINT  
i-esimo punto presente in P.
- *P.Unione(Q: POINT):* POINT  
genera l'unione dei due insiemi di punti.

Si suppone definito anche l'operatore di uguaglianza:  $P = Q$ .

# Un modello logico di riferimento per Geo-DBMS

## Spatial Data Types (SDT)

### LINE

*Def:* E' un insieme di spezzate  $L$  di  $\mathcal{R}^2$  tali che nessuna coppia di spezzate abbia come intersezione una spezzata.

*Boundary of L:* è l'insieme di punti  $\mathcal{R}^2$  costituito dagli estremi delle spezzate tolti gli estremi che appartengono a più di una spezzata.

*Interior of L:* è l'insieme di punti  $\mathcal{R}^2$  costituito dalle spezzate senza gli estremi che costituiscono il Boundary of L.

*Proprietà:*

Data  $L$ : LINE:

- *L.Boundary*: POINT;  
boundary di  $L$ .
- *L.NumSpezzate*: INTEGER  
numero di spezzate di  $L$ .
- *L.Lunghezza*: REAL  
lunghezza di  $L$  ottenuta sommando la lunghezza di tutti le spezzate che la compongono.
- *L.Semplice*: BOOLEAN  
true se l'unione delle spezzate di  $L$  è una spezzata semplice.
- *L.Chiusa*: BOOLEAN  
true se l'unione delle spezzate di  $L$  è una spezzata chiusa.

- *L.Unione*(K: LINE): LINE  
genera un valore di tipo LINE che rappresenta l'unione delle due linee L e K.
- *L.IntersezioneL*(K: LINE): LINE  
genera un valore di tipo LINE che rappresenta l'intersezione delle due linee L e K. Se l'intersezione è costituita solo da punti allora produce l'insieme vuoto.
- *L.IntersezioneP*(K: LINE): POINT  
genera un valore di tipo POINT che rappresenta l'intersezione delle due linee L e K. Se l'intersezione è costituita solo da spezzate allora produce l'insieme vuoto. Se nell'intersezione sono presenti spezzate e punti isolati, restituisce solo questi ultimi.

Si suppone definito anche l'operatore di uguaglianza:  $L = L'$ .

# Un modello logico di riferimento per Geo-DBMS

## Spatial Data Types (SDT)

### POLYGON

*Def:* E' un insieme  $G$  di poligoni bucati di  $\mathcal{R}^2$ , tali che ogni coppia di poligoni costituisce un insieme di punti non connesso.

*Boundary of G:* è costituito dalle spezzate che delimitano i poligoni di  $G$ .

*Interior of G:* è il poligono senza le spezzate che lo delimitano.

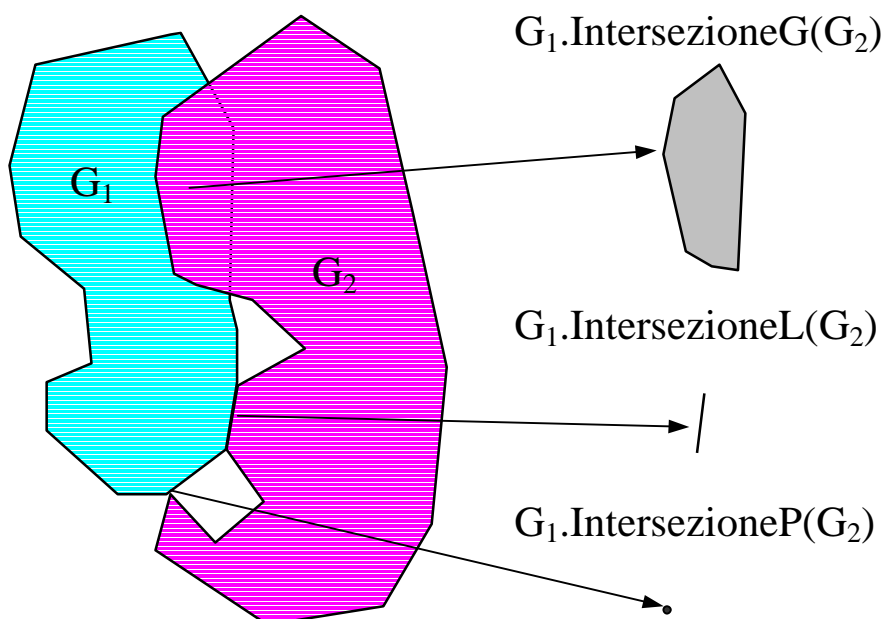
*Proprietà:*

Data  $G$ : POLYGON:

- *G.Frontiera*: LINE;  
valore geometrico di tipo LINE che costituisce la frontiera esterna del poligono.
- *G.Boundary*: LINE  
valore geometrico di tipo LINE che costituisce la frontiera del poligono.
- *G.Semplice*: BOOLEAN  
true se  $G$  è semplice.
- *G.NumBuchi*: INTEGER  
numero di buchi presenti nel poligono  $G$ .
- *G.Perimetro*: REAL  
lunghezza del perimetro di  $G$  ottenuto sommando la lunghezza di tutte le spezzate che lo delimitano.
- *G.Area*: REAL  
misura dell'area del poligono  $G$ .

- $G.Unione(H: POLYGON): POLYGON$   
genera un valore di tipo POLYGON che rappresenta l'unione dei due poligoni G e H.
- $G.IntersezioneG(H: POLYGON): POLYGON$   
genera un valore di tipo POLYGON che rappresenta l'intersezione dei due poligoni G e H. Se l'intersezione è costituita solo da punti e/o linee allora produce l'insieme vuoto.
- $G.IntersezioneL(H: POLYGON): LINE$   
genera un valore di tipo LINE che rappresenta l'intersezione dei due poligoni G e H. Se l'intersezione è costituita solo da punti e/o da poligoni allora produce l'insieme vuoto. Se l'intersezione presenta oltre a spezzate anche poligoni e/o punti restituisce solo le spezzate.
- $G.IntersezioneP(H: POLYGON): POINT$   
genera un valore di tipo POINT che rappresenta l'intersezione dei due poligoni G e H. Se l'intersezione è costituita solo da linee e/o poligoni allora produce l'insieme vuoto. Se l'intersezione presenta oltre a punti isolati anche linee e/o poligoni restituisce solo i punti isolati.

Si suppone definito anche l'operatore di uguaglianza:  $G = G'$ .



# Un modello logico di riferimento per Geo-DBMS

## Le relazioni (tabelle) del modello geo-relazionale

### Domini di base:

#### *Tradizionali*

- INTEGER: numeri interi,
- REAL: numeri reali approssimati (rappresentazione Mantissa/Esponente),
- STRING: stringhe di caratteri.
- BOOLEAN: valori booleani (vero, falso)

#### *Geometrici* (vedi definizioni precedenti)

- POINT,
- LINE,
- POLYGON.

### Relazioni (tabelle):

Schema:  $R(X_1: D_1, \dots, X_n: D_n)$

dove  $D_1, \dots, D_n$  rappresentano domini di base tradizionali o geometrici.

Istanza:  $r$  di  $R(X_1: D_1, \dots, X_n: D_n)$

$$r \subseteq (D_1 \times \dots \times D_n)$$

### Esempio

COMUNE(      Nome: STRING,  
                  NumAbitanti: INTEGER,  
                  Estensione: POLYGON)

# Un modello fisico di riferimento per Geo-DBMS

## La rappresentazione dei valori geometrici in un Geo-DBMS

(A)  
Come rappresentare  
i VALORI GEOMETRICI  
dei domini SDT del modello  
geo-relazionale

(B)  
Come INTEGRARE  
i VALORI GEOMETRICI e  
il dato ALFANUMERICO

### (A) Rappresentazione dei valori geometrici

Problemi:

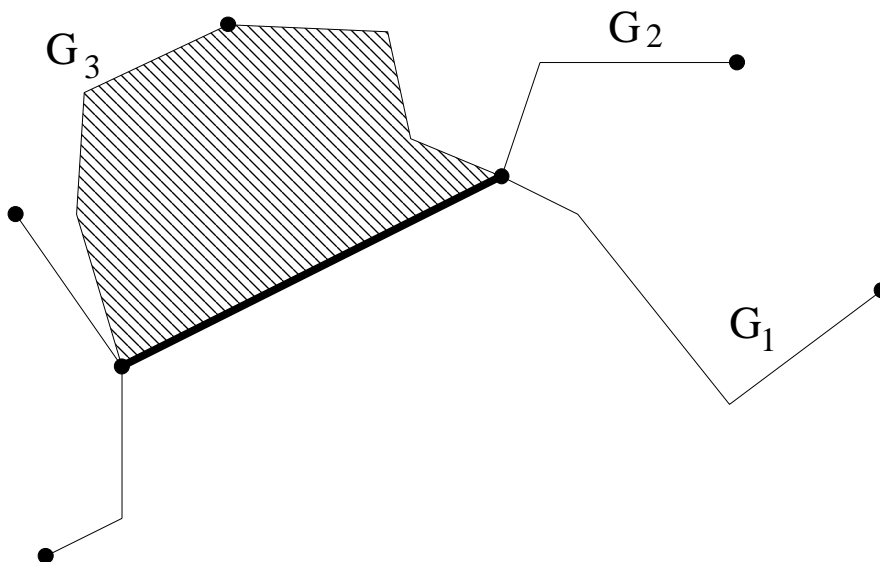
1. L'occupazione in memoria di un valore geometrico può variare molto: da una coppia di coordinate per il punto a centinaia di coordinate per le spezzate e i poligoni
2. Imprecisione nelle coordinate: rappresentazione approssimata dei numeri reali;
3. I valori geometrici sono fortemente correlati in quanto inseriti nel medesimo spazio di riferimento: spesso i valori geometrici condividono degli elementi, spesso esistono vincoli spaziali che i valori devono rispettare.

# Un modello fisico di riferimento per Geo-DBMS

## La rappresentazione dei valori geometrici in un Geo-DBMS

Metodo 1: insiemi di liste di coordinate

Esempio



---

*Livello fisico*

$G1 = (\text{ln}, (<0,0>, <1,2>, <1,7>, <13.11>, <15,10>, <20,4>, <24,6>))$

$G2 = (\text{ln}, (<-2,10>, <1,7>, <13,11>, <14,13>, <20,13>))$

$G3 = (\text{pg}, (<5,14>, <10,13.5>, <11,12>, <13,11>, <1,7>, <0,10>, <0,12.5>))$

### Vantaggi:

- 1) Indipendenza tra i valori geometrici  $\Rightarrow$  facilità di inserimento di nuovi valori geometrici. Verifica vincoli spaziali a posteriori.
- 2) Buone prestazioni dell'operazione di visualizzazione grafica della geometria di un sottoinsieme limitato di valori.

### Svantaggi:

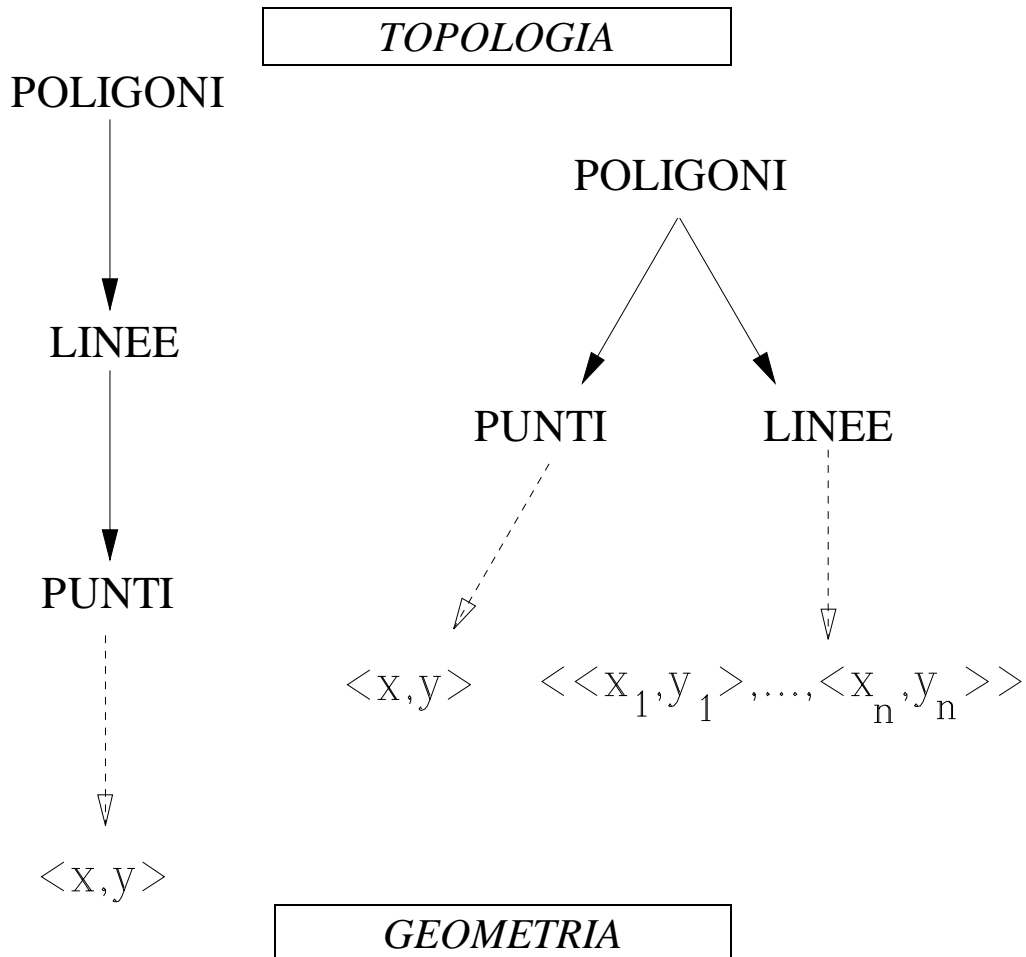
- 1) Tutte le selezioni spaziali richiedono l'applicazione di algoritmi di geometria computazionale.
- 2) Tutte le operazioni di manipolazione (ad esempio il calcolo di intersezioni) richiede algoritmi di geometria computazionale.
- 3) Ridondanza nella rappresentazione della geometria.



# Un modello fisico di riferimento per Geo-DBMS

## La rappresentazione dei valori geometrici in un Geo-DBMS

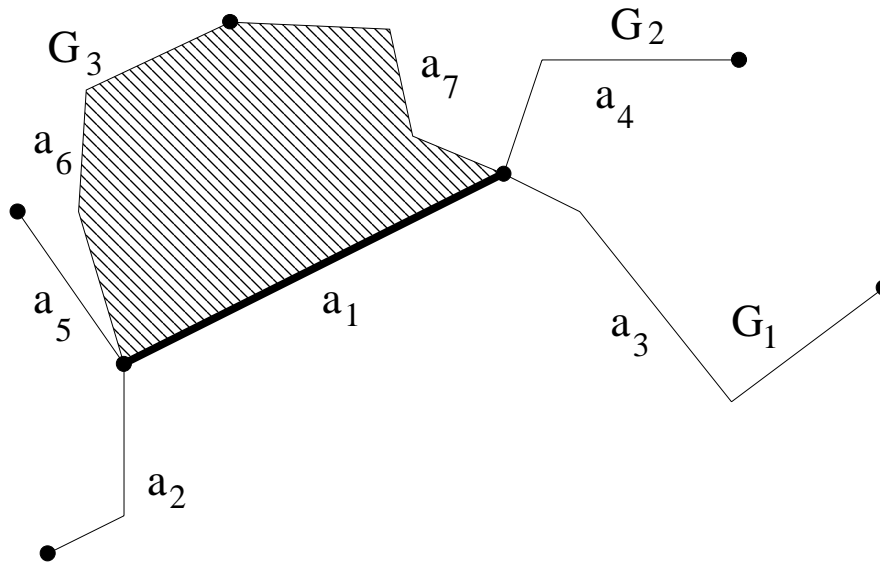
Metodo 2: struttura basata sulla topologia



# La rappresentazione dei valori geometrici in un Geo-DBMS

Metodo 2: struttura basata sulla topologia

Esempio



*Livello fisico*

$$G1 = \langle L, \{a_1, a_2, a_3\} \rangle$$

$$G2 = \langle L, \{a_1, a_4, a_5\} \rangle$$

$$G3 = \langle P, \{a_1, a_6, a_7\} \rangle$$

$$a_1 = (P_2, (), P_3)$$

$$a_2 = (P_1, (\langle 1, 2 \rangle), P_2)$$

$$a_3 = (P_3, (\langle 15, 10 \rangle, \langle 20, 4 \rangle), P_4)$$

$$a_4 = (P_3, (\langle 14, 13 \rangle), P_5)$$

$$a_5 = (P_6, (), P_2)$$

$$a_6 = (P_2, (\langle 0, 10 \rangle, \langle 0, 12.5 \rangle), P_7)$$

$$a_7 = (P_7, (\langle 10, 13.5 \rangle, \langle 11, 12 \rangle), P_3)$$

$$P_1 = \langle 0, 0 \rangle \quad P_2 = \langle 1, 7 \rangle \quad P_3 = \langle 13, 11 \rangle$$

$$P_4 = \langle 24, 6 \rangle \quad P_5 = \langle 20, 13 \rangle \quad P_6 = \langle -2, 10 \rangle$$

$$P_7 = \langle 5, 14 \rangle$$

# La rappresentazione dei valori geometrici in un Geo-DBMS

## Metodo 2: struttura basata sulla topologia

### Vantaggi:

1. Eliminazione della ridondanza nella rappresentazione della geometria.
2. Riduzione di alcune operazioni geometriche a selezione di un insieme di identificatori o puntatori a valori geometrici elementari precalcolati.
3. Riduzione della ricerca via predicati topologici ad un confronto tra identificatori o puntatori a valori geometrici elementari precalcolati.
4. Verifica vincoli spaziali immediata e supportata dalla struttura topologica.

### Svantaggi:

1. Ripristino della topologia corretta a valle di ogni inserimento di un nuovo valore geometrico.
2. La pura visualizzazione di un valore geometrico richiede il ritrovamento della geometria attraverso una catena di puntatori (indirettezze).

## Un modello fisico di riferimento per Geo-DBMS

Quali strutture dati sono adottate nei Geo-DBMS presenti sul mercato?

*Passato:*

I sistemi più diffusi adottavano strutture basate sulla topologia (Arc/Info ESRI, OpenSicad SIEMENS) o consentivano la generazione di strutture simili (Mge INTERGRAPH).

*Attuale tendenza:*

I sistemi attuali (Spatial Data Engine e ArcView ESRI oppure GeoMedia INTERGRAPH) hanno abbandonato in parte le strutture basate sulla topologia preferendo la soluzione meno strutturata delle liste di coordinate accoppiate a metodi d'accesso spaziali (indici) più potenti.

### **PROBLEMA**

Come suddividere e memorizzare gli insiemi di valori geometrici presenti in una base di dati geografica?

Osservazioni:

- Non è possibile costruire un'unica struttura dati di tipo topologico e/o un unico indice spaziale per tutti i dati contenuti nel sistema.
- Non è sempre utile costruire una struttura topologica/indice per ogni attributo di tipo POINT, LINE o POLYGON contenuto in una relazione (tabella).

### **SOLUZIONE**

Suddivisione della base di dati in **strati** (layer).

Per i dati del medesimo strato viene mantenuta una struttura topologica e/o indice spaziale.

non esiste topologia/indice tra strati diversi.  
(Es. Arc/Info della ESRI, Mge INTERGRAPH)

# Un modello logico/fisico di riferimento per GEO-DBMS

## Riepilogo dei costrutti presenti nel modello logico/fisico di riferimento

### Domini di base

*Schema*: INTEGER, REAL, STRING, POINT, LINE e POLYGON.

### Relazione (tabella)

*Schema*: è un costruttore di tipo per tuple (record), dove gli attributi (campi) hanno come dominio un dominio di base.

*Istanza*: è un insieme di tuple (record), dove i valori sono presi dai domini degli attributi (campi).

### Strato

*Schema*: è una struttura dati con indice per la rappresentazione di insiemi di valori geometrici di uno o più tipi (POINT, LINE, POLYGON).

*Istanza*: è un insieme di valori geometrici con indice spaziale.

### Vincoli tra Strati e Attributi con dominio geometrico in uno schema:

- ogni strato presente in uno schema contiene solo i valori geometrici di uno o più attributi geometrici appartenenti a relazioni dello schema.
- i valori di un attributo geometrico sono contenuti in uno e un sol strato.

Da ciò deriva che:

- in uno schema che presenti relazioni (tabelle) con attributi geometrici deve essere presente almeno uno strato.
- in uno schema che presenti relazioni (tabelle) con attributi geometrici, va indicata la corrispondenza attributo geometrico/strato.

# Un modello logico/fisico di riferimento per GEO-DBMS

## Esempio di schema logico/fisico di una base di dati geografica

### Relazioni

BOSCO (Tipo: STRING, Pg: POLYGON)

PRATO (Pascolo: BOOLEAN, Pg: POLYGON)

### Strati

S\_MONTAGNA ({POLYGON})

### Mapping Attributi geometrici/Strati

BOSCO.Pg → S\_MONTAGNA

PASCOLO.Pg → S\_MONTAGNA

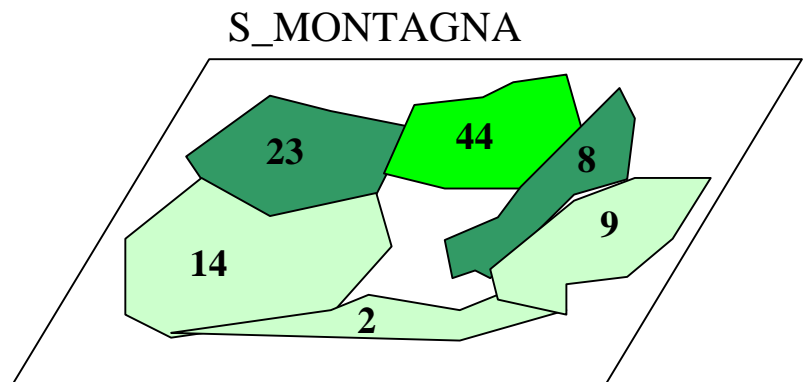
## Esempio di istanza di una base di dati geografica

### BOSCO

Tipo	Pg
ceduo	8
conifere	23
misto	44

### PRATO

Pascolo	Pg
true	9
false	2
false	44
true	14



# Un modello logico/fisico di riferimento per GEO-DBMS

## (B) Implementazione del legame (link) tra valori negli attributi geometrici e valori negli strati.

Tecniche utilizzate nei GEO-DBMS

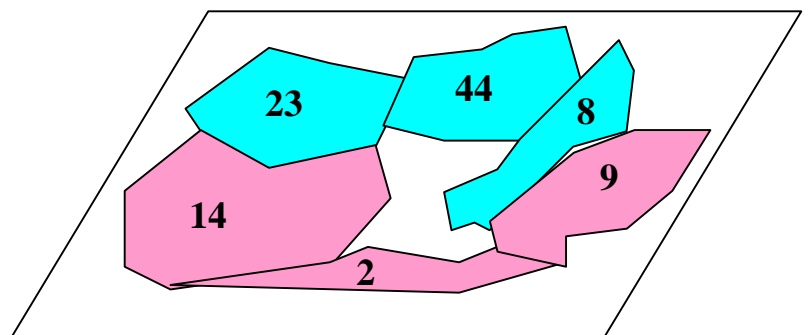
- i valori geometrici vengono memorizzati negli strati e negli attributi geometrici si riporta un “puntatore” al valore geometrico presente nello strato;
- il puntatore è solitamente un identificatore numerico (Geo-ID o GID);
- negli strati è possibile che un valore geometrico abbia più di un identificatore: in tal modo è possibile costruire valori geometrici per aggregazione (composizione) di valori geometrici più elementari.

Esempio

E<sub>1</sub>

Attributo <sub>1</sub>	Pg
A	1000
B	1001

STRATO<sub>1</sub>



E<sub>2</sub>

Attributo <sub>1</sub>	Pg
123	9
23	2
4	44
121	23
101	8
22	14

Tabella di aggregazione

GIDaggr	GID
1001	9
1001	2
1000	44
1000	23
1000	8
1001	14

# Un modello logico/fisico di riferimento per GEO-DBMS

## Specifica delle aggregazioni nello schema logico.

Per tenere conto della presenza di un vincolo di composizione tra attributi geometrici, estendiamo la sintassi del mapping attributi geometrici/strati. Aggiungiamo in particolare la possibilità di indicare che un attributo geometrico contiene valori che si ottengono dall'aggregazione di altri valori geometrici presenti su uno strato introducendo la parola chiave:

**Aggregate**(<Nome Strato>).

Esempio

### *Relazioni*

E<sub>1</sub> (Attributo<sub>1</sub>: STRING, Pg: POLYGON)

E<sub>2</sub> (Attributo<sub>1</sub>: INTEGER, Pg: POLYGON)

### *Strati*

STRATO<sub>1</sub> ({POLYGON})

### *Mapping Attributo geometrico/Strato*

E<sub>1</sub>.Pg → Aggregate(STRATO<sub>1</sub>)

E<sub>2</sub>.Pg → STRATO<sub>1</sub>



## Un modello logico/fisico di riferimento per GEO-DBMS

### Implementazione del legame (link) tra valori negli attributi geometrici e valori negli strati (soluzione alternativa).

Nel caso di valori geometrici di tipo LINE, alcuni sistemi realizzano il legame con gli strati sulla base di una ascissa curvilinea definita sulle linee che appartengono allo strato.

E' un modo alternativo di realizzare l'aggregazione di elementi geometrici.

Esempio

*Relazioni*

STRADA (Codice: STRING, Percorso: LINE)

TRATTO (N\_Corsie: INTEGER, Tracciato: LINE)

*Strati*

S\_STRADE ({LINE})

*Mapping Attributo geometrico/Strato*

STRADA.Percorso → Aggregate(S\_STRADE)

TRATTO.Tracciato → S\_STRADE

STRADA

Codice	Percorso
SS11	23
SP23	10

TRATTO

N_Corsie	Ln
1	23(41-100)
2	23(0-18)
3	23(18-41)
1	10(23-100)
2	10(0-23)

S\_STRADE

