

# Les modèles numériques des données spatiales

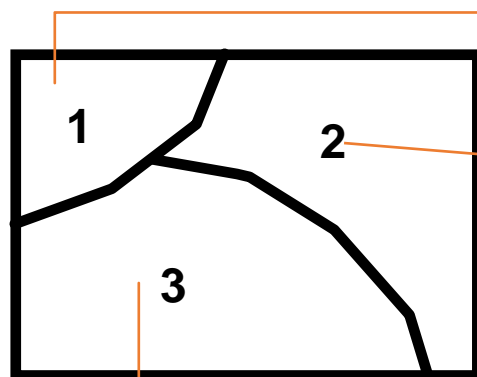
# Le mode vecteur

# Le mode vecteur

- Les entités de références du monde réel sont représentées sous forme de points, de lignes ou de polygones.
  - Les points identifient des lieux sous forme de coordonnées.
  - Les lignes connectent les points.
  - Les polygones sont formés par les lignes connectées entre elles.
- Il est plus proche d'une vision des phénomènes sous forme d'entité (mais pas exclusivement).

# Structure par couches vecteur

- Un identifiant commun permet de relier les identifiants des objets géométriques à des tables d'attribut :
- La structure la plus classique d'une base de données géographiques vecteur est de la forme : couches géométriques vecteurs ↔ tables SGBD
- A chaque entité géométrique correspond une ligne d'une table dans le SGBD
- Ex : Table d'attribut des objets polygones

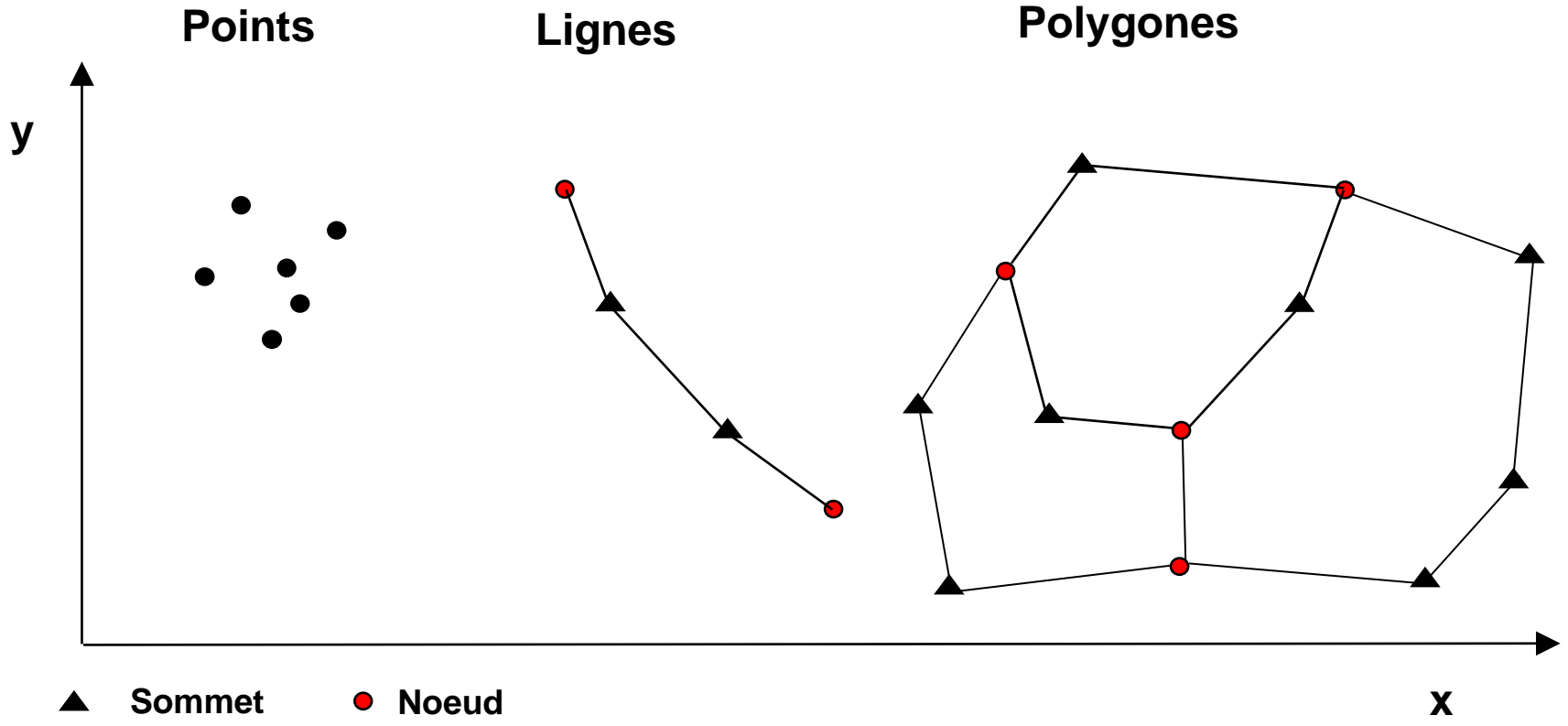


*Table (SGBD)*

<b>Poly id</b>	<b>Aire</b>	<b>Pop.</b>	<b>Tx. Cr.</b>
<b>1</b>	<b>297</b>	<b>4583</b>	<b>2.7</b>
<b>2</b>	<b>607</b>	<b>3927</b>	<b>3.1</b>
<b>3</b>	<b>806</b>	<b>9271</b>	<b>3.8</b>

- De la même manière, on pourra définir une table d'attributs de points ou de lignes

# Le mode vecteur

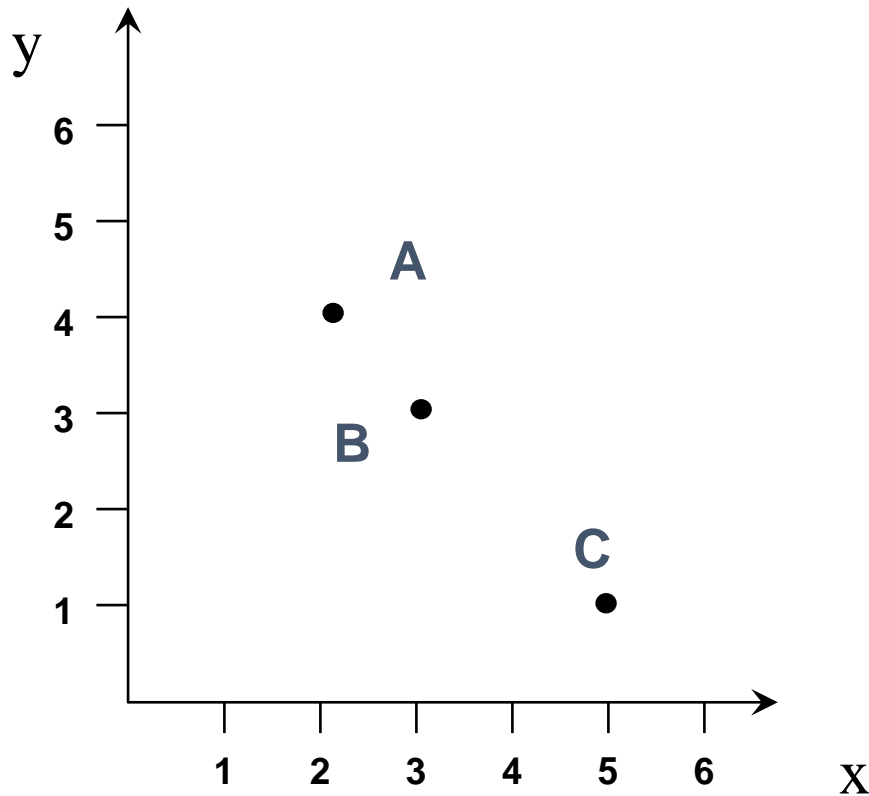


# Le mode vecteur

- Les entités spatiales sont définies par les coordonnées de leur contour dans un système de coordonnées cartésiennes
  - Les points sont stockés comme des paires de coordonnées  $x,y$
  - Les lignes sont des séries de paires de coordonnées  $x,y$
  - Un polygone est une série de coordonnées  $x,y$  dont la première et la dernière paire de coordonnées sont identiques.
- La précision dans la localisation de ces coordonnées est virtuellement infinie et ne dépend que de la puissance de la machine
- Elle est dans la pratique limitée par les sources des données.

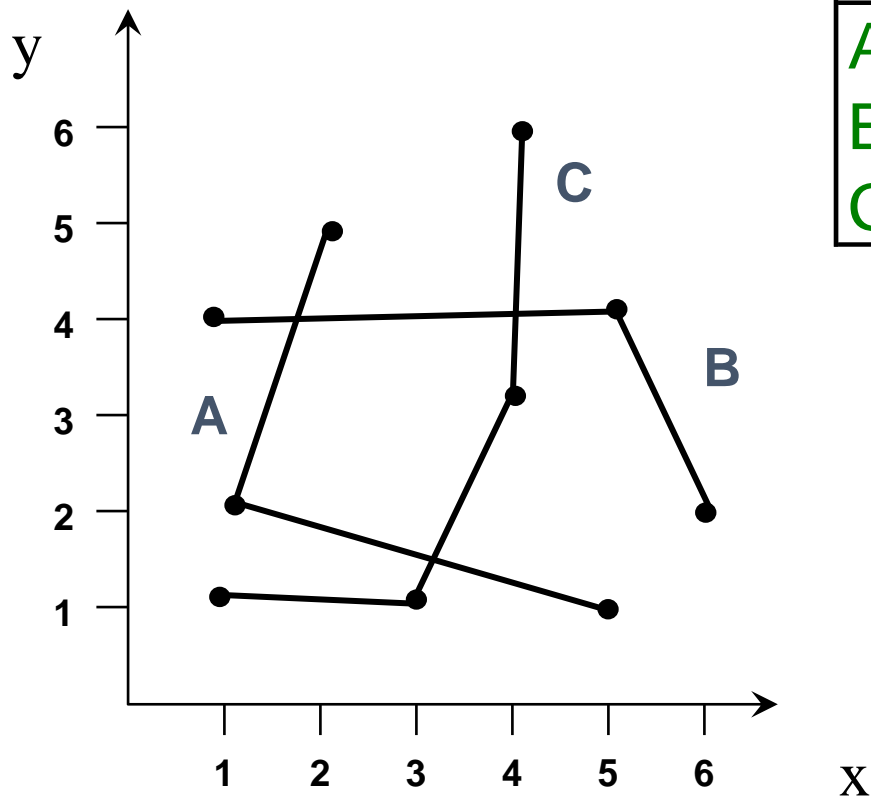
# Stockage des objets ponctuels en mode vecteur

Point	Coordonnées
A	(2,4)
B	(3,3)
C	(5,1)



***Les points sont stockés comme des paires de coordonnées x,y***

# Stockage des objets linéaires en mode vecteur



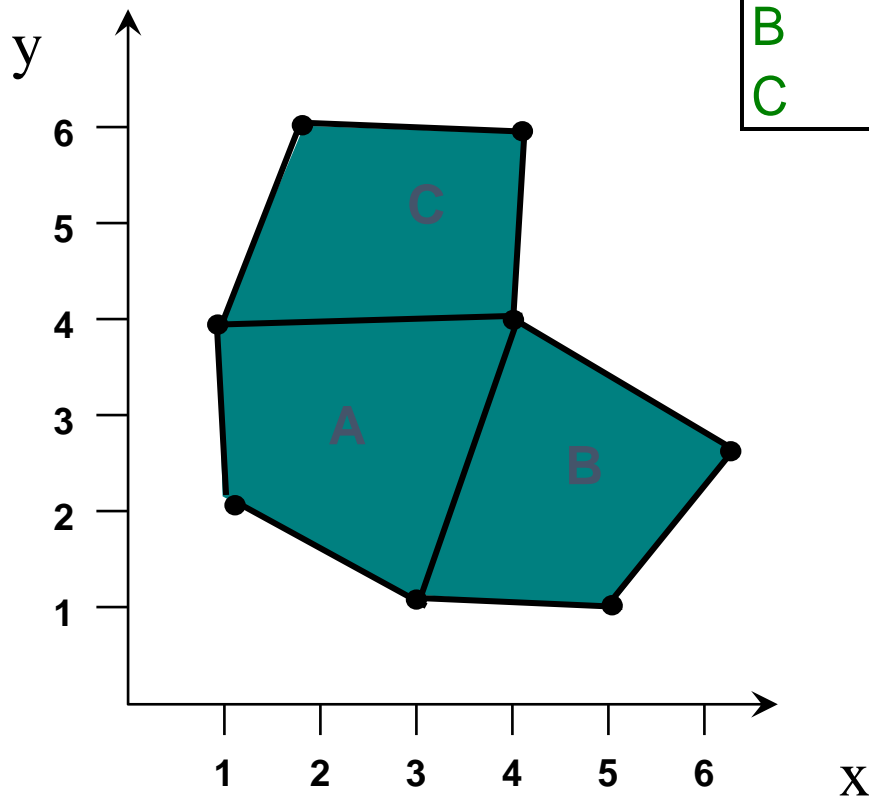
Ligne	Coordonnées
A	(2,5), (1,2), (5,1)
B	(1,4), (5,4), (6,2)
C	(1,1), (3,1), (4,3), (4,6)

*Les lignes sont des séries de paires de coordonnées x,y*



# Stockage des objets surfaciques en mode vecteur

Polygone	Coordonnées
A	(1,4), (4,4), (3,1),(1,2),(1,4)
B	(4,4), (6,3), (5,1),(3,1),(4,4)
C	(4,4), (1,4), (2,6),(4,6),(4,4)



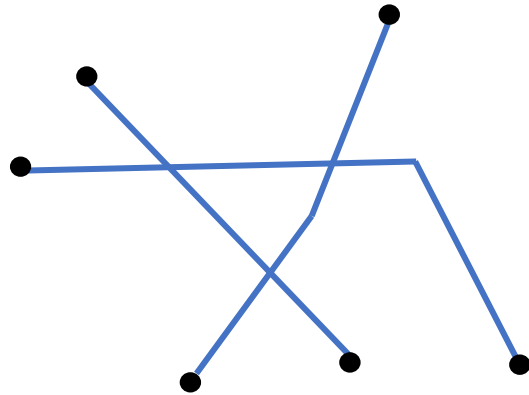
***Les surfaces sont stockées comme une série de coordonnées x,y dont la première et la dernière paire de coordonnées sont identiques.***

# Différentes structures de données vecteur

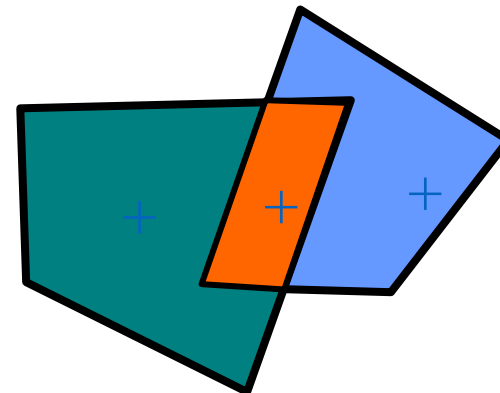
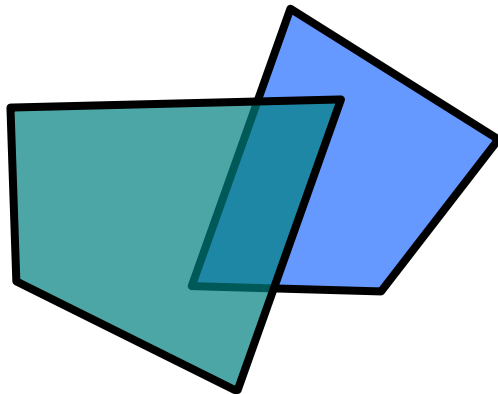
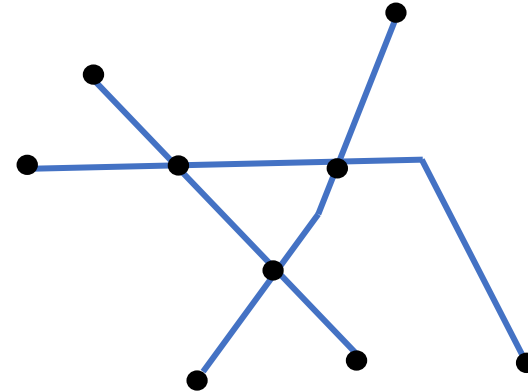
- Structure de données spaghetti
  - Pas de relations spatiales explicites entre les entités
- Structure de données topologique
  - Relations spatiales explicites (intersections, contigüité, superposition)

# Topologie planaire et non-planaire

Non planaire

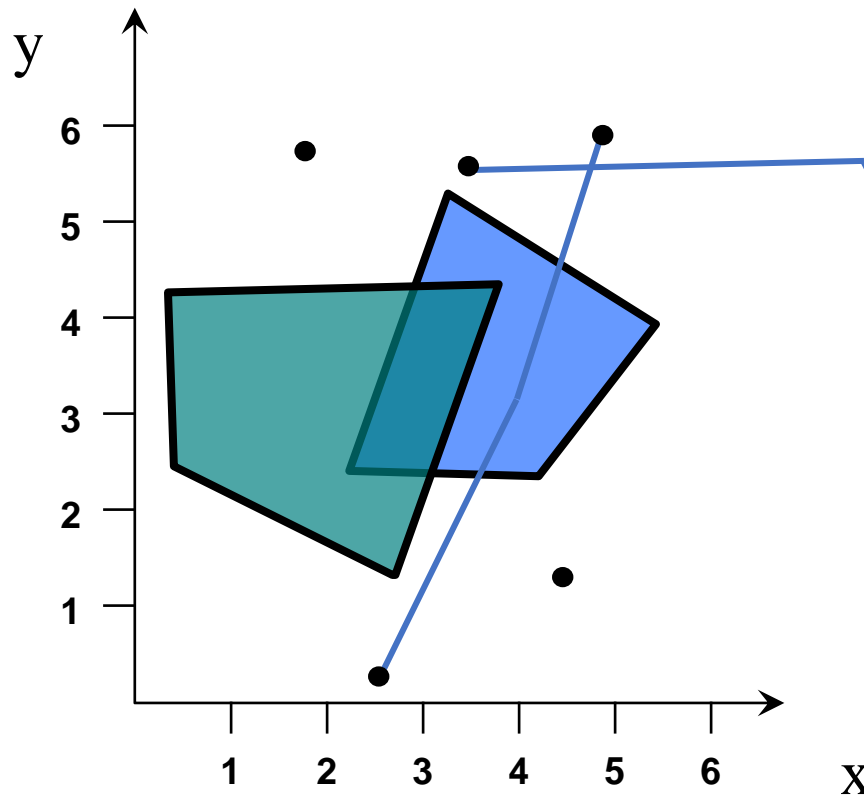


Planaire



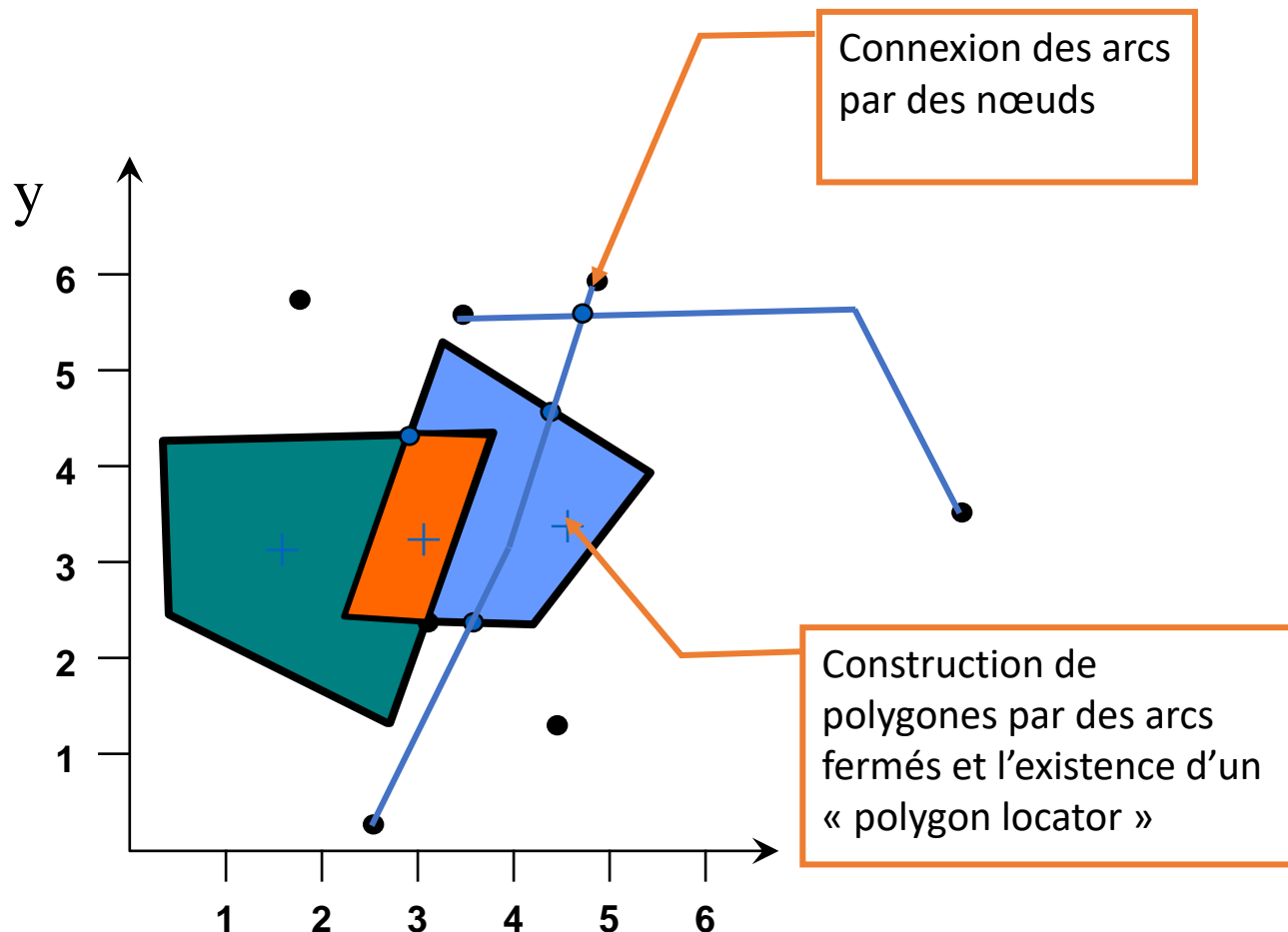
# Structure de données spaghetti

On appelle modèle de données "Spaghetti" une structuration de données qui considère les objets de différents types (points, lignes, surfaces) comme isolés et n'entretenant aucune relation

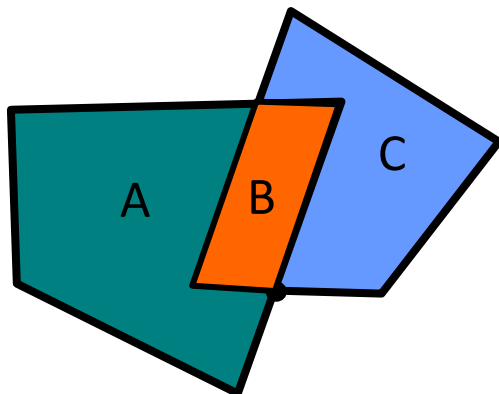


# Structure de données topologique

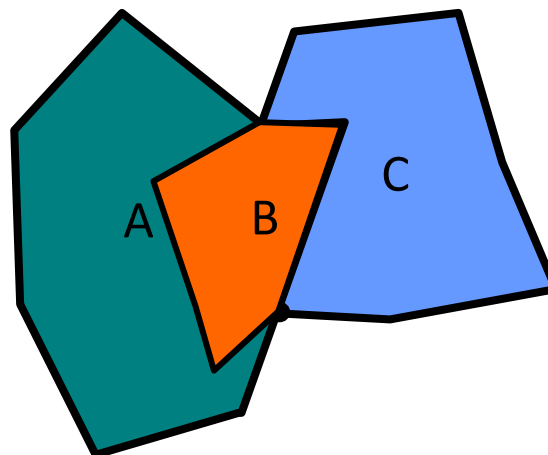
Les entités spatiales sont séparées comme les relations qu'elles entretiennent dans le plan



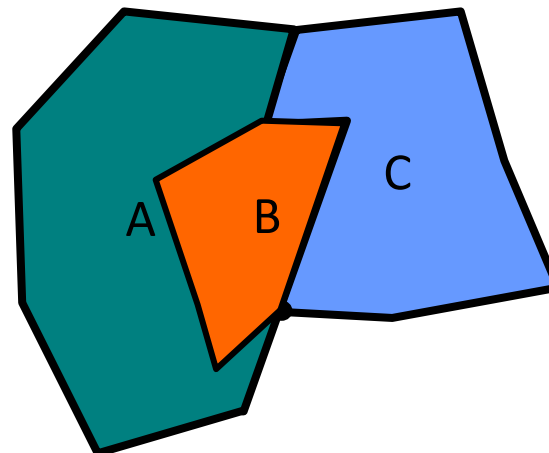
# Structure de données topologique



A voisin de B  
C voisin de B



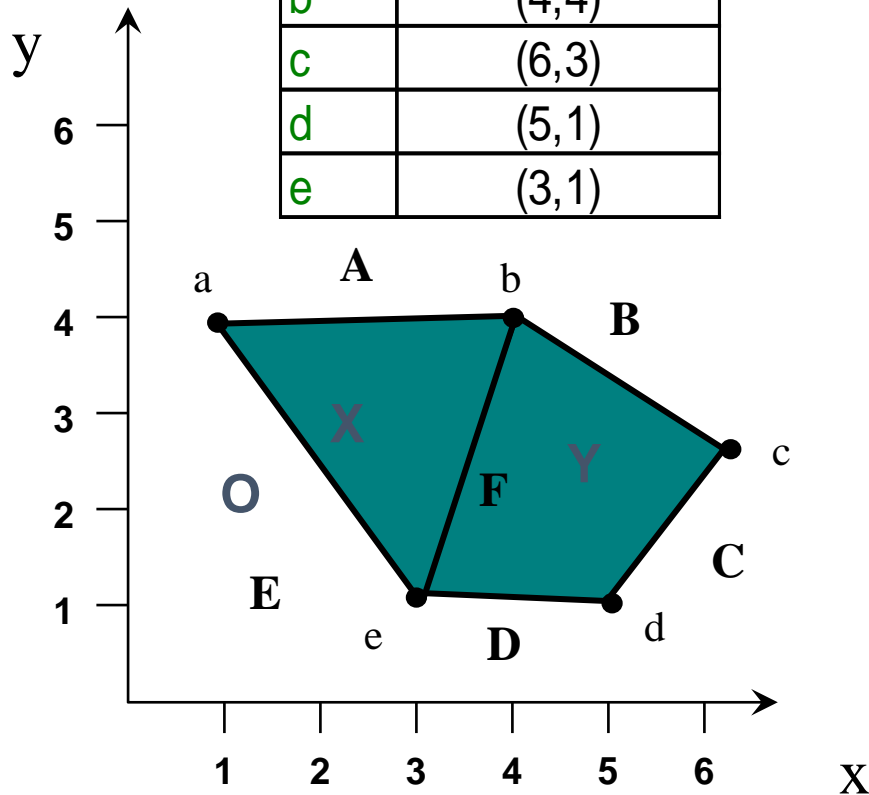
A voisin de B  
C voisin de B



A voisin de B  
C voisin de B  
A voisin de C

# Exemple de structure topologique

Point	Coordonnées
a	(1,4)
b	(4,4)
c	(6,3)
d	(5,1)
e	(3,1)



Arc	de	à	A droite	A gauche
A	a	b	X	O
B	b	c	Y	O
C	c	d	Y	O
D	d	e	Y	O
E	e	a	X	O
F	b	e	X	Y

X	A, F, E
Y	B, C, D, F
O	A, B, C, D, E

- Modèle topologique d'objets surfaciques type Arc-Nœud-Polygone (Arcinfo)

# L'intérêt d'une structure topologique de données

- Elle évite les redondances (pas d'arcs ou de nœuds dupliqués) et économise la place mémoire
- Elle facilite les traitements spatiaux fondés sur des relations topologiques (connexion, contiguïté) :
  - Quelles sont les routes qui débouchent sur la place centrale ?
  - Quelles sont les routes que je dois prendre pour aller à la mairie ?
  - Quelle est l'occupation du sol de la parcelle voisine ?



# Quelle structure de données vecteur choisir ?

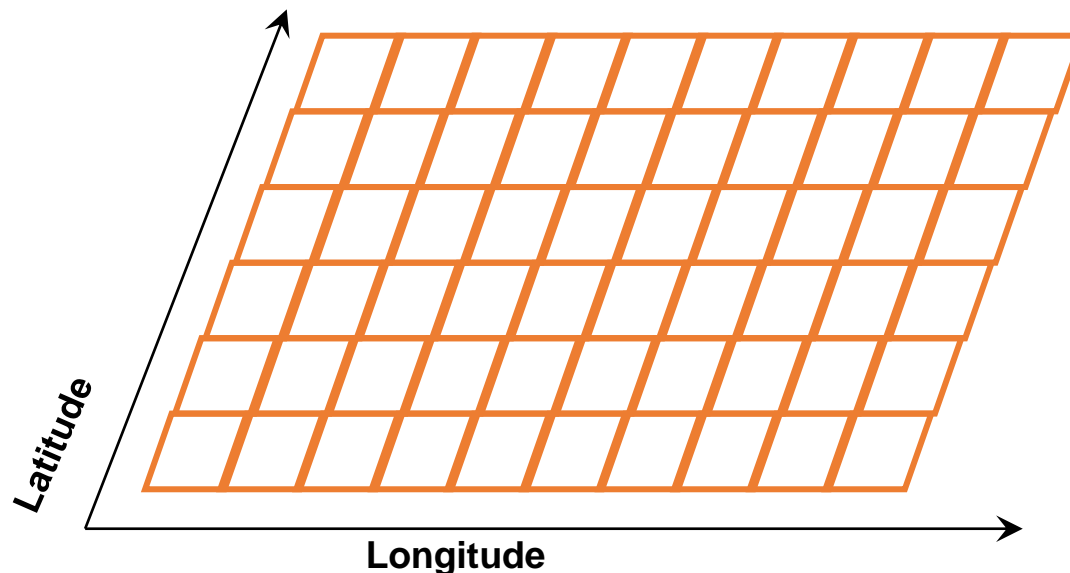
- Structure Spaghetti
  - plus simple
  - permet les manipulations graphiques standards
  - les relations spatiales de type connectivité, contiguïté doivent être effectuées à chaque analyse.
- Structure topologique
  - stocke explicitement les relations spatiales
  - l'analyse spatiale peut se faire sans accéder aux coordonnées
  - La structure de données est plus complexe
  - La topologie doit être rétablie à chaque mise à jour.

# Le mode raster

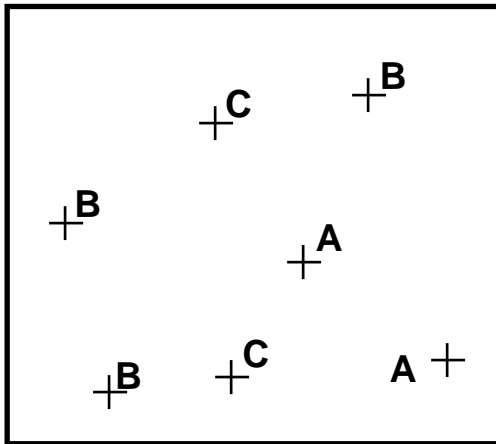
# Le mode de données raster (maillé)

Dans ce modèle, l'espace est découpé selon une grille régulière de taille prédéterminée (la résolution)

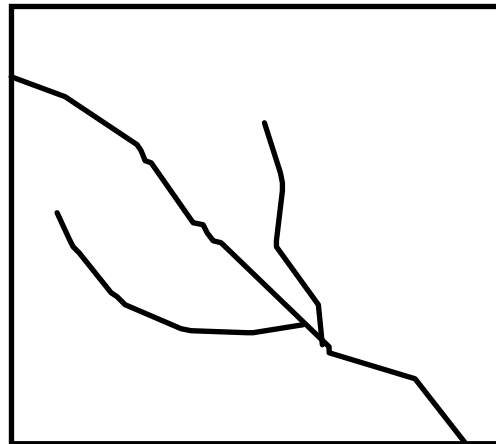
- A chaque maille est attribuée une valeur numérique et une seule, pouvant correspondre à une mesure (pollution, altitude), à une catégorie (type de végétation) ou à l'identifiant d'un objet (numéro d'une école, d'une commune, d'une route,...)



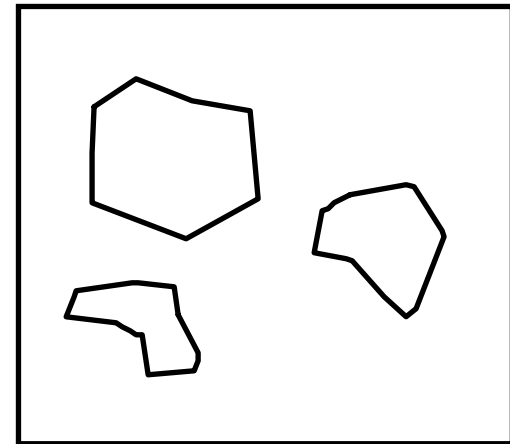
# Le mode de données raster (maillé)



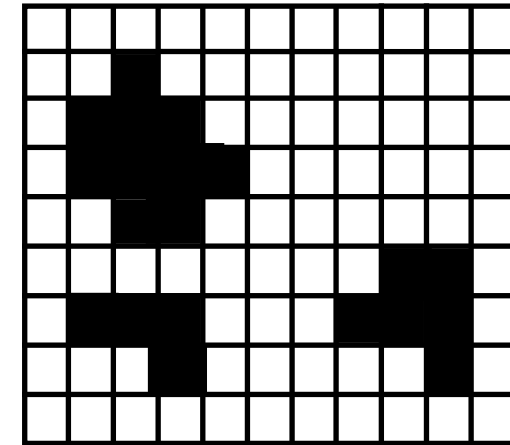
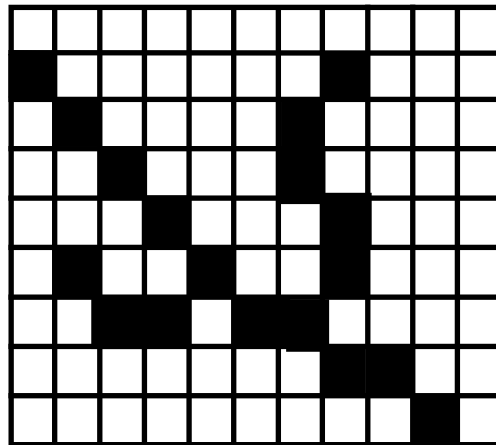
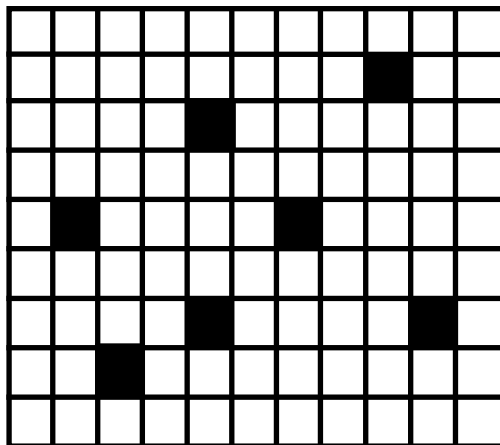
**Points**



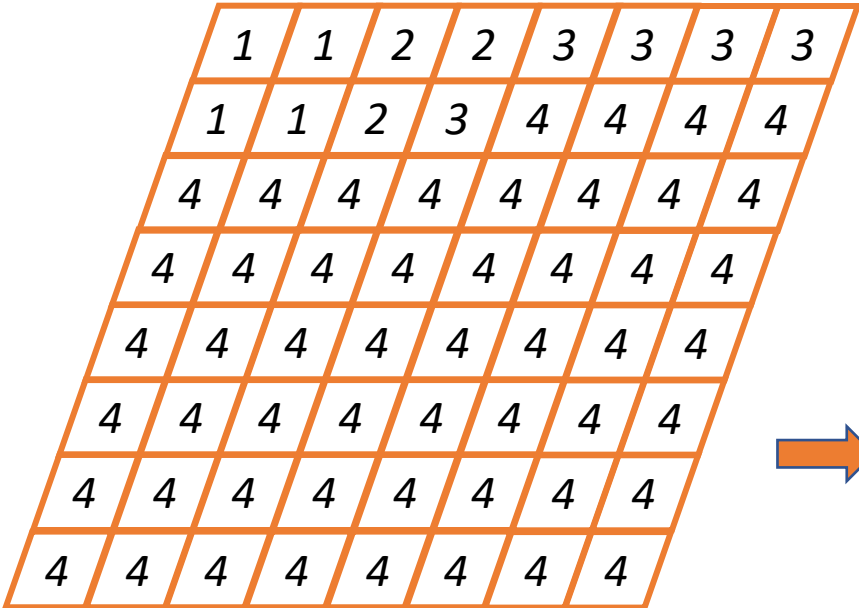
**Lignes**



**Polygones**

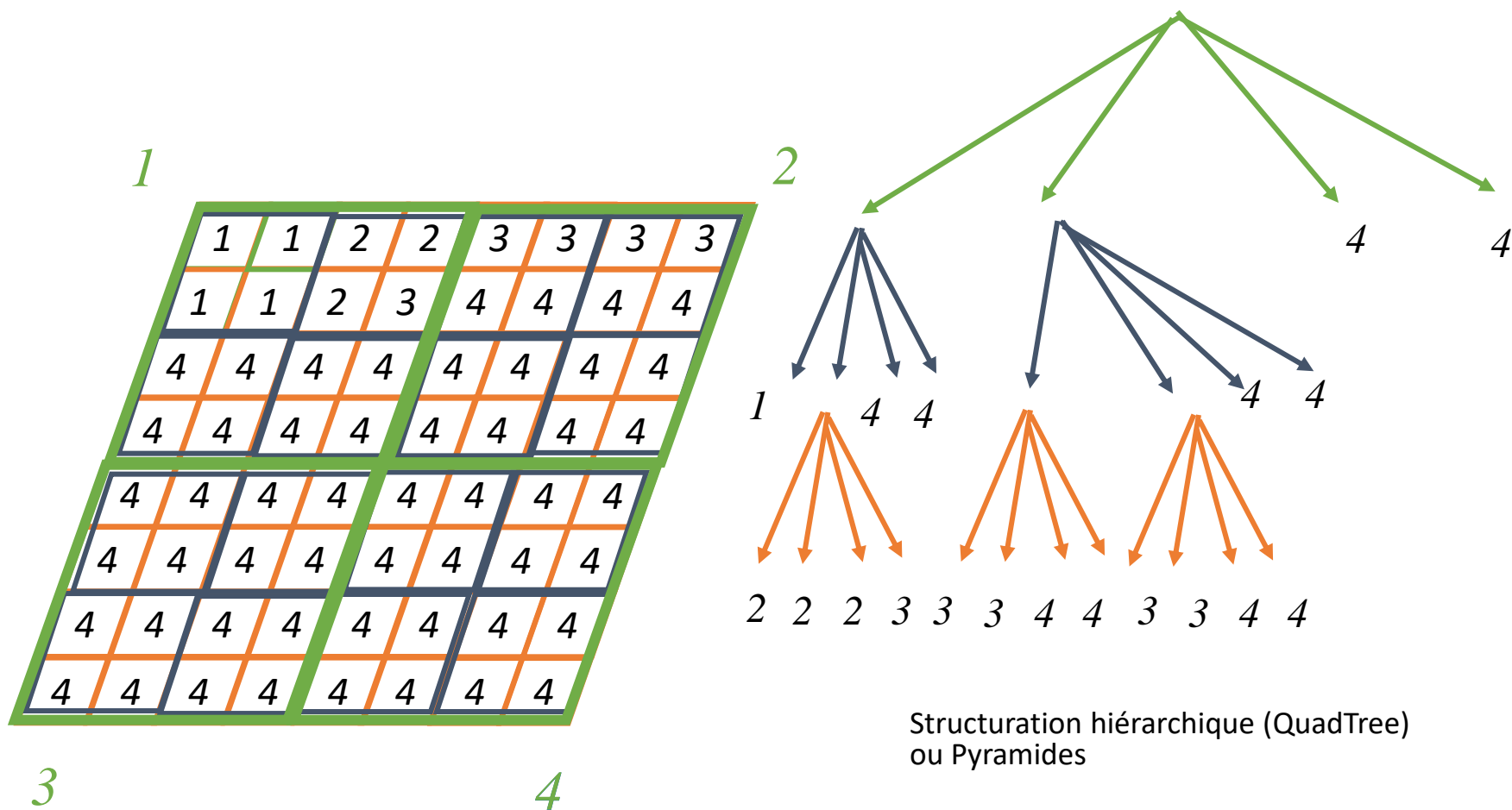


# Structures de données raster



1		
1	<i>2x1</i>	<i>2x1</i>
2	<i>2x2</i>	<i>2x2</i>
3	<i>4x3</i>	<i>2x3</i>
3	<i>2x1</i>	<i>2x1</i>
3	<i>1x2</i>	<i>1x2</i>
3	<i>1x3</i>	<i>1x3</i>
1	<i>4x4</i>	<i>52x4</i>
1		
2	<i>8x4</i>	
3	<i>8x4</i>	Fichier avec compression d'ensemble
4	<i>8x4</i>	
4	<i>8x4</i>	
4	<i>8x4</i>	
4	<i>8x4</i>	
4	<i>8x4</i>	Fichier avec compression par ligne
4		
4		
4		Fichier simple (les valeurs l'une après l'autre)
4		
4		
4		
4		
...		

# Structures de données raster



# Comparaison et combinaison des modes vecteurs et raster

# Avantages/inconvénients du modèle vecteur

- Avantages
  - logique naturelle de gestion par objet
  - qualité de la cartographie
  - interface simplifiée avec les bases de données
  - stockage peu volumineux
  - Adapté quand on a de multiples attributs décrivant une même couches d'entités géométriques
- Inconvénients
  - logiciels plus sophistiqués
  - difficile de gérer les données à variation spatiale continue



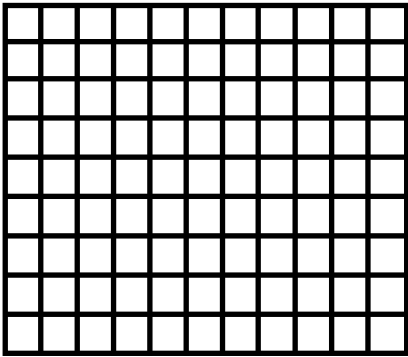
# Avantages/inconvénients du modèle maillé

- Avantages :
  - simplicité
  - Plus faible coût des logiciels (certains gratuits)
  - intègre les images scannées, les images satellitaires
  - Adapté quand on a de multiples couches d'entités géométriques, avec peu d'attributs associés à chaque couche.
- Inconvénient :
  - taille des fichiers
  - cartographie peu agréable
  - résolution à fixer a priori

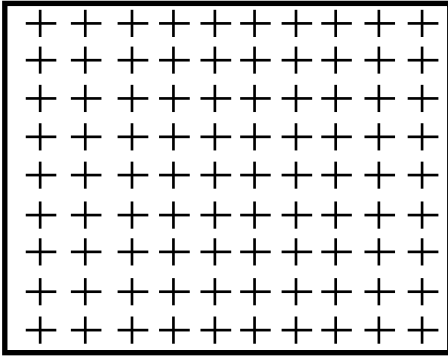
# Gérer les champs (données à variation spatiale continue)

- On peut stocker les données aussi bien en raster qu'en vecteur
- Le modèle maillé est bien adapté à la gestion des champs
- Le modèle vecteur est plus orienté gestion d'entités
  - Mais il existe des méthodes de gestion de données à variation continue en mode vecteur
  - Et l'on peut gérer des entités en mode maillé.

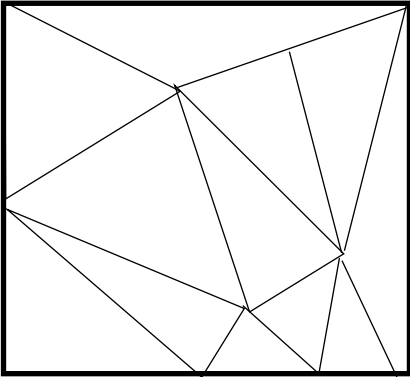
# Gérer les champs (données à variation spatiale continue)



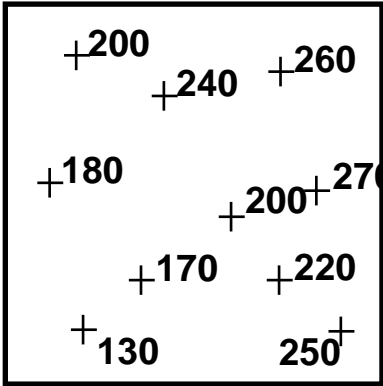
**Grille raster**



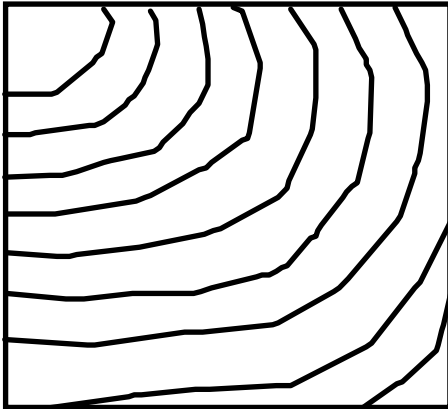
**Points réguliers**



**Réseau de triangles  
Irréguliers. TIN**

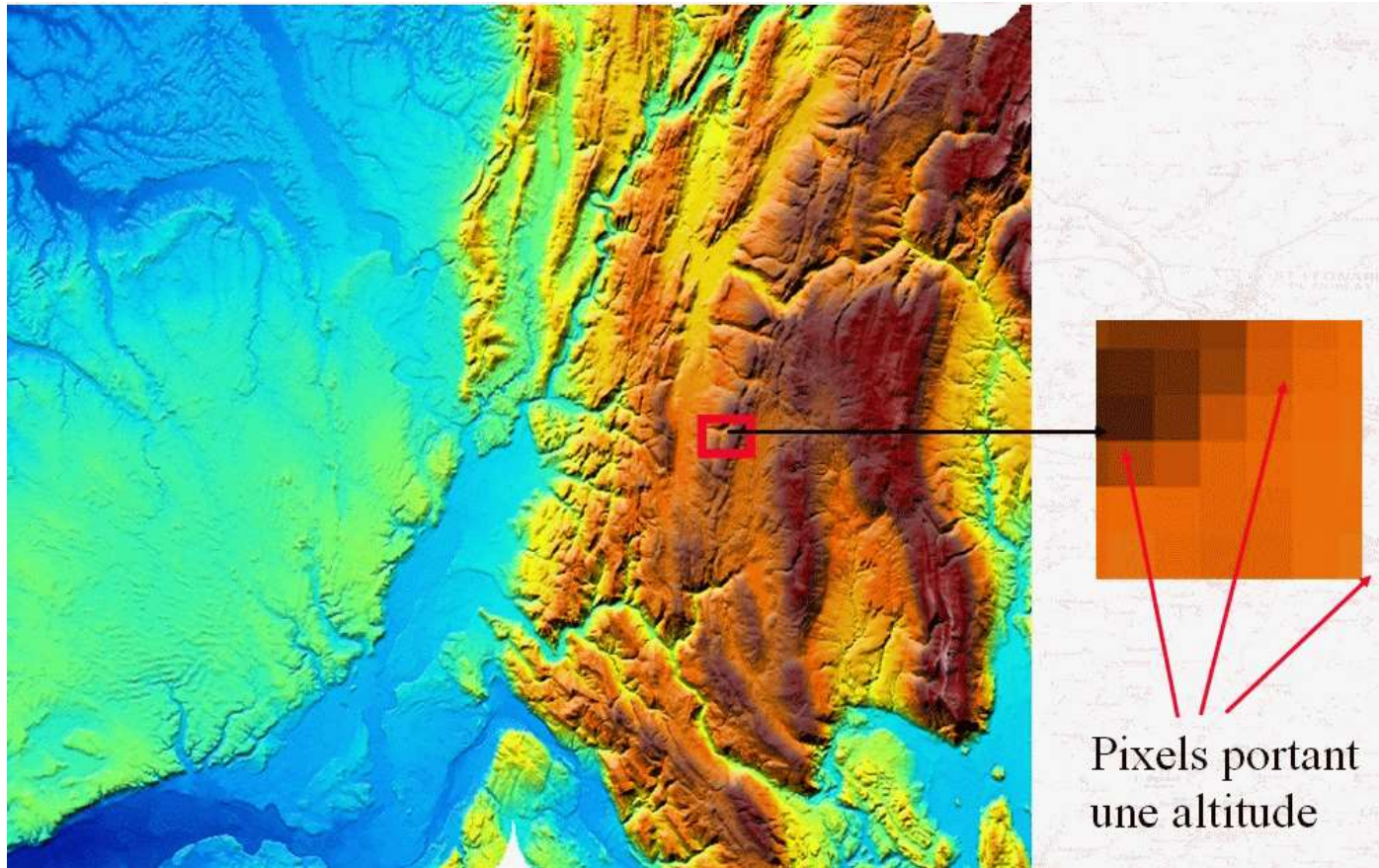


**Points irréguliers**



**Isolignes**

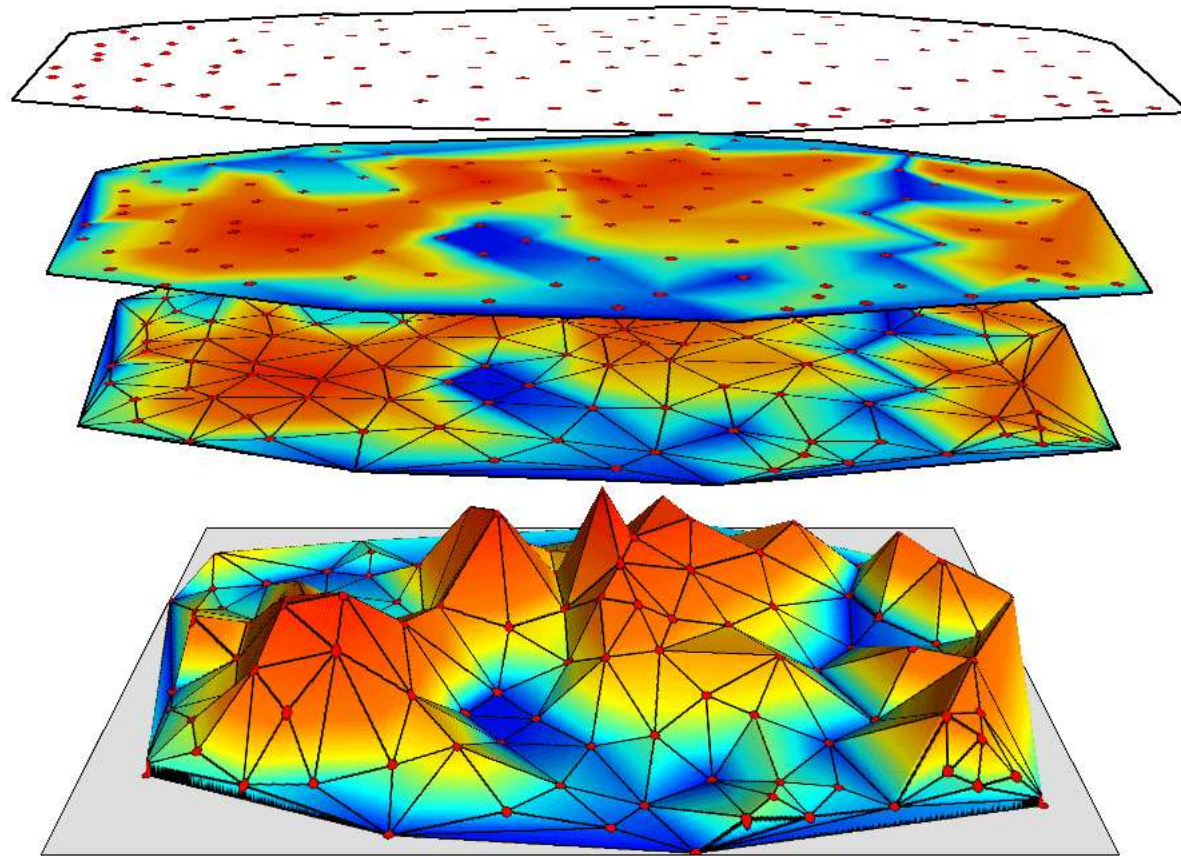
# Altitude (modèle raster)



[http://www.ente-aix.fr/documents/118-demoGeo/demo/4\\_BasesIG/res/i20\\_DonneesRaster3\\_2.gif](http://www.ente-aix.fr/documents/118-demoGeo/demo/4_BasesIG/res/i20_DonneesRaster3_2.gif)

Thierry Joliveau. Concepts, méthodes et usages des SIG .UJM 2002-2024

# Altitude (modèle TIN)



[http://www.geosolutions.com/3d/analyse/images/interpolate\\_tin.gif](http://www.geosolutions.com/3d/analyse/images/interpolate_tin.gif)

# Les modèles de données attributaires

- Les données sont stockées séparément des coordonnées.
- Le niveau de structuration peut être plus ou moins élevé :
  - Fichier texte
  - Fichier Excel (Tableur)
  - Table d'un SGBD

# Les modèles de données attributaires

# La structure texte

- Avantages

- Lisible par tous les logiciels
- Simple à modifier

- Inconvénients

- Peu structurée donc interrogation et mise à jour difficiles
- Interrogation longue (accès séquentiel)
- Mal adapté à la gestion de grandes masses de données
- Intéressante pour les transferts de données (Format .csv)



# Structure SGBD

- Plusieurs types :
  - Hiérarchique,
  - Réseau,
  - Relationnel
  - Objet
- Les Systèmes de Gestion de Base de Données relationnelles (SGBD-R) sont les plus utilisés
  - normalisation : séparation des données en autant de tables qu'il y a d'entités logiques

# La structure relationnelle

- Avantages

- Données organisées dans des tables correspondant à des entités ou des relations entre entités
- Gestion de bases de données complexes
- Elimination des redondances des bases de données
- Mise à jour sécurisée

- Inconvénients

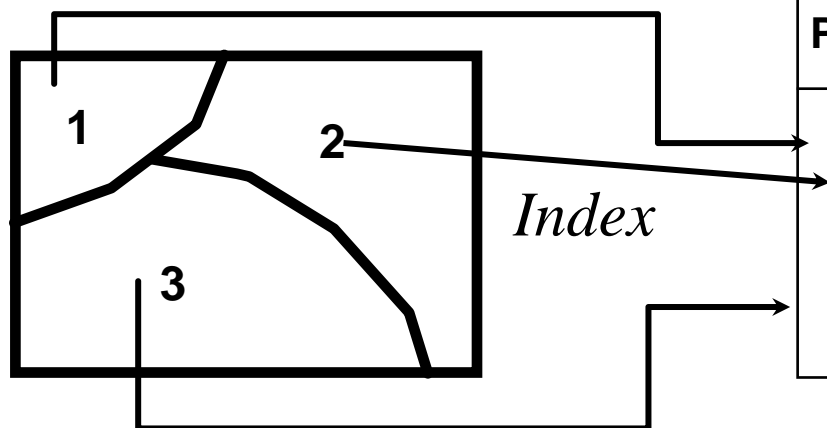
- Lentes quand les bases de données deviennent importantes
- Mal adapté à la gestion des relations spatiales entre les entités

# Stocker les données attributaires

- Une bonne organisation des données attributaires est cruciale dans un projet SIG.
- Dans certaines applications (par exemple les SIG socio-économiques), la composante attributaire est en volume plus importante que la composante spatiale.
- Le temps passé à décrire et organiser correctement sa base de données est toujours plus que compensé par le gain de temps au moment de la gestion et de l'exploitation.

# Structure par couches vecteur <> table

*Couche*



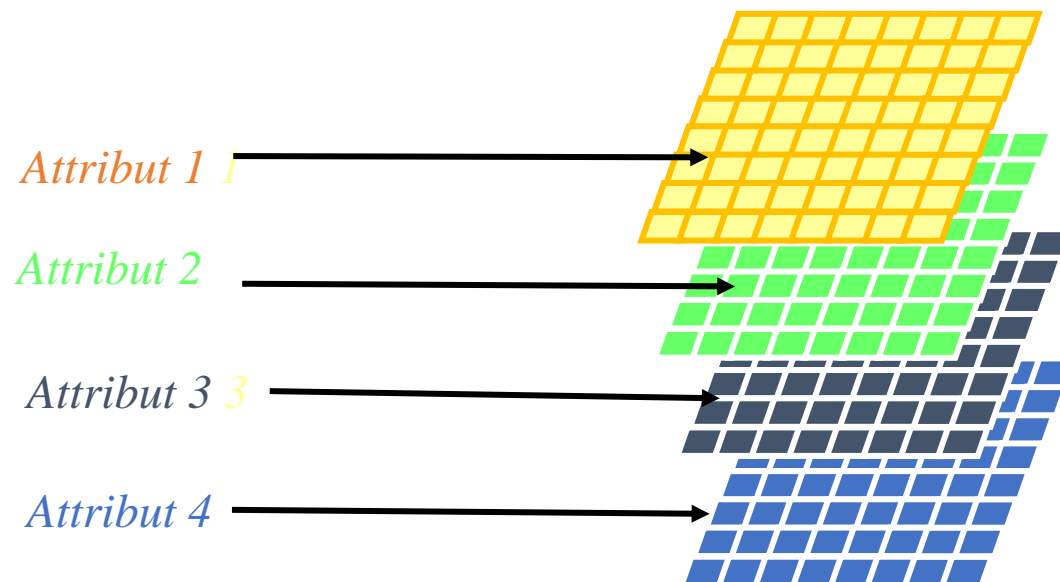
*Table*

Poly id	Aire	Pop.	Tx. Cr.
1	297	4583	2.7
2	607	3927	3.1
3	806	9271	3.8

- Modèle logique multifichiers (3 au moins) donc complexe et risqué à gérer

# Structure par couches raster

- En mode raster, les données attributaires sont placées directement dans les grilles sous forme de valeurs numériques.
- Il y a virtuellement autant de grilles que d'attributs

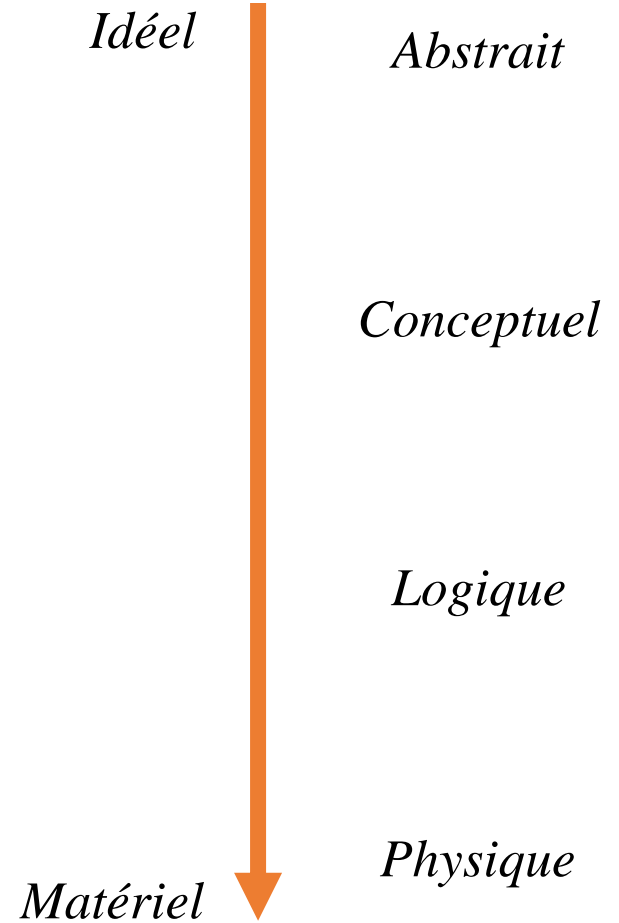


# Structure SGBD

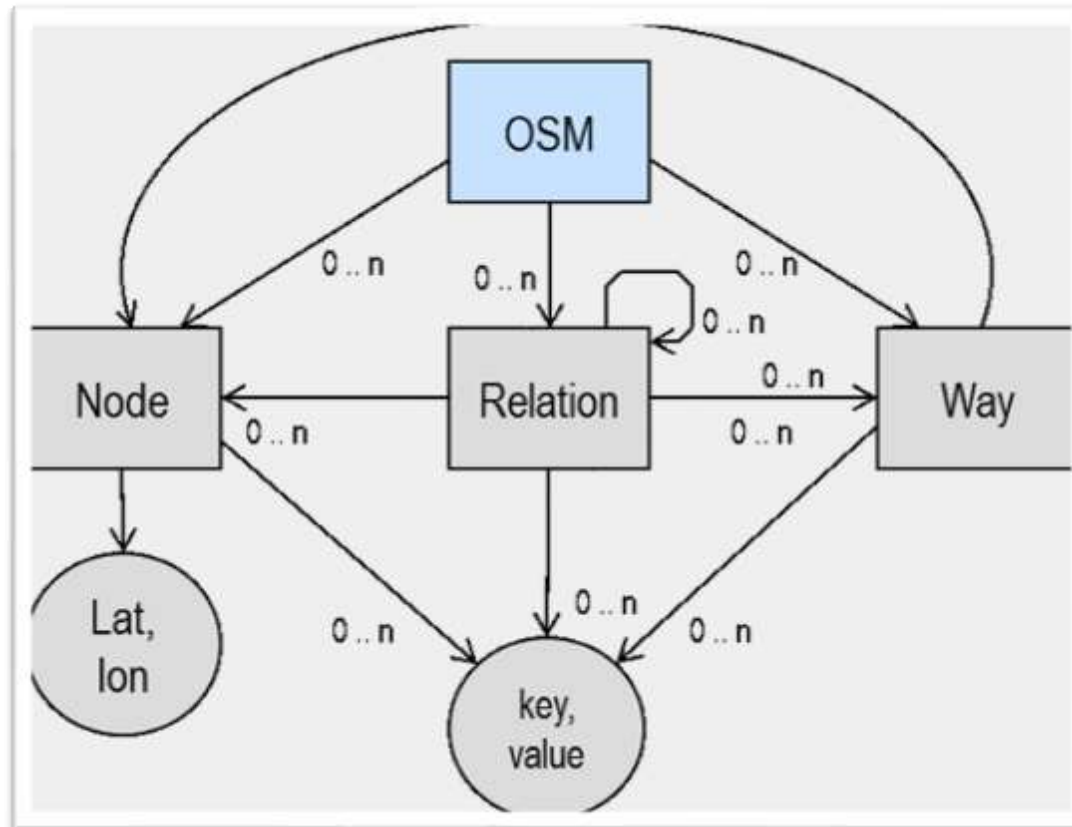
- Une structuration intégrée apparaît pour gérer des bases de données importantes :
  - Les structures de bases de données orientées objets permettent de gérer les objets complexes, appartenant à plusieurs thèmes et de mieux définir les relations spatiales (peu de produits commerciaux).
  - Les bases de données relationnelles ont élargi les structures supportés (vecteur, raster) et offrent la possibilité de gérer les données spatiales dans la même structure que les données attributaires :
    - Oracle Spatial, SDE, PostGIS
  - Avantage : cohérence de la maintenance de la base

# Les modèles de données

- Représentation du phénomène
  - Entités et relations
  - Champs
- Type de structuration des données
  - Données spatiales
    - mode vecteur topologique ou non
    - mode maillé (raster, trame)
  - Données attributaires
    - Fichier
    - Table SGBD
    - Objet
- Format des données
  - Binaire, ASCII
  - Propre à chaque logiciel
- Implémentation dans l'ordinateur



# Le modèle conceptuel d'OpenStreetmap



<https://learn.opengeoedu.de/en/opendata/vorlesung/freiwillig-erhobene-daten/openstreetmap/datenmodell>



# Un modèle logique : le format des fichiers Keyhole Markup Language (KML)

- **Format texte à balise XML**





```
<?xml version="1.0" encoding="UTF-8"?>  
<kml xmlns="http://www.opengis.net/kml/2.2">  
<Document>  
  <Placemark>  
    <name>New York City</name>  
    <description>New York City</description>  
    <Point>  
      <coordinates>-74.006393,40.714172,0</coordinates>  
    </Point>  
  </Placemark>  
</Document>  
</kml>
```

[http://en.wikipedia.org/wiki/Keyhole\\_Markup\\_Language](http://en.wikipedia.org/wiki/Keyhole_Markup_Language)

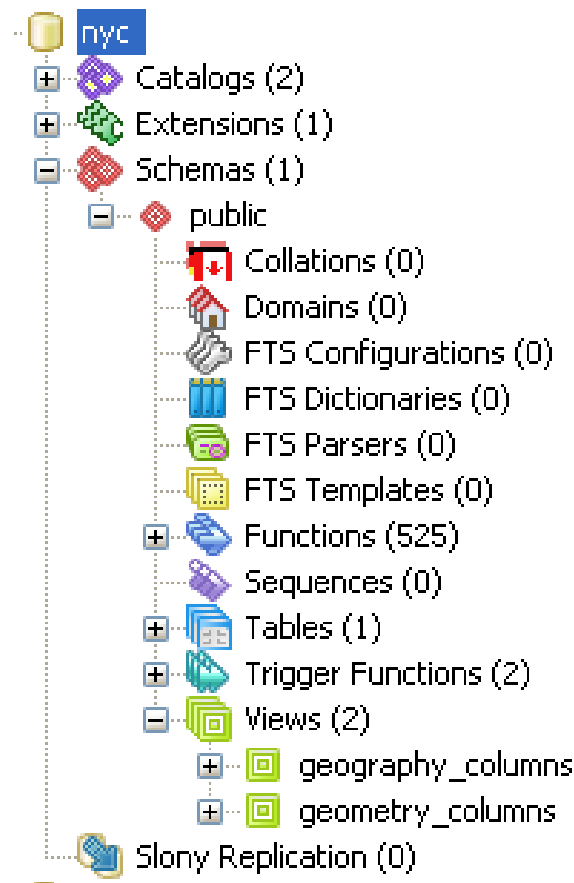


# Un autre modèle logique : le format GeoJSON

<https://www.codethemap.fr/2019/07/exploitez-des-donnees-libres-geojson.html>

Type	Exemples	
Point		<pre>Code {   "type": "Point",   "coordinates": [30, 10] }</pre>
Segments		<pre>Code {   "type": "LineString",   "coordinates": [     [30, 10], [10, 30], [40, 40]   ] }</pre>
Polygones		<pre>Code {   "type": "Polygon",   "coordinates": [     [[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]   ] }</pre>
		<pre>Code {   "type": "Polygon",   "coordinates": [     [[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]],     [[20, 30], [35, 35], [30, 20], [20, 30]]   ] }</pre>

# Un autre modèle logique : les schémas d'objets dans PostGIS



[http://workshops.boundlessgeo.com/postgis-intro/creating\\_db.html](http://workshops.boundlessgeo.com/postgis-intro/creating_db.html)