Modélisation logique et physique

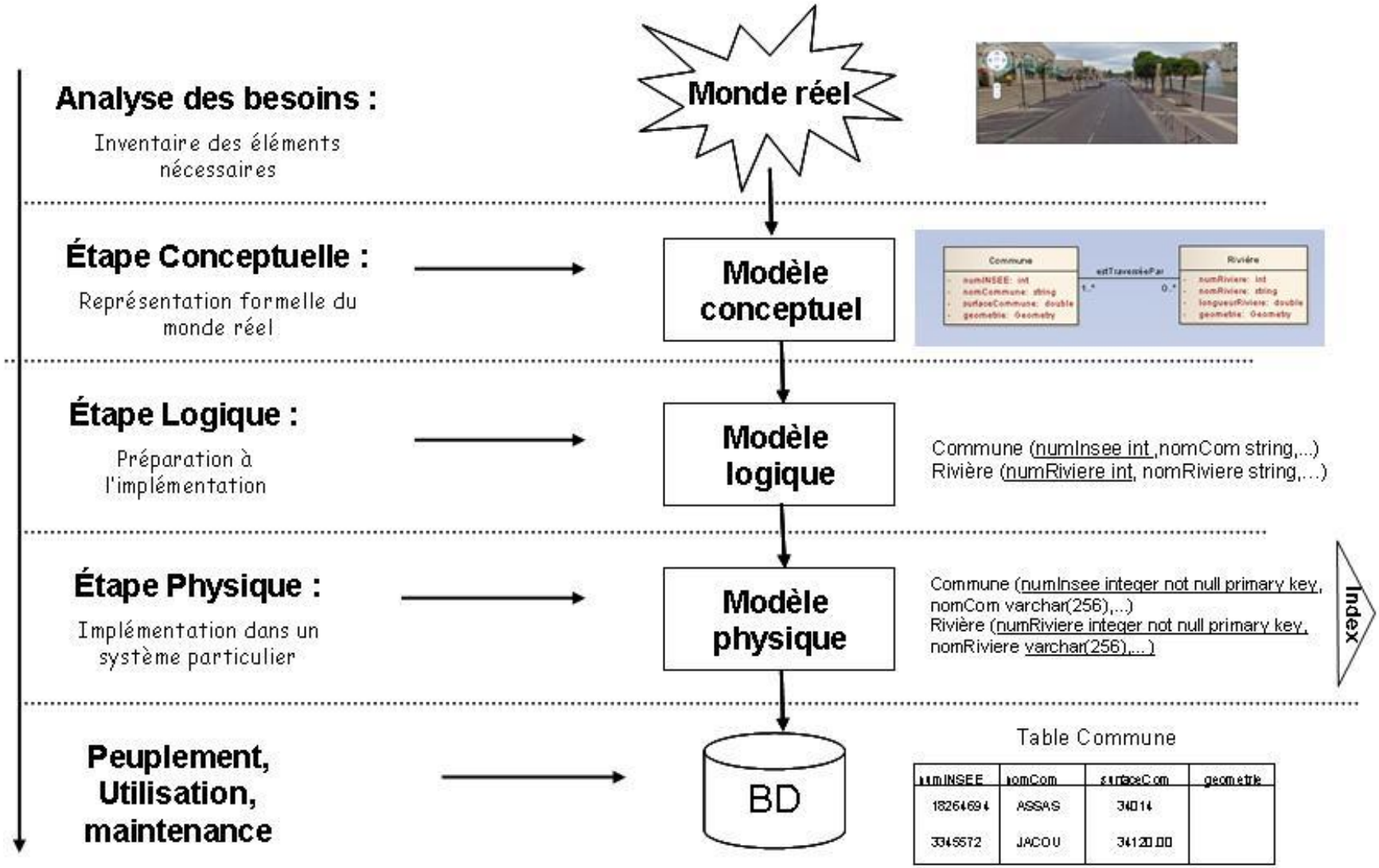
Christelle Pierkot, Jean-Christophe Desconnets

UE BD Spatiales

Février 2012

Plan

Cycle de vie d'une base de données



Modèle logique

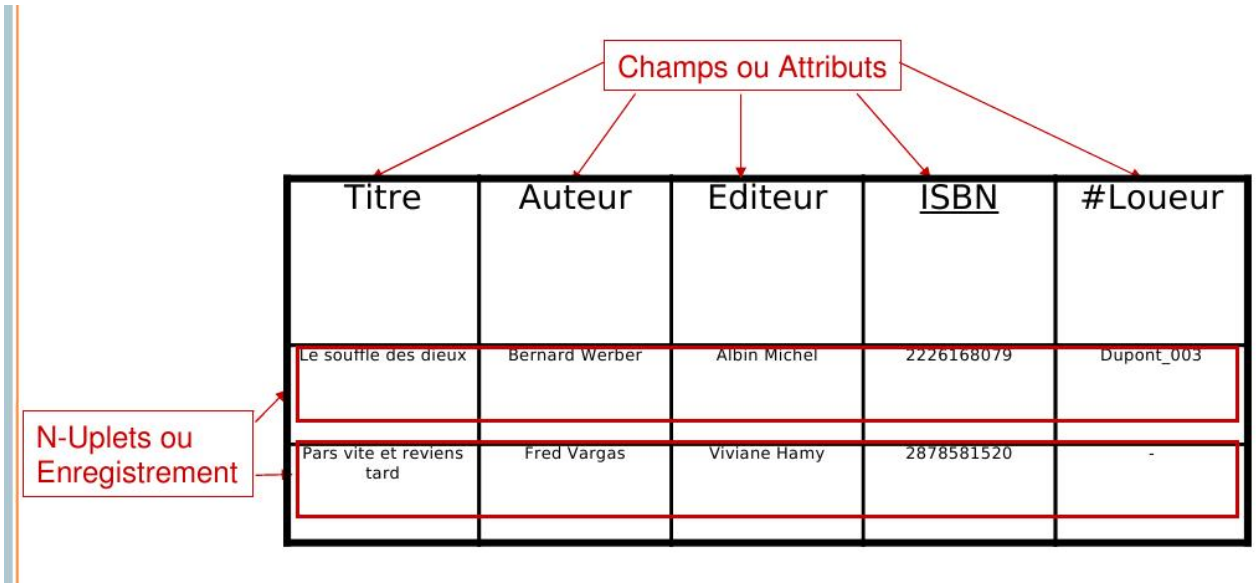
- But : Décrire la structure de données utilisée sans faire référence à un langage de programmation.
- C'est une représentation du système tel qu'il sera implémenté dans un ordinateur
- Transformation du modèle conceptuel dans le formalisme spécifique au SGBD
- Dépend du type de modèle de données utilisé dans le SGBD.
- Formalismes de modélisation
  - Modèle relationnel
    - Modèle basé sur l'algèbre linéaire
  - Modèle objet-relationnel
    - Relationnel + structures complexes + attributs multivalués
  - Modèle objet
    - données structurées en classes et instances de classes

Le modèle relationnel

- Défini par Edgar F. Codd en 1970
- **Schéma informatique** des données
  - Modèle d'organisation sous forme de tables (tableaux à deux dimensions).
  - Simplicité de la structure de données
- Manipulation des données selon le concept mathématique de **relation de la théorie des ensembles**
  - Simplicité des opérateurs
  - Toute opération relationnelle sur une table (union, intersection, différence, ...) génére une autre table

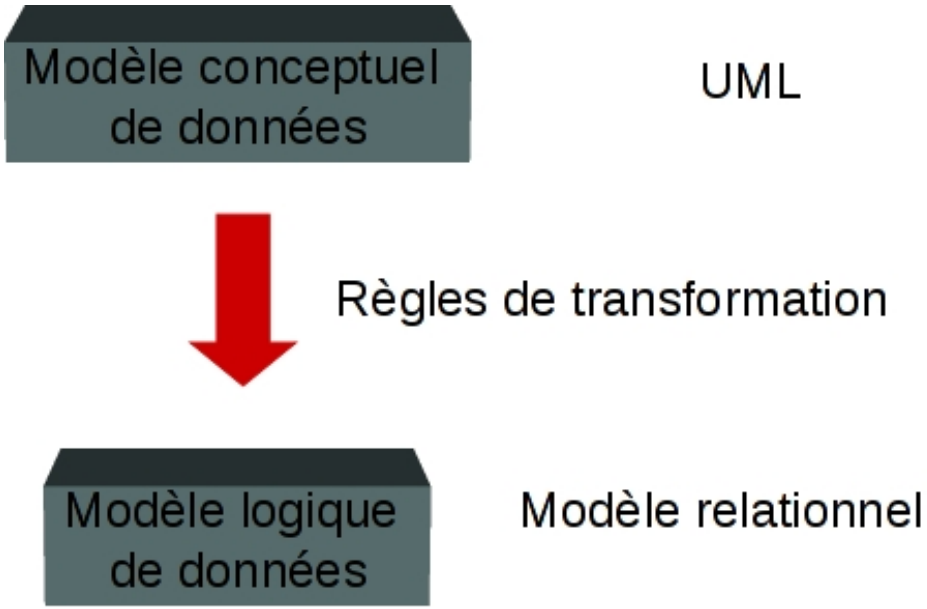
Exemple

- Relation Livre
  - Schéma : Livre(ISBN, Titre, Auteur, Editeur, #Loueur)
  - Représentation



Passage du modèle conceptuel au modèle logique

- Dépend du formalisme utilisé pour le modèle conceptuel et du formalisme utilisé pour le modèle logique
- Transformation grace à des règles



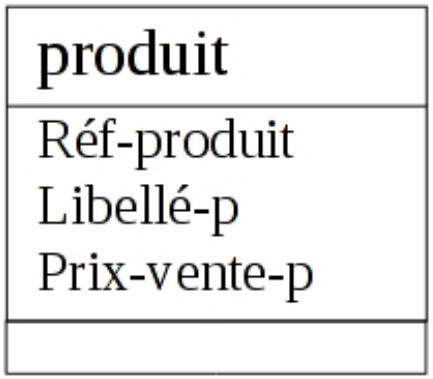
Règles de transformation de UML au modèle relationnel

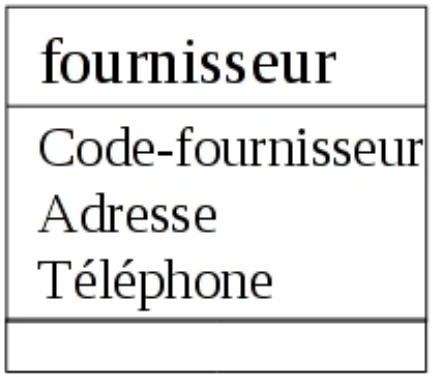
- **Règles de transformation**
  - Nom de la classe UML → Nom de la relation (table)
  - Attributs de la classe UML → Champs de la table
  - Relation entre classes → Création d'un nouveau champ ou d'une nouvelle table selon la multiplicité

- **Comment définir la clé primaire ?**
  - Il existe un attribut de la classe qui est unique et pérenne et qui ne prend jamais la même valeur pour des objets différents, alors il peut servir de clé primaire. *Exemple : ISBN pour un livre ou numéro sécurité sociale pour une personne*
  - Il n'existe pas d'attribut dans la classe pouvant remplir ces conditions, il faut alors générer un identifiant.

Exemple classes et attributs

Classe





Schémas Relationnels

Produit (Réf-produit, Libellé-p, Prix-vente-p)

Fournisseur (Code-fournisseur, Adresse, Téléphone)

Convertir les relations de type Association, agrégation et composition

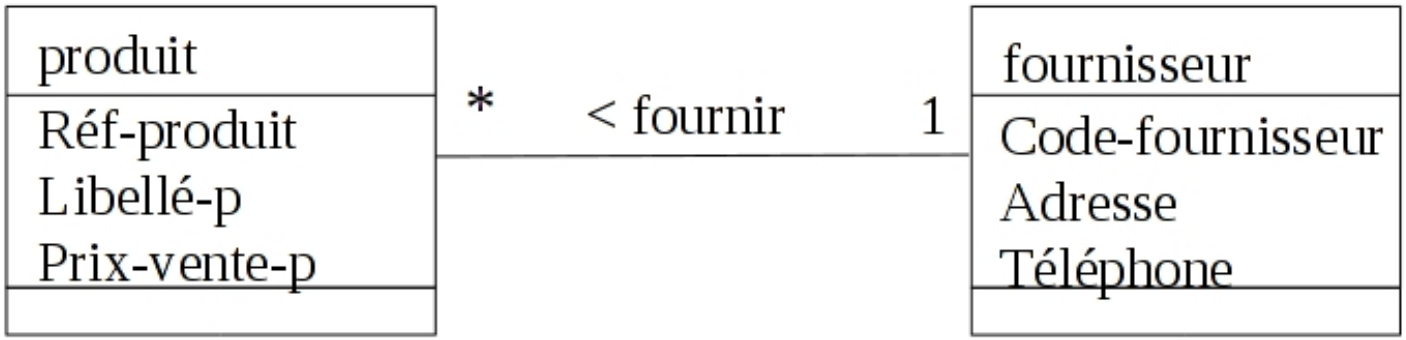
- **Relation de cardinalité 1-1 entre deux classes A et B** : Création d'un attribut représentant l'identité de la classe B (resp. A) dans la table définie par A (resp. B) i.e. création d'une clé étrangère référençant B (resp. A) dans la table définie par A (resp. B)

- **Relation de cardinalité 1-n ou 1-* entre A et B** : Création d'un attribut représentant l'identité de la classe A dans la table définie par B i.e. création d'une clé étrangère référençant A dans la table définie par B. (idem si 1 remplacé par (0..1)

- **Relation de cardinalité n-m ou *-* entre A et B** : Création d'une table supplémentaire composé des deux attributs représentant l'identité des deux classes. La clé primaire de la nouvelle table est composée des deux attributs.

Exemple relations 1-* ou 1-n

Diagramme de Classe :



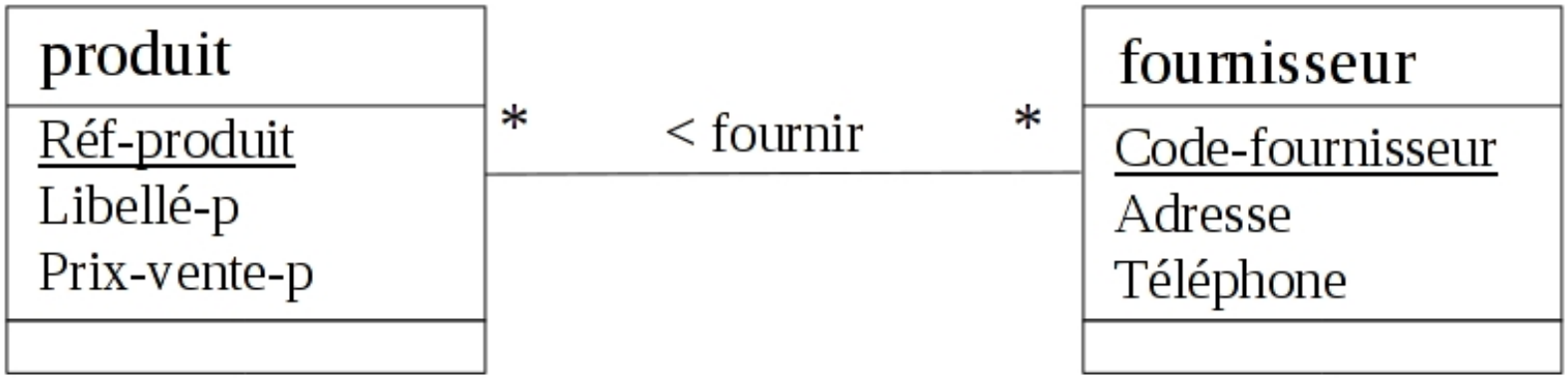
Schémas Relationnels :

Produit (Réf-produit, Libellé-p, Prix-vente-p, #Code-fournisseur)

Fournisseur (Code-fournisseur, Adresse, Téléphone)

Exemple relations *-* ou *-n ou n-m

Diagramme de Classe :



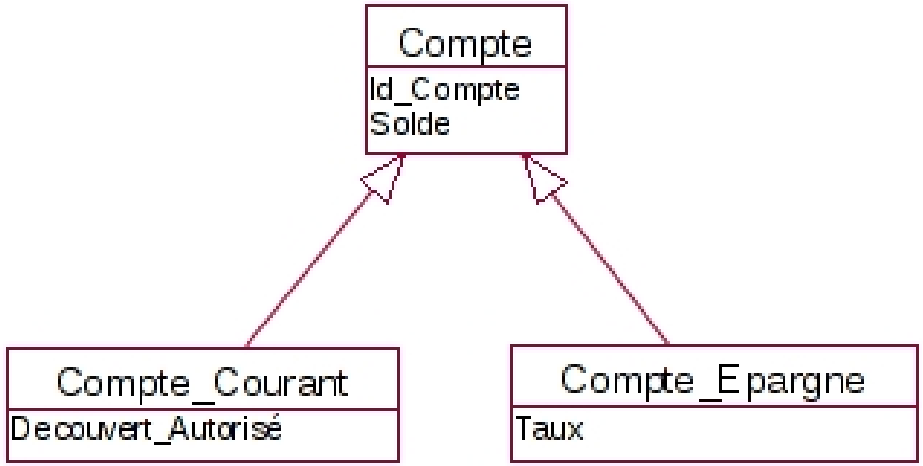
Schémas Relationnels

Produit (Réf-produit, Libellé-p, Prix-vente-p)

Fournisseur (Code-fournisseur, Adresse, Téléphone)

Fournir (Réf-produit, Code-fournisseur)

Convertir les relations de type généralisation/spécialisation



1 Chaque classe possède sa propre table

- Les sous classes ont la même clé primaire que la super classe
- Eventuellement, un attribut type est ajouté à la super classe
- *Exemple :*

  Compte(Id_Compte, solde, type) avec type = courant, épargne

  Courant(Id_Compte, decouvert_autorisé)

  Epargne(Id_Compte, taux)

Convertir les relations de type généralisation/spécialisation

2 On ne représente que les sous classes

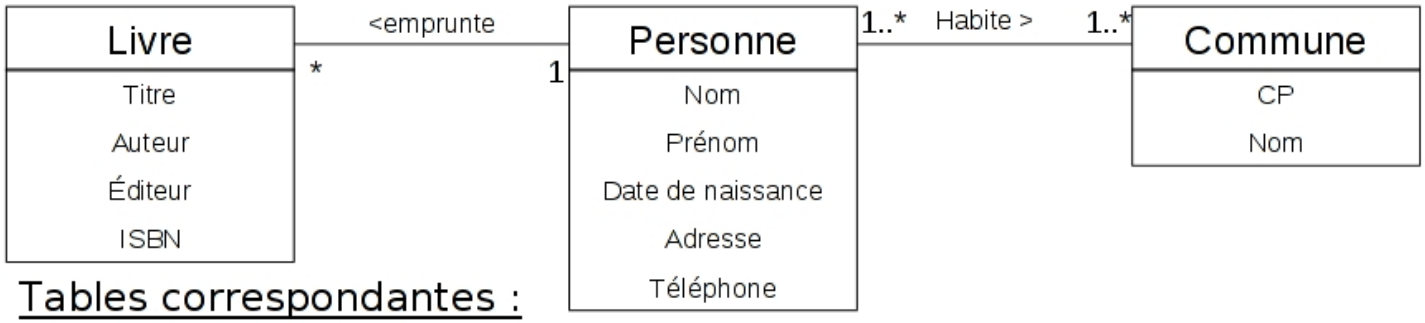
- Les attributs de la super classe descendent dans chaque sous-classe
- *Exemple :*
  Courant(Id_Compte, solde, decouvert_autorisé)
  Epargne(Id_Compte, solde, taux)

3 Une seule classe regroupe tout

- Problème de gestion des valeurs nulles pour les attributs exclusifs à l'une ou l'autre des sous classes
- *Exemple :*
  Compte(Id_Compte, solde, decouvert_autorisé, taux)

Exemple complet

- Schéma UML :



- Tables correspondantes :

tbl_Livre
Titre
Auteur
Éditeur
ISBN
#Id_Personne

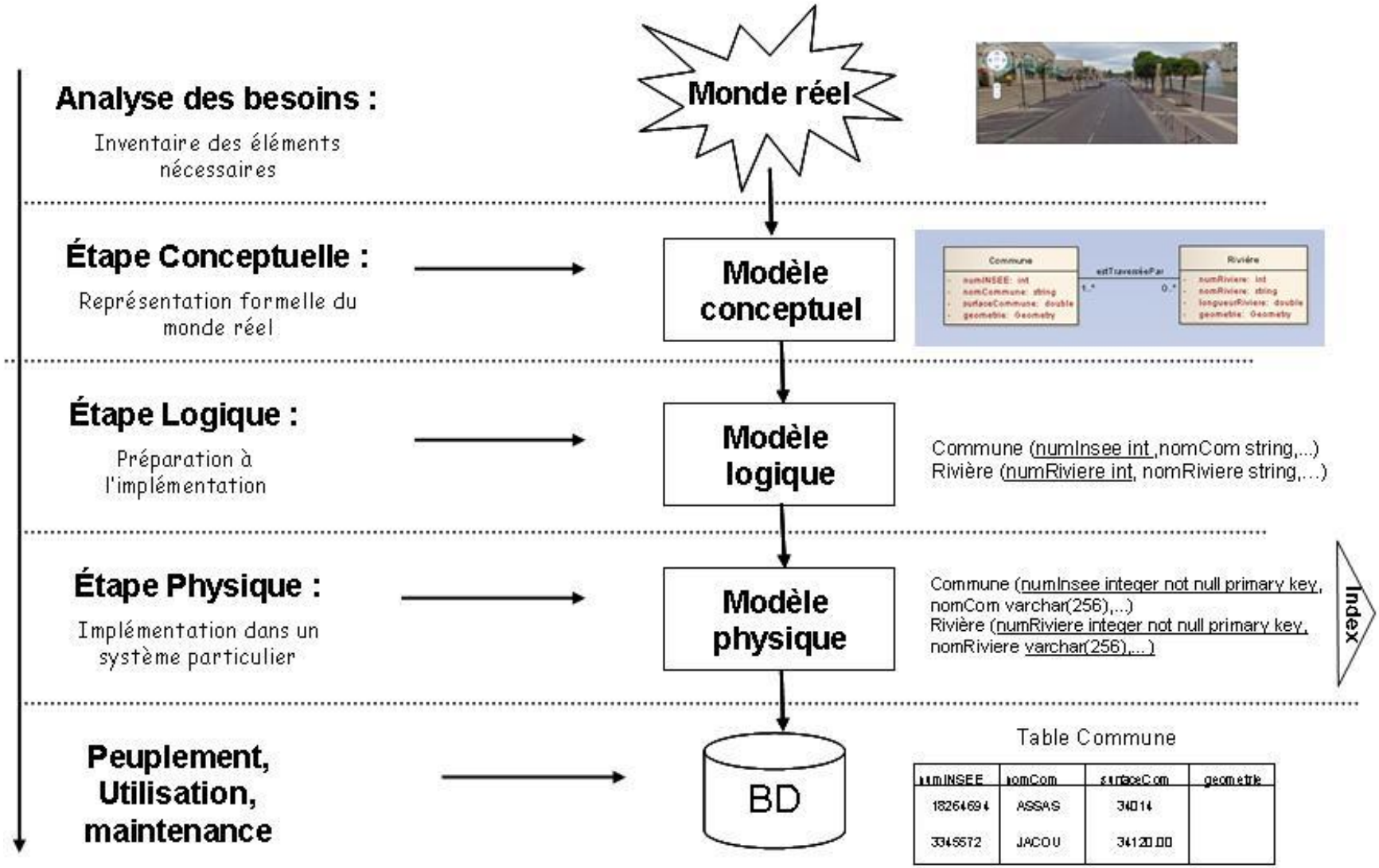
tbl_Personne
Id_Personne
Nom
Prénom
Date de naissance
Adresse
Téléphone

tbl_lien_Personne_Commune
Id_Commune
Id_Personne

tbl_Commune
Id_Commune
CP
Nom

- Schémas correspondants :
  - tblLivre(ISBN,titre, auteur, éditeur, #Id_Personne)
  - tbl_Personne(Id_Personne, nom, prénom, date_naissance, adresse, téléphone)
  - tbl_Commune(Id_Commune, CP, nom)
  - tbl_lien_Personne_Commune(Id_Personne, Id_Commune)

Cycle de vie d'une base de données



Modélisation physique

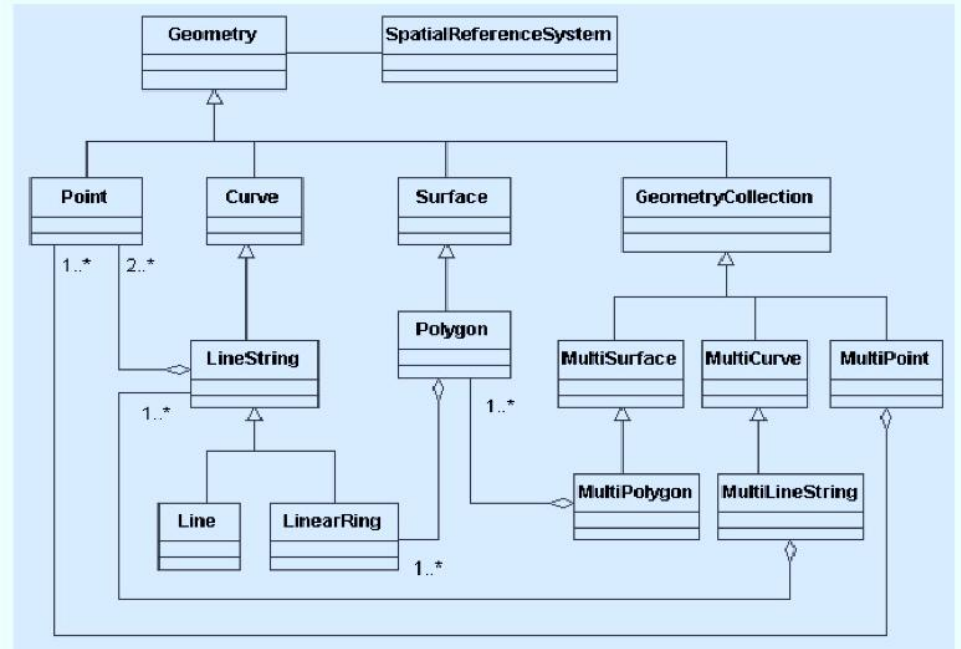
- Répond à la question : comment ?
- But : Implémenter le modèle logique dans la BD
  - Définition des systèmes de stockages employés : Types de données, index, triggers, ...
  - Traduction dans un langage de définition de données (SQL par exemple)
- Gestion du stockage et de l'accès aux données
- Les SGBD spatiaux :
  - PostGis, Oracle spatial, DB2 Spatial, MySQL Gis et SQL Server Spatial

PostGreSQL/PostGis

- **PostGreSQL** est un SGBD de type objet-relationnel
- La cartouche **PostGis** permet d'en faire un SGBD spatial
- **Bibliothèques complémentaires**
  - PROJ4 pour la reprojection des coordonnées
  - GEOS pour effectuer les opérations géométriques et topologiques (intersection, buffer, ...)

Types géométriques et Systèmes de référence

- **Types d'objets géométriques**
  - Conformes à ceux définis par l'OpenGIS Consortium (OGC).



  - Point, Multipoint, Linestring, Multilinestring, Polygon, Multipolygon et GeometryCollections

- **SRID (Spatial Referencing system IDentifier)**
  - Identifiant du système de référence des objets géographiques
  - Obligatoire pour stocker les objets géographiques dans la BD

Formats de stockage

- **Format WKT (Well Know Text) de l'OpenGis**
  - Format de stockage "simple" des objets géométriques
  - *Exemples :*

    POINT(0 0)

    LINESTRING(0 0,1 1,1 2)

    POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))

    MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))

    GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))

- **Format WKB (Well Know Binary) de l'OpenGis**
  - Format de stockage "avancé" des objets géométriques
  - Représentation par un ensemble d'octets.
  - *Exemple : La valeur WKB correspondante à POINT(1 1) est la séquence suivante :*

    01010000000000000000F03F000000000000F03F avec :

    Byte order : 01 (Type de stockage)

    WKB type : 01000000 (code qui indique le type de géométrie (Point, LineString, Polygon, ...))

    Coord. X : 000000000000F03F

    Coord. Y : 000000000000F03F

Les tables SPATIAL_REF_SYS et GEOMETRY_COLUMNS

- Tables de métadonnées spécifiées par l'OpenGis

- **La table Spatial_Ref_Sys**
  - Liste les systèmes de référencement spatial et les systèmes de projection cartographiques planaires et géodésiques.
  - Schéma :
    SRID : Un entier qui identifie de façon unique le système de références spatiales (SRS) de la base.
    AUTH_NAME : Le nom du système de référence.
    AUTH_SRID : L'identifiant du SRS définie par l'autorité cité dans le AUTH_NAME.
    SRTEXT : La représentation Well-Known Text (WKT) du SRS.
    PROJ4TEXT : Définitions de coordonnées Proj4 pour un SRID particulier.
  - Un grand nombre de systèmes de références répertoriés par défaut (WGS84 Long Lat, RGF 93 Lambert 93,..)

- **La table Geometry_Column**
  - Liste les tables dont les fonctionnalités spatiales ont été activées
  - Schéma :
    F_TABLE_CATALOG, F_TABLE_SCHEMA, F_TABLE_NAME : le nom totalement qualifié de la table de propriétés contenant la colonne géométrique.
    F_GEOMETRY_COLUMN : le nom de la colonne géométrique dans la table propriété.
    COORD_DIMENSION : la dimention spatiale (2,3 ou 4) de la colonne.
    SRID : l'identifiant du système de référence spatiale utilisée pour les coordonnées géométiques dans cette table. C'est une clef étrangère faisant référence à la table SPATIAL_REF_SYS.
    TYPE : le type d'objet géographique.

Créer des tables et des objets spatiaux

- **Créer une table**
  - CREATE TABLE riviere (numRiviere integer not null PRIMARY KEY, nomRiviere varchar(256), longueur real,numINSEE integer, FOREIGN KEY (numINSEE) references commune) ;
- **Ajouter une colonne géométrique**
  - Syntaxe :SELECT AddGeometryColumn(", 'nomTable','nomColGeom','numSRID','typeGeom',dimension) ;
  - SELECT AddGeometryColumn(",'riviere','the_geom','2154','MULTILINESTRING',2) ;
- **Insérer une donnée spatiale dans la table**
  - Syntaxe :INSERT INTO nomTable (att1, att2, nomColGeom) VALUES (valAtt1, valAtt2, GeomFromText('typeGeom(listeCoord)', numSRID)) ;
  - INSERT INTO riviere (numRiviere,nomRiviere,longueur,the_geom) VALUES ('5530','fosse la transide','3623.1201171875000000',GeomFromText('MULTILINESTRING((769986.7073358211 6290779.744180864, 769989.6601997252 6290799.728719401, 769986.7073358211 6290779.744180864))',2154)) ;