UMR 8199
GÉNOMIQUE INTÉGRATIVE ET MODÉLISATION DES MALADIES MÉTABOLIQUES

Optimiser l'utilisation de sa base de données avec SQL -

premiers pas avec SQL

Stefan.Gaget@cnrs.fr

Objectifs :

- Utiliser des bases de données
  - Pour stocker des informations
  - Pour en faciliter l'utilisation
    - c'est à dire l'extraction selon différents critères
- Compétences à acquérir
  - Modélisation
    - Structurer les informations
  - Langage SQL
  - Utilisation d'un logiciel de base de données
    - PostgreSQL sous Windows

Organisation de la formation

- Principe des bases de données
  - Modélisation d'une base
    - Des données à la structure
    - Codification et contrôle
  - Du modèle conceptuel au modèle relationnel
    - Niveau de modélisation
    - Relation entre les tables
    - Valider la modélisation
  - Langage SQL
    - Éléments du langage
    - Fonctions

- Création de bases de données sur machine
  - Implémentation d'une base
    - Stratégie
    - Erreurs / Astuces
  - Importer les données
    - Solutions
    - Applications
  - Exporter les données
    - Solutions
    - Applications

- Historique :
  - SGBD, outil libre, licence BSD
  - Conçu par Michael Stonebraker
  - Version 1 en 1995
  - Version 9.6.1 27/10/2016
- Application client / serveur
- Large communauté +++
- Nombreuses extentions
  - PostGis : moteur de données spatiales

- Versions de PostgreSQL :
  - Linux / Windows / Mac OS X
  - Portable
- Édition de code :
  - Avec un éditeur de texte : Emacs, Sublime Text, notepad++, …
  - De préférence, avec complétion automatique et colorisation spécifique au langage
- Interfaces utilisateurs (client) :
  - Console : psql
  - Graphique : pgAdmin, phpPgAdmin, HeidiSQL, Dbeaver, ...

- Pour la formation :
  - Monoposte windows
  - PostgreSQL Version 9.xxx
  - Notepad++
  - Psql
  - PgAdmin
  - HeidiSQL

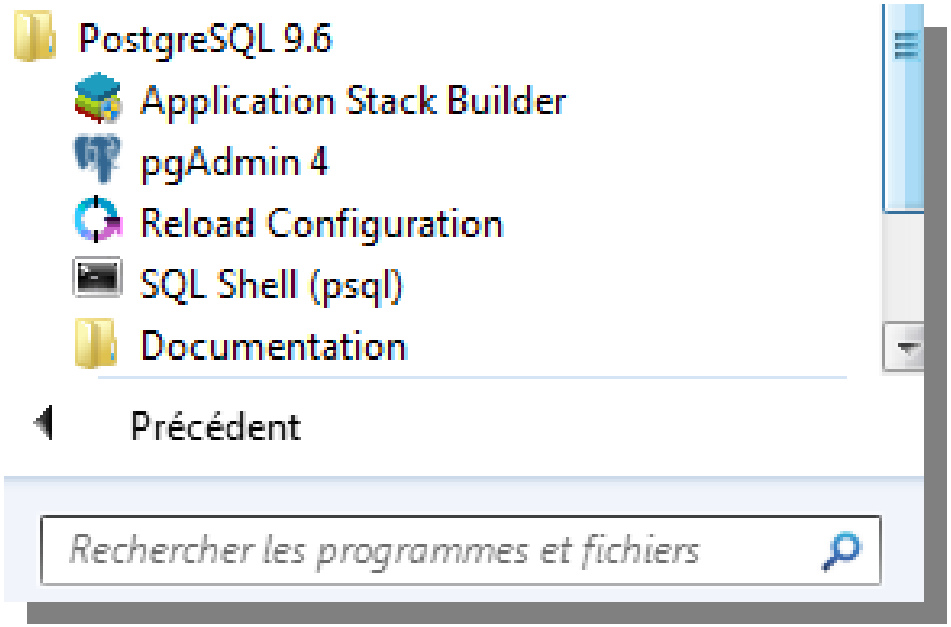
- Weborama :
  - Home page : https://www.postgresql.org
  - Documentation en ligne : https://www.postgresql.org/docs/manuals/
  - Documentation en français : http://docs.postgresqlfr.org/
  - Tutoriels « Premiers pas avec PostgreSQL version 9,4 », http://www.postgresqltutorial.com …
- Livres

Avant la 1ère Partie

Premiers pas avec PostgreSQL

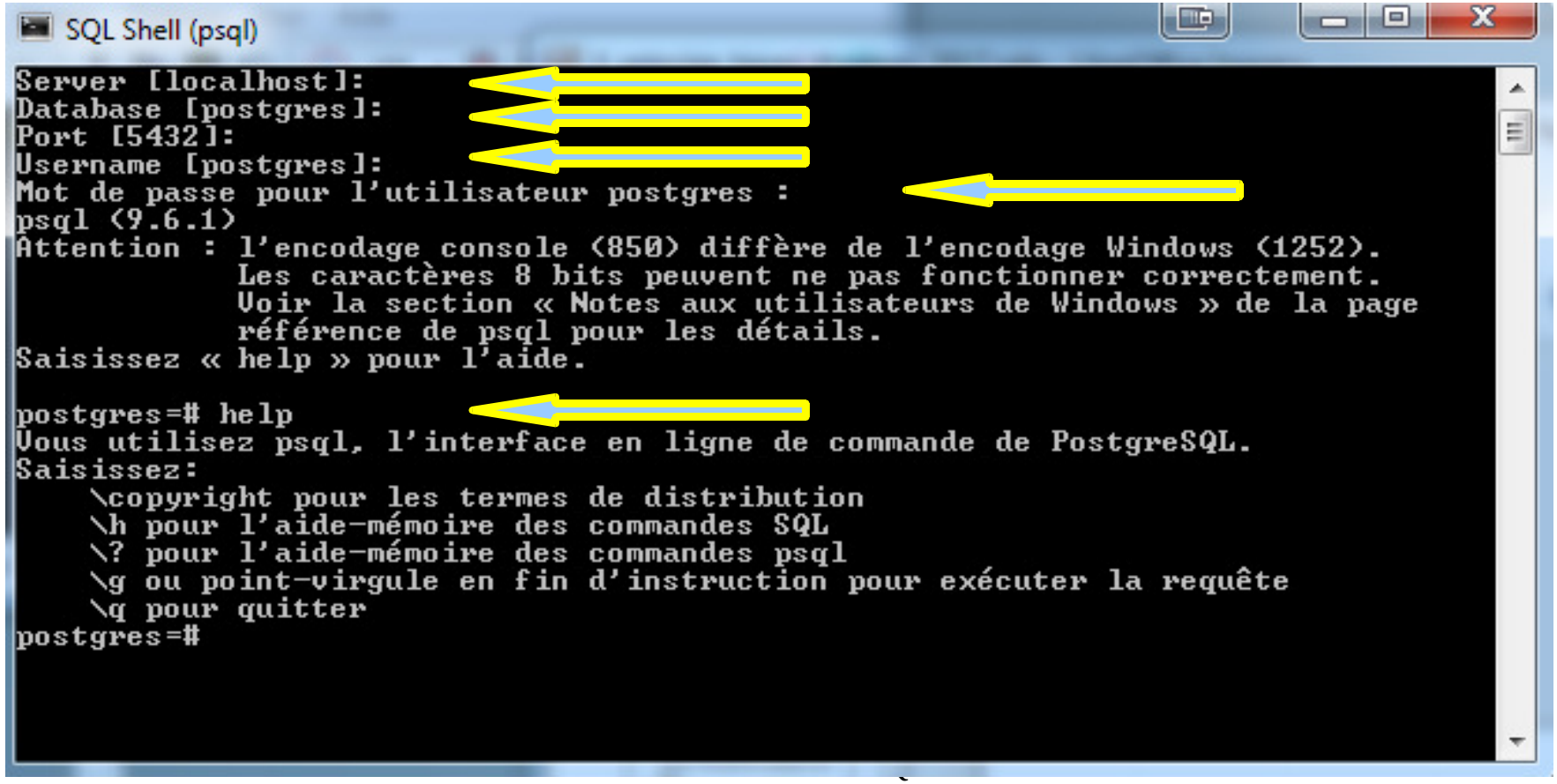
Post-Installation

- Un super-utilisateur « ***postgres*** », mot de passe « ***pgsql*** », port de connexion « ***5432*** »
- Le super-utilisateur dispose de tous les droits, éviter de l'utiliser, sauf si impératif
- Dans le menu démarrer, se trouvent tous les outils nécessaires



Premières manipulations

- Lancez psql et acceptez les réponses par défaut aux questions.



Console Windows et psql

- La console Windows est par défaut dans un encodage compatible DOS.

```
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
            Les caractères 8 bits peuvent ne pas fonctionner correctement.
            Voir la section « Notes aux utilisateurs de Windows » de la page
            référence de psql pour les détails.
```

- Il est préférable de modifier l'encodage de la console. Pour cela, éditez =~ C:\Program Files\PostgreSQL\9.6\scripts\runpsql.bat pour y ajouter la ligne « chcp 1252 » avant de lancer psql

Console Windows et psql

```
1   @echo off
2   REM Copyright (c) 2012-2016, EnterpriseDB Corporation.  All rights reserved
3
4   REM PostgreSQL server psql runner script for Windows
5   chcp 1252
6   SET server=localhost
7   SET /P server="Server [%server%]: "
8
9   SET database=postgres
10  SET /P database="Database [%database%]: "
11
12  SET port=5432
13  SET /P port="Port [%port%]: "
14
15  SET username=postgres
16  SET /P username="Username [%username%]: "
17
18  for /f "delims=" %%a in ('chcp ^|find /c "932"') do @ SET CLIENTENCODING_JP=%%a
19  if "%CLIENTENCODING_JP%"=="1" SET PGCLIENTENCODING=SJIS
20  if "%CLIENTENCODING_JP%"=="1" SET /P PGCLIENTENCODING="Client Encoding [%PGCLIENTENCODING%]: "
21
22  REM Run psql
23  "C:\Program Files\PostgreSQL\9.6\bin\psql.exe" -h %server% -U %username% -d %database% -p %port%
24
25  pause
26
27
```

Premières manipulations

- Commandes psql
  - **\?** pour l'aide des commandes psql (si vous deviez n'en retenir qu'une)
  - **\q** quitter
  - **\h** aide des commandes sql
  - **\l** liste des bases de données
  - **\c** se connecter à une base
  - **\d [nom]** pour la description d'une table, d'un index, séquence, vue
  - **\d** liste des relations (tables, vues et séquences)
  - **\i nom_fichier.sql** exécuter un fichier de commandes SQL

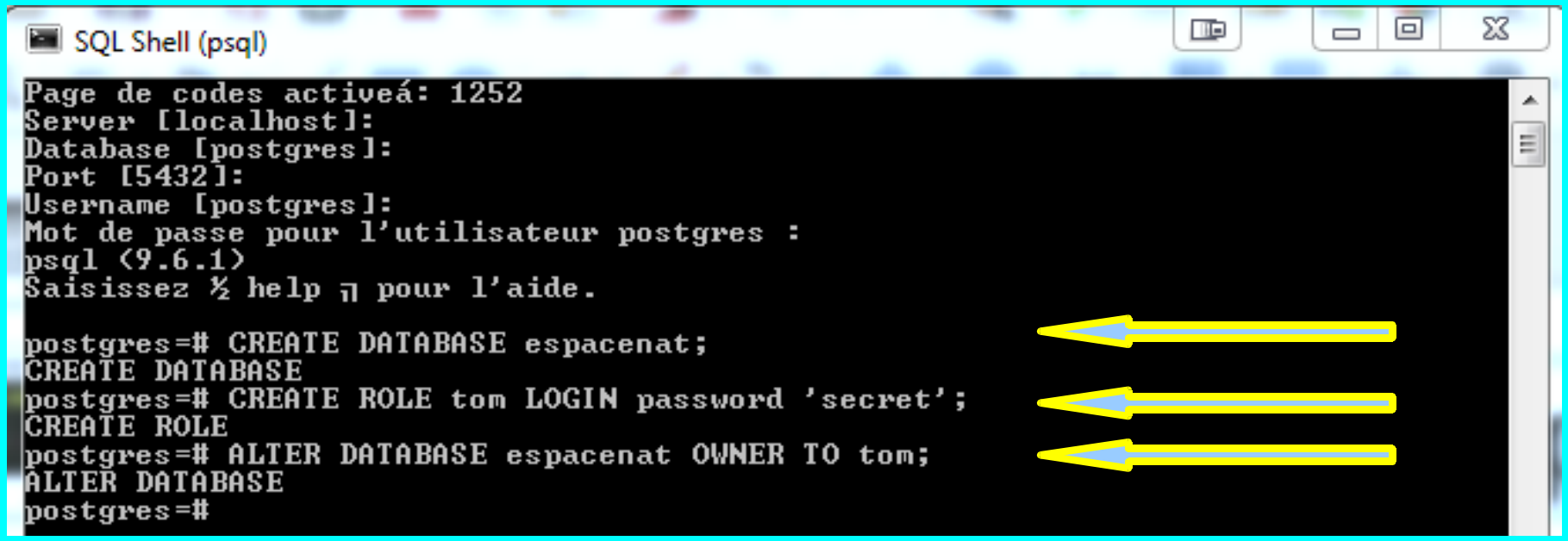
Premières manipulations

- Comment créer dans une base ***espacenat***, une table ***ville***, y insérer des données et utilisable par ***tom*** ?

id_v	nom
0	Marquise
1	Samer

Premières manipulations

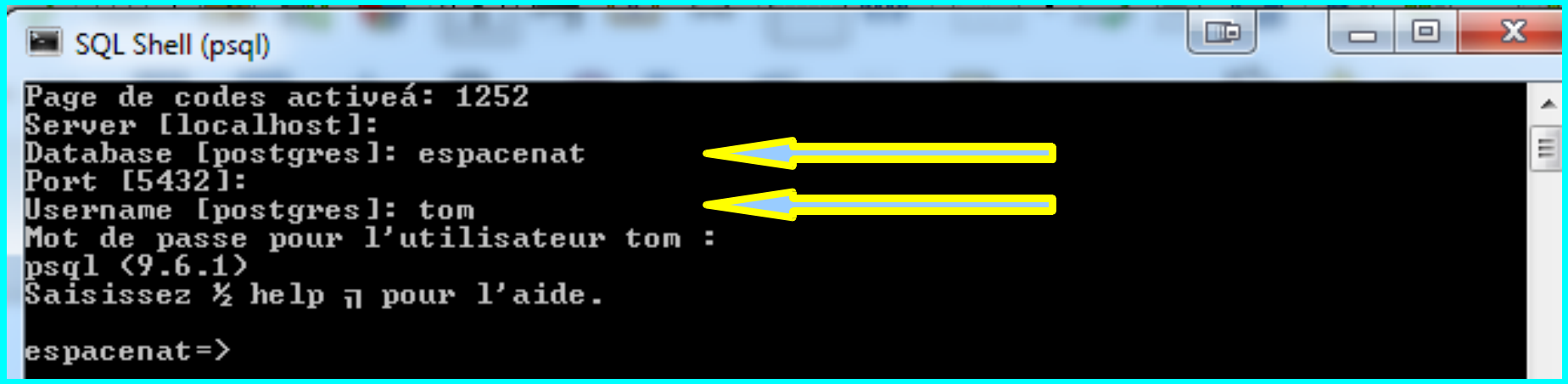
- Créer une base ***espacenat*** et en faire de ***tom*** le propriétaire de la base (afin qu'il puisse l'utiliser comme bon il lui semble)



- Utilisation des commandes du langage SQL : CREATE DATABASE, CREATE ROLE et ALTER DATABASE

Premières manipulations

- Connexion de ***tom*** sur la base ***espacenat***
- Fermez psql (**\q**), relancez-le et re-connectez-vous



- Attention à bien préciser le nom de la base et l'utilisateur. (si vous ne précisez pas le nom de la base, psql essaie de se connecter à la base de même nom que l'utilisateur. Si vous ne précisez pas le nom d'utilisateur, c'est le nom de l'utilisateur du système qui est utilisé.)

Premières manipulations

- Créer la table ***ville*** avec seulement un seul champ

```
espacenat=>
espacenat=> CREATE TABLE ville ( id_v integer NOT NULL, PRIMARY KEY (id_v));
CREATE TABLE
espacenat=> SELECT * FROM ville;
 id_v
------
(0 ligne)
```

- Puis ajout d'un second champ

```
espacenat=> ALTER TABLE ville ADD COLUMN nom CHAR (30) DEFAULT NULL;
ALTER TABLE
espacenat=> SELECT * FROM ville;
 id_v | nom
------+-----
(0 ligne)


espacenat=>
```

- Utilisation des commandes du langage SQL : CREATE TABLE, SELECT et ALTER TABLE

Premières manipulations

- Ajout d'une ligne de valeur
- Puis de la seconde

```
espacenat=>
espacenat=>
espacenat=> INSERT INTO ville(id_v, nom) values (0, 'Marquise');
INSERT 0 1
espacenat=> SELECT * FROM ville;
 id_v |              nom
------+-------------------------------
    0 | Marquise
(1 ligne)


espacenat=> INSERT INTO ville(id_v, nom) values (1, 'Samer');
INSERT 0 1
espacenat=> SELECT * FROM ville;
 id_v |              nom
------+-------------------------------
    0 | Marquise
    1 | Samer
(2 lignes)


espacenat=> _
```

- Utilisation des commandes du langage SQL : INSERT INTO et SELECT

1ère Partie

Modélisation

Données => Structure

Exemple de données

- Parcs ou Espaces Naturels :
  - Parc Naturel Régional Avesnois, Scarpe-Escaut
  - Parc Naturel Transfrontalier du Hainaut
  - Espace Naturel Régional Lille Métropole
- Organismes gestionnaires
  - Département du Nord
  - Région Hauts-de-France
  - Belgique
  - Europe

1er essai d'organisation

Zone	Organismes gestionnaires
PNR des Caps et Marais d'Opale	France, *Région HdF*, Pas-de-Calais
PNR Scarpe-Escaut	France, *Région HdF*, Nord
PNR Avesnois	France, Ré*gion HdF*, Nord
PNT du Hainaut	Europe, France, *Région HdF*, Nord, Belgique, Région Wallone
ENR Lille Métropole	France, *Région HdF*, Nord, Métropole Européenne de Lille

- Remarques :
  - Une zone est gérée par plusieurs organismes
  - Un organisme gère plusieurs zones
  - Certaines informations sont dupliquées
  - => problème de redondance de l'information

A quelles questions peut-on répondre ?

Zone	Organismes gestionnaires
PNR des Caps et Marais d'Opale	France, *Région HdF,* Pas-de-Calais
PNR Scarpe-Escaut	France, *Région HdF,* Nord
PNR Avesnois	France, Rég*ion HdF,* Nord
PNT du Hainaut	Europe, France, *Région HdF*, Nord, Belgique, Région Wallone
ENR Lille Métropole	France, *Région HdF,* Nord, Communauté Urbaine de Lille

- Liste des zones ?
- Liste des organismes ?
- Liste des zones gérées par le Nord ?

Organisation des données en entités et relations

Entité 1

Zone
PNR des Caps et Marais d'Opale
PNR Scarpe-Escaut
PNR Avesnois
PNT du Hainaut
ENR Lille Métropole

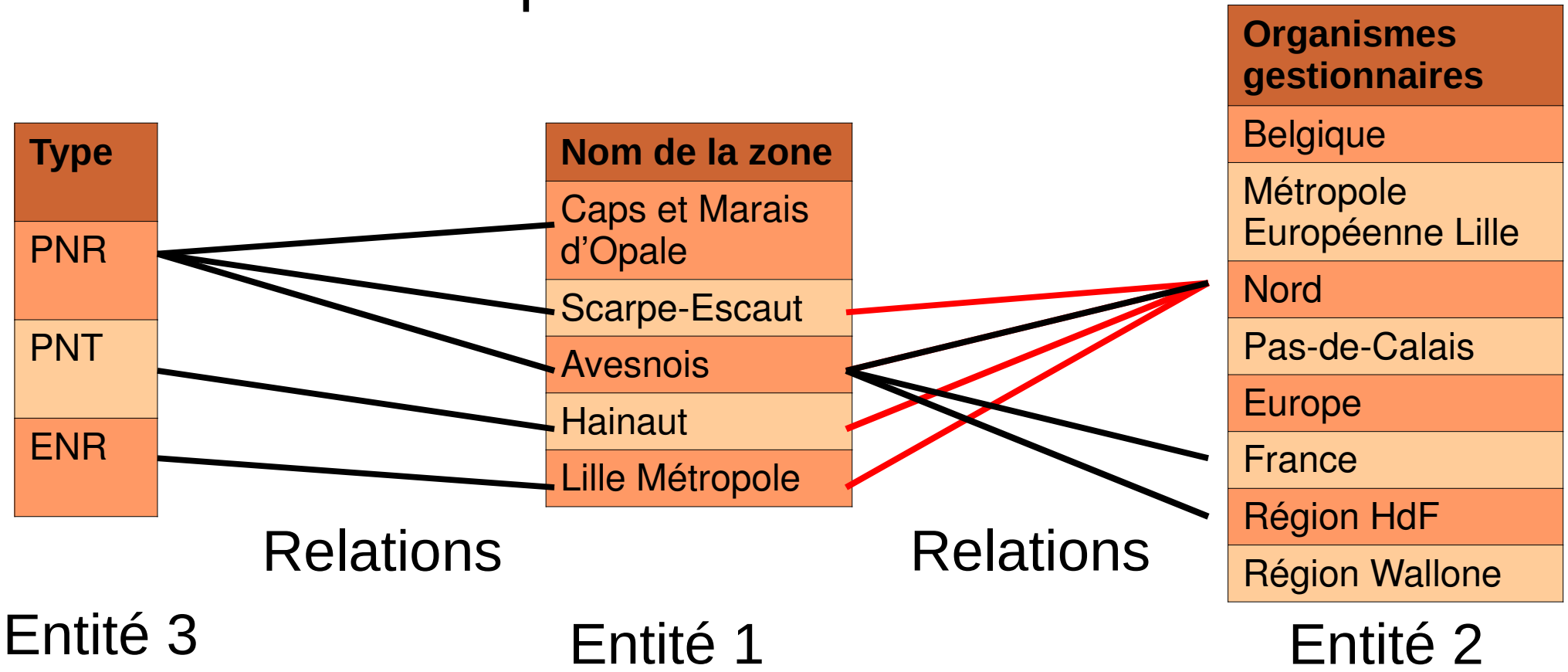
Relations

Entité 2

Organisme gestionnaire
Belgique
Métropole Européenne Lille
Nord
Pas-de-Calais
Europe
France
Région HdF
Région Wallone

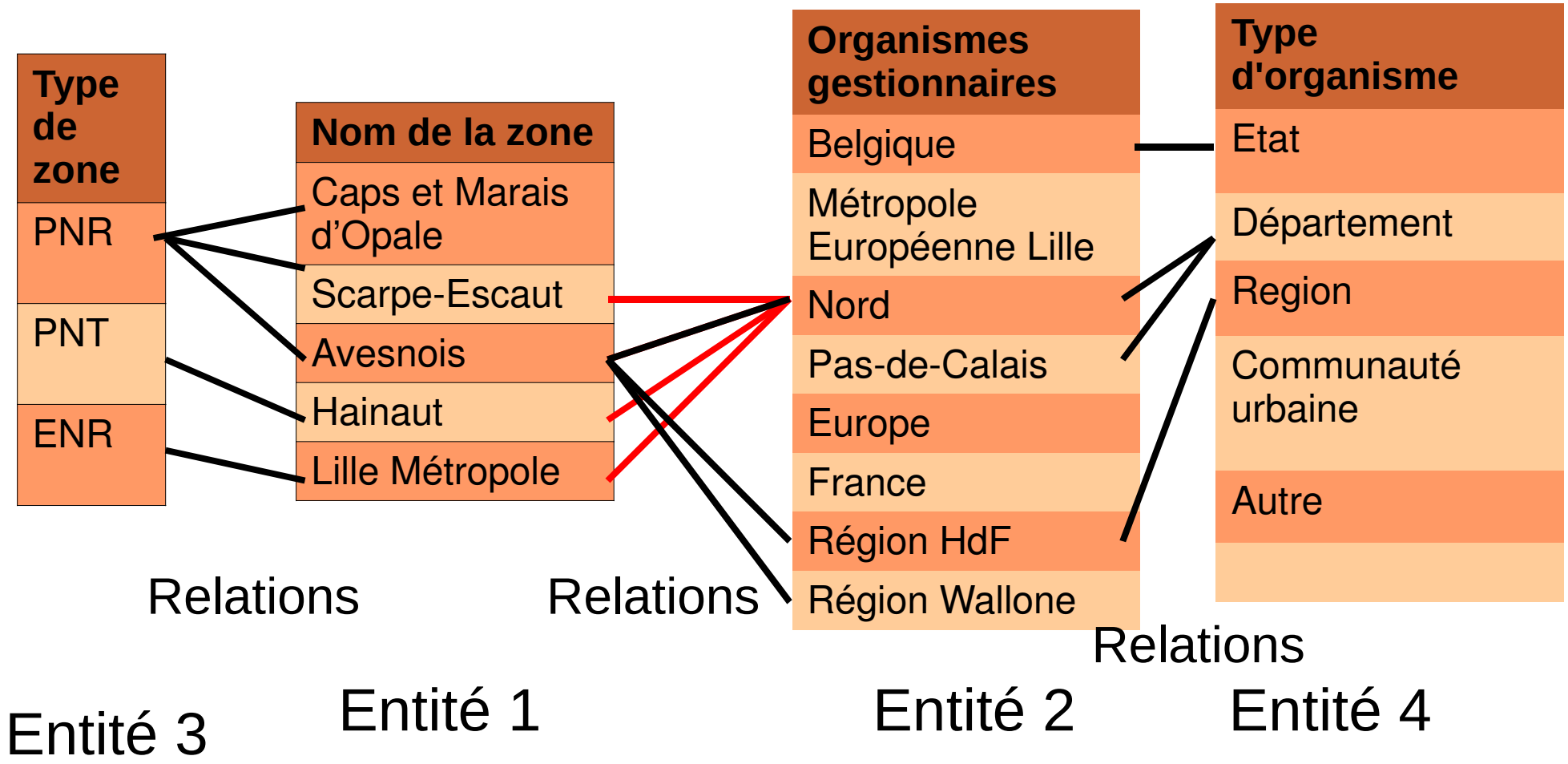
- Redondance d'informations ?
- Liste des zones ?
- Liste des organismes ?
- Liste des zones gérées par le Nord?
- Liste des zones de type PNR gérées par le Nord?

Décomposition en entités + fines



- Liste des zones de type PNR gérées par le Nord?
- Liste des zones gérées par un département ?
- Type d'organisme gérant des zones PNR ?

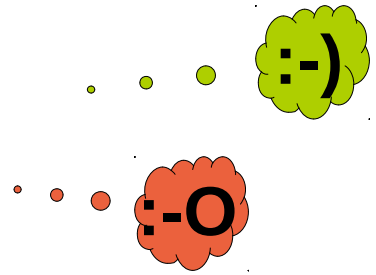
Décomposition en entités + fines

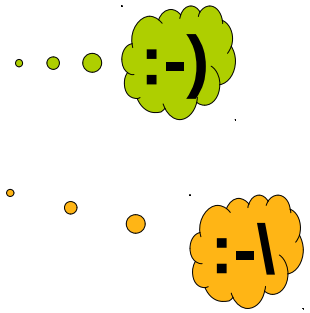


- Liste des zones gérées par un département ? ✔
- Type d'organisme gérant des zones PNR ? ✔

Où s'arrêter ?

- + de détails : Modèle complexe
  - - de redondance, + de souplesse et de puissance
  - Requêtes + compliquées, interrogation + longue (+ d'opérations)
- - de détails : Modèle simple
  - Interrogation + rapide
  - Questions limitées





- Faire une DB = choisir le bon compromis pour répondre à une liste de questions (=> les fonctionnalités souhaitées)
- Une DB ce n'est pas juste des données
- La liste des fonctionnalités fait partie du cahier des charges

La démarche générale

- Faire une DB =
  - Organiser les données en vue d'un traitement
  - Il est NÉCESSAIRE de s'interroger sur ces traitements
- Ces traitements conditionnent
  - La structuration des données (modélisation)
  - La « normalisation » des données
    - Codification et description (méta-données)
  - Le contrôle de qualité
  - L'inter-opérabilité (connexion avec d'autres outils)
  - « L'interface utilisateur »
    - Qui peut faire quoi (saisir/modifier/exporter, quelles parties des données)

Codification

- Données sont extraites pour être traitées
- Souvent par des méthodes statistiques

- Traitement automatique => codification
  - Pays = code international, Région ?
  - Localisation = données GPS
  - Personne = Initiales, autre code
  - Espèce = n° taxon, nom binomial (genre espèce), autre code
  - Publication = PMID

- Codification des données manquantes

Méta-données

- Exemples
  - Date d'acquisition de la donnée
  - Auteur de la donnée
  - Unité de mesure / donnée numérique (mètres ou miles ?)
  - Méthode ou protocole utilisée pour l'acquisition de la donnée
  - Niveau de confiance ou de qualité, statut de l'information
    - certain/incertain, direct/indirect, 1 mesure/moyenne,
    - local/importé d'une base de donnée publique
    - Ex : observation d'un animal : Incertain sur l'espèce ou sur le stade de développement, quantité exacte ou estimée, estimation indirecte (trace, cris).
- Utilité
  - Gérer des données de provenance et de qualité hétérogènes
    - Plusieurs équipes, plusieurs protocoles
  - Filtrer ou convertir les données avant traitement

Contrôle de qualité des données 1/4

- Gestion des données manquantes
  - Évaluer leur fréquence, leur provenance
    - Ex : concernent un lieu, un type de lieu, une période de temps ...
  - Filtrer les données manquantes
    - Ex : calculer le nombre de données «complètes»
  - Calculer des estimations pour les données manquantes
    - Ex si localisation GPS manquante, prendre la localisation de la commune et mettre « estimateur=commune » comme niveau de confiance

Contrôle de qualité des données 2/4

- Contrôle des valeurs
  - Données qualitatives
    - Listes de valeurs possibles (cf codification)
  - Données quantitatives
    - Plages de valeurs
  - Données corrélées
    - Vérifications de certaines relations
      - Ex : heure début < heure fin

Contrôle de qualité des données 3/4 => DB : contraintes d'intégrité

- **<u>Contrainte d'intégrité</u>** = expression logique qui doit être vérifiée sur la base (ie qui doit toujours être vrai)

- Ex1 : toute observation doit avoir une date (ie pas de date manquante)
  - *Contrainte de non nullité*
- Ex2 : chaque observateur a un identifiant unique
  - *Contrainte d'unicité ou de clé primaire*
- Ex3 : Il ne peut y avoir 2 cours dans la même salle à la même heure
  - *Contrainte d'unicité : clé = numéro de salle + heure*
- Ex4 : un numéro de salle est un entier
  - *Contraintes de domaine*
- Ex5 : un numéro de salle est compris entre 0 et 23
  - *Contraintes de vérification*

Contrôle de qualité des données 4/4 => DB : contraintes d'intégrité

- **<u>Contrainte d'intégrité</u>** = expression logique qui doit être vérifiée sur la base (ie qui doit toujours être vrai)

- Ex6 : une zone est de type PNR, PNT ou ENT
  - *Contrainte d'intégrité de référence* (table contenant les valeurs)
- Ex7 : heure de fin > heure de début
  - *Contrainte de table*, portant sur plusieurs attributs de la table

- Spécifier les contraintes
  - Permet la vérification automatique par le SGBD
  - Fait partie du cahier des charges et de la modélisation

Exemple de contraintes d'intégrité

	A	B	C	D	E	F	G	H
1	ID	date_naissance	sexe	age_exam	taille	poids	IMC	PP_grossesse
2	20715	09/04/1931	Feminin	67	157	56	22,72	Non
3	20633	28/09/1954	Masculin	44	163	86	32,37	Oui
4	20713	24/01/1955	Masculin	43	176	140	45,20	
5	20673	19/05/1956	Feminin	42	165	133	48,85	Oui
6	20630	01/08/1948	Feminin	50	257	78	11,81	Oui
7	20618	13/06/1948	Masculin	50	124	278	180,80	
8	20591	22/12/1953	Feminin	44	164	102	37,92	Oui
9	20590	21/04/1953	Masculin	45	177	98	31,28	
10	20752	10/08/1922	Masculin	76	168	84	29,76	
11	20753	01/09/1923	Feminin	74	162	66	25,15	Oui
12	20719	04/10/1925	Masculin	73	168	80	28,34	
13	20720	08/02/1927	Feminin	71	158	72	28,84	Oui
14	20629	12/07/1980	Feminin	18	169	91	31,86	Non
15	20085	31/01/1958	Masculin	39	178	141	44,5	
16	20371	18/09/1959	Masculin	39	175	80	26,12	

Interface utilisateur

- Qui peut faire quoi ?
  - Ajouter / modifier / supprimer / extraire
  - Sur quelles parties des données
  - Y-a-t-il différents rôles, plusieurs types d'utilisateurs
- Faciliter / Orienter / Limiter la Saisie
  - Masques de saisie différents selon les types d'utilisateurs
  - Limite les champs et les tables accessibles
  - Fixe l'ordre de saisie (différent de l'ordre des attributs)
  - Aide grâce à des menus déroulants
  - Ajout d'Aide / Documentation
- Faciliter l'extraction de données
  - Différents filtrages des données
  - Différents formats de sortie

Exemple Interface utilisateur (1/3)

Tableau de Bord | Patients | Tubes | Volet Medical

Logged in as sgaget (Logout) Fran◆ais | English

You are here: Patients :: Liste

My Profile Change Password

détails | liste | trouver

nouvelle entrée | tout voir | copier le résultat | mettre à jour le résultat | importer des données

Found 268 records 1 2 3 4 5 6 Next Showing 30 Results per page

	Patient	Initiales du pr◆nom	Initiales du nom	Elucid◆	Sexe	Num◆ro de famille	Date de naissance	Lien familial	Ethnie	Pays	Provenance	Medecin	Diagnostic	D◆tails du diagnostic
	25168	MAT	ALL	non	feminin	DN-161	2007-09-17	cas index	antilles	antilles			TNDM	a vérifier, indiqué ND à naissance
	25170	GUY	ART	non	masculin	DN-161	1972-11-24	parent	AD	NC			NON DIAB	
	25169	TAN	ALL	non	feminin	DN-161	1979-05-28	parent	AD	NC			NON DIAB	
	33377	ALE	FEJ	non	feminin	DN-199	2008-12-06	cas index	asie	turquie			PNDM	hypoplasie pancreas
	19262	DAN	BIT	non	masculin	DN-112	2004-11-20	cas index	maghreb	israel			PNDM	convulsions.
	31621	PEN	ZEI	non	feminin	DN-189	1992-12-25	cas index	AD	AD			EOD	
	22277	KAS	ASS	non	masculin	DN-147	1988-03-07	cas index	AD	AD		suivi à la Réunion	PNDM	sous insuline
	21681	PAU	PAR	non	feminin	DN-142	1997-04-26	cas index	AD	AD			EOD	DT1 à confirmer (info du 11-06-2009)
	21210	SHL	HAR	non	masculin	DN-130	1987-11-26	cas index	maghreb	israel			PNDM	pas de diabete dans la famille
	20925	INE	SOB	non	feminin	DN-127	2006-07-11	cas index	AD	AD			EOD	
	20881	MAR	GAZ	non	feminin	DN-126	1973-07-18	cas index	AD	AD			EOD	pas de diabète dans la famille
								cas						hernie diaphragmatique

Exemple Interface utilisateur (2/3)

Patient ■

Entrer le num◆ro du patient

Initiales du pr◆nom ■

Entrer les 2 initiales

Initiales du nom ■

Entrer les 2 initiales

Elucid◆ non

Sexe ■ SVP sélectionnez ...

- SVP sélectionnez ...
- masculin
- feminin

Num◆ro de famille

Date de naissance

Lien familial SVP sélectionnez ...

Par rapport au Cas Index

Ethnie SVP sélectionnez ...

Exemple Interface utilisateur (3/3)

Patient Information

Sex		◉ Male ○ Female
Year of birth		1930
Country of birth		Malta
Family participation to the study	1st member of the family participating to the study	○ No ◉ Yes ○ Unknown
	If No :	First member's sample code
		Family relationship with the first member
Other family members with diabetes	Father	○ No ○ Yes ◉ Unknown
	Mother	○ No ○ Yes ◉ Unknown
	Brother	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ◉ Unknown
	Sister	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ◉ Unknown
	Twin	○ No ○ Monozygotic ○ Dizygotic ◉ Unknown
	Other (uncle, aunt, cousin, ...)	
Consanguineous family		○ No ○ Yes ◉ Unknown

Comment faire une base de données

- Modélisation à 3 niveaux
  - Modèle conceptuel
    - Identifie les entités et les relations
  - Modèle relationnel
    - Identifie les éléments d'une base de données relationnelle
    - Tables, attributs, clés, contraintes etc...
  - Niveau externe (ie interface utilisateur)
    - Masques de saisie
    - Requêtes d'interrogation et d'extraction
- Implémentation
  - Utilisation d'un outil logiciel de SGBD + langage SQL
  - Création des tables, attributs, clés, contraintes
  - Création des interfaces utilisateurs

2ème Partie

Modélisation

Modèle conceptuel => modèle relationnel

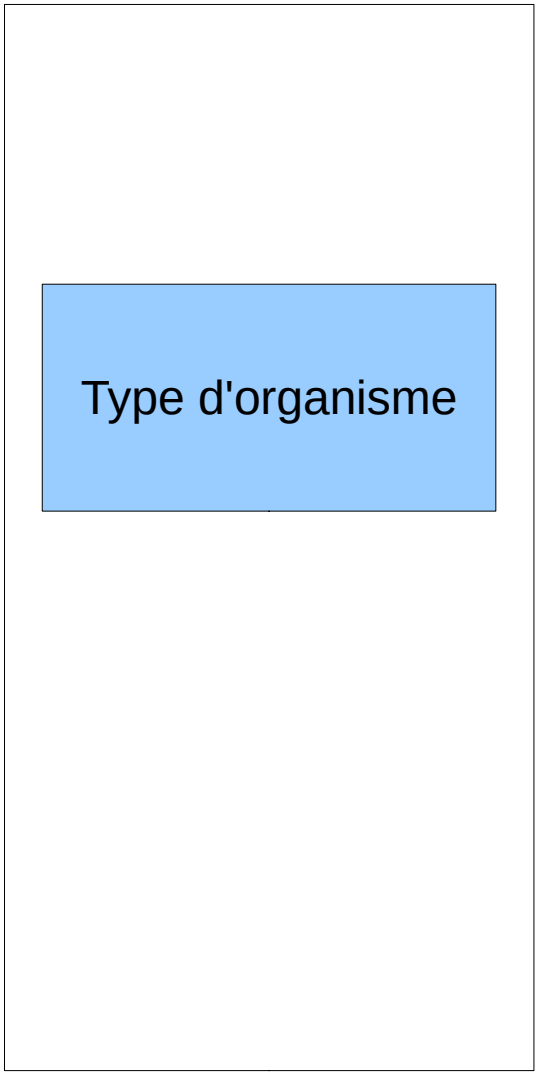
Niveaux de Modélisation

1 colonne => 1 entité

1 entité => 1 table

Col X	**Type d'organisme**	Col Y
...	Etat	...
...	Département	...
	Region	
...	Communauté urbaine	...
...	Autre	...

Exemple, tableau Excel



Modèle conceptuel

Table *Attributs*

type_org(id_type, nom)

Contrainte : Clé primaire unique

id_type	**nom**
1	Etat
2	Département
3	Region
4	Communauté urbaine
5	Autre

Modèle relationnel

Exemple => Modèle conceptuel : relation 1-N

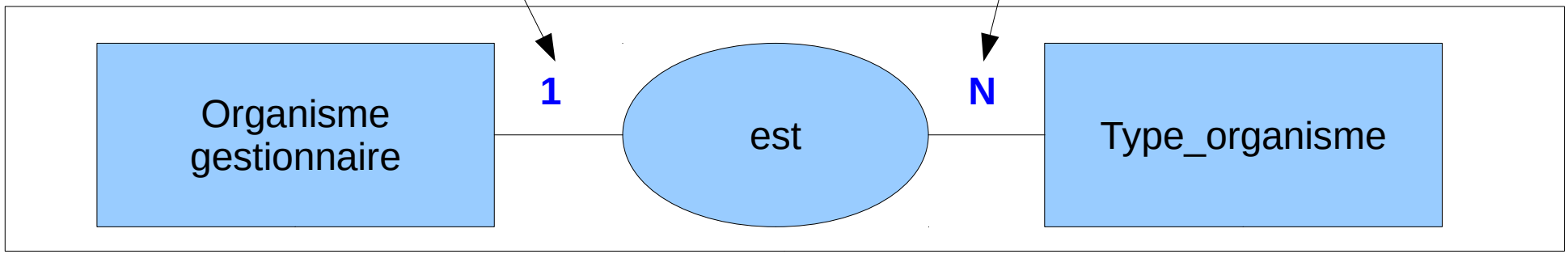
Organismes gestionnaires
Belgique
Métropole Européenne Lille
Nord
Pas-de-Calais
Europe
France
Région HdF
Région Wallone

Type d'organisme
Etat
Département
Region
Communauté urbaine
Autre

Entité 1

Relations

Entité 2



Modèle conceptuel => modèle relationnel

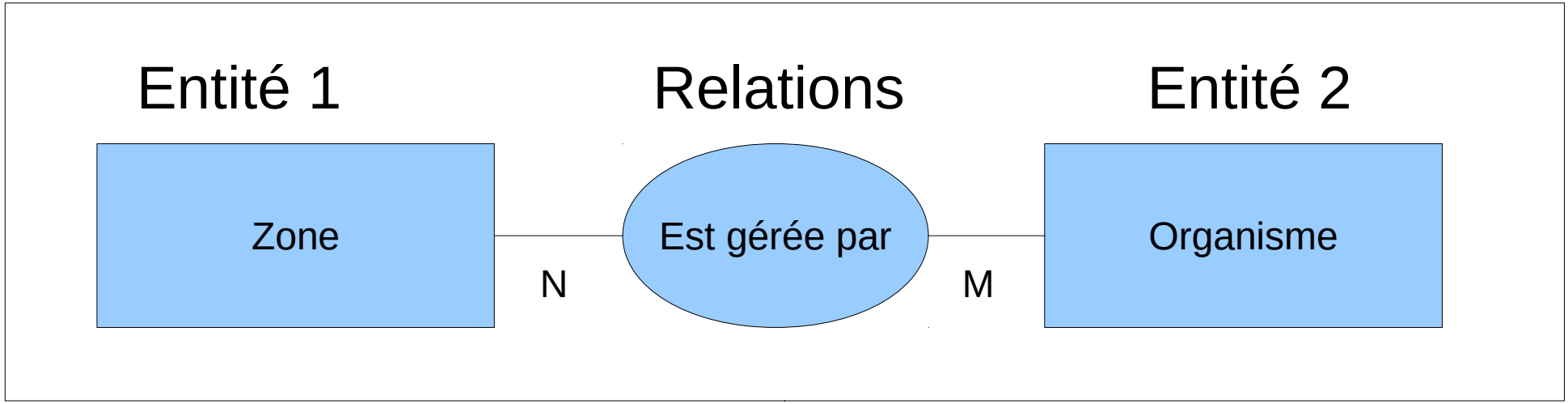


table *attributs*

zone(id_zone, nom)

org(id_org, nom)

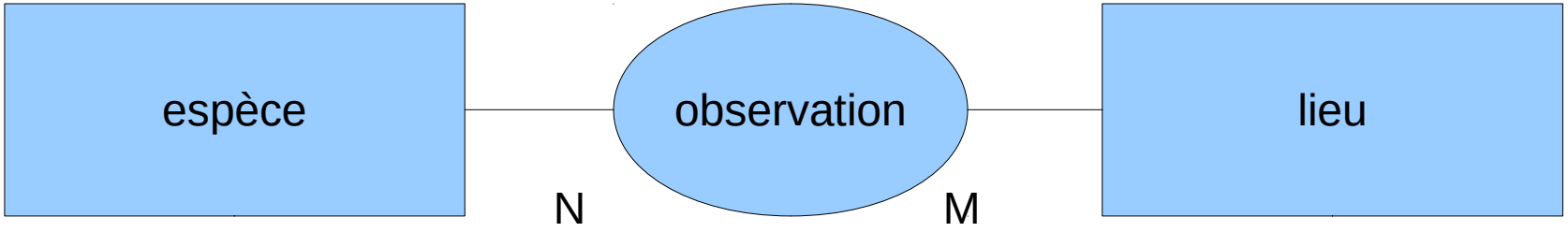
gerePar(id_zone, id_org)

Clé primaire

Table d'entités

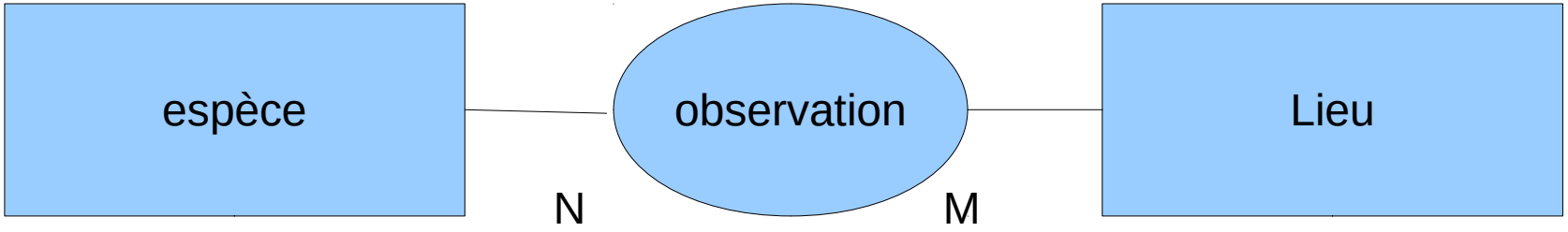
Table de relations

2ème exemple – relation N - M



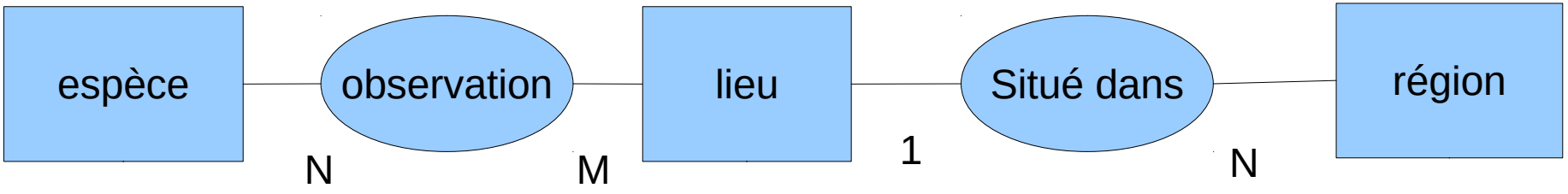
- espece(id_espece, nom)
- lieu(id_lieu, nom)
- observation(id_espece, id_lieu)

Relation N – M avec champs



- espece(id_espece, nom)
- lieu(id_lieu, nom)
- observation(id_espece, id_lieu, date, effectif)

Relation 1 – N

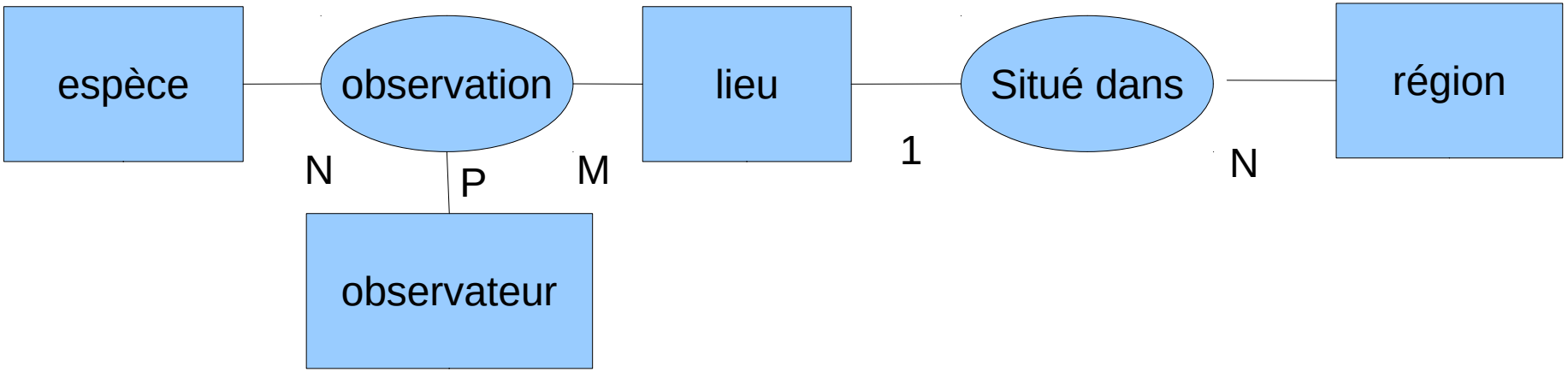


- region(<u>id_region</u>, nom)

Clé étrangère = clé primaire d'une autre table

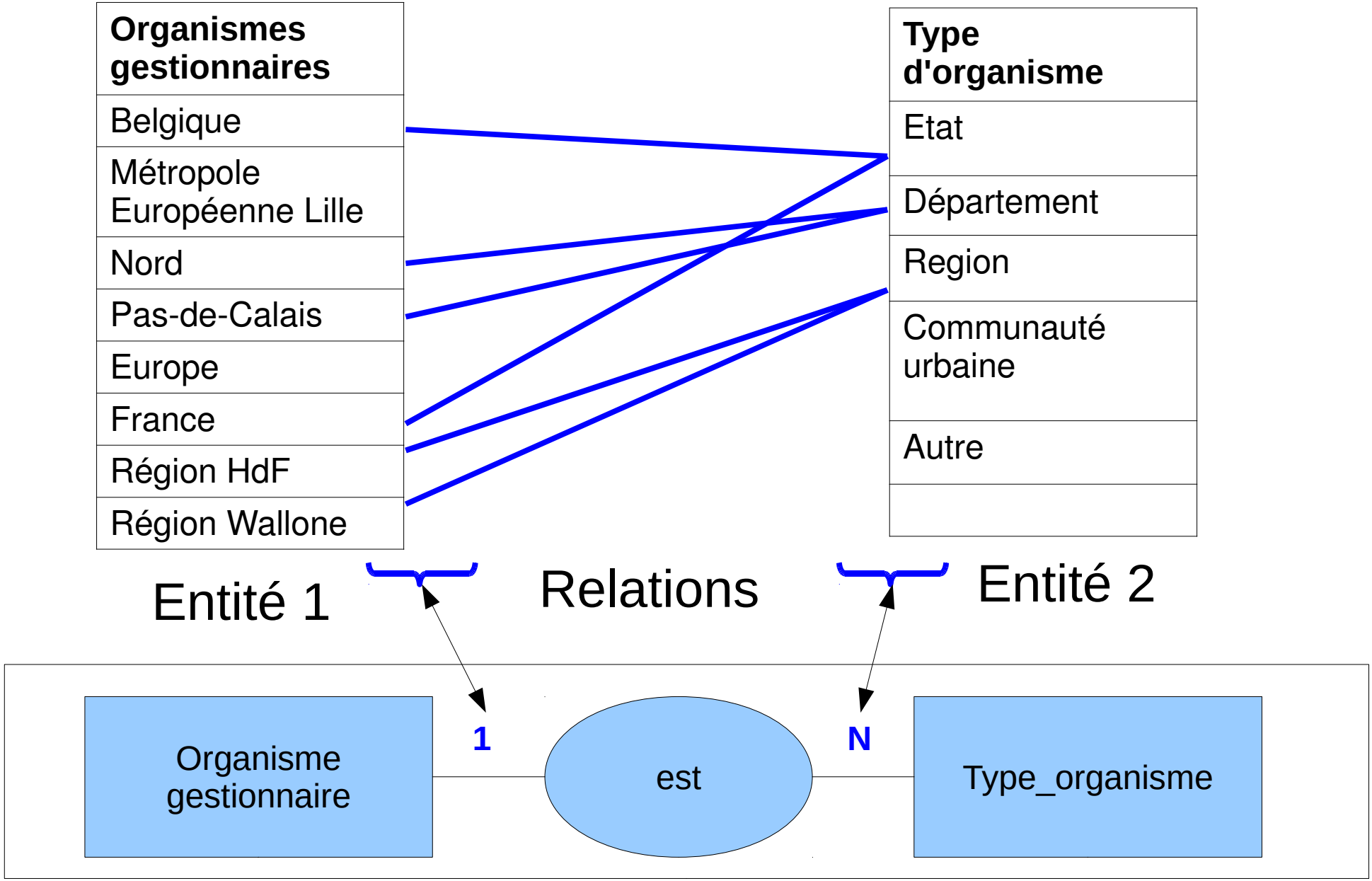
- lieu(<u>id_lieu</u>, nom, id_region)
- especе(<u>id_especе</u>, nom)
- observation(<u>id_especе, id_lieu, date</u>, effectif)

Combinaison de relations

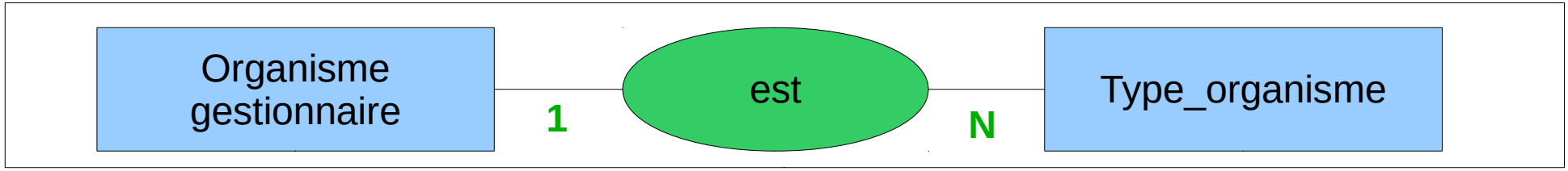


- region(<u>id_region</u>, nom)
- lieu(<u>id_lieu</u>, nom, id_region)
- especе(<u>id_especе</u>, nom)
- observateur(<u>id_obs</u>, nom, prénom)
- observation(<u>id_especе, id_lieu, id_obs, date</u>, effectif)

Exemple => Modèle conceptuel : relation 1-N



Modèle conceptuel => modèle relationnel
relation 1-N => clé étrangère



SGBD lien FK / PK

id_org	nom	id_type
1	Belgique	1
2	Métropole Euro Lille	4
3	Nord	2
4	Pas-de-Calais	2
5	Europe	5
6	France	1
7	Région HdF	3
8	Région Wallone	3

id_type	nom
1	Etat
2	Département
3	Region
4	Communauté urbaine
5	Autre

Clé primaire PK

Clé étrangère FK

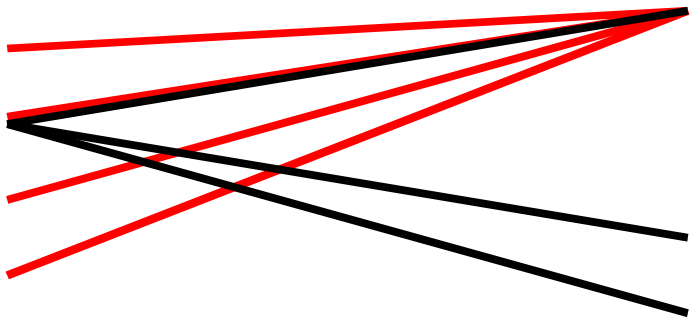
Clé primaire

org(id_org, nom, id_type) ← *Tables* → **type_org(id_type, nom)**

Exemple => Modèle conceptuel : relation N-M

Nom de la zone
PNR des Caps et Marais d'Opale
PNR Scarpe-Escaut
PNR Avesnois
PNT du Hainaut
ENR Lille Métropole

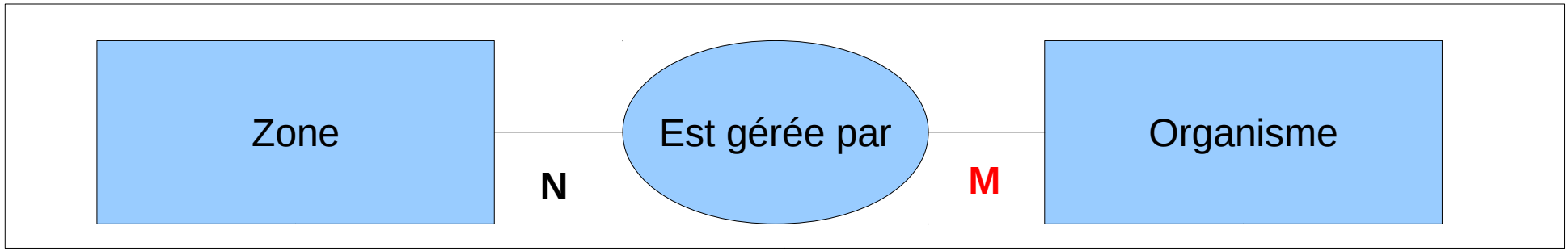
Entité 1



Relations

Organismes gestionnaires
Belgique
Métropole Européenne Lille
Nord
Pas-de-Calais
Europe
France
Région HdF
Région Wallone

Entité 2



Modèle conceptuel => modèle relationnel
relation N-M => table

id	nom
1	PNR des Caps et Marais d'Opale
2	PNR Scarpe-Escaut
3	PNR Avesnois
4	PNT du Hainaut
5	ENR Lille Métropole

id zone	id org
1	3
4	3
5	3
3	3
3	6
3	7

id	nom
1	Belgique
2	Métropole Européenne Lille
3	Dpt59
4	Dpt62
5	Europe
6	France
7	Région HdF
8	Région Wallone

Clé primaire de la table de relation
= couple de clés étrangère

zone(id_zone, nom) **gere_par(id_zone, id_org)** **org(id_org, nom)**

Tables

Importance de la validation des modèles

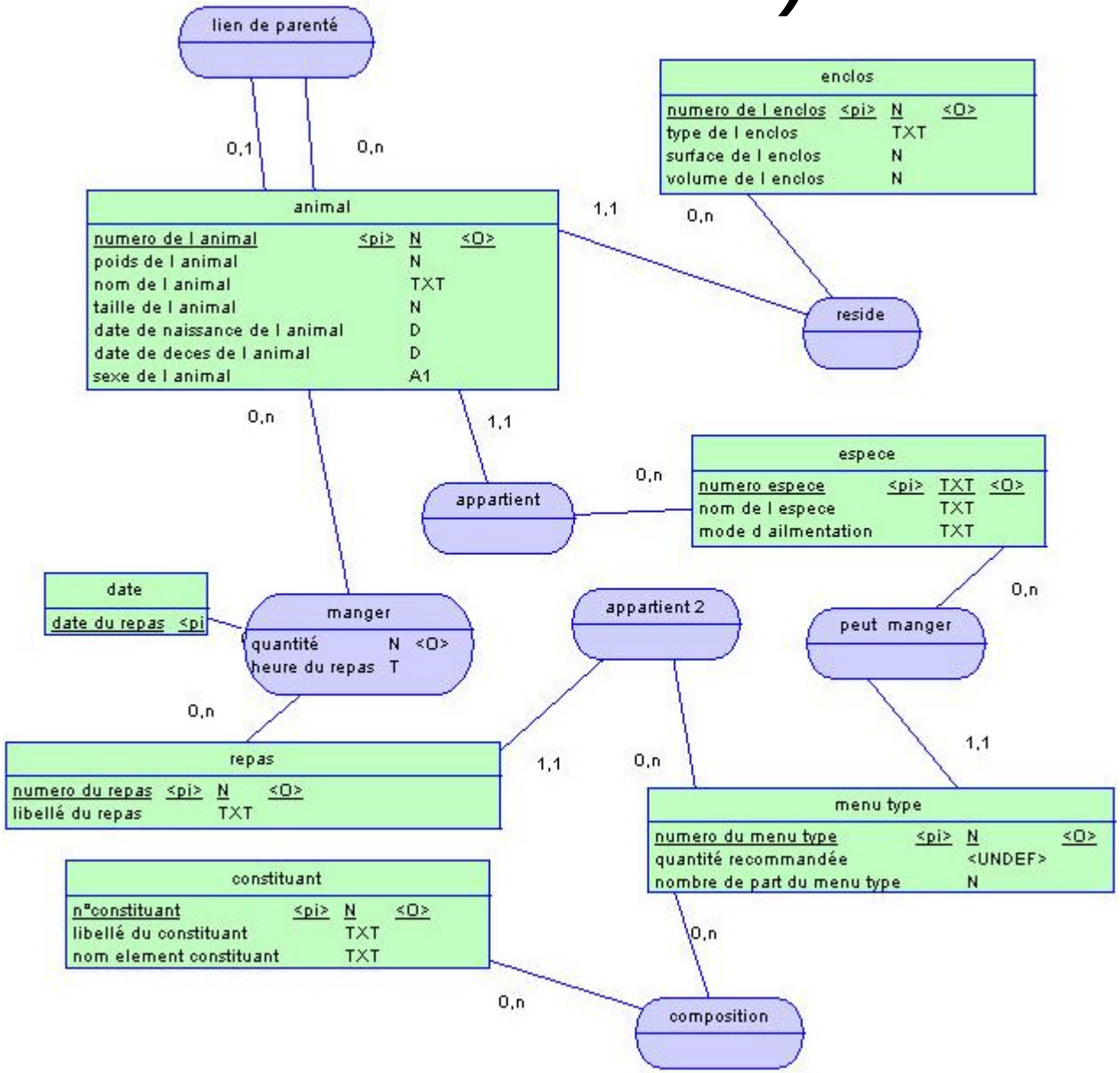
- Modélisation à 3 niveaux
  - Modèle conceptuel
    - Identifie les entités et les relations
  - Modèle relationnel
    - Identifie les éléments d'une base de données relationnelle
    - Tables, attributs, clés, contraintes etc...
  - Niveau externe (ie interface utilisateur)
    - Masques de saisie
    - Requêtes d'interrogation et d'extraction
- Implémentation
  - Utilisation d'un outil logiciel de SGBD + langage SQL
  - Création des tables, attributs, clés, contraintes
  - Création des interfaces utilisateurs

+ Validation à chaque étape

Comment valider la modélisation ?

- Modèle conceptuel
  - Est-ce qu'il répond aux attentes
    - Toutes les données sont intégrées ?
    - Toutes les fonctionnalités peuvent être réalisées ?
  - Peut-on faire plus simple sans perdre des fonctionnalités ?
  - Le schéma graphique est-il facilement utilisable ?
  - Inter opérabilité (ie facilité de connexion avec d'autres applications) ?
  - Reste-t-il des redondances ?
- Modèle relationnel
  - Toutes les contraintes sont-elles spécifiées ?
  - Quelles prévisions de monté en charge ?

Exemple MCD (Modèle Conceptuel des Données)



Résumé : Démarche de modélisation

- Déterminer la liste des entités, pour chacune
  - Établir la liste des attributs
  - Déterminer un identifiant
- Déterminer la liste des relations, pour chacune
  - Donner la liste des attributs
  - Vérifier la dimension (binaire, ternaire)
  - Définir la cardinalité (1-N, N-M)
- Vérifier le schéma
  - Supprimer les redondances
  - Vérifier qu'il supporte les fonctionnalités

APPLICATIONS

1 – Étude des espaces naturels

Espaces naturels

Sur une carte régionale, on a représenté des zones correspondant aux différents Parcs Naturels Régionaux (PNR), Parcs Naturels Transfrontaliers (PNT) et Espaces Naturels Régionaux (ENR) des Hauts-De-France. Chacune de ces zones est repérée par un numéro à 2 chiffres. On souhaite associer des informations complémentaires à ces zones :

- nom de la zone (PNR des Caps et Marais d'Opale, ENR Lille Métropole…)

- type (PNR, PNT, EN)

- liste de villes se trouvant sur le territoire de la zone

- liste des organismes prenant part à la gestion du parc

Voici les zones se trouvant sur votre carte (les listes de villes et d'organismes sont volontairement non exhaustives) :

N°	Nom	Type	Villes	Organismes
01	PNR des Caps et Marais d'Opale	PNR	Marquise, Samer, Guines, Licques	France, Région HdF, Département du Pas-de-Calais
02	PNR Scarpe-Escaut	PNR	Saint-Amand-les-Eaux, Raismes, Marchiennes	France, Région HdF, Département du Nord
03	PNR Avesnois	PNR	Maroilles, Avesnes/Helpe, Le Quesnoy	France, Région HdF, Département du Nord
04	PNT du Hainaut	PNT	Saint-Amand-les-Eaux, Raismes, Marchiennes	Europe, France, Région HdF, Département du Nord, Belgique, Région Wallone
05	ENRLille Métropole	ENR	Croix, Leers, Marcq-en-Barœul, Roubaix, Tourcoing, Wasquehal, Wattrelos	Etat, Région HdF, Département du Nord, Métropole Européenne de Lille

Construisez le modèle entité-relation

Vous organiserez au mieux vos données pour éviter les redondances d'informations et faciliter les recoupements d'informations...

Indiquez pour chacune des entités et des relations la liste des attributs

APPLICATIONS

2 – Étude de lignes ferroviaires

Lignes ferroviaires

On veut constituer une base de données avec les lignes ferroviaires de la région. Chaque ligne est repérée par un numéro.

On souhaite associer des informations complémentaires à ces lignes :

- nom de la ligne (exemples : "Lille-Dunkerque", "Lille-Calais-Londres"...)

- type de la ligne (exemples : "TER", "TGV"...)

- liste des gares desservies par la ligne, et pour chaque gare, la ville dans laquelle elle se trouve

Voici la liste des lignes (les listes de gares sont volontairement non exhaustives) :

N°	Nom	Type	Gares (Ville)
08	Lille-Dunkerque	TER	Lille Flandres (Lille), Armentières (Armentières), Hazebrouck (Hazebrouck), Dunkerque (Dunkerque)
101	Lille-Paris	TGV	Lille Flandres (Lille), Paris Nord (Paris)
02	Lille-Douai-Arras-Amiens-Rouen	TER	Lille Flandres (Lille), Douai (Douai), Arras (Arras), Amiens (Amiens), Rouen Rive Droite (Rouen)
201	Lille-Calais-Londres	Eurostar	Lille Europe (Lille), Calais Frethun (Frethun), Londres Waterloo (Londres)
12	Lille-Hazebouck-Calais-Boulogne	TER	Lille Flandres (Lille), Armentières (Armentières), Hazebrouck (Hazebrouck), Calais Ville (Calais), Calais Frethun (Frethun), Boulogne Tintelleries (Boulogne-sur-Mer), Boulogne Ville (Boulogne-sur-Mer)

Construisez le modèle entité-relation

Vous organiserez au mieux vos données pour éviter les redondances d'informations et faciliter les recoupements d'informations...

Indiquez pour chacune des entités et des relations la liste des attributs

APPLICATIONS

3 – Immeubles

Immeubles

On veut constituer une base de données d'immeubles en construction. Pour chacun de ces bâtiments, on connait :

- le numéro de permis de construire
- le nom du bâtiment
- la ville dans laquelle se réalise le projet
- le porteur du projet de construction
- les architectes
- les entreprises qui participent au chantier

Voici la liste des bâtiments :

N° de permis	Nom du bâtiment	Ville	Porteur du projet	Architectes	Entreprises
45XV5	Résidence Camus	Lille	M. Dupont	Cabinet Lorem	SGEN (électricité), SGO (gros-œuvre), SPN (plomberie)
23GC4	Résidence Vinci	Douai	Mme Durand	Cabinet Sic	DOUELEC (électricité), SGO (gros-œuvre), PLOMBONOR (plomberie)
01SD6	Résidence Europe	Lille	M. Paul	Cabinet Ipsum Cabinet Lorem	DOUELEC (électricité), GROUVRE (gros-œuvre), SPMPLM (plomberie)
87PX9	Résidence Soleil	Arras	M. Dupont	Cabinet Sic	SGEN (électricité), SGO (gros-œuvre), PLOMBONOR (plomberie)

Construisez le modèle entité-relation

Vous organiserez au mieux vos données pour éviter les redondances d'informations et faciliter les recoupements d'informations...

Indiquez pour chacune des entités et des relations la liste des attributs

APPLICATIONS

4 – Étude des lérots

Lérots

On veut constituer une base de données des observations relevées pour le lérot, dans la région des Hauts-De-France sur plusieurs années.

Les observateurs fournissent des informations sur l'age, l'effectif et les conditions dans lesquelles l'observation a eu lieu (type de contact). Attention, une personne peut faire plusieurs observations sur le même lieu le même jour.

Voici un jeu de données minimales pour le projet.

Observateur	Date	Commune	Dept	Âge	Effectif	Type de conta
P Olivier	01/03/2009	LILLE	59		1	mort
G José	01/09/2000	SAINT-AYBERT	59		2	cris
G José	01/10/2009	SAINT-AYBERT	59		1	traces de dent
G José	02/09/2000	SAINT-AYBERT	59		2	cris
H David	08/09/2002	DOUAI	59	Juvénile	1	obs. visuelle
G José	10/11/2009	SAINT-AYBERT	59	Adulte	1	obs. visuelle
S Hubert	15/08/2006	TAISNIERES-EN-THIERACHE	59	Adulte	1	obs. visuelle
S Hubert	15/08/2006	TAISNIERES-EN-THIERACHE	59	Juvénile	4	obs. visuelle
H David	21/08/2002	DOUAI	59	Juvénile	1	obs. visuelle
G Antoine	27/04/2003	WILLERVAL	62	Inconnu	1	mort

Construisez le modèle entité-relation

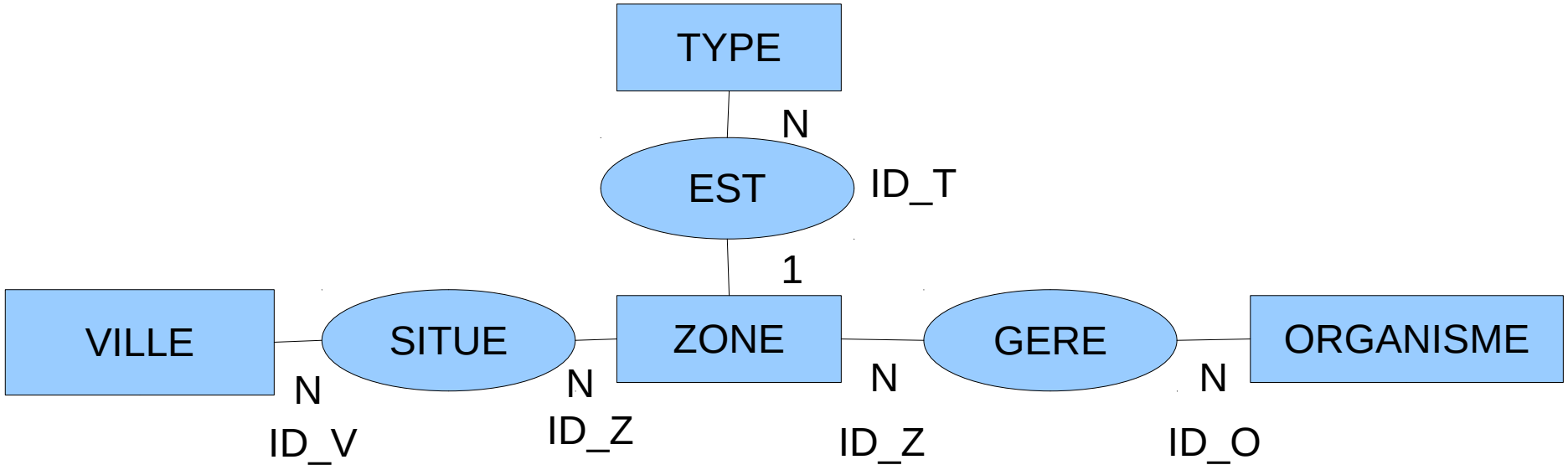
Vous organiserez au mieux vos données pour éviter les redondances d’informations et faciliter les recoupements d’informations...

Indiquez pour chacune des entités et des relations la liste des attributs

ANNEXES

1 – Étude des espaces naturels

Espaces Naturels

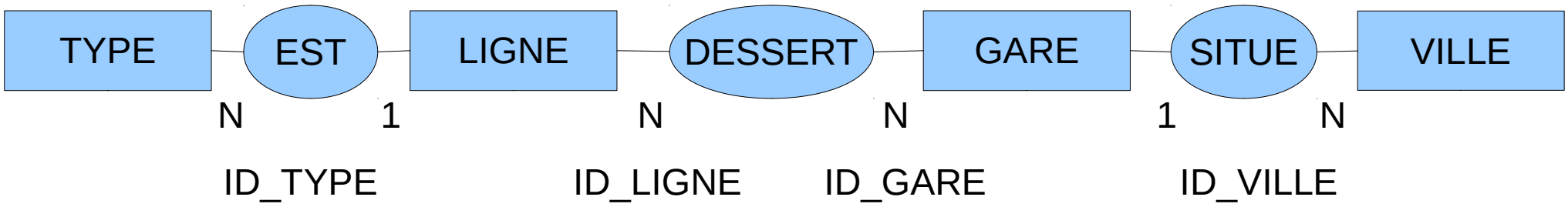


- VILLE(<u>ID_V</u>, NOM)
- ORG(<u>ID_O</u>, NOM)
- TYPE(<u>ID_T</u>, NOM)
- ZONE(<u>ID_Z</u>, NOM, ID_T)
- SITUE(<u>ID_Z, ID_V</u>)
- GERE(<u>ID_O, ID_Z</u>)

ANNEXES

2 – Étude de lignes ferroviaires

Lignes Ferroviaires

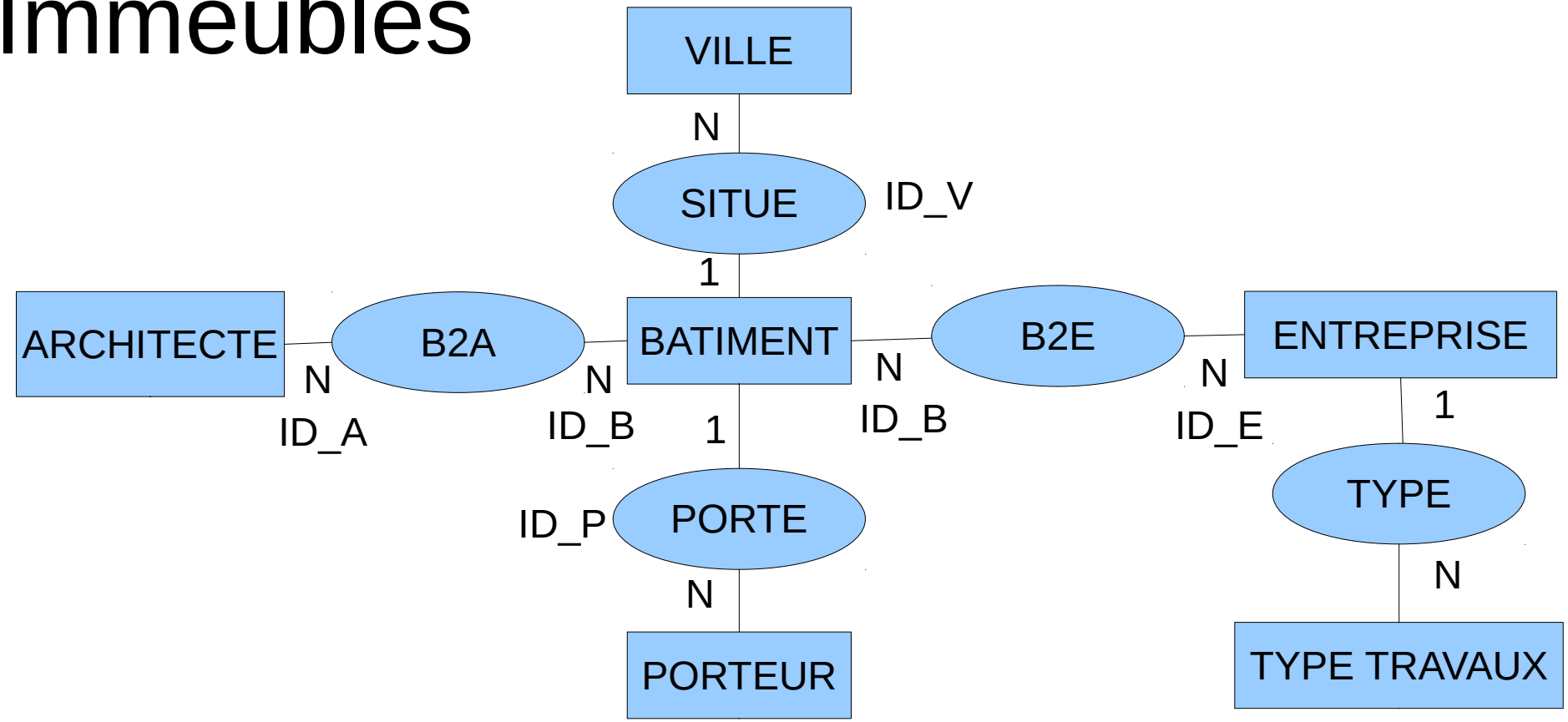


- TYPE(<u>ID_TYPE</u>, NOM)
- VILLE(<u>ID_VILLE</u>, NOM)
- LIGNE(<u>ID_LIGNE</u>, NOM, NUMERO, ID_TYPE)
- GARE(<u>ID_GARE</u>, NOM, ID_VILLE)
- DESSERT(<u>ID_LIGNE, ID_GARE</u>)

ANNEXES

3 – Immeubles

Immeubles

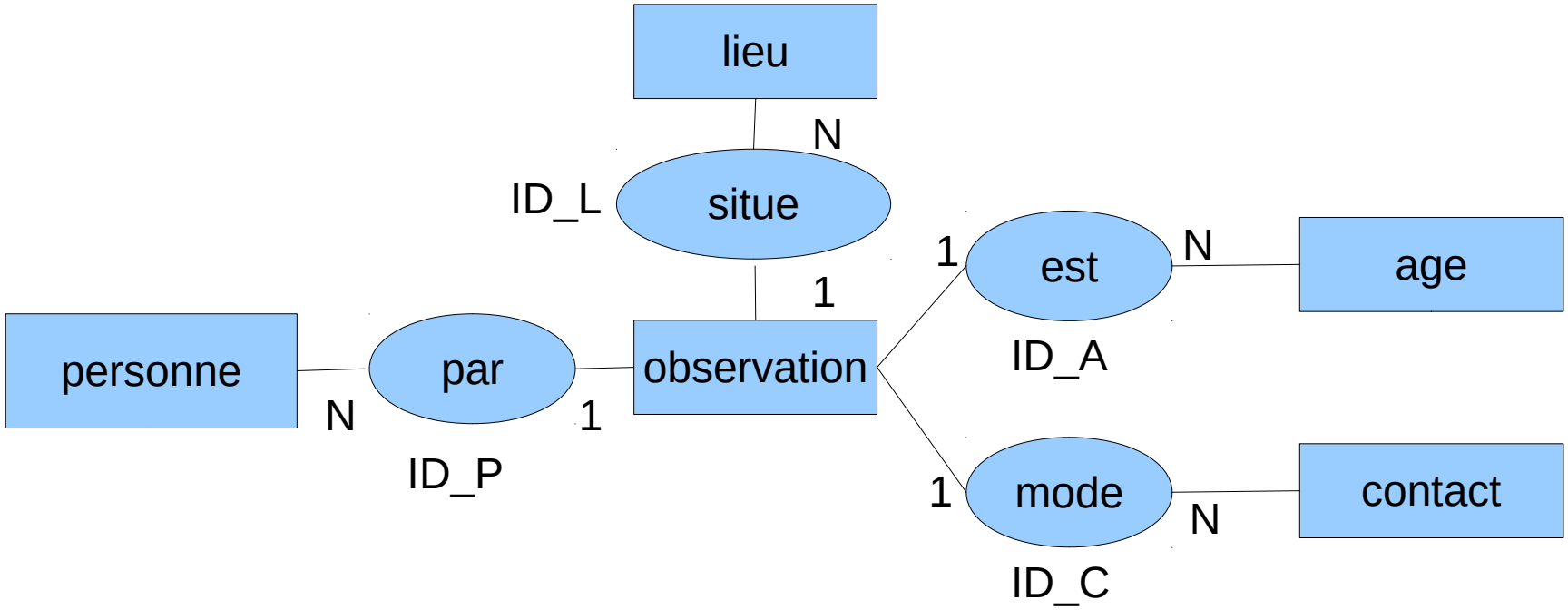


- VILLE(<u>ID_V</u>, NOM)
- ARCH(<u>ID_A</u>, NOM)
- PORTEUR(<u>ID_P</u>, NOM)
- TT(<u>ID_TT</u>, NOM)
- ENTR(<u>ID_E</u>, NOM, ID_TT)
- BAT(<u>ID_B</u>, NOM, ID_P, ID_V, PERMIS)
- B2A(<u>ID_B, ID_A</u>)
- B2E(<u>ID_B, ID_E</u>)

ANNEXES

4 – Étude des lérots

Lérots



- PERSONNE(ID_PERSONNE, NOM)
- LIEU(ID_LIEU, COMMUNE, DEP)
- AGE(ID_AGE, NOM)
- CONTACT(ID_CONTACT, NOM)
- OBSERVATION(ID_O, ID_P, ID_L, ID_C, ID_A, DATE, EFECTIF)

Premières manipulations

- Comment créer dans une base ***mabase***, une table ***matable***, y insérer des données et utilisable par ***tom*** ?

ID	Value
2	5
3	64

Premières manipulations

- Créer une base ***mabase*** et en faire de ***tom*** le propriétaire de la base (afin qu'il puisse l'utiliser comme bon il lui semble)

```
SQL Shell (psql)
Page de codes activeá: 1252
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Mot de passe pour l'utilisateur postgres :
psql (9.6.1)
Saisissez ½ help ╗ pour l'aide.

postgres=# CREATE DATABASE mabase;
CREATE DATABASE
postgres=# CREATE ROLE tom LOGIN password 'secret';
CREATE ROLE
postgres=# ALTER DATABASE mabase OWNER TO tom;
ALTER DATABASE
postgres=#
```

- Utilisation des commandes du langage SQL : CREATE DATABASE, CREATE ROLE et ALTER DATABASE

Premières manipulations

- Connexion de ***tom*** sur la base ***espacenat***
- Fermez psql, relancez-le et re-connectez-vous

- Attention à bien préciser le nom de la base et l'utilisateur. (si vous ne précisez pas le nom de la base, psql essaie de se connecter à la base de même nom que l'utilisateur. Si vous ne précisez pas le nom d'utilisateur, c'est le nom de l'utilisateur du système qui est utilisé.)

Premières manipulations

- Créer la table ***matable*** avec seulement un seul champ
- Puis ajout d'un second champ

```
mabase=>
mabase=> CREATE TABLE "matable" ("id" INTEGER NULL);
CREATE TABLE
mabase=> SELECT * FROM matable;
 id
----
(0 ligne)


mabase=> ALTER TABLE "matable" ADD "value" INTEGER NULL DEFAULT NULL;
ALTER TABLE
mabase=> SELECT * FROM matable;
 id | value
----+-------
(0 ligne)


mabase=>
```

- Utilisation des commandes du langage SQL : CREATE TABLE, SELECT et ALTER TABLE

Premières manipulations

- Ajout d'une ligne de valeur
- Puis de la seconde

```
mabase=> insert into matable(id, value) values (1, 5);
INSERT 0 1
mabase=> SELECT * FROM matable;
 id | value
----+-------
  1 |     5
(1 ligne)


mabase=> insert into matable(id, value) values (2, 65);
INSERT 0 1
mabase=> SELECT * FROM matable;
 id | value
----+-------
  1 |     5
  2 |    65
(2 lignes)


mabase=> _
```

- Utilisation des commandes du langage SQL : INSERT INTO et SELECT