- **1. La conversion des modèles de données géographiques**
  - » Le double aspect des données géographiques – géométrique et attributaire – peut se traduire par :
    - Une double implémentation dans un modèle hybride (ou dual) : un SGBD relationnel pour les attributs « classiques » et un « autre système » pour les données géographiques.
    - Une implémentation dans un modèle unique ou intégré, où toutes les données sont rassemblées dans un modèle de type relationnel (étendu).
  - » Aujourd'hui, la présence d'architectures client-serveur (ou multiclients-multiserveurs) favorise un modèle unique relationnel ou objet-relationnel. Cette tendance est renforcée :
    - D'une part, par l'usage généralisé des données géographiques dans **toute** l'organisation.
    - D'autre part, par le coût de création/maintenance et les risques **d'incohérences** en cas de duplication des données dans des modèles duaux.
  - » Conséquences : de multiples variantes d'implémentation en évolution depuis le début des années 2000 :
    - depuis des solutions hybrides propriétaires légataires,
    - vers des solutions objet-relationnelles et une architecture répartie.

- **2. Modèle logique hybride sans topologie (shapefile)**
  - **2.1. Définition**
    - » Un modèle hybride, parfois qualifié de dual, se caractérise par une implémentation séparée des données géométriques, d'une part, et des données attributaires, d'autre part.
      - Les données **géométriques** sont gérées par un simple système de gestion de fichiers (SGF).
      - Les données **attributaires** sont gérées de manière relationnelle
    - » Un modèle hybride sans topologie regroupe parmi les données géométriques :
      - Les données de **position** (*ex.* x, y) des entités géographiques.
      - Les relations **logiques** (*ex. composition*) entre les entités.
    - » Les deux ensembles de données sont reliés par la notion d'identifiant d'entités géographiques.
    - » Modèle du Shapefile (ESRI)

– 2.2. Les données géométriques

» Les données géométriques sont gérées par un Système de Gestion de Fichiers (SGF) propriétaire :

- Les types d'entités sont définis par leur géométrie : points, lignes, polygones.
- Les entités composées sont généralement admises, au moins pour les polygones (enclaves, îles), plus rarement les réseaux, grâce à l'exploitation des relations logiques entre les entités.
- Les phénomènes spatialement continus sont souvent supportés selon les modèles vectoriels TIN et/ou *Lattice* / *Grid*.

» Les entités sont regroupées par couches géométriques (« ***layers*** »).

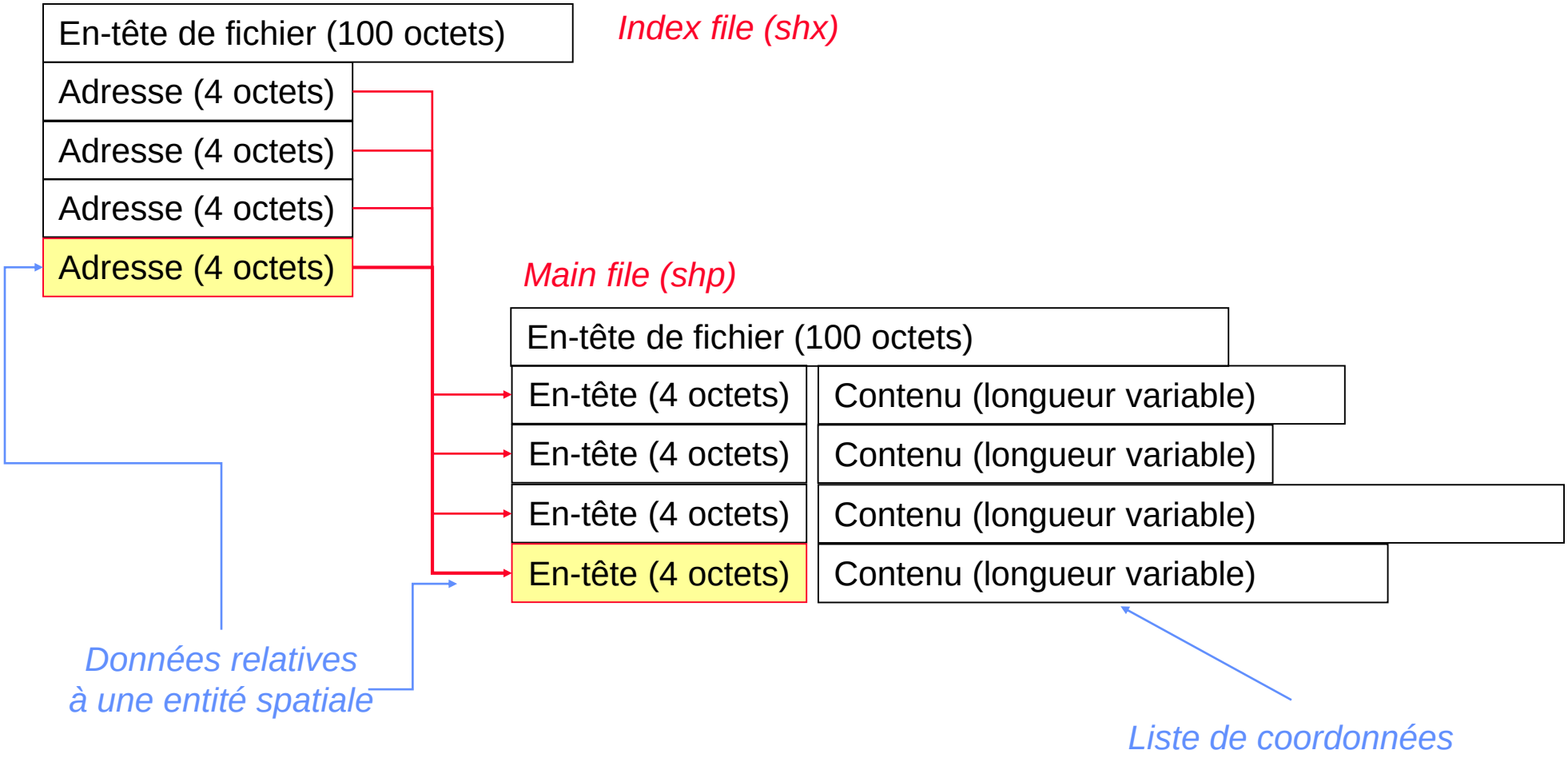
- Une couche ne peut contenir qu'un seul type géométrique d'entités à la fois.

» Les entités sont identifiées par un numéro unique (identifiants).

- Ce numéro permet d'associer les géométries des entités aux tables d'attributs.

» *Exemple du shapefile (binaire):*

- *Main file* (suffixe shp) : collections de coordonnées 2 (x, y), 3 $(x, y, z/m)$ ou 4 (x, y, z, m) dimensions, regroupées par entités, présentées de manière séquentielle, avec un seul type d'entité spatiale par fichier (mais de nombreux types d'entités sont disponibles).
- *Index file* (suffixe shx) : reprend les adresses (pointeurs) de chaque début d'entité dans le *main file* associé et permet un accès direct aux entités.



Formats des fichiers d'entités spatiales d'un Shapefile (ArcView)

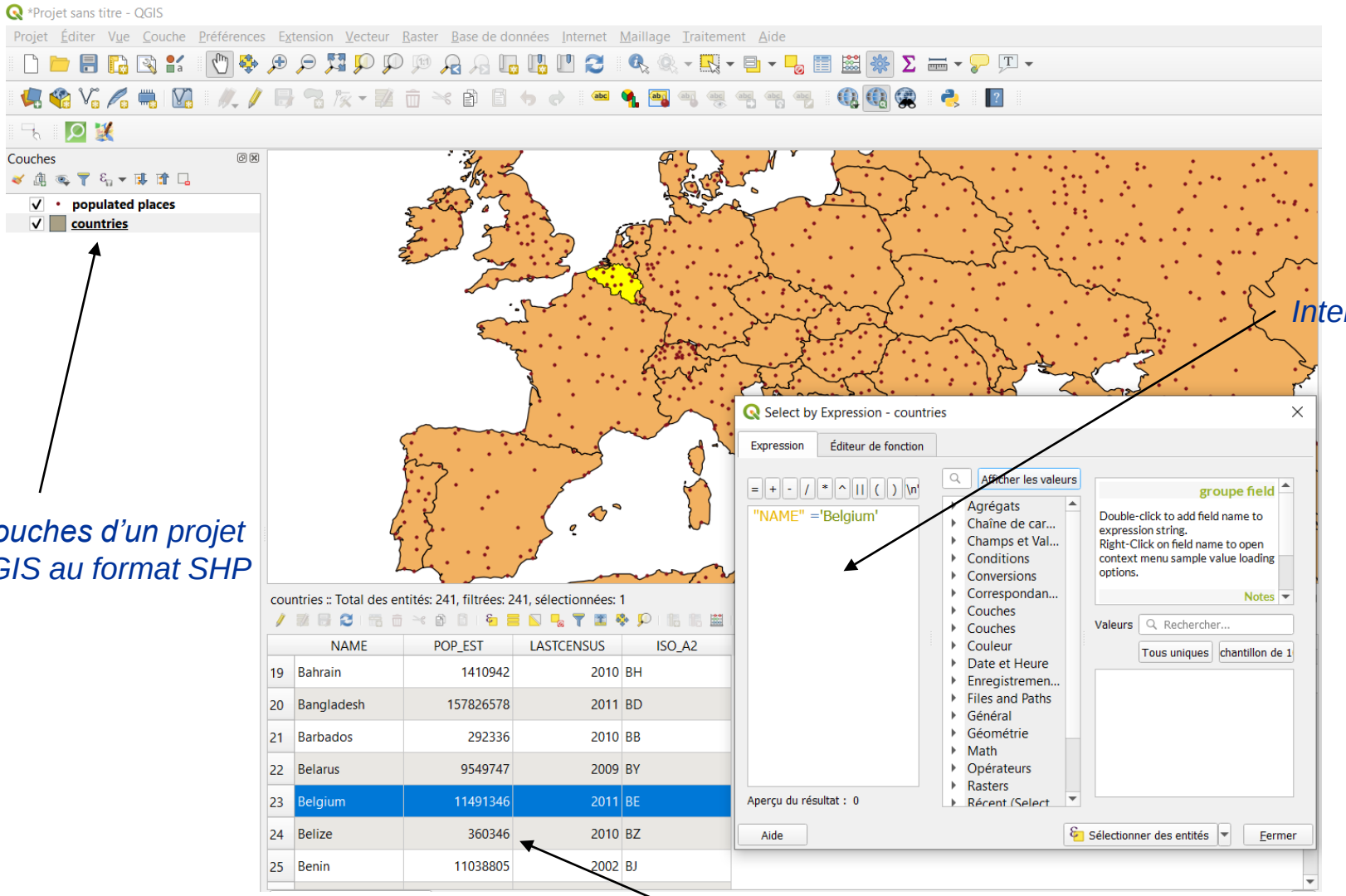
– 2.3. Les données attributaires

» Les données attributaires sont gérées par un SGBD selon un modèle relationnel.

- Une table est systématiquement associée à chaque couche de données géométriques, où l'identifiant des entités figure dans une colonne (table principale).
  - L'identifiant des entités peut servir de **clé** primaire de la table principale.
  - Les autres colonnes de la table principale sont complétées par l'utilisateur et conservent un ou plusieurs attributs, parmi lesquels une ou plusieurs clés externes permettant l'association par jointures aux autres tables d'attributs.
- Toutes les requêtes et transactions portant sur les attributs peuvent être effectuées via le langage de requête du SGBD (compatible **SQL**).

» *Dans un shapefile, elles sont sauvegardées dans une table dBase (fichier du Shapefile de suffixe dbf).*

- L'association 1-1 entre la géométrie et les attributs est basée sur l'ordre d'enregistrement (numéro implicite).
- Chaque tuple (enregistrement) de la table correspond à une entité spatiale et ils sont identifiés et présentés par le même numéro d'ordre que les entités spatiales dans le fichier principal des données géométriques (*main file*) .
- Note: le shapefile comprend également un fichier prj comprenant les infos sur le système de coordonnées de référence (voir plus loin dans le cours)



Interface de requêtes

Couches d'un projet QGIS au format SHP

Table principale (dbf) du Shapefile courant

- **3. Modèle logique hybride géo-relationnel (avec topologie)**
  - **3.1. Définition**
    - » Le modèle géo-relationnel **conserve** une structure hybride, séparant les données géométriques d'une part, et les attributs d'autre part.
    - » Le modèle repose sur l'exploitation des relations topologiques entre les entités spatiales.
      - Suppose des données exemptes d'erreurs topologiques (saisie avec topologie ou construction de la topologie après la saisie).
    - » L'information géographique est structurée en couches thématiques (« *coverages* »), et non plus géométriques.
      - Concept plus proche des applications.
    - » L'implémentation repose sur un modèle conceptuel relationnel des données, tant pour les attributs que pour la topologie, mais **pas** la géométrie de position.

– 3.2. Les données géométriques

» Les données de localisation se ramènent aux coordonnées des points (primitives graphiques).

- Les coordonnées sont définies en 2 dimensions (x, y), plus rarement 3D (x, y, h) ou pseudo-3D (x, y, m) ou pseudo-4D (x, y, h, m).
  - Pseudo 3D et 4D sont utilisées pour les collections de points cotés échantillonnés sur un phénomène spatialement continu en surface (pseudo-3D) ou en volume (pseudo-4D), avec m = valeur de l'attribut quantitatif.
- Chaque **point** reçoit automatiquement un identifiant distinct (numéro interne) indépendamment de l'entité spatiale à la définition de laquelle il participe.

» Les points participent à la définition des entités topologiques :

- Les nœuds, définissant la propriété de connexité.
- Les arcs, définissant la propriété de contiguïté.

» Les entités spatiales simples et composées (ponctuelles, linéaires et polygonales) **sont définies sur bases des entités topologiques** .

– 3.3. Les relations topologiques

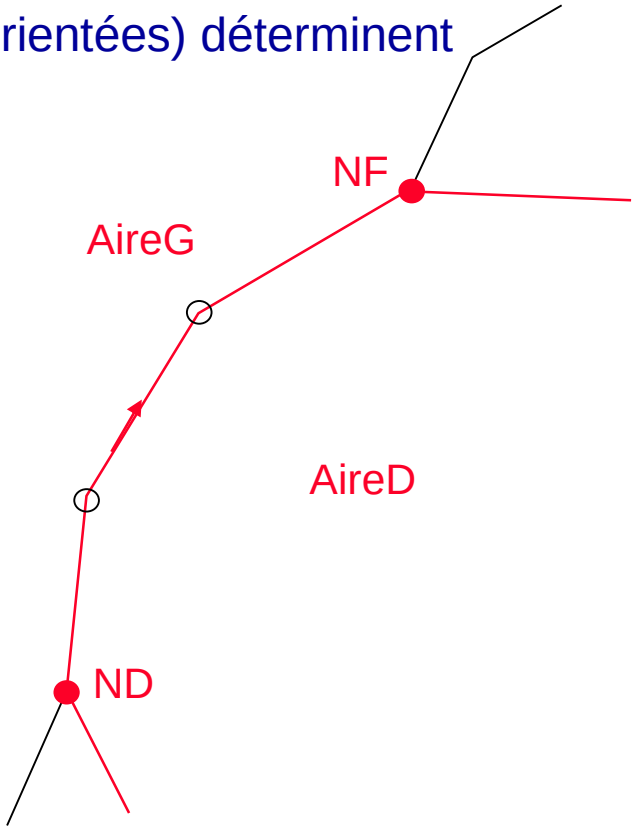
» 3.3.1. Structure arc-nœud

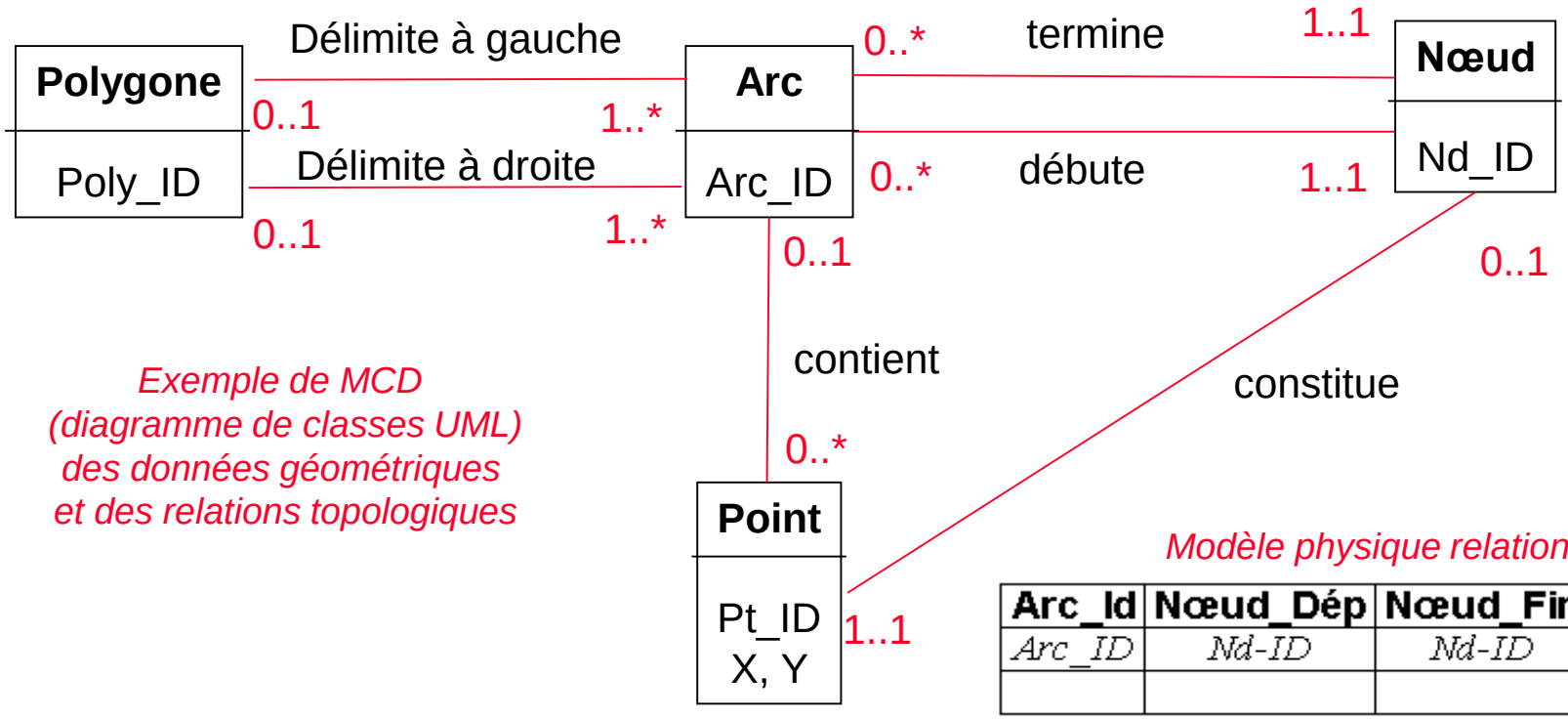
- Les entités discrètes simples sont reconstituées au départ des points grâce aux relations topologiques d'une structure arcs-nœuds.
- Les caractéristiques suivantes des arcs (arêtes orientées) déterminent les relations topologiques de base :
  - Correspondent à des polylignes formées d'un ou plusieurs segments dont les extrémités correspondent à des **points**.
  - Sont délimités par un **nœud** de départ (ND) et un **nœud** final (NF).
  - Sont orientés (→).
  - Laissent une aire à **gauche** (AireG) et une aire à **droite** (AireD), éventuellement identiques (même identifiant) ou virtuelle (surface entourant la zone d'intérêt).

Contiguïté

  - Ne se rejoignent / recoupent qu'à l'endroit des nœuds (**graphe planaire**).

Connexité





Exemple de MCD (diagramme de classes UML) des données géométriques et des relations topologiques

Modèle physique relationnel des arcs

Arc_Id	Nœud_Dép	Nœud_Fin	Poly_G	Poly_D
Arc_ID	*Nd-ID*	*Nd-ID*	*Poly_ID*	*Poly_ID*

Modèle logique relationnel correspondant

POLYGONE (Poly_ID)
ARC (Arc_ID, #Nœud_Dép, #Nœud_Fin, #Poly_G, #Poly-D)
NŒUD (Nd_ID, #Pt_ID)
POINT (Pt_ID, X, Y)

– 3.4. Les données attributaires

» 3.4.1. À quoi se rapportent les attributs ?

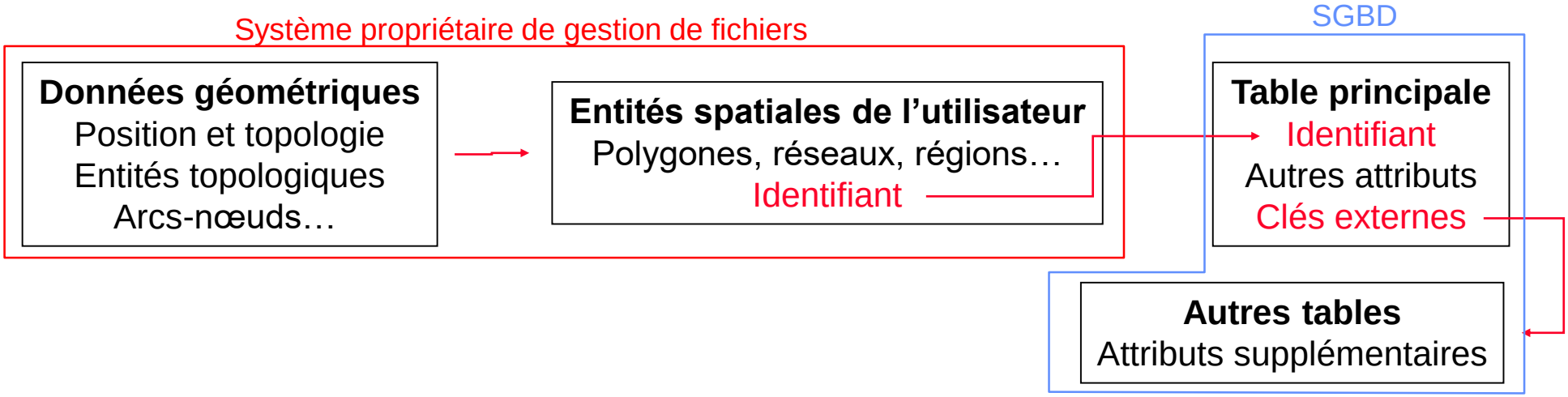
- Les attributs se rapportent aux entités spatiales définies par l'utilisateur, et non pas directement aux entités topologiques arcs-nœuds car :
  - Une même entité topologique peut servir de support à plusieurs entités spatiales de l'utilisateur.

    Exemple : une même collection d'arcs peut participer à la constitution soit d'un réseau de voirie, soit d'une ligne de bus, ou encore servir à délimiter des îlots.
  - Certaines entités topologiques ne sont pas nécessairement porteuses d'attributs.

    Exemple : les nœuds peuvent être considérés comme des entités ponctuelles porteuses d'attributs (ex. carrefours dans un réseau de voirie) ou au contraire comme de simples extrémités, sans attributs, d'un ou plusieurs arcs.
- Il est donc nécessaire de construire, **par combinaison des entités topologiques**, et d'identifier les entités spatiales de l'utilisateur auxquelles vont se référer les attributs.

» 3.4.2. Les tables d'attributs

- Les attributs sont présentés dans des tables, gérées par un SGBD relationnel.
- Une table principale comporte parmi ses attributs, l'identifiant des entités spatiales de l'utilisateur.
  - Elle permet d'établir une association 1-1 avec la définition géométrique des entités spatiales.
  - Elle est souvent engendrée par le logiciel SIG, lors de la construction des entités spatiales.
- Les autres attributs de la table principale peuvent être utilisés comme clés externes, permettant d'établir des jointures avec d'autres tables.

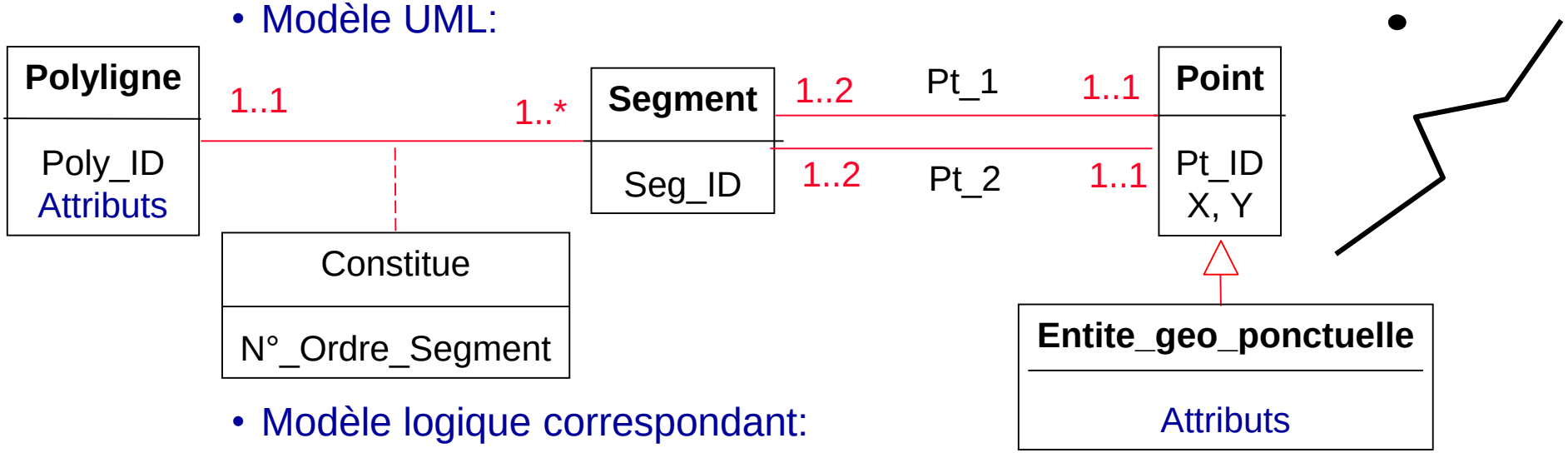


- **4. Modèle intégré relationnel**
  - **4.1. Du MCD (UML) au MLD relationnel**
    - » Il est possible de décrire complètement le MCD d'une base de données géographiques au moyen d'un formalisme relationnel (*ex. UML ou E-A*), en tenant compte ou non des relations topologiques entre les entités spatiales (*cf. § 3*).
      - Le passage d'un MCD au modèle logique relationnel utilise seulement les **cardinalités** des relations (dépendances fonctionnelles, relations récursives, etc)
      - Il est dès lors possible d'envisager l'implémentation complète de la base de données (données spatiales et attributaires) sous une forme relationnelle.
    - » L'intérêt de confier l'implémentation complète de la base à un SGBD relationnel est évident :
      - Disponibilité des fonctions d'administration de données (cohérence, non redondance, partage, sécurité, etc.).
      - Usage d'un SGBD unique, standard et robuste, **sans** solutions propriétaires, pour toute les données de l'organisation.

– 4.2 Transpostion en modèle relationnel

» 4.2.1. Entités géographiques ponctuelles et linéaires

- Modèle UML:



- Modèle logique correspondant:

Polyligne (Poly_ID, Attribut_1, …, Attribut_n)

TablePS (#Poly_ID, #Seg_ID, N°_Ordre_Segment) traduit l'association « constitue »

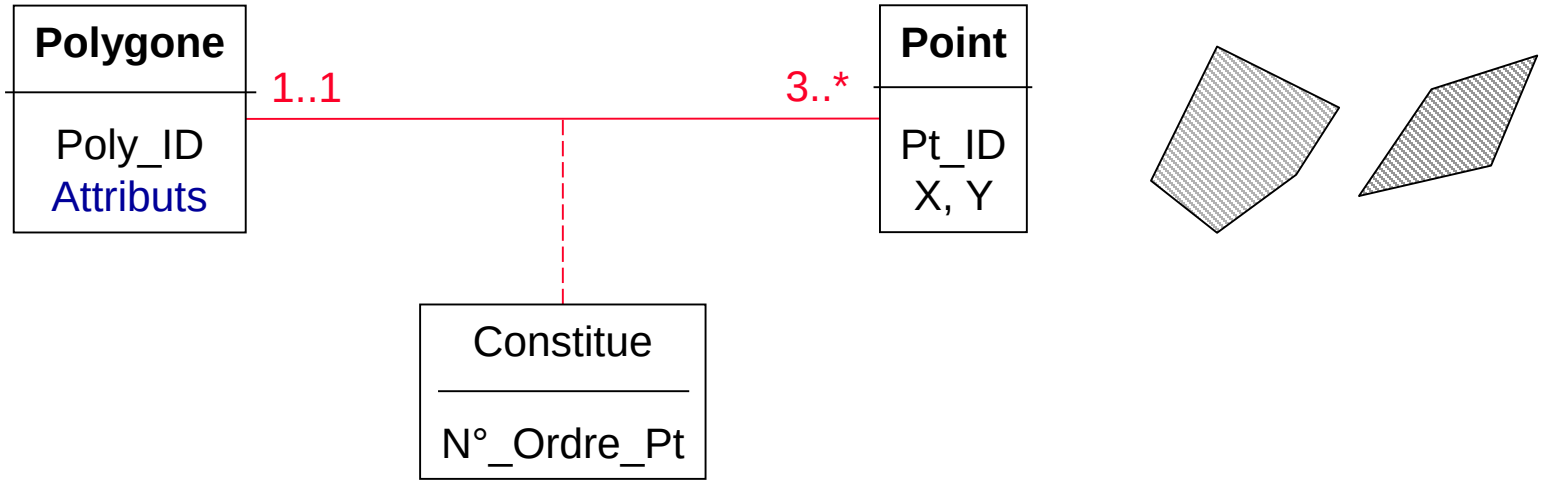
Segment (Seg_ID, #Pt_1, #Pt_2)

Point (Pt_ID, X, Y)

Entite_geo_ponctuelle (Pt_ID, Attribut_1, …, Attribut_n)

Note: la table « Entite_geo_ponctuelle » permet d'éviter des valeurs nulles pour les nombreux points ne définissant pas des entités géo ponctuelles

» 4.2.2. Entités polygonales simples et isolées, sans topologie

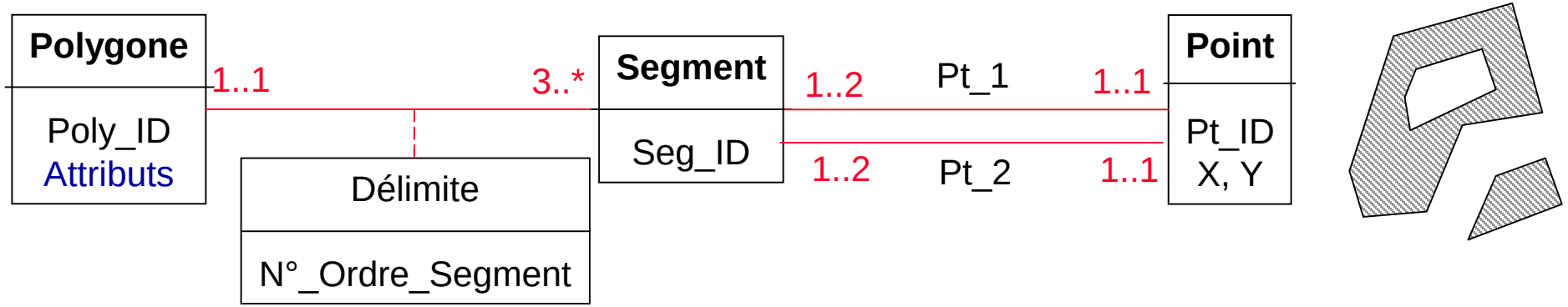


Polygone (Poly_ID, Attribut_1, …, Attribut_n)

TablePP (#Poly_ID, #Pt_ID, N°_Ordre_Pt) traduit l'association « constitue »

Point (Pt_ID, X, Y)

» 4.2.3. Entités polygonales composées et isolées, sans topologie



Polygone (Poly_ID, Attribut_1, …, Attribut_n)

TablePS (#Poly_ID, #Seg_ID, N°_Ordre_Segment)

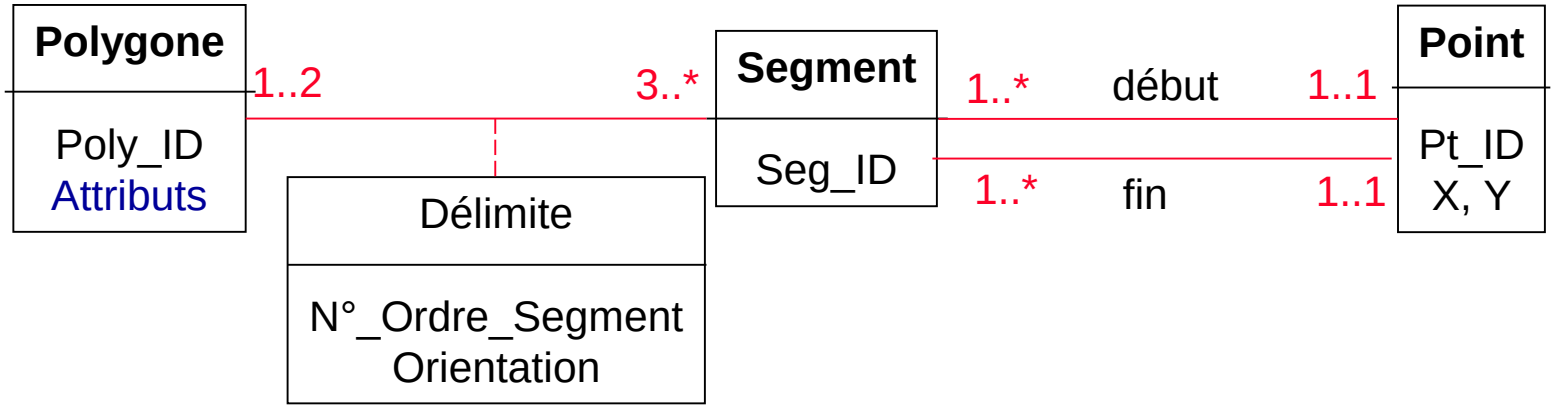
Segment (Seg_ID, #Pt_1, #Pt_2)

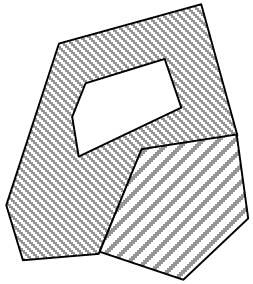
Point (Pt_ID, X, Y)

traduit l'association « délimite »

- La double dépendance fonctionnelle entre les segments et leurs extrémités est utilisée pour éviter une relation supplémentaire « segment_point »

» 4.2.4. Entités polygonales composées jointives, avec topologie





- L'orientation correspond à un simple entier (binaire ou signé) traduisant que le segment est parcouru du point-début vers le point-fin ou inversement, lors du parcours de la frontière du polygone (parcours systématiquement effectué dans un même sens).

traduit la relation « délimite »

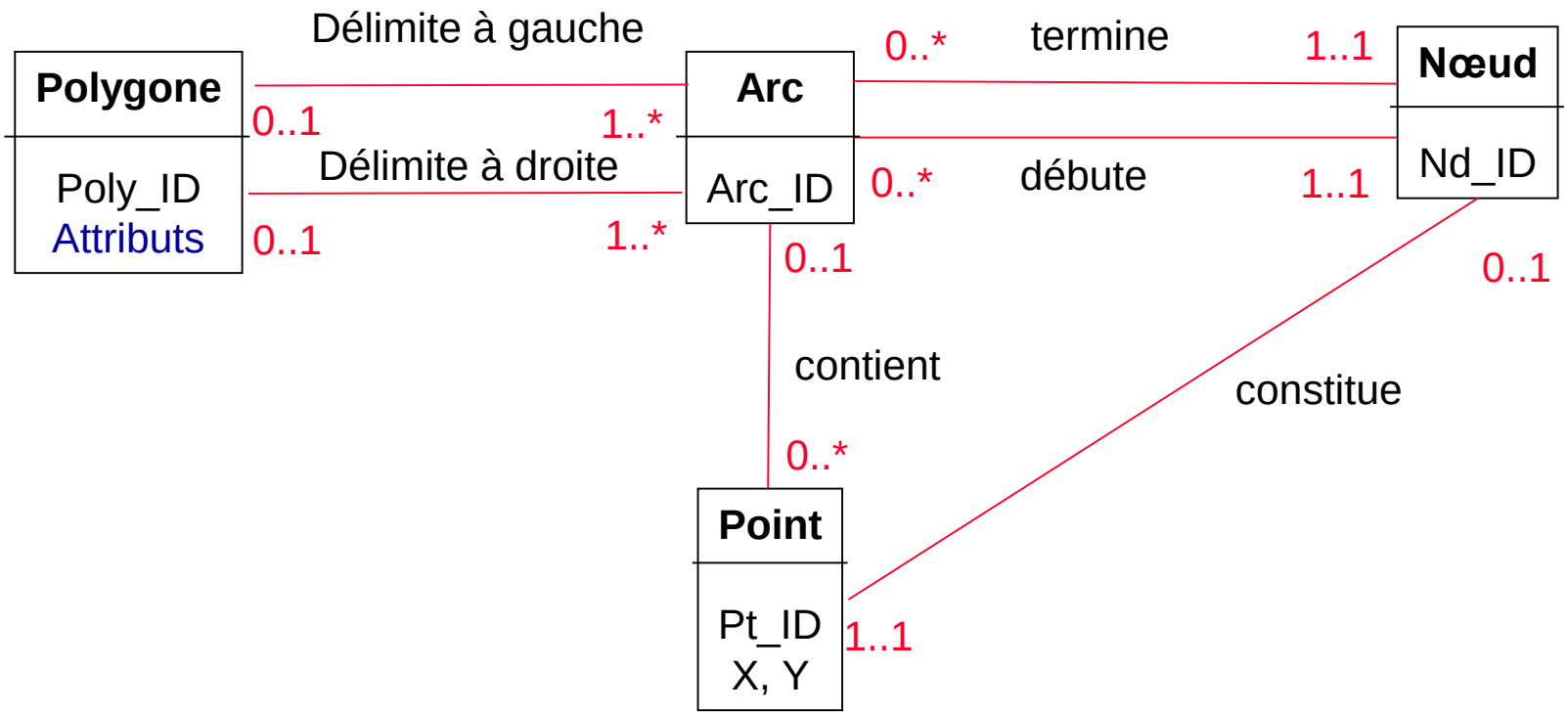
Polygone (Poly_ID, Attribut_1, …, Attribut_n)

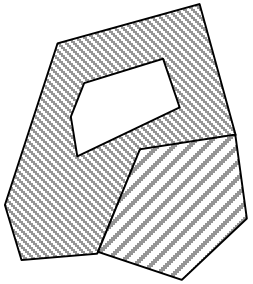
TablePS (#Poly_ID, #Seg_ID, N°_Ordre_Segment, Orientation)

Segment (Seg_ID, #Pt_Début, #Pt_Fin)

Point (Pt_ID, X, Y)

- Alternative: application directe du modèle arc-nœud vu précédemment





POLYGONE (Poly_ID, Attribut_1, …, Attribut *n*)
ARC (Arc_ID, #Nœud_Dép, #Nœud_Fin, #Poly_G, #Poly-D)
NŒUD (Nd_ID, #Pt_ID)
POINT (Pt_ID, X, Y)

5. Modèle objet-relationnel

5.1. L'objet à différents niveaux

» L'approche objet apparaît beaucoup plus naturelle pour définir les entités géographiques:

- Caractère spatial de l'information géographique
- En amont, pour le développeur : MCD Orienté Objet (UML).
- En aval, pour l'utilisateur : vues / applications Orientées Objet (OO).

» Mais, au niveau physique, les SGBD relationnels (SGBD-R) paraissent incontournables :

- Caractère sémantique (attributaire) de l'information géographique
- La table reste la meilleure manière de rendre persistantes les entités

» Il faut donc concilier les modèles internes (R) et externes (OO) :

- Modèle logique O-R (ou relationnel-étendu).
- Enrichissement du langage SQL.
- Interfaces développées dans un langage de programmation OO.

5.2. Avantages du modèle objet-relationnel

» Le modèle relationnel étendu, incorporant des fonctionnalités du modèle objet, permet **en théorie de résoudre la plupart des problèmes de performances** du modèle relationnel classique :

- Création de types (classes) d'attributs.
- Déclaration de fonctions (méthodes) dans les types.
- Identités d'objets (pointeurs)
- Partage par référence
- Héritage
- Attributs multivalués et collections structurées d'attributs.

» Parallèlement, l'évolution du langage de requête **SQL** (version 3) permet d'exploiter ces extensions et de développer des requêtes ciblées et propres aux informations spatiales.

- On peut donc espérer implémenter, par exemple, une structure géo-relationnelle complète et l'exploiter de manière efficace dans un SGBD objet-relationnel.

5.3. Exemples d'extensions du modèle relationnel

5.3.1. Types

- Un type créé par l'utilisateur peut être utilisé pour créer une table d'objets et être référencé par un autre type d'objet.

(TYPE) **Point** (Pt-ID Int, X Float, Y Float)

(TABLE) **Points** of Point

(TYPE) **Nœud** (Nd-ID Int, Position **REF**(Point))

- SQL3 offre plusieurs patrons de base pour construire des objets complexes, tels que la liste par exemple.

(TYPE) **Polyligne** (Pll-ID Int, **List**(Point), Attribut-1 …)

(TYPE) **Arc** (Arc-ID Int; **List**(Point), NdD **REF**(Noeud),
NdF **REF (**Noeud**)**, PolG **REF**(Polygone), PolD **REF**(Polygone))

(TYPE) **Polygone** (Plg-ID Int, **List**(Arc), Attribut-1 …)

- Le partage référentiel (**REF**) permet d'éviter les jointures entre les tables d'objets composés-composants en exploitant des pointeurs

5.3.2. Fonctions encapsulant le type

- La déclaration d'un type permet de définir des fonctions publiques, utilisables lors de requêtes ultérieures.

Procédure écrite en langage 00

(TYPE) **Segment** (Seg-ID Int, Pt1 **REF**(Point), Pt2 **REF**(Point),

FUNCTION Calcul_Longueur(S Segment) **RETURNS** (Float)

… **END FUNCTION**)

Exemple de requête :

SELECT Calcul_Longueur (S)

FROM TableSegment

- Il est possible d'implémenter les diverses règles intentionnelles facilitant les requêtes spatiales courantes.

Procédure écrite en langage 00

(TYPE) **Segment** (Seg-ID Int, Pt1 **REF**(Point), Pt2 **REF**(Point),

a Float, b Float, c Float,

FUNCTION AppartientSegment (S Segment, P Point, Tolerance Float)

RETURNS (Boolean) … **END FUNCTION**)

5.3.3. Héritage et spécialisation

- Un type peut hériter de la structure d'un autre type d'objet, et surcharger ou spécialiser ses propriétés ou ses méthodes.

  (TYPE) **Polyligne** (PLL-ID Int, **List**(Point))

  (TYPE) **AxeVoirie UNDER** Polyligne(Nom String)

- Le même type d'héritage existe également entre les tables.

  (TABLE) **Polygone** (Pol-ID Int, **List**(Point))

  (TABLE) **Parcelle UNDER** Polygone **WITH** Adresse String

  (TABLE) **Quartier UNDER** Polygone **WITH** Nom String

5.3.4. Objets longs (BLOB)

- Introduit dans les années 90 dans les SGBD relationnels, un objet long (« *Binary Large Object* ») est un champ de données codé sous forme d'une séquence de bits (en binaire).
  - Bien qu'illisible par l'homme, le format binaire est plus facile à interpréter par la machine ➔ gain de performance
- Forme de stockage simple de données non formatées.
  - Peut être lu et écrit comme une valeur d'attribut dans une base.
  - Peut atteindre de très grandes tailles (> Go).
- Pour que le BLOB puisse être traité par l'application, sa structure doit être interprétable par le SGBD
- C'est à travers des BLOB que sont notamment stockées les géométries (ou les rasters) des données géographiques. Pour une entité vectorielle, le BLOB contient alors:
  - Les métadonnées de l'objet nécessaire à sa lecture: type d'entité (point, ligne, polygone, …), système de coordonnées de référence et nombre de dimensions des coordonnées
  - Les données de l'objet

6. Schéma spatial ISO 19107

6.1. Norme ISO 19107

» ISO = **International Organization for Standardization**

» L'ISO propose un schéma spatial standardisé pour les entités géographiques « vectorielles » définies par des géométries et des topologies jusqu'à 3 dimensions.

» Modèle conceptuel permettant la description et la manipulation, grâce à des opérateurs spatiaux, des caractéristiques spatiales des entités géographiques.

» Les caractéristiques spatiales sont définies par :

- Une géométrie : fournit les moyens d'une description quantitative (dimensions, position, taille, forme, orientation) au moyen de coordonnées et de fonctions mathématiques.
- Une topologie : fournit les moyens d'une description de la connectivité d'un graphe à n dimensions.
- Les opérateurs spatiaux permettent d'accéder, rechercher, gérer, traiter et échanger des entités spatiales définies selon ce schéma.
- Des opérateurs topologiques permettent notamment de dériver la connectivité des entités géographiques, depuis leurs géométries.

6.2. La géométrie

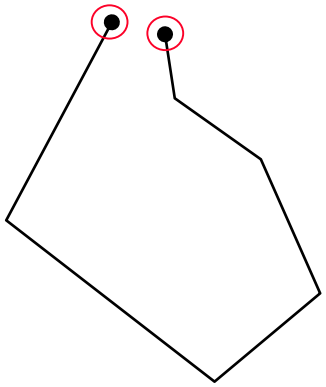
6.2.1. Primitives géométriques

- Objet géométrique non décomposable représentant un élément unique, connecté et homogène de l'espace et qui présente une information relative à la configuration géométrique.
  - **simple:** sans intersection d'aucune sorte;
  - **connecté:** 2 positions *successives* de l'objet peuvent avoir une frontière commune (voir slide suivant);
  - **Homogène:** tant par le système de référence utilisé, que par les valeurs des autres attributs dont pourrait être porteur l'objet.
- Les différentes primitives géométriques sont distinguées selon leurs dimensions :
  - Dimension 0 : point : représentant une position.
  - Dimension 1 : ligne (« *curve* ») : représentant l'image continue d'une ligne.
  - Dimension 2 : polygone (« *surface* ») : représentant localement l'image continue de la région d'un plan .
  - Dimension 3 : volume (« *solid* ») : représentant l'image continue d'un espace euclidien borné à 3 dimensions.

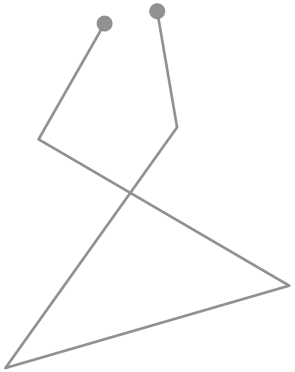
6.2.2. Frontière (« *boundary* ») d'un objet géométrique

- Ensemble de primitives géométriques, de dimensions **inférieures** à celle de l'objet, qui limitent l'extension d'un objet géométrique.
  - La frontière d'un objet de type « **point** » est vide.
  - La frontière d'une « **ligne** » est formée des 2 « points » situés à ses extrémités (point de départ et point terminal).
    - On considère qu'une ligne qui se referme sur elle-même (anneau ou « *ring* ») **n'a pas** de frontière.
  - La frontière d'une « **surface** » est l'ensemble des « lignes », orientées et fermées, qui délimite la surface.
    - La frontière d'une surface isolée peut être réduite à un « *ring* ».
  - La frontière d'un « **solide** » est l'ensemble des « surfaces » orientées et fermées qui délimite le solide.
    - On considère que la surface d'une sphère (coquille ou « *shell* ») **n'a pas** de frontière.
- Cycle : objet géométrique n'ayant pas de frontière (« *ring* » ou « *shell* »).

Frontière

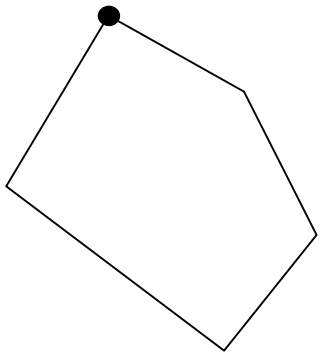


Curve
simple



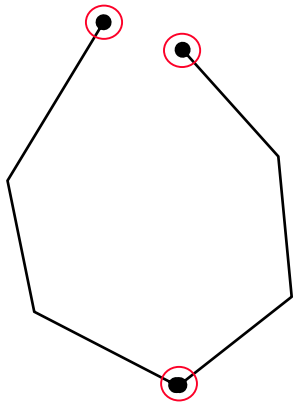
Curve
non-simple

Pas de frontière



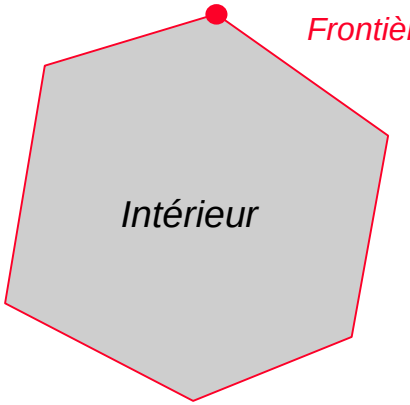
Curve fermée ou *Ring*
simple

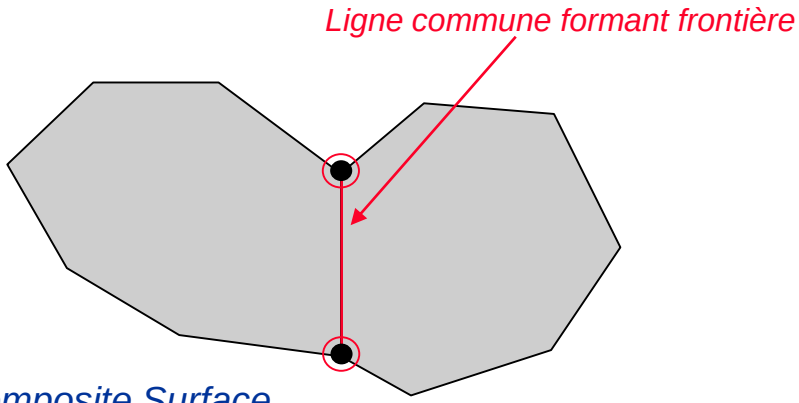
Frontière



Composite Curve
simple

Surface
simple





Composite Surface

6.2.3. Autres propriétés des géométries

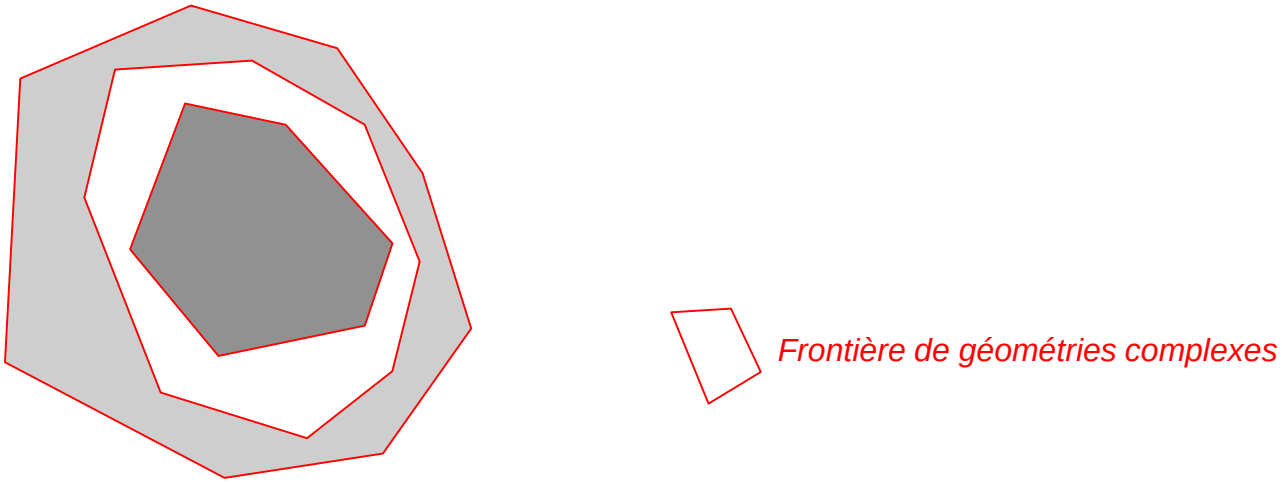
- Intérieur (« *interior* ») d'un objet géométrique : ensemble des positions qui figurent sur un objet géométrique mais **pas** sur sa frontière.
- Fermeture (« *closure* ») d'un objet géométrique: union de l'intérieur et de la frontière d'un objet géométrique.
- Extérieur (« *exterior* ») d'un objet géométrique: différence entre l'univers et la fermeture de l'objet.

6.2.4. Objets géométriques composites (« joints »)

- Ligne composite (« *composite curve* ») : séquence de lignes telle que chaque ligne, à l'exception de la première, commence au point terminal de la ligne précédente dans la séquence.
- Surface composite : ensemble de surfaces jointes les unes aux autres par des lignes partagées formant les frontières.
- Solide composite : ensemble des solides joints les uns aux autres par des surfaces communes formant les frontières.

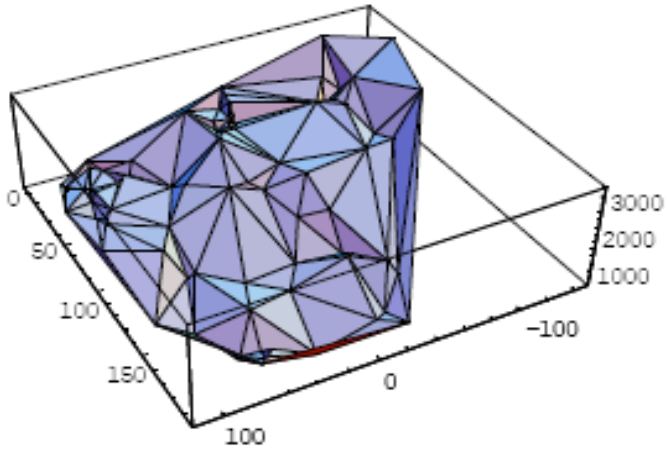
6.2.5. Géométries complexes

- Ensemble de primitives géométriques disjointes, où la **frontière** de chaque primitive géométrique peut être représentée par l'union d'autres primitives géométriques de plus petites dimensions présentes dans le même ensemble.
  - **Disjoint** : aucune position n'est intérieure à plus d'un seul d'objet.
  - La frontière d'une géométrie complexe est une autre géométrie complexe rassemblant les primitives géométriques formant les frontières de chacun des éléments du complexe.



6.2.6. Autres combinaisons de primitives géométriques

- « *Surface patch* » : objet géométrique connecté à 2 dimensions, utilisé pour représenter une portion continue d'une surface, en faisant appel à des méthodes homogènes de définition et d'interpolation.



6.3. La topologie

» Un **objet topologique** est un objet présentant des caractéristiques spatiales qui ne varient pas lorsque l'espace est déformé de manière élastique ou continue *(par ex. par transformation de coordonnées).*

6.3.1. Primitive topologique

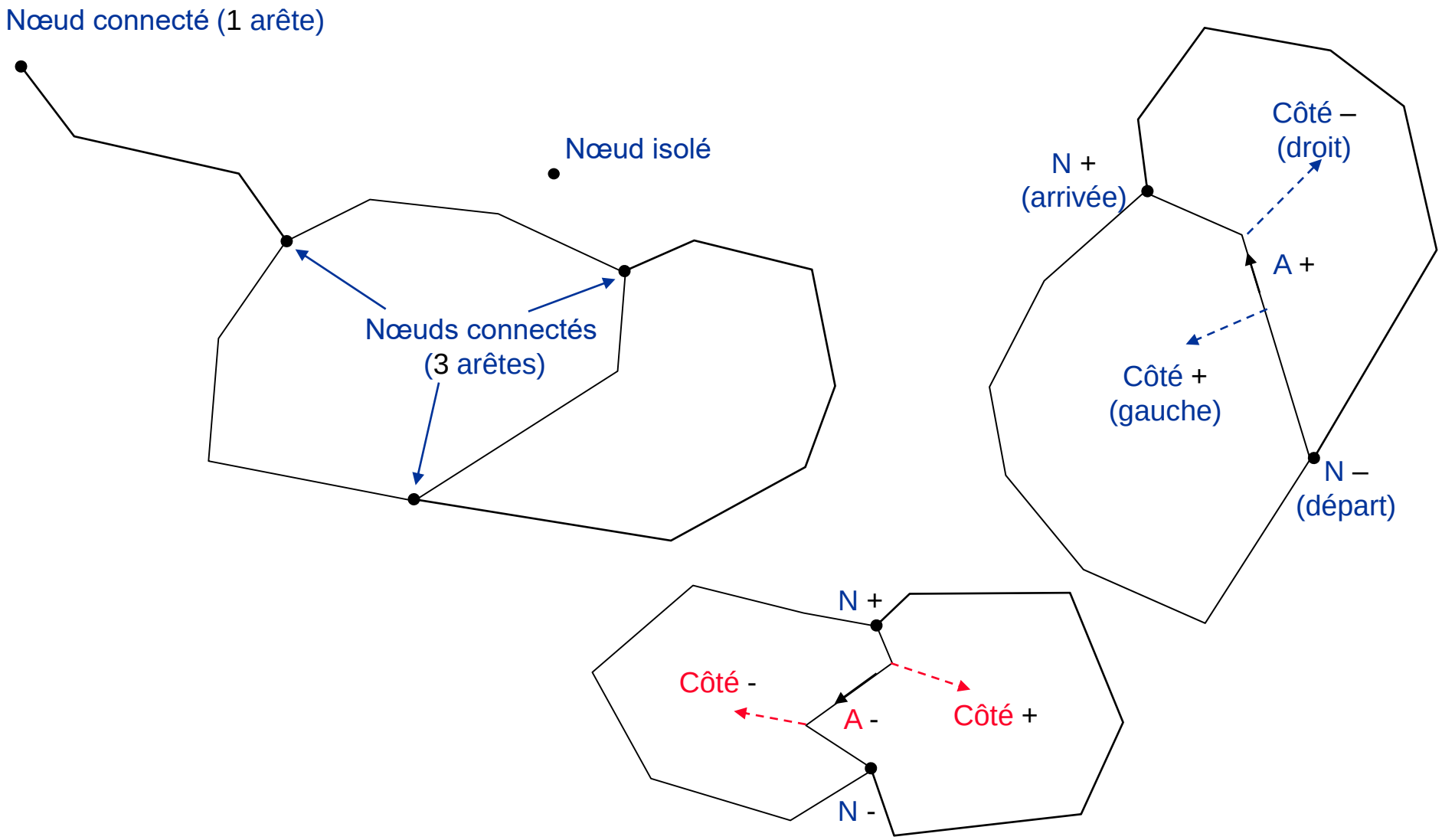
- Une primitive topologique est un objet topologique simple et non décomposable.
- Les primitives sont déclinées selon leur **dimensionnalité** et, dans une réalisation géométrique, elles correspondent à **l'intérieur** des primitives géométriques de même dimension :
  - 0 Dimension : nœud (« *node* ») ;
  - 1 Dimension : arête (« *edge* ») ;
  - 2 Dimensions : face ;
  - 3 Dimensions : solide topologique.
- À l'exception du nœud, les objets topologiques ont une **frontière**, et ils peuvent tous (y compris les nœuds) être associés à une **orientation**.

6.3.2. Les nœuds (topologie de connexité)

- Nœud **isolé** : nœud qui n'est lié à aucune arête ;
- Nœud **connecté** : nœud qui commence ou termine une ou plusieurs arêtes.
- Nœud **orienté** : objet topologique orienté qui représente une association entre un nœud et l'**une** de ses orientations.
  - L'orientation d'un nœud par rapport à une arête est notée « - » s'il s'agit du nœud de départ et « + » s'il s'agit du nœud terminal.

6.3.3. Les arêtes (topologie de contiguïté)

- La frontière d'une arête correspond au complexe topologique formé d'un ou de deux **nœuds** associés à l'arête.
- Arête **orientée** (ou arc) : objet topologique orienté qui représente une association entre une arête et l'**une** de ses orientations.
  - Une arête orientée est notée « + » si son orientation est conforme au sens de parcours de l'arête, entre son nœud de départ et son nœud terminal ; elle est notée « - » sinon.
  - L'arête orientée permet de définir un côté gauche (noté « + ») et un côté droit (noté « - ») de part et d'autre de l'arête orientée.


Nœud connecté (1 arête)
Nœud isolé
Nœuds connectés
(3 arêtes)
Côté –
(droit)
N +
(arrivée)
A +
Côté +
(gauche)
N –
(départ)
N +
Côté -
A -
Côté +
N -

6.4. Les relations topologiques entre entités spatiales

» Les relations sont retrouvées grâce aux 3 propriétés de toute entité :

- Intérieur (I), frontière (B) et extérieur (E).

– 6.4.1. Modèle DE-9IM *(Dimensionnally Extended nine-Intersection Model)*

» Deux entités présentent une intersection vide (F) ou non nulle (T) selon les 9 combinaisons possibles de leurs propriétés.

- Les intersections entre entités sont figurées dans une matrice qui spécifie quelles sont les dimensions des intersections entre les 3 propriétés des entités. Ainsi pour 2 entités a et b :

$$\mathrm{DE9IM}(a,b) = \begin{bmatrix} \dim(I(a)\cap I(b)) & \dim(I(a)\cap B(b)) & \dim(I(a)\cap E(b)) \\ \dim(B(a)\cap I(b)) & \dim(B(a)\cap B(b)) & \dim(B(a)\cap E(b)) \\ \dim(E(a)\cap I(b)) & \dim(E(a)\cap B(b)) & \dim(E(a)\cap E(b)) \end{bmatrix}$$

où dim est :

– Soit la dimension maximale de l'intersection ($\cap$) de l'intérieur (I), la frontière (B) et l'extérieur (E) des géométries a et b (domaine {0,1,2, F }).

– Attention : dans ce cas, 0 signifie une intersection ponctuelle !

– Soit une valeur binaire (1) si la dimension de l'intersection vaut {0,1,2}, et (0) sinon (matrice binaire; domaine {0,1} ou { F, T }).

- Le résultat de la matrice peut être consigné dans une chaîne de caractères (*string*), tant en termes de dimensions qu'en binaire (*DE-9IM string code*).

	Interior	Boundary	Exterior
Interior	 $dim(I(a) \cap I(b)) = 2$	 $dim(I(a) \cap B(b)) = 1$	 $dim(I(a) \cap E(b)) = 2$
Boundary	 $dim(B(a) \cap I(b)) = 1$	 $dim(B(a) \cap B(b)) = 0$	 $dim(B(a) \cap E(b)) = 1$
Exterior	 $dim(E(a) \cap I(b)) = 2$	 $dim(E(a) \cap B(b)) = 1$	 $dim(E(a) \cap E(b)) = 2$

Dimensions de l'intersection en rouge

String code (dimensions) : '212101212'

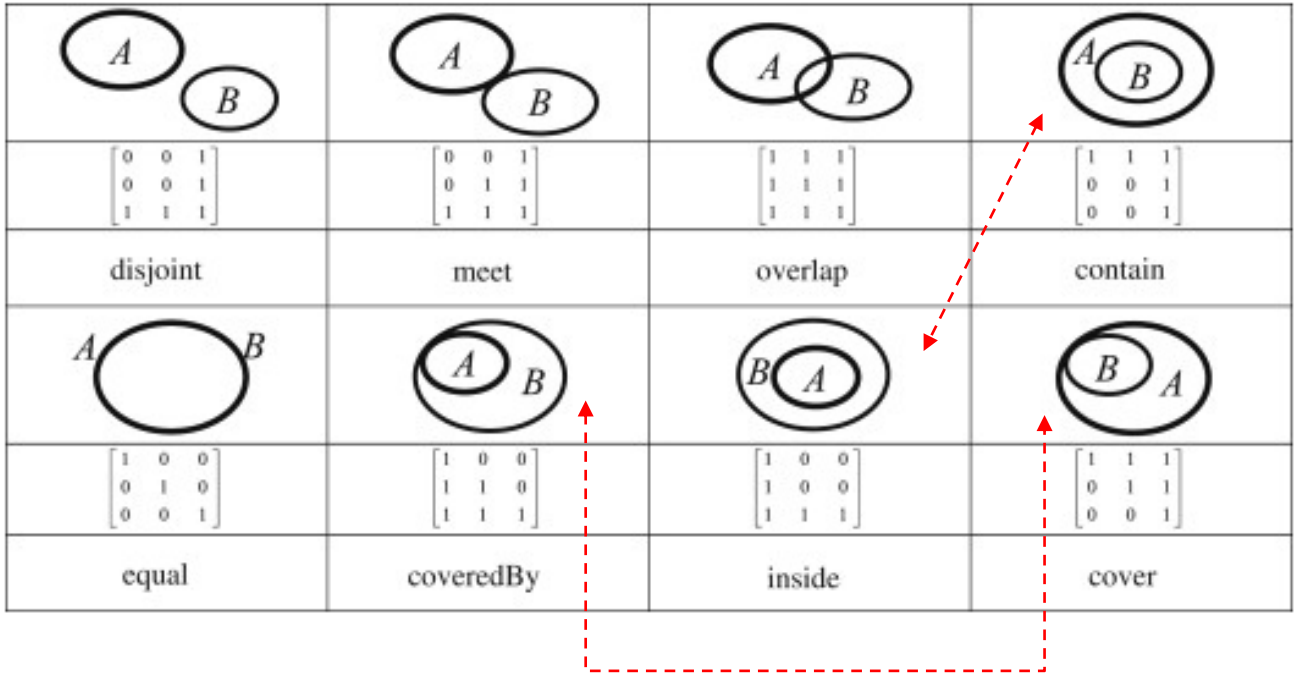
– 6.4.2. Prédicats spatiaux

» Pour faciliter l'usage des relations topologiques dans les requêtes, les relations spatiales sont traduites par des fonctions, intitulées « **prédicats** », et traduites par des **masques binaires matriciels**.

» Le masque binaire est construit sur la même structure de matrice : $\begin{bmatrix} II & IB & IE \\ BI & BB & BE \\ EI & EB & EE \end{bmatrix}$

» *Exemples de prédicats entre 2 entités polygonales :*

Fonctions inverses



7. Schéma spatial de l'OGC

7.1. *Open Geospatial Consortium* (ex Open GIS Consortium ou Open GIS)

» L'OGC est une organisation, à but non lucratif, fondée en 1994, regroupant plusieurs centaines de membres (principales compagnies privées actives en géomatique, agences publiques, représentants du monde académique).

- L'objectif est de favoriser l'usage des données spatiales dans la technologie de l'information (« *geo-enabled technology* »), tout en améliorant la productivité des développeurs et des utilisateurs.
- Son action principale consiste à promouvoir une plus grande interopérabilité en définissant divers types de standards en matière de données, d'échanges et de traitements.

» Parmi les standards déposés, l'OGC a défini :

- Des entités spatiales simples et en collections (« *OGC spatial features* »).
- Les spécifications SQL permettant de manipuler ces entités à travers des types abstraits de données (« *Abstract Data Type* » ou ADT).

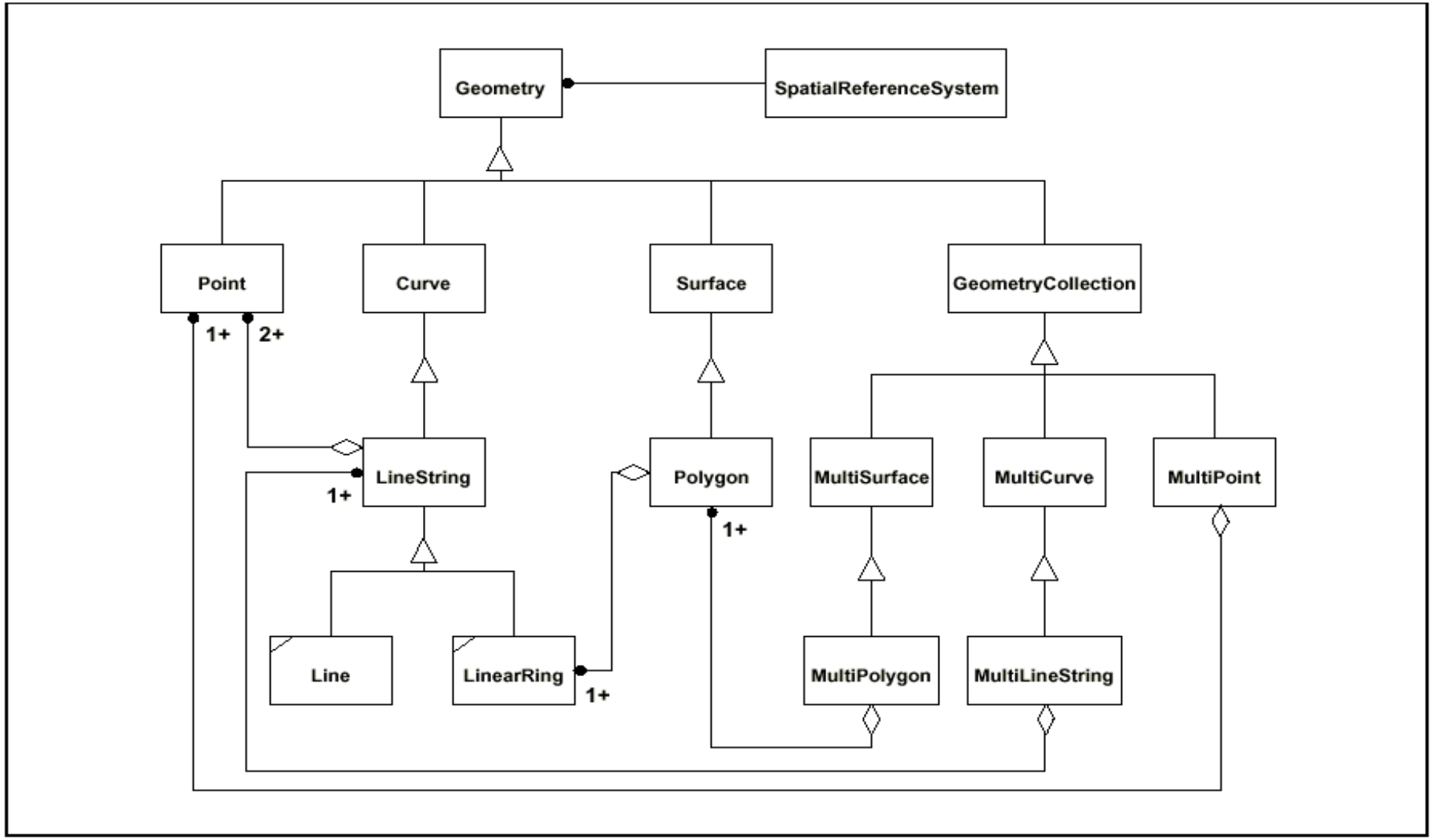
7.2. Propriétés des entités spatiales OGC

» Une entité est représentée par une géométrie, constituée d'un point ou d'un ensemble de points symbolisant une entité au sol.

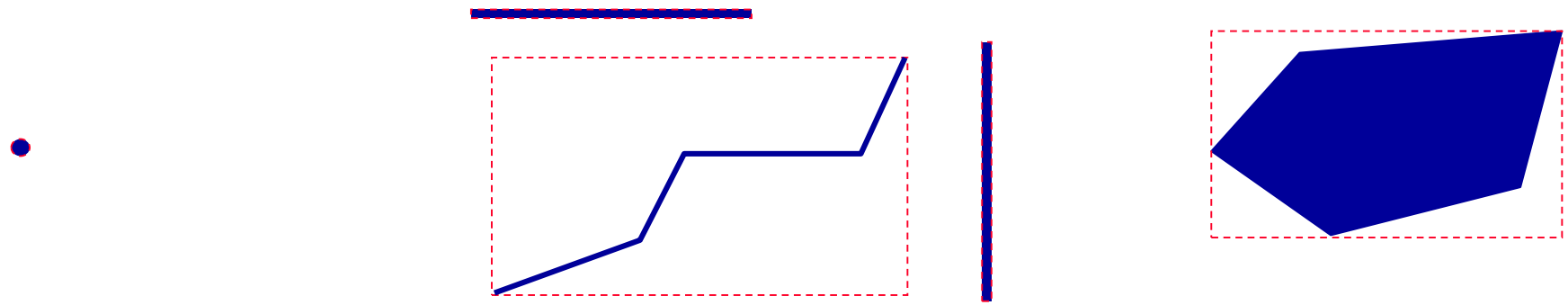
» La géométrie peut être considérée comme une super-classe dont sont dérivées des sous-classes de géométries par spécialisation.

» Chaque géométrie possède plusieurs propriétés :

- Elle est positionnée dans l'espace par son intérieur, sa frontière (« *boundary* ») et son extérieur.
- Elle contient 0 (vide), 1 ou plusieurs points définis en planimétrie dont les coordonnées sont intitulées « mesures ».
- Elle possède une enveloppe: point, ligne verticale ou horizontale ou cadre capable (« bounding box »).
- Sa dimension (0/1/2) détermine ses relations spatiales possibles.
- Les sous-classes de géométrie sont simples ou non-simples selon leur topologie **d'intersection**.
- Le système de référence spatiale de chaque géométrie est défini et modifiable par une matrice de transformation de coordonnées.



Hiérarchie des classes de géométries OGC

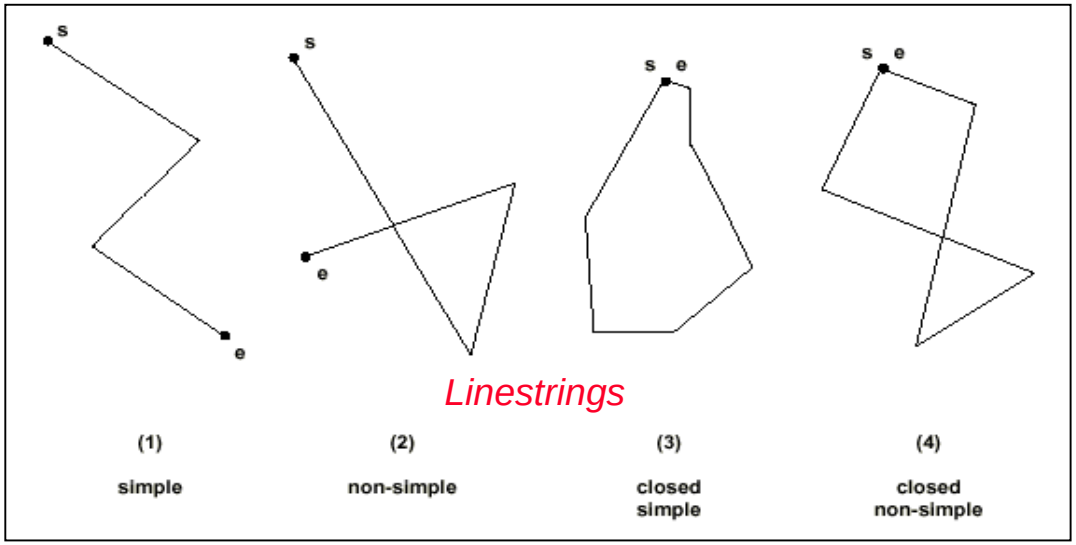


Dimension (0) *Dimension (1)* *Dimension (2)*

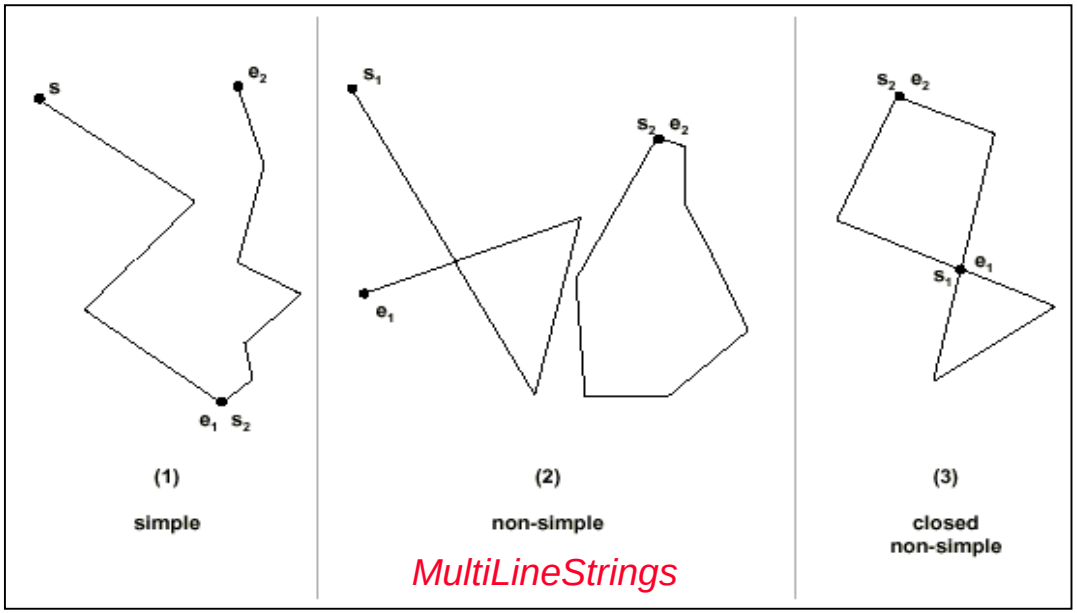
Enveloppe

Dimensions et enveloppes des « Simple Features » de l'OGC

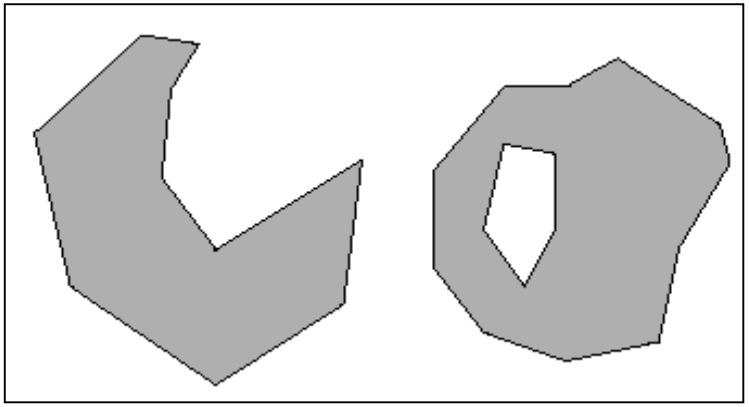
- L'enveloppe d'un point est le point en question
- L'enveloppe d'une polyligne (« LineString ») est un rectangle (cadre capable) sauf dans deux cas particuliers:
  - L'enveloppe d'une ligne horizontale est la ligne en question
  - L'enveloppe d'une ligne verticale est la ligne en question
- L'enveloppe d'un polygone est un rectangle (cadre capable)



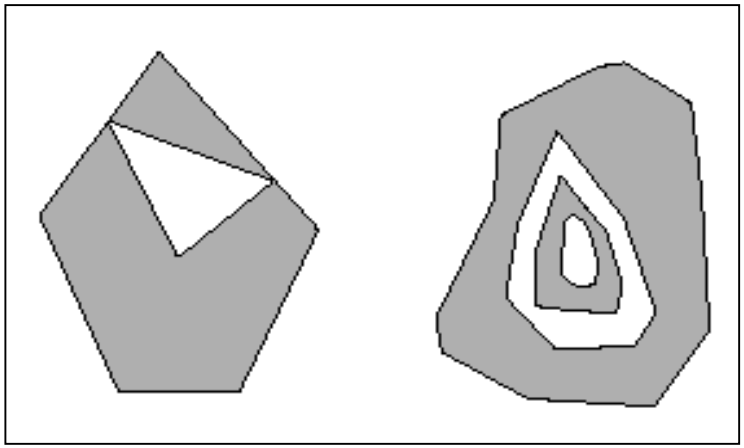
Linestrings



MultiLineStrings



Exemples de Polygons à 1 et 2 rings



Exemples de MultiPolygons

7.3. Formats WKB et WKT de l'OGC

» L'OGC a défini un format d'échange des géométries en mode binaire intitulé « *Well-known binary representation* » ou WKB.

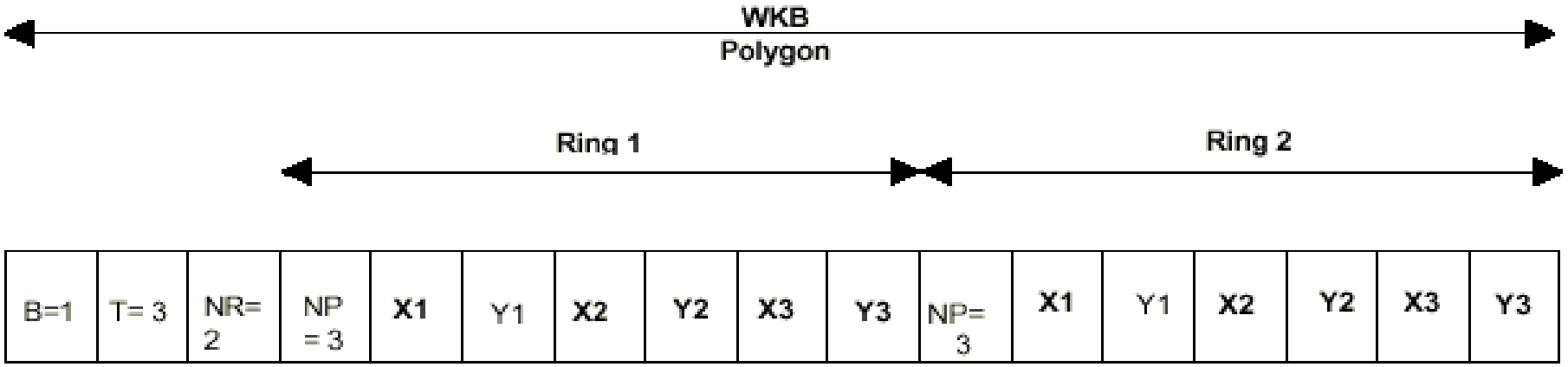
» Représentation des géométries par une suite continue d'octets :

- Le mode d'encodage des valeurs (*cf. ci-dessous*).
- Le type de géométrie : *Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeoCollection*.
- Le nombre de points.
- Les valeurs des coordonnées (x, y) sauvegardées en :
  - Entiers non signés sur 4 octets {0, 4294967295}.
  - Double précision sur 8 octets (IEEE 754).

    susceptibles d'être encodés selon les modes :
    - XBR « big endian » : octet le + significatif / signe d'abord.
    - NBR « little endian » : octet le - significatif d'abord / signe à la fin.

» Le format WKB ne supporte pas les entités présentant des intersections, ni les éventuelles valeurs de z associées aux points.

» L'OGC a aussi défini un format WKT (*Text*) utilisant une séquence de caractères Unicode-ASCII.



Nombre d'octets par champ

1	4	4	4	8	8	8	8	8	8	4	8	8	8	8	8	8

Exemple d'encodage en WKB d'un polygone composé de 2 limites (avec 1 enclave ou 1 île)
Mode NBR (B=1) – Type Polygon (T=3) – Nombre de Rings (NR=2) – Nombre de points par Ring (NP=3)

Geometry Type	SQL Text Literal Representation	Comment
Point	'POINT (10 10)'	a Point
LineString	'LINESTRING (10 10, 20 20, 30 40)'	a LineString with 3 points
Polygon	'POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))'	a Polygon with 1 exterior ring and 0 interior rings
Multipoint	'MULTIPOINT (10 10, 20 20)'	a MultiPoint with 2 point
MultiLineString	'MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))'	a MultiLineString with 2 linestrings
MultiPolygon	'MULTIPOLYGON (((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))'	a MultiPolygon with 2 polygons
GeomCollection	'GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20))'	a GeometryCollection consisting of 2 Point values and a LineString value

Collection « hétérogène » dans cet exemple !

Représentation des géométries sous forme de texte dans le format Well-Known Text (WKT)

8. « *SRID* » (« *Spatial Reference Identifier* »)

8.1. Objet de la géo-référenciation

» Un logiciel SIG doit intégrer et gérer des données géographiques diverses, tant par la thématique que par la géométrie, et permettre leur traitement simultané par des applications « métier ».

- Pour que les données portant sur un même territoire soient cohérentes, il faut que leurs géométries soient définies dans un **même système de coordonnées de référence (SCR)**, ou au moins que leurs systèmes de coordonnées soient parfaitement connus pour permettre les transformations de l'un à l'autre.

» La **géo-référenciation** consiste à établir l'association entre les coordonnées des géométries d'un jeu de données géographiques (vectorielles ou maillées), et un système de coordonnées terrestres de référence.

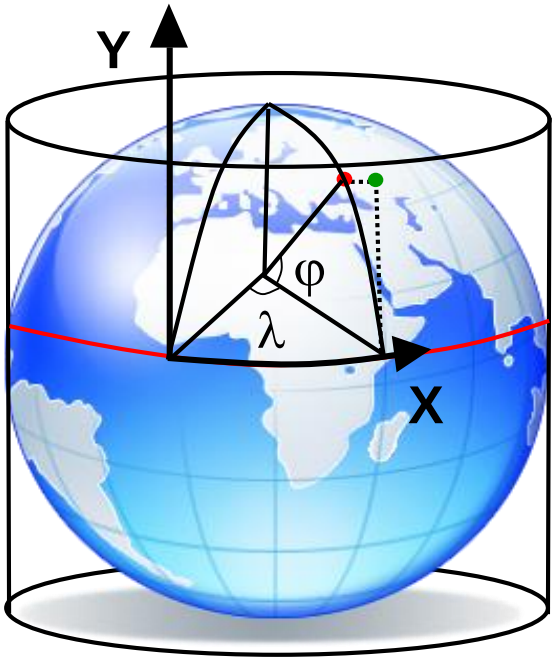
- Pour être complet, un système de coordonnées de référence devrait inclure les coordonnées sur une surface de référence (plan de projection « 2D » ou ellipsoïde « 3D »), et une hauteur par rapport à cette surface de référence (« 3D »).

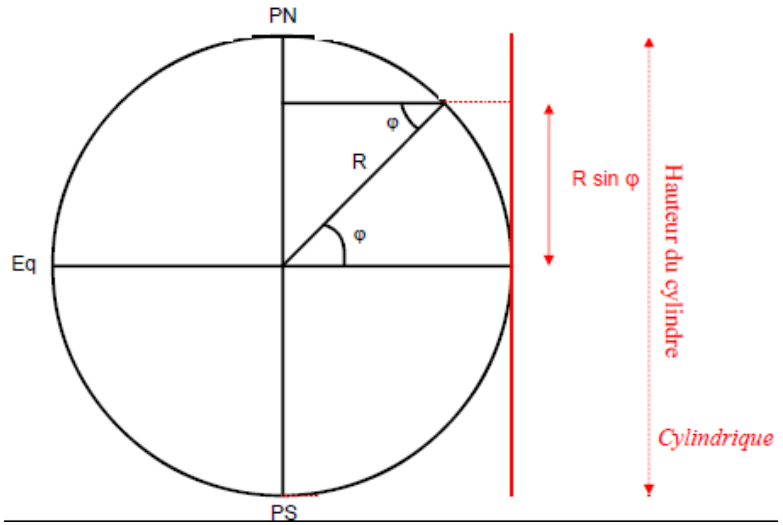
8.2. Paramètres d'un système de référence

» La documentation d'un SRID est standardisée par l'OGC (reprise par l'ISO) et décrite sous forme textuelle (format **WKT**), mais son implémentation peut varier selon les SGBD.

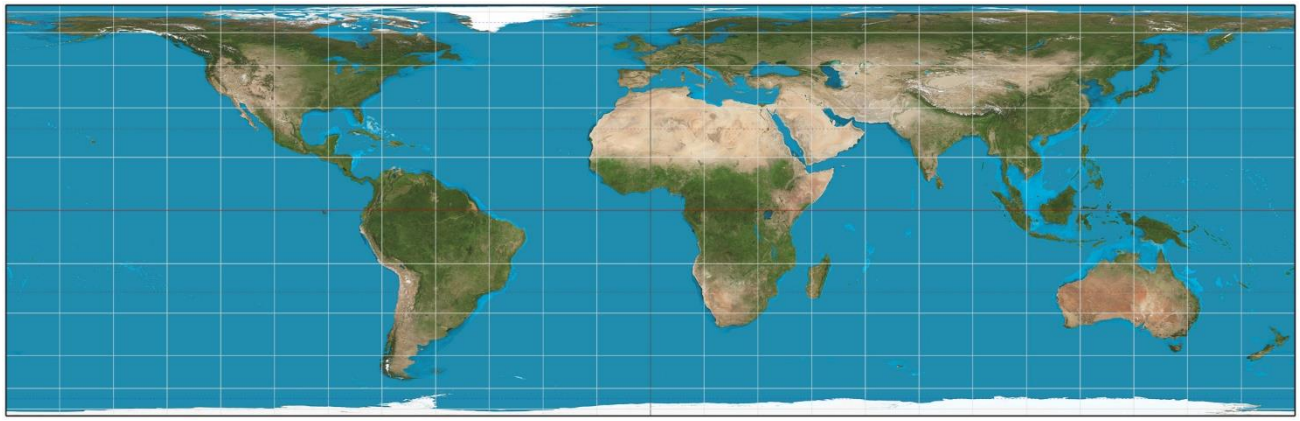
» Les paramètres décrivant le SRID portent **au moins** sur :

- Le système géodésique (*datum*), dit « **non projeté** » : nom, ellipsoïde, demi-grand axe, coefficient d'aplatissement, méridien central, unité d'angles et unité de longueur;
- Le quadrillage cartographique (si le SRID concerne un système **projeté**) : nom de la projection, latitude origine, méridien central, translations en x et y de la fausse origine.
  - Certains systèmes prévoient une documentation sur la troisième dimension z.
  - Un système projeté réclame l'identification parallèle d'un système non projeté.





$x=R*\lambda$
$y=R*\sin(\varphi)$



Exemple de projection cartographique: projection cylindrique équivalente de Lambert en aspect direct tangent

8.3. Codification des systèmes de référence

» SRID = valeur unique, codée par un nombre entier, utilisée pour identifier sans ambiguïté un système de coordonnées géographiques.

- Le système peut être défini en 2D ou en 3D, **projeté** (coordonnées rectangulaires) ou **non projeté** (coordonnées géodésiques).

» Les producteurs de SGBD-spatiales ont défini leurs propres séries de codes SRID ou, plus généralement, font référence à une nomenclature existante.

- La plus connue est celle de l'**EPSG** (« *European Petroleum Survey Group* ») gérée depuis 2005 par l'*International Association of Oil & Gas Producers (OGP) Surveying and Positioning Committee* (http://www.epsg.org/)
- Le code EPSG est devenu une référence absolue dans le domaine SIG et est exploité par tous les outils proposant des transformations de SCR

» Quelques exemples de codes EPSG et du SCR associé:

Systèmes locaux
- 31370: Lambert belge 72 (projeté avec *x, y* en mètres)
- 3812: Lambert belge 2008 (projeté avec *x,y* en mètres)

Systèmes globaux
- 4326: WGS84 (non-projeté avec longitude et latitude en degrés)
- 3857: Spherical Mercator ou « Google Mercator » (projeté avec x,y en mètre)
  - SRID officieux: 900913

8.4. Implémentation sous forme de métadonnées

» Le SRID fait partie des propriétés des objets spatiaux

- Ex: dans PostGIS, le SRID est renseigné comme propriété du type « geometry »

» Une métatable du SGBD spatial contient la définition de chaque SCR associée à son code SRID (EPSG)

- Ex: PostGIS reprend les définitions aux formats WKT et Proj4
- Il est généralement possible de définir son propre SRID (nouvel enregistrement dans la table des SRID), en choisissant une valeur non encore existante et en fournissant tous les paramètres requis.

	srid [PK] integer	auth_name character varying(256)	auth_srid integer	srtext character varying(2048)	proj4text character varying(2048)
2140	4324	EPSG	4324	GEOGCS["WGS 72BE",DATUM["WGS_1972_Tra	+proj=longlat +ellps=WGS72 +tow
2141	4326	EPSG	4326	GEOGCS["WGS 84",DATUM["WGS_1984",SPHE	+proj=longlat +datum=WGS84 +no_
2142	4328	EPSG	4328	GEOCCS["WGS 84 (geocentric)",DATUM["W	+proj=geocent +datum=WGS84 +uni
2143	4330	EPSG	4330	GEOCCS["ITRF88 (geocentric)",DATUM["I	+proj=geocent +ellps=GRS80 +uni
2144	4331	EPSG	4331	GEOCCS["ITRF89 (geocentric)",DATUM["I	+proj=geocent +ellps=GRS80 +uni
2145	4332	EPSG	4332	GEOCCS["ITRF90 (geocentric)",DATUM["I	+proj=geocent +ellps=GRS80 +uni
2146	4333	EPSG	4333	GEOCCS["ITRF91 (geocentric)",DATUM["I	+proj=geocent +ellps=GRS80 +uni
2147	4334	EPSG	4334	GEOCCS["ITRF92 (geocentric)",DATUM["I	+proj=geocent +ellps=GRS80 +uni
2148	4335	EPSG	4335	GEOCCS["ITRF93 (geocentric)",DATUM["I	+proj=geocent +ellps=GRS80 +uni

Métatable spatial_ref_sys de PostGIS

Table spatialisée

Autres attributs non spatiaux		Colonne spatiale
…	…	Coordonnées des géométries + SRID + autres métadonnées spatiales

Exemple d'implémentation d'une table des SRID documentant une base de données spatiales

Métatable

Table des SRID

SRID	Auth_Name	Auth_SRID	SRText
Integer	CharVar (256)	Integer	CharVar (2048)

EPSG

2029

PROJCS["NAD27(76) / UTM zone 17N",GEOGCS["NAD27(76)",DATUM["North_American_Datum_1927_1976" ,
SPHEROID["Clarke 1866",6378206.4,294.9786982139103,AUTHORITY["EPSG","7008"]],AUTHORITY ,
PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]] ,UNIT["degree",0.01745329251994328,AUTHORITY["EPSG","9122"]],AUTHORITY ,
UNIT["metre",1,AUTHORITY["EPSG","9001"]] ,
PROJECTION["Transverse_Mercator"],PARAMETER["latitude_of_origin",0],PARAMETER["central_meridian",-81] ,PARAMETER["scale_factor",0.9996],
PARAMETER["false_easting",500000],PARAMETER["false_northing",0],AUTHORITY

Description standardisée (OGC) d'un système de coordonnées projetées (UTM Zone 17N) du datum NAD27 (USA) dans le format WKT

9. SGBD PostGIS

9.1. PostGreSQL et PostGIS

» PostGIS est une implémentation du schéma spatial standardisé de l'OGC (*OGC Simple Features for SQL Specification*) pour l'enregistrement de données géographiques au sein du SGBD relationnel « PostgreSQL ».

- Projet **Open Source** développé par *Refractions Research* (Canada).

» **PostGIS** est développé comme un jeu de **fonctions** et de **types de données** permettant de spatialiser les tables du SGBD relationnel PostgreSQL et de les manipuler avec des requêtes spatiales (objet relationnel).

9.2. Les géométries

» Les types de géométries reconnus correspondent aux *Simple Features* de l'OGC, **plus** les collections hétérogènes, étendus aux dimensions 3D et 4D :

- *Point, LineString, Polygon*
- *MultiPoint, MultiString, MultiPolygon*
- *GeometryCollection*

» *Ces types peuvent être définis commes des « **geometry** » pour les systèmes projetés ou comme des « **geography** » pour les système sphériques (non-projetés)*

9.3. Les fonctions spatiales

- Les fonctions supportées par PostGIS couvrent les fonctionnalités proposées par l'OGC *(cf. chapitre « Traitements : Fonctions spatiales »)* :
  - De gestion : création et suppression de colonnes spatiales, assignation d'un SRID par géométrie.
  - Topologiques : basées sur la proximité (*ex. distance*) et les relations (prédicats) d'algèbre topologique (*disjoint*, *intersect*…).
  - De traitement géométrique : centroïde, longueur, surface, polygone convexe, espace tampon…
  - D'accès aux propriétés géométriques (SRID, enveloppe…).
  - De construction des géométries au départ de fichiers WKT ou WKB.
    - Propose aussi une fonction d'importation de shapefiles (shp2pgsql)
- D'autres extensions peuvent dépendre de PostGIS
  - *Exemple: PGRouting pour les calculs de distance dans un graphe spatial (« routing »).*

9.4. L'indexation

- L'indexation d'une base de données permet d'accélérer les requêtes impliquant plusieurs tables et les requêtes de recherche.
- Tous les SGBD proposent des méthodes d'indexation
- PostGres et PostGIS:
  - Les données **alphanumériques** peuvent être indexées à l'aide d'un arbre binaire (index standard de PostGreSQL).
  - Les données **spatiales** peuvent être indexées par un arbre R (jugé non optimal) ou par un arbre de recherche généralisé « GiST » (*Generic Index Structure*)
    - Les données spatiales sont réparties selon les relations topologiques de voisinage, de couverture et d'inclusion, dans un arbre de recherche de format R^+(GiST).

Table spatialisée

Autres attributs non spatiaux		Colonne spatiale (geometry)
…	…	**Coordonnées des géométries et métadonnées**

Contraintes sur les données spatiales

Vue Geometry_Column

F_Table_ Catalog	F_Table_ Schema	F_Table_ Name	F_Geometry_ Column	Coord_ Dimension	SRID	Type
Varchar(256)	**Varchar(256)**	**Varchar(256)**	**Varchar(256)**	**Integer**	**Integer**	**Varchar(30)**
Non utilisé par PostGIS	Par défaut dans PostGIS	Nom de la table spatialisée	Nom de la colonne spatiale	Dimensions: 2, 3 ou 4	N° du SRID	Type de géométrie

Table Ref_Sys	SRID	Auth_Name	Auth_SRID	SRText	Proj4Text
	Integer	**Varchar(256)**	**Integer**	**Varchar(2048)**	**Varchar(2048)**
		Organisme ayant défini le SR	N° du SR donné par l'organisme	Définition du SR en format WKT	Possibilité de changement de coordonnées

Pour plus de détails, voir :
Manuel PostGIS sur Internet

10. SGBD et logiciels SIG

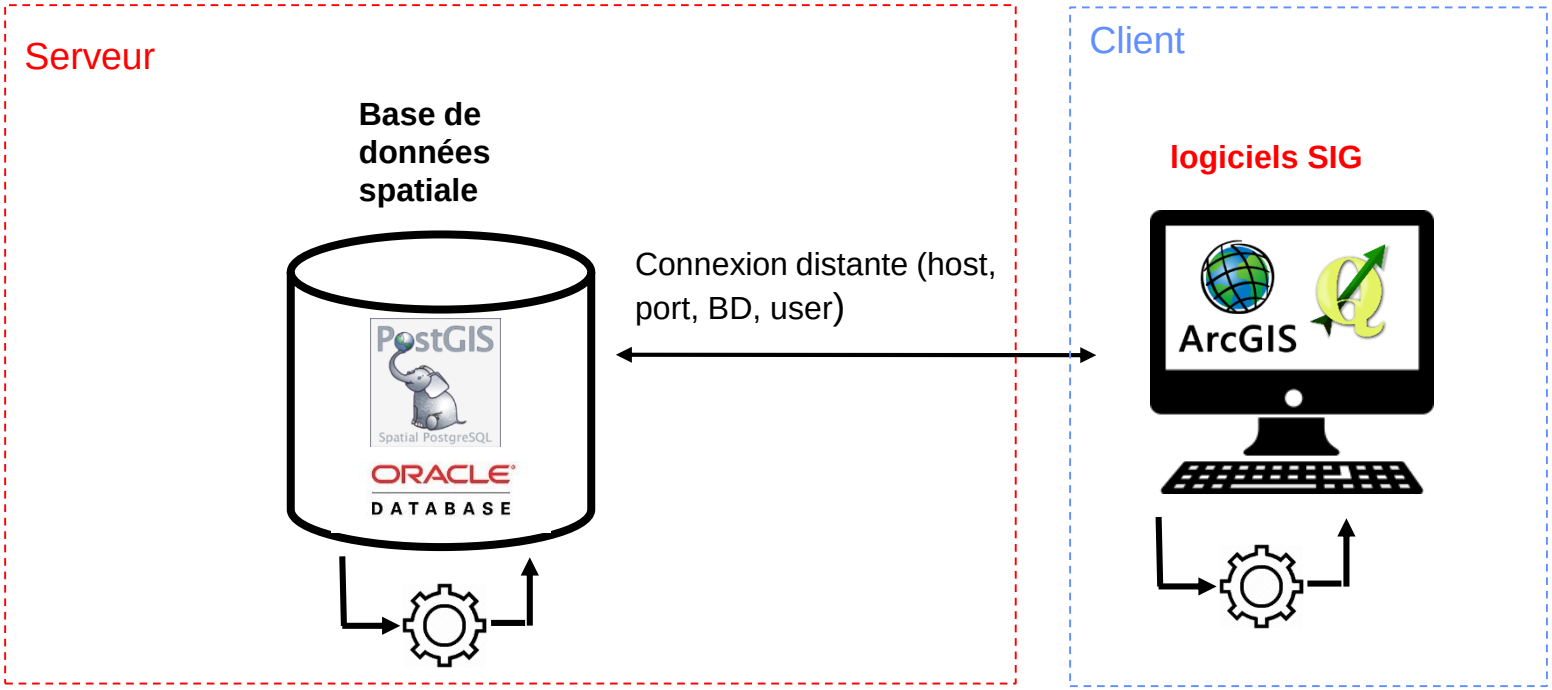
10.1. Pourquoi une couche supplémentaire ?

» Les solutions offertes par les fournisseurs principaux de SGBD (Oracle, IBM-Informix, PostgreSQL, etc.) pour stocker et gérer l'information géographique sont appréciées par les grandes organisations :

- Utilisent les **mêmes SGBD** pour les autres SI de l'organisation.
- Approche **traditionnelle** des SGBD déjà maîtrisée.
- **Fiabilité** des tâches d'administration de données, notamment en matière de **grandes bases de données** (« *backups* », multibases et bases réparties, « *warehouses* », etc.) et **sécurité**.

» Ces solutions ne prennent cependant pas en charge certaines fonctions du SIG et du SOG de l'organisation, telles que :

- Saisie de données géographiques et pré-traitements (géoréférencement, …).
- Outils de conception de bases de données spatiales (AGL) et d'intégration de données (ETL).
- Communication des données géographiques, soit la production cartographique, y compris interactive.
- Tâches spécialisées relevant de l'analyse spatiale au sens large.



Connexion entre une base de données spatiale et un logiciel SIG client

10.2. Rôles du logiciel médiateur

10.2.1. Vis-à-vis du client

- Offre au client une vue externe la plus évoluée possible (OO) de la base de données spatiales (même si le SGBD sous-jacent ne repose pas sur ce modèle).
- Offre à l'utilisateur une vue homogène des informations spatiales en présence de multiples SGBD et/ou SDGF.
- Offre à l'utilisateur une interface unique pour effectuer des requêtes spatiales (SQL enrichi) et des traitements géométriques sur les objets spatiaux.
  - Répartit la charge des traitements entre clients et serveur selon les capacités des clients.
  - Gère les requêtes et transactions distribuées entre plusieurs serveurs (de données spatiales ou SGBD) (« *multithreads* »).
- Permet de transmettre des requêtes attributaires classiques.
- Retourne aux clients les résultats des requêtes et traitements, éventuellement sous forme de cartes.

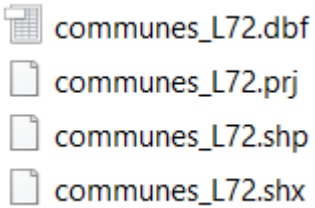
10.2.2. Vis-à-vis du SGBD

- Effectue la transposition entre le modèle de l'utilisateur (vue externe) et le schéma propre du SGBD.
  - Convertit les données alimentant la base dans le modèle propre du SGBD (« *data loading* »).
  - Extrait les données requises pour la résolution des divers traitements spatiaux, et effectue certaines opérations.
- Se contente de transmettre les requêtes attributaires ne requérant pas ses services.
- Peut dans certains cas assurer les fonctions d'administration propres aux données spatiales que n'effectuerait pas le SGBD, par exemple :
  - Indexation spatiale pour optimiser l'accès aux données.

10.2.3. Exploitation des capacités spatiales du SGBD

- Toute extension du modèle du SGBD pour la prise en compte de données spatiales facilite (réduit) le rôle du serveur de données spatiales.
  - Les fonctionnalités de spatialisation du médiateur « s'effacent » au profit du SGBD dès que celui-ci offre les fonctionnalités adéquates (en théorie…).

- **11. Exercice dans PostGIS**
  - » Création d'une base de données spatiale
    - Ouvrir PGAdmin 4 (mot de passe: postgres)
    - Connexion au serveur postgres local
    - Créer une base de données « td_sig »
    - Activer l'extension PostGIS pour la BD « td_sig »
      - Dans la console SQL de PGAdmin, taper la commande:
        - CREATE EXTENSION postgis
  - » Insertion de données spatiales depuis un shapefile
    - Téléchargement des données
      - http://geomatics.ulg.ac.be/download/communes_be.rar
    - Décompresser le fichier communes_be.rar dans le dossier D:\etudiants
      - Shapefile des communes belges dans SCR Lambert 72



- Ouvrir l'invite de commande dans le dossier D:\etudiant
  - Taper « cmd » dans l'explorateur windows une fois dans le bon dossier
- Convertir le shapefile en fichier SQL en utilisant la commande shp2pgsql de postgres

SRID — Fichier shp à convertir — Fichier SQL en output

```
D:\etudiants>shp2pgsql -s 31370 communes_L72.shp > communes_be.sql
Shapefile type: Polygon
Postgis type: MULTIPOLYGON[2]
```

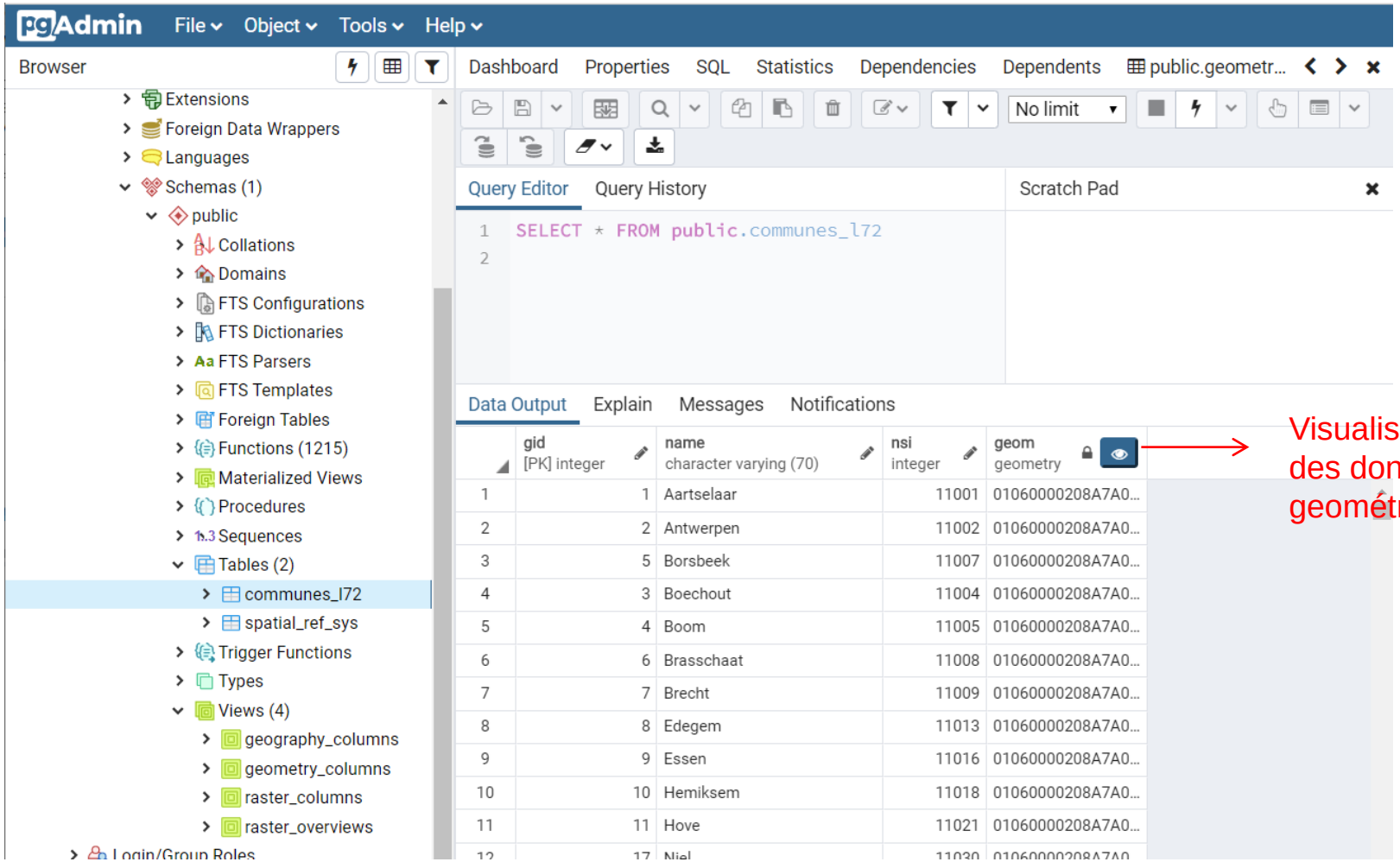
- Lecture du fichier sql avec la fonction psql

Utilisateur — database — host — port — Fichier SQL à exécuter

```
D:\etudiants>psql -U postgres -d td_sig -h localhost -p 5432 -f communes_be.sql
Mot de passe pour l'utilisateur postgres :
SET
SET
BEGIN
CREATE TABLE
ALTER TABLE
                        addgeometrycolumn
-----------------------------------------------------------------
 public.communes_l72.geom SRID:31370 TYPE:MULTIPOLYGON DIMS:2
(1 ligne)


INSERT 0 1
INSERT 0 1
```

» Visualisation de la table créée dans PGAdmin



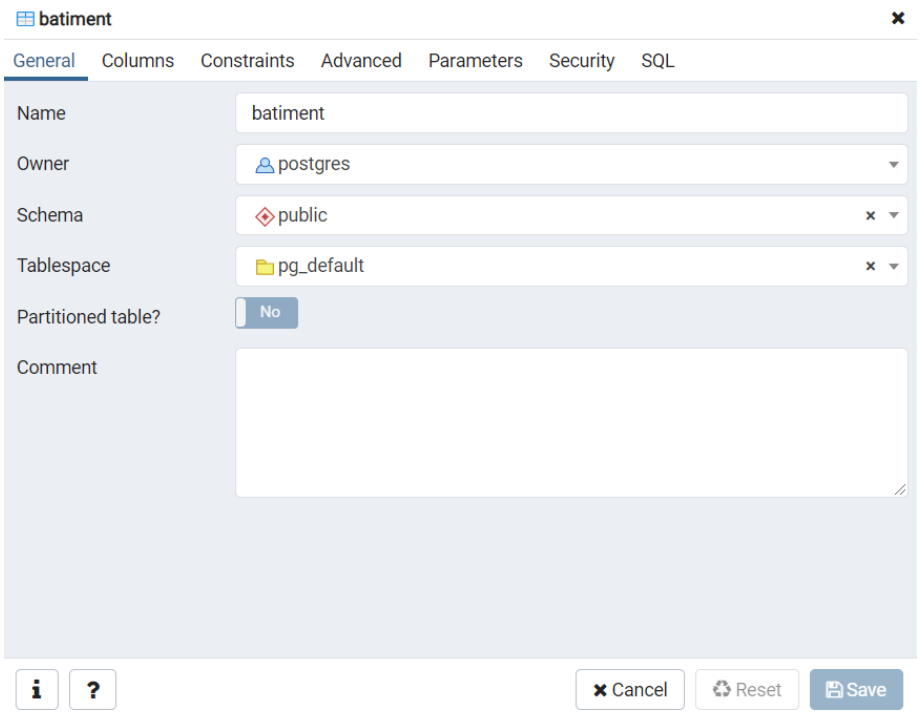
» Visualisation des données géométriques de la table « communes_l72 »

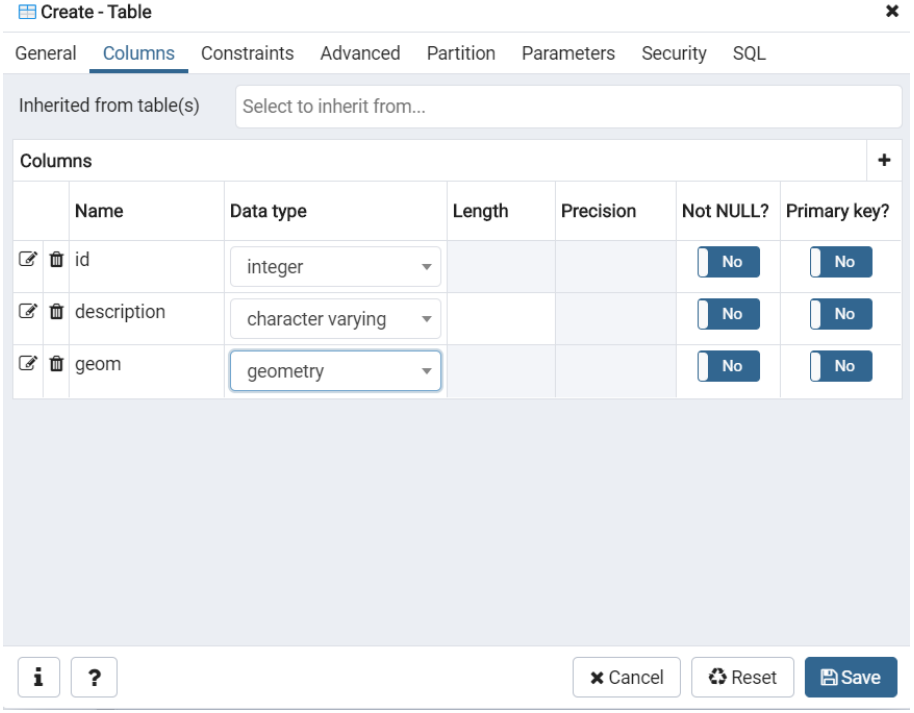
» Visualisation des contraintes spatiales dans la vue « geometry_columns »

	f_table_catalog character varying (256)	f_table_schema name	f_table_name name	f_geometry_column name	coord_dimension integer	srid integer	type character varying (30)
1	td_sig	public	communes_l72	geom	2	31370	MULTIPOLYGON

» Création d'une table vide « localisation » avec attribut « geom « de type « geometry »

- Ne pas oublier de définir la contrainte de clé primaire sur l'attribut « id »





- En SQL, cela donne:

```
CREATE TABLE public.communes_l72
(
    gid integer NOT NULL DEFAULT nextval('communes_l72_gid_seq'::regclass),
    name character varying(70) COLLATE pg_catalog."default",
    nsi integer,
    geom geometry(MultiPolygon,31370),
    CONSTRAINT communes_l72_pkey PRIMARY KEY (gid)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.communes_l72
    OWNER to postgres;
```

```
CREATE TABLE public.batiment
(
    id integer NOT NULL,
    description character varying COLLATE pg_catalog."default",
    geom geometry,
    CONSTRAINT pk PRIMARY KEY (id)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.batiment
    OWNER to postgres;
```

» Insertion d'un enregistrement dans la table « batiment »

- Id: 1
- Description: Bâtiment de Physique
- Latitude: 50.582°
- Longitude: 5.566°

» Requête SQL dans la console de PGAdmin:

Reprojection de WGS84 vers Lambert 72 (SRID 31370)

Définition du SCR WGS84 (SRID 4326)

Conversion d'un point WKT en géométrie PostGIS

```
INSERT INTO batiment(id, description, geom)
VALUES(1,'Bâtiment de Physique',
    st_transform(
        st_setsrid(
            st_geomfromtext(
                'POINT(5.566 50.582)'
            ),
            4326
        ),
        31370
    )
)
```

» Visualisation de la table « batiment »

	id [PK] integer	description character varying	geom geometry
1	1	Bâtiment de Physique	01010000208A7A0...

» Dans quelle commune se trouve le bâtiment de physique?

- Requête SQL dans PGAdmin

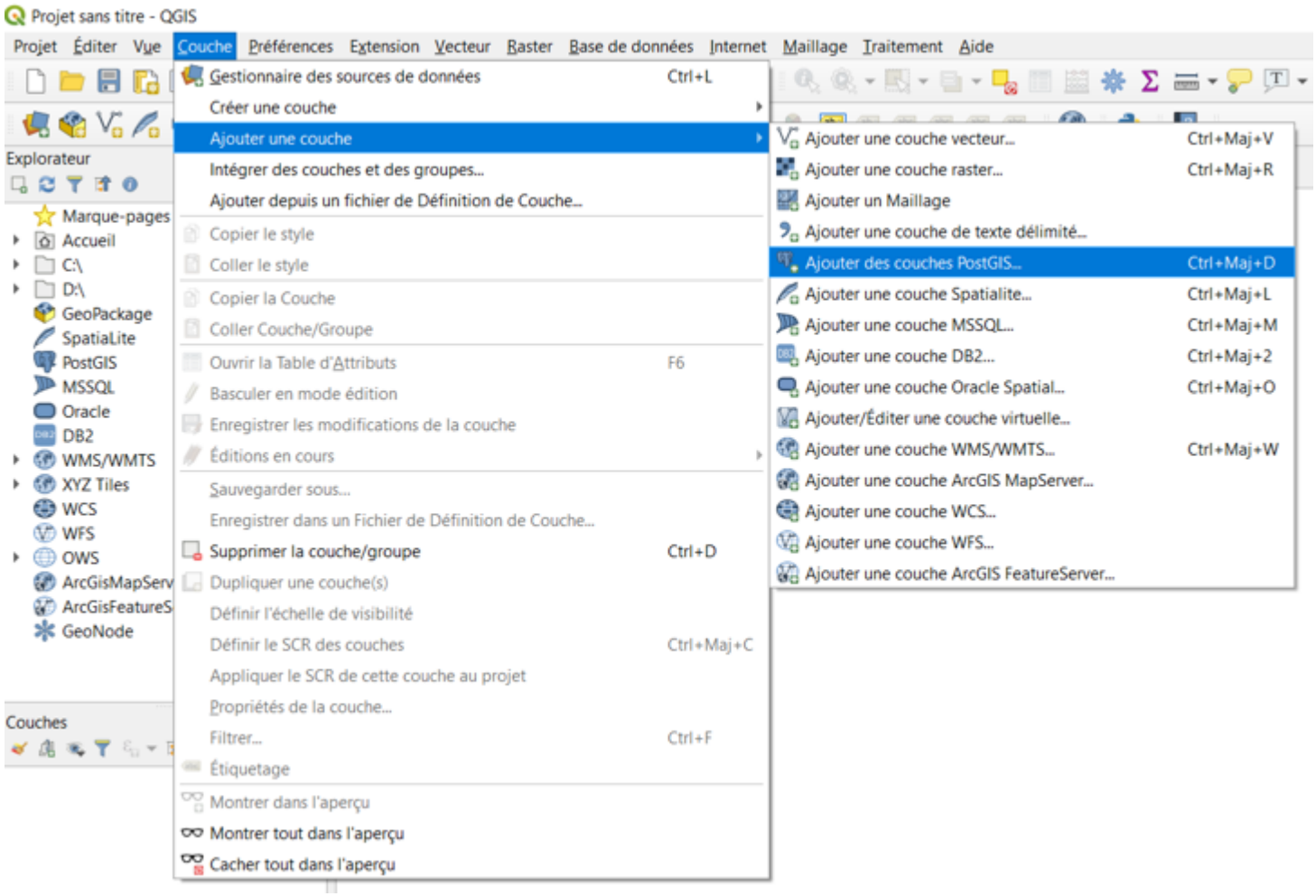
Intersection

```
SELECT communes_l72.name
FROM batiment, communes_l72
WHERE st_intersects(batiment.geom, communes_l72.geom)=true
```

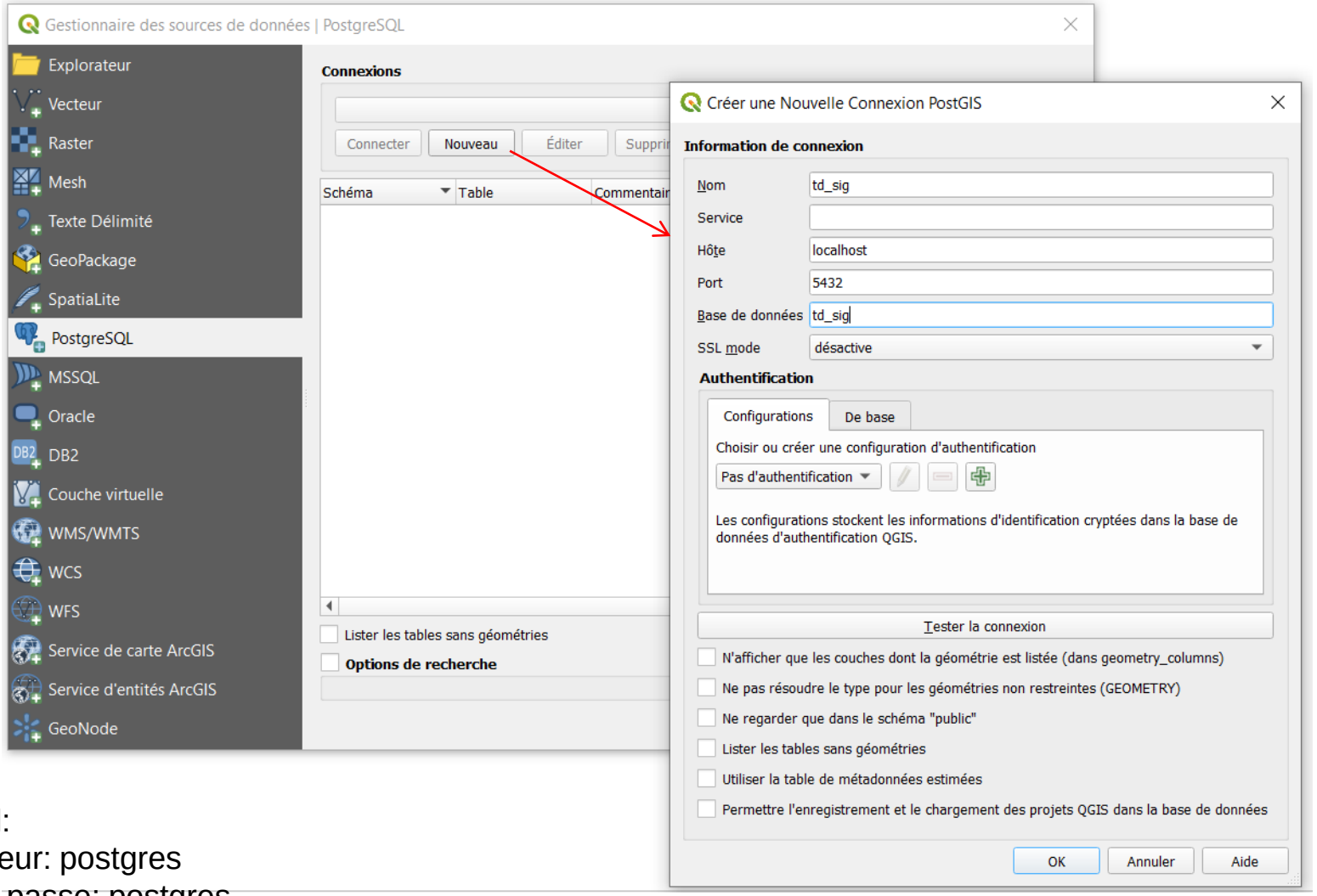
Data Output | Explain | Messages | Notifications

	name character varying (70)
1	Liège

» Connexion avec QGIS

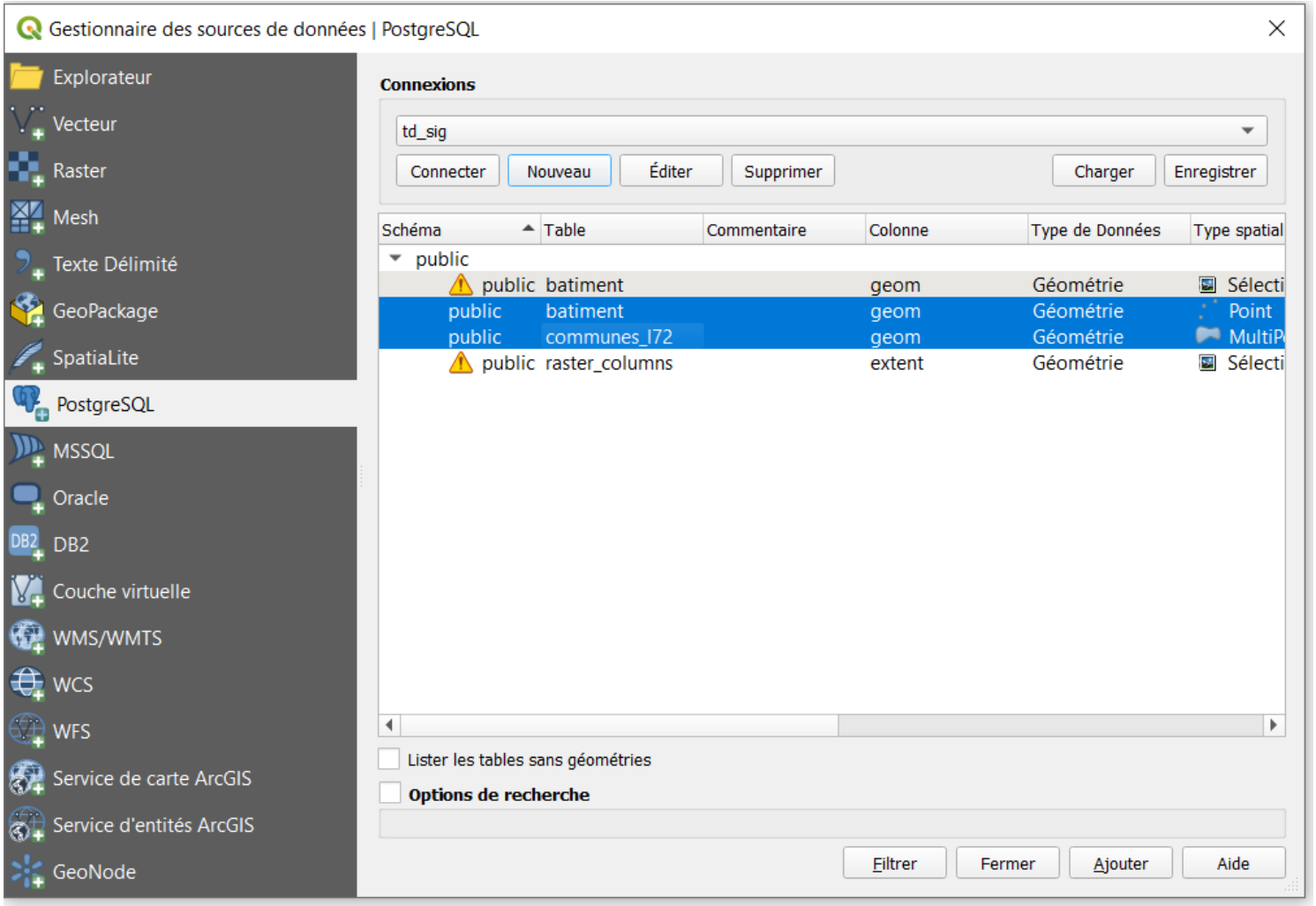


- Définition d'une nouvelle connexion vers la BD « td_sig »

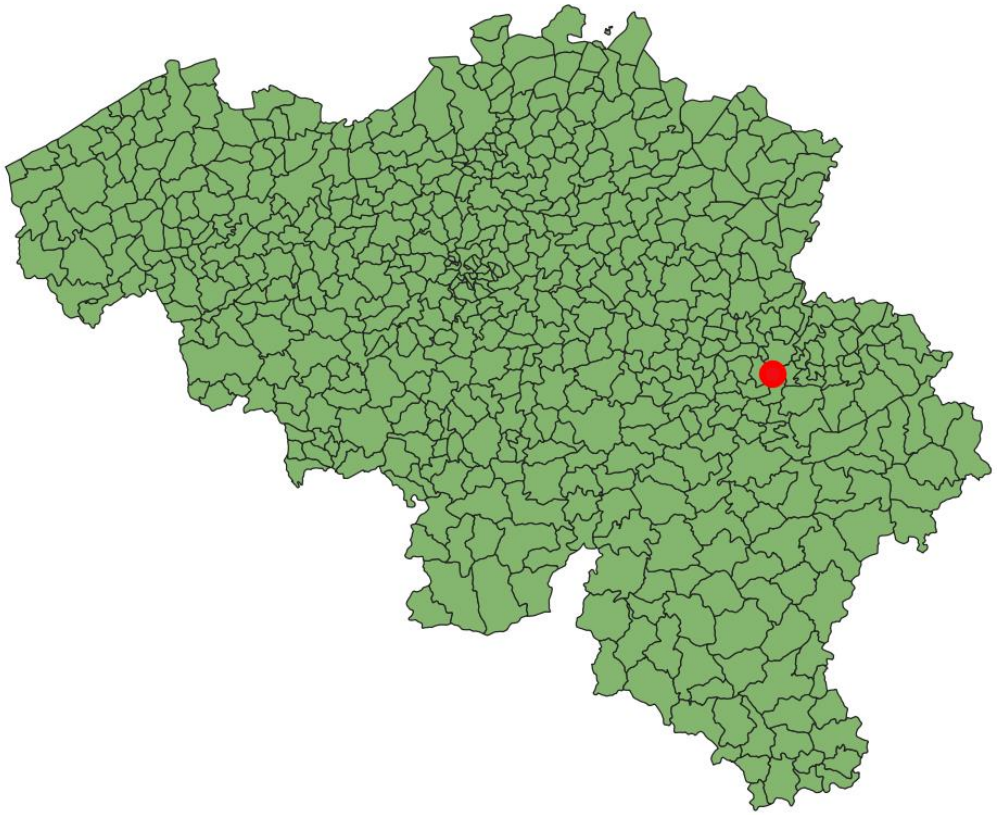


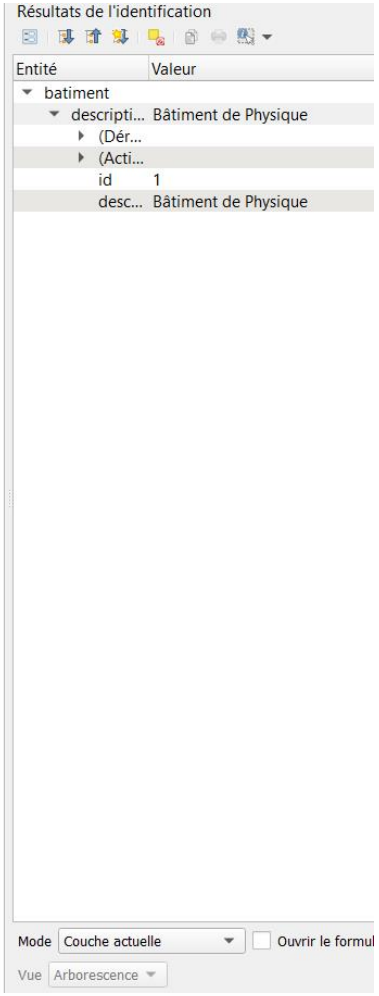
Rappel:
Utilisateur: postgres
Mot de passe: postgres

- Après connexion, sélectionner les deux tables créées dans PostGIS



- Résultat:




Résultats de l'identification
Entité
Valeur
batiment
descripti...
Bâtiment de Physique
(Dér...
(Acti...
id
1
desc...
Bâtiment de Physique
Mode
Couche actuelle
Ouvrir le formul
Vue
Arborescence