

# Stage SIG : Introduction aux bases de données

Karim Lidouh  
<klidouh@gmail.com>

Certains des slides de cette présentation sont inspirés de l'exposé fait par Moritz Lennert



1. Structuration de données

2. Création de Base de données

3. Interrogation de Base de données

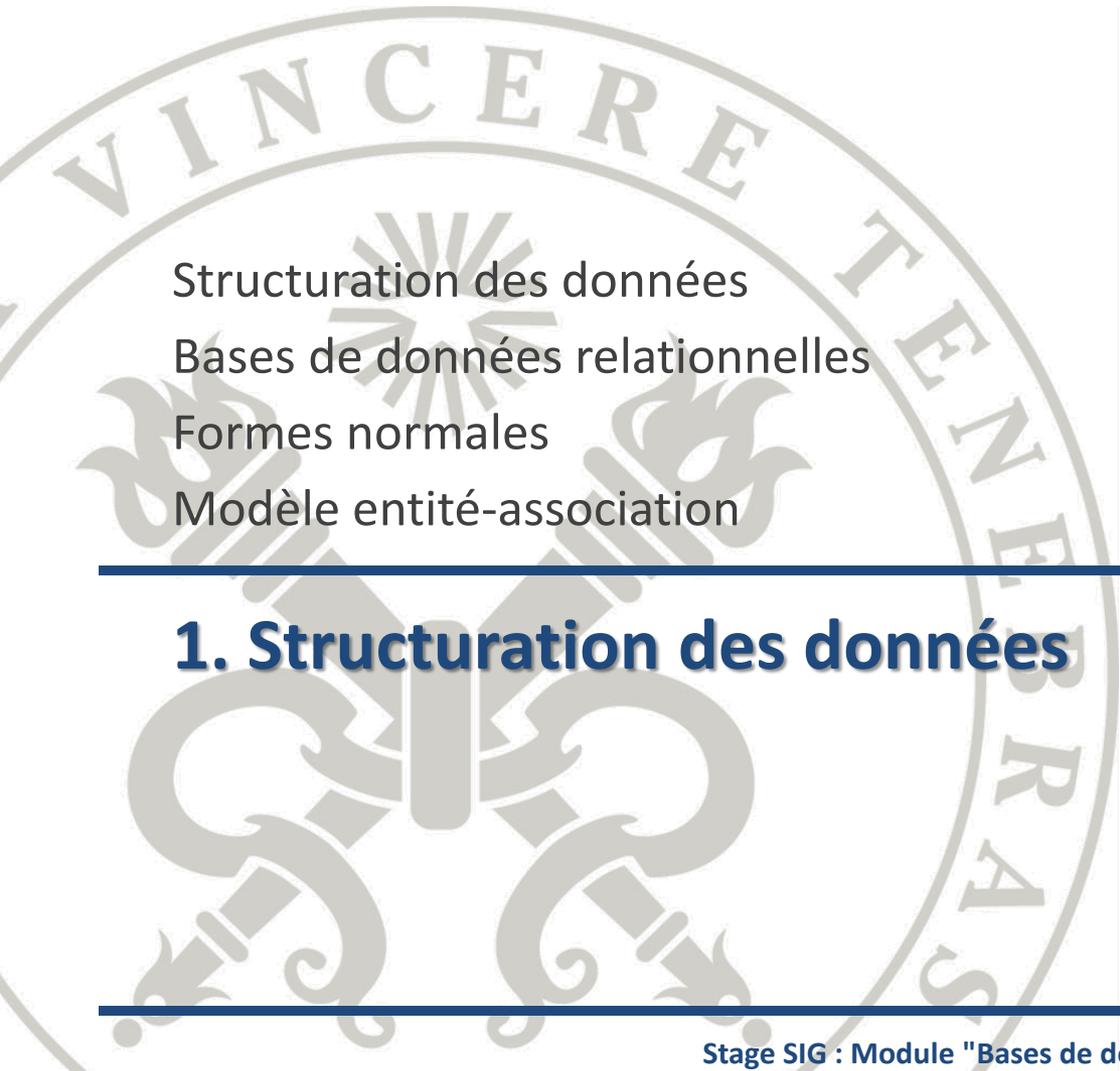
4. Maintenance et accès

5. Interfaces

## Objectifs

---

1. Structuration des données
2. Installation de PostgreSQL et insertion des données structurées
3. Interrogation de la base de données en utilisant le langage SQL
4. Maintenance et accès à la base de données
5. Programmation et utilisation d'interfaces

The background features a large, faint watermark of the ULB logo, which is a circular seal containing the Latin motto 'VINCIERE TENEBRAS' and a central emblem with a sunburst and two crossed keys.

Structuration des données  
Bases de données relationnelles  
Formes normales  
Modèle entité-association

---

## **1. Structuration des données**

---

# Structuration des données

---

Les données d'une entreprise ou organisation sont au centre de la gestion et de l'information.

Afin d'en rendre le traitement efficace, d'en assurer la pérennité et la transférabilité, il sera nécessaire de bien les structurer.

Dans le cadre de **bases de données relationnelles** (structurées autour de la notion de relation), nous réaliserons cela à l'aide d'un **schéma entité-association**.

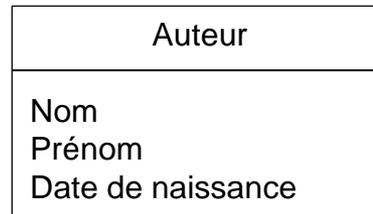
Les étapes suivantes permettent d'élaborer un modèle complet :

- Identifier les besoins (les règles de gestion)
- Elaborer le dictionnaire de données (identifier les **entités** du cas étudié)
- Décrire les liens entre les données (les **associations**)
- Tracer le **modèle conceptuel de données** ou **schéma entité-association**

# Schéma entité-association

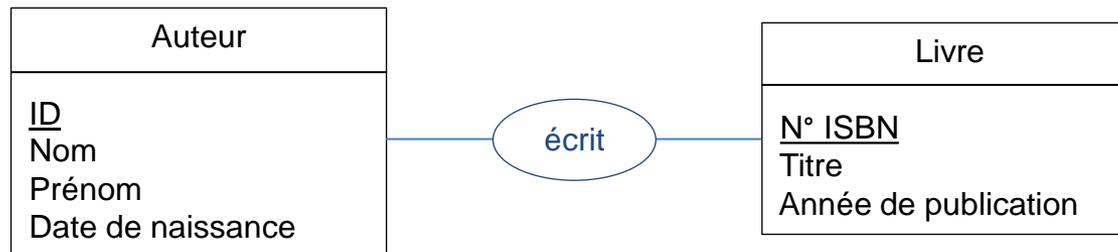
## Entités :

- Chaque entité est décrite par un ensemble de propriétés appelées attributs ou caractéristiques.



## Associations :

- Une association définit un lien sémantique entre une ou plusieurs entités.



# Identifiants et cardinalités

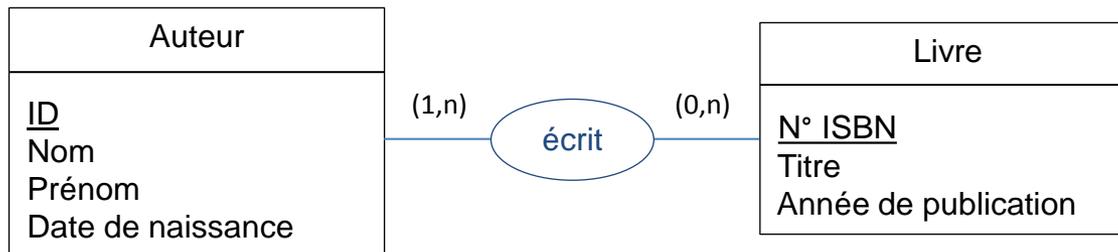
## Identifiants :

- Chaque entité doit posséder un identifiant : un (ou plusieurs) attribut(s) permettant de différencier les différentes instances d'une même entité.



## Cardinalités :

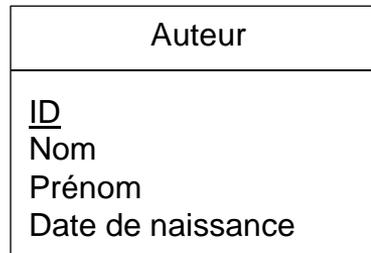
- Les cardinalités indiquent les nombres minimum et maximum de liens d'une entité vers les autres (0 = aucune, 1 = une seule, n = plusieurs).



# Conversion du schéma vers des tables

## Entités :

- Chaque entité va donner lieu à une table contenant une colonne pour chaque attribut.



Auteur			
<u>ID</u>	Nom	Prénom	Date de naissance
2	Camus	Albert	07-11-1913
8	Hugo	Victor	26-02-1802
15	Rowling	Johanne	31-07-1965
23	Tyson	Mike	30-06-1966

- Les identifiants devant être uniques, ils donneront lieu à une contrainte d'unicité appelée **clé primaire**.

# Conversion du schéma vers des tables

## Associations :

- Les association simple-simple ou simple-complexe seront converties en ajoutant une référence dans une des tables pour les entités.



Auteur				
<u>ID</u>	Nom	Prénom	Date de naissance	Pays d'origine
2	Camus	Albert	7-11-1913	1
8	Hugo	Victor	26-02-1802	3
15	Rowling	Johanne	31-07-1965	4
23	Tyson	Mike	30-06-1966	2

Pays	
ID	Nom
1	Algérie
2	Etats-Unis
3	France
4	Royaume Uni

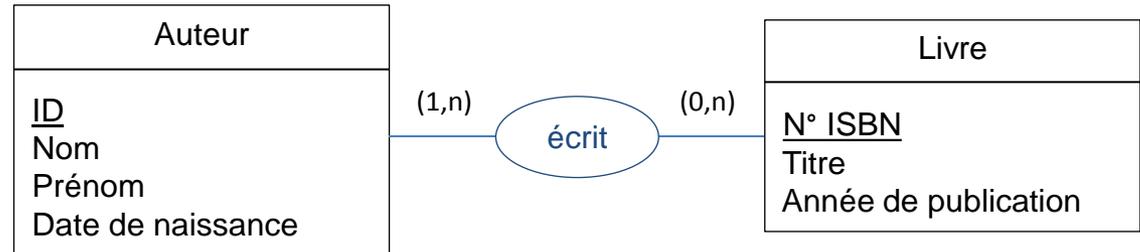
- Les identifiants devant exister dans la table référencée, ils donneront lieu à une contrainte d'intégrité nommée **clé étrangère**.

# Conversion du schéma vers des tables

## Associations :

- Les association complexe-complexe donne lieu à une nouvelle table.

Auteur			
<u>ID</u>	Nom	Prénom	Date de naissance
2	Camus	Albert	07-11-1913
8	Hugo	Victor	26-02-1802
15	Rowling	Johanne	31-07-1965
23	Tyson	Mike	30-06-1966



Auteur écrit Livre	
Auteur	Livre
2	978-2070322886
8	978-2211215350
15	978-2070643028
23	978-2352042839

Livre		
<u>N°ISBN</u>	Titre	Année de publication
978-2070322886	Le Mythe de Sisyphe	1942
978-2211215350	Les Misérables	1862
978-2070643028	Harry Potter à l'école des sorciers	1997
978-2253158356	Le Livre Sans Nom	2011
978-2352042839	La vérité et rien d'autre	2013

## Exercice (Besoins)

---

Pour les besoins de la gestion d'un aéroport, on souhaite mémoriser dans une base de données les informations nécessaires à la description des faits suivants :

- chaque avion géré est identifié par un numéro d'immatriculation. On doit connaître le nom, l'adresse et le numéro de téléphone du propriétaire ;
- chaque avion est d'un certain type, celui-ci étant caractérisé par son nom, le nom du constructeur et le nombre de places ;
- la maintenance des avions est assurée par les mécaniciens de l'aéroport. Pour toute intervention effectuée, on conserve son numéro unique, son objet, sa date et sa durée ;
- pour chaque mécanicien on connaît son nom, son adresse, son numéro de téléphone et les types d'avion sur lesquels il est habilité à intervenir.

Le tableau suivant reprend un extrait des données collectées par le chef mécanicien. Etablissez un modèle entité-association pour ce système.

## Exercice (Données)

N°	Objet	Date	Dur.	Matr. Avion	Type	Constr.	Nb pl.	Proprio	Adr. Proprio	Tel. Proprio	Mécano	Adr. Mécano	Tél. Mécano	Habilitation
126	Turbine	21/08/2013	45	T112	Airbus A320	Airbus	150	Jean KAGINA	99, rue des dunes	12345678	Martin ALONSI	Chemin des Rois, 18	11223344	Airbus A310, Airbus A320
127	Air conditionné	21/08/2013	30	M125	AASI Jetcruzer	AASI	5	Lionel Noblet	1, Place de la Concorde	87654321	Tiery Mitchell	Avenue Jacobi 152	26262626	AASI Jetcruzer, Gulfstream G500
128	Moteur	22/08/2013	180	S175	Gulfstream G500	General Dynamics	19	Lionel Noblet	1, Place de la Concorde	87654321	Jacques Dupont	Boulevard Lucas, 4	98989898	Gulfstream G500
129	Moteur	22/08/2013	210	T459	Airbus A310	Airbus	150	Jean KAGINA	99, rue des dunes	12345678	Martin ALONSI	Chemin des Rois, 18	11223344	Airbus A310, Airbus A320
130	Tableau de bord	23/08/2013	90	SN115	Boeing 747	Boeing	404	Jean KAGINA	rue des dunes 99	12345678	Yvain Alonso	Chaussée du pauvre, 5	12121212	Boeing 747, Airbus A310, Airbus A320
131	Eclairage	23/08/2013	5	T489	Airbus A320	Airbus	150	Guillaume Van Bergen	45, rue des plaines	88800000	Martin Alonsi	18 Chemin des Rois	11223344	Airbus A310, Airbus A320
132	Ajustement siège	24/08/2013	15	M125	AASI Jetcruzer	AASI	5	Lionel Noblet	1, Place de la Concorde	87654321	Tiery Mitchell	Avenue Jacobi 152	26262626	AASI Jetcruzer, Gulfstream G500
133	Tableau de bord	24/08/2013	30	SN116	Boeing 747	Boeing	404	Guillaume VanBergen	45, rue des plaines	88800000	Yvain Alonso	Chaussée du pauvre, 5	12121212	Boeing 747, Airbus A310, Airbus A320

## Formes normales

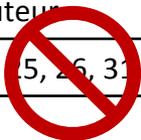
Les formes normales sont des niveaux de normalisation (des propriétés que doivent respecter les modèles conceptuels pour être considérés valides).

Les trois premières sont les plus répandues :

- **1<sup>ère</sup> Forme Normale (1FN) :**

Les attributs d'une relation doivent être atomiques et doivent être en dépendance fonctionnelle avec la clef primaire de cette dernière.

Auteur écrit Livre	
Auteur	Livre
{5, 26, 31}	Le Roman de Renard



**Pas d'attributs composés**

Auteur écrit Livre	
Auteur	Livre
25	Le Roman de Renard
26	Le Roman de Renard
31	Le Roman de Renard

## Formes normales

- **2<sup>ème</sup> Forme Normale (2FN) :**

Il faut être en 1FN et que toutes les dépendances fonctionnelles entre la clef primaire et les autres attributs de la relation soient élémentaires.

**Les attributs doivent dépendre de l'entière de la clé**

Livres		
<u>Titre</u>	<u>Editeur</u>	Pays
Harry Potter à l'école des sorciers	Assimil GmbH	Allemagne
Harry Potter à l'école des sorciers	Gallimard Jeunesse	France

**Clé primaire**

## Formes normales

- **3<sup>ème</sup> Forme Normale (3FN) :**

Il faut être en 2FN et que les attributs qui ne font pas partie de la clé ne dépendent que de celle-ci.

**Pas de dépendances transitives**

Livres					
<u>N°ISBN</u>	Titre	Année de publication	Prénom	Nom	Date naissance
978-2070322886	Le Mythe de Sisyphe	1942	Albert	Camus	07-11-1913
978-2211215350	Les Misérables	1862	Hugo	Victor	26-02-1802
978-2070643028	Harry Potter à l'école des sorciers	1997	Rowling	Johanne	31-07-1965
978-2253158356	Le Livre Sans Nom	2011			
978-2352042839	La vérité et rien d'autre	2013	Tyson	Mike	30-06-1966

Installation de PostgreSQL et pgAdminIII  
Création d'une base de données  
Insertion des données structurées dans la base de données

---

## **2. Installation de PostgreSQL et insertion des données structurées**

# Systemes de gestion de bases de données

---

Les systèmes de gestion de bases de données (SGBD) sont des logiciels permettant le stockage et le partage de données.

Ils servent d'intermédiaires entre les utilisateurs et les bases de données et assurent la qualité, la pérennité et la transmissibilité des données en cachant la complexité des opérations.

Il existe plusieurs systèmes de ce type :

- IBM DB2
- Microsoft SQL Server
- Oracle MySQL
- Oracle Database
- PostgreSQL

## Installation de PostgreSQL et pgAdmin III

---

### Sur Ubuntu :

- Installation via la logithèque de *PostgreSQL* et *pgAdmin3*
- Installation via le gestionnaire de paquets Synaptic de *postgresql*, *pgadmin3* et *pgadmin3-data* ainsi que tous les paquets qui en dépendent
- Installation via la commande apt-get dans le terminal
  - `sudo apt-get update`
  - `sudo apt-get install postgresql pgadmin3`

### Sur Windows :

- Télécharger l'installateur graphique sur : <http://www.postgresql.org/download/> (inclut PostgreSQL et pgAdmin III)

# Création d'un super-utilisateur pour PostgreSQL

---

Dans le terminal :

- Pour créer un utilisateur avec le même nom que le login système :
  - `sudo -u postgres createuser -s $USER -P`
  - Entrer un mot de passe (attention à la touche majuscule)
- Pour choisir un autre nom :
  - `sudo -u postgres createuser -s userulb -P`
  - Entrer un mot de passe

## Création d'une connexion depuis pgAdmin III

- Lancer pgAdmin III
- Dans la barre d'outils, cliquer sur le bouton « Ajouter une connexion à un serveur »



- Dans la fenêtre qui s'ouvre, entrer les informations suivantes :
  - Nom : au choix (MaMachine, par ex.)
  - Hôte : localhost
  - Port : 5432
  - Nom utilisateur : login système (ou celui choisi, par ex. userulb)
  - Mot de passe : celui choisi
- Cliquer sur OK / Valider

A screenshot of the 'New Server Registration' dialog box in pgAdmin III. The dialog has a title bar with a close button (X) and the text 'New Server Registration'. It contains several fields and checkboxes:

- Name:** MaMachine
- Host:** localhost
- Port:** 5432
- SSL:** (empty dropdown)
- Maintenance DB:** postgres
- Username:** mlennert
- Password:** (masked with dots)
- Store password:**
- Restore env?:**
- DB restriction:** (empty text area)
- Service:** (empty text field)
- Connect now:**
- Colour:** (empty text field with a color selection button)

At the bottom, there are three buttons: 'Aide', 'Valider', and 'Annuler'.

# Création d'une base de données dans pgAdmin III

- Après avoir établi la connexion, un clic droit sur « Bases de données » dans l'arborescence permet d'en créer une nouvelle
- Il suffit d'en choisir le nom (par ex. gestion\_aeroport)

The screenshot shows the 'New Database...' dialog box in pgAdmin III. The 'Name' field is highlighted with a red border and contains the text 'remembrement'. The 'Encoding' dropdown is set to 'UTF8'. The 'Tablespace' dropdown is set to '<default tablespace>'. The 'Connection Limit' field contains '-1'. The 'Collation' field has a mouse cursor over it. The 'Comment' field is empty. The 'Aide', 'Valider', and 'Annuler' buttons are visible at the bottom.

## Insertion des données

---

Pour stocker ses données :

- Créer des tables pour chaque entité et association qui le nécessite
- Remplir les tables avec les valeurs de chaque attribut

En pratique, plusieurs possibilités pour y arriver :

- En utilisant l'interface graphique de pgAdmin III
- En utilisant le langage SQL
- En utilisant la commande `psql` dans le terminal

## Création de tables dans pgAdmin III

- Dans l'arborescence de la base de données créée, l'élément Tables vous indique les tables existantes (pour l'instant 0)
- Un clic droit permet de créer une nouvelle table :
  - Choisir un nom pour la table
  - Créer toutes les colonnes de la table (en précisant leur type)
  - Ajouter les contraintes de clé primaire et de clés étrangères si c'est nécessaire
- Une fois la table créée, grille d'édition permet de voir et d'entrer des données dans la table ligne par ligne
- La fenêtre SQL vous permet quand à elle d'exécuter des requêtes SQL sur votre base de données



# Création de tables avec le langage SQL

---

## Structure type :

```
CREATE TABLE {nom de la table}(  
    {nom d'attribut} {type de l'attribut},  
    {nom d'attribut} {type de l'attribut}  
);
```

## Exemple :

```
CREATE TABLE type_avion(  
    id SERIAL PRIMARY KEY,  
    nom VARCHAR(50),  
    constructeur VARCHAR(50),  
    nb_places INT  
);
```

TypeAvion			
id	nom	constructeur	nb_places
1	Airbus A320	Airbus	150
2	AASI Jetcruzer	AASI	5
3	Gulfstream G500	General Dynamics	19
4	Airbus A310	Airbus	150
5	Boeing 747	Boeing	404

## Types de données

- Types d'attributs dans PostgreSQL et leur équivalent dans le langage SQL :

Types PostgreSQL	Types SQL	Description
Boolean		Type logique ( <b>TRUE</b> , <b>FALSE</b> )
Character varying	<b>VARCHAR</b> ( )	Chaîne de caractères (longueur)
Date	<b>DATE</b>	Date du calendrier
Double precision	<b>DOUBLE PRECISION</b>	Nombre à virgule flottante
Integer	<b>INT</b>	Nombre entier
Real	<b>REAL</b>	Nombre réel de simple précision
Serial		Entier à incrémentation automatique
Time without timezone	<b>TIME</b>	Temps sur 24 heures

- PRIMARY KEY** indique la clé primaire
- NOT NULL** indique qu'une cellule ne peut être vide
- REFERENCES** indique les clés étrangères et leur lien

## Création de tables avec le langage SQL

---

- Exemple :

```
CREATE TABLE avion(  
    matricule VARCHAR(10) PRIMARY KEY,  
    type INT REFERENCES type_avion(id),  
    propriétaire INT REFERENCES propriétaire(id)  
);
```

Avion		
<u>matricule</u>	type	propriétaire
T112	1	1
M125	2	2
S175	3	2
T459	4	1
SN115	5	1
T489	1	3
SN116	5	3

# Insertion dans des tables avec le langage SQL

---

## Structure type :

```
INSERT INTO {nom de la table}({liste des attributs})  
VALUES ({liste des valeurs pour une ligne}),  
{liste des valeurs pour une ligne};
```

## Exemples :

```
INSERT INTO type_avion(id, nom, constructeur,  
nb_places) VALUES (1, 'Airbus A320', 'Airbus', 150);
```

- Ou encore, si id est un serial qui s'auto-incrémente :

```
INSERT INTO type_avion(nom, constructeur, nb_places)  
VALUES ('AASI Jetcruzer', 'AASI', 5),  
( 'Gulfstream G500', 'General Dynamics', 19),  
( 'Airbus A310', 'Airbus', 150),  
( 'Boeing 747', 'Boeing', 404);
```

## Remplissage de tables avec la commande COPY

---

- Nécessite d'avoir créé un fichier texte ou csv avec les données de la table et que ce fichier soit sur la même machine que le serveur
- Les données de chaque cellule doivent avoir été séparées par un caractère spécial qui n'apparaît pas dans les données (par ex. point virgule, barre verticale, ...)
- Cette requête n'est implémentée que dans PostgreSQL

### Structure type :

```
COPY {nom de la table} FROM {chemin vers fichier}  
WITH DELIMITER {le caractère séparateur};
```

### Exemple :

```
COPY mecanicien FROM '/home/userulb/mecanicien.csv'  
WITH DELIMITER ';' ;
```

## Remplissage de tables avec la commande COPY

---

- L'usage de la commande COPY sans préciser les attributs suppose que ces derniers se trouvent tous dans le fichier csv et apparaissent dans le même ordre que dans la table
- Si ce n'est pas le cas, il est possible de préciser quels sont les attributs se trouvant dans le fichier

### Structure type :

```
COPY {nom de la table} ({liste des attributs})  
FROM {chemin vers fichier}  
WITH DELIMITER {le caractère séparateur};
```

### Exemple :

```
COPY mecanicien(nom, adresse, num_telephone)  
FROM '/home/userulb/mecanicien.csv'  
WITH DELIMITER ';' ;
```

## Exercice Remembrement

---

Suite à la construction d'une nouvelle autoroute, la région doit procéder à un remembrement des terres agricoles pour permettre à tous les agriculteurs d'avoir toutes leurs terres du même côté de la nouvelle autoroute. En même temps il ne faut pas non plus punir les propriétaires des terrains.

La région possède un fichier avec les données brutes, identifiant pour chaque parcelle le ou les propriétaires et l'exploitant. Mais ce fichier ne s'avère pas très pratique pour la gestion au quotidien, et la région demande donc de restructurer tout cela en une base de données propre selon les règles de normalisation et dans une structure relationnelle.

## Exercice Remembrement

N°	adresse parcelle	nom proprio	prénom proprio	adresse proprio	téléphone proprio	Exploitant	adresse exploitant	type de terre	Surf. (en ha)	nombre de points remembrement	pts/ha
1	Au Champ 8, 5430 Beaulieu	Dupont	Bernard	Rue de la Longue Vie 15, 1050 Bruxelles	02/5432187	Dupont, Ferdinand	Rue de l'Ecole, 5430 Beaulieu	Limons à tâches	120	90000	750
2	Chant d'Oiseau 15b, 5430 Beaulieu	Decocq	Micheline	Rue de l'Eglise 3, 7777 Beauvillage	067/12345	George, Pierre	Rue du Château 1, 5430 Beaulieu	Sable argileux	110	44000	400
3	Chemin de la Haie 9, 5430 Beaulieu	Decocq	Micheline	Rue de l'Eglise 3, 7777 Beauvillage	0474/89.56.87	Laferme, Hilaire	10, Rue Ronde, 5431 Mochelieu	Sables argileux	450	180000	400
4	Au Champ 5, 5430 Beaulieu	Dupont	Bernard	15, Rue de la Longue Vie, 1050 Bruxelles	02/5432187	Laferme, Hilaire	Rue de la Poste 15, 5430 Beaulieu	Limons à tâches	340	255000	750
5	Rue du Chemin de Fer 9, 5431 Mochelieu	Dufer	Piet	Quai de la brique 65, 6654 Montusine	045/335241	Laferme, Hilaire	Rue de la Poste 15, 5430 Beaulieu	Argile lourde	67	53600	800
6	Chemin vert 11, 5430 Beaulieu	Dufer	Piet	Quai de la Brique 65, 6654 Montusine	045/335241	Pierre George	Rue du Château 1, 5430 Beaulieu	Argile lourde	220	176000	800
7	7, Au Champ, 5430 Beaulieu	Dupont	Bernard	Rue de la Longue Vie 15, 1050 Bruxelles	02/5432186	Dupont, Ferdinand	Rue de l'Ecole, 5430 Beaulieu	Limons à tâches	123	92250	750
8	C. de la Haie 24, 5430 Beaulieu	Decoc	Micheline	Rue de l'Eglise 3, 7777 Beauvillage	067/12345	George, Pierre	Rue du Château 1, 5430 Beaulieu	Argile lourde	90	72000	800
9	Rue courte 1, 5430 Beaulieu	Hilo	Martin	6649 Kempenhout, Kerkstraat 7	023/776643	George, Pierre	Rue du Château 1, 5430 Beaulieu	Argiles lourdes	66	52800	800
10	Rue Longue 23, 5431 Mochelieu	Hilo	Martin	Kerkstraat 7, 6649 Kempenhout	023/776643	George, Pierre	Rue du Château 1, 5430 Beaulieu	Argiles lourdes	307	245600	800
10	Rue Longue 23, 5431 Mochelieu	Keyser	Marc	Rue du village 8, 5922 Chezmoi	043/739122	George, Pierre	Rue du Château 1, 5430 Beaulieu	Argiles lourdes	307	245600	800

Utilisation de la base de données complète  
Extraction d'informations en utilisant le langage SQL

---

### **3. Interrogation de la base de données en utilisant le langage SQL**

---

## Langage SQL

---

SQL (Structured Query Language) est un langage standardisé permettant de communiquer avec une base de données. Il présente l'avantage d'être utilisable avec une multitude de systèmes.

Il permet de :

- rechercher
- ajouter
- modifier
- supprimer

des données dans les bases de données relationnelles.

## Construction d'une requête de sélection

---

### Structure type :

```
SELECT {liste des attributs ou * pour tous}  
FROM {une ou plusieurs tables}  
[WHERE {liste de conditions ou critères}];
```

### Exemples :

- Sélectionner tous les attributs de la table Mécanicien :

```
SELECT * FROM mecanicien;
```

- Sélectionner les noms et constructeurs de la table TypeAvion :

```
SELECT nom, constructeur FROM type_avion;
```

- Sélectionner les noms de la table TypeAvion dont le nombre de places est supérieur à 100 :

```
SELECT nom FROM type_avion WHERE nb_places > 100;
```

## Critères de sélection

---

- Opérateurs de comparaison (signes au sein des conditions) :

=	Égal
< (ou >)	Inférieur (ou supérieur)
<= (ou >=)	Inférieur ou égal (ou supérieur ou égal)
<> ou !=	Différent de

- Opérateurs logiques (liens entre plusieurs conditions) :

<b>AND</b>	ET (les deux conditions doivent être vraies)
<b>OR</b>	OU (une des deux conditions doit être vraie)
<b>NOT</b>	Négation (la condition doit être fausse)

## Tri des résultats d'une requête

---

### Structure type :

```
SELECT {liste des attributs ou * pour tous}
FROM {une ou plusieurs tables}
[WHERE {liste de conditions ou critères}]
[ORDER BY {attribut(s)} [DESC (si décroissant)]];
```

### Exemples :

- Sélectionner les types d'avion dont le nombre de places est supérieur à 100 et triés par ordre alphabétique de leur nom :

```
SELECT * FROM type_avion WHERE nb_places > 100
ORDER BY nom;
```

- Sélectionner toutes les interventions par ordre décroissant de leur durée :

```
SELECT * FROM intervention ORDER BY duree DESC;
```

## Fonctions sur les résultats

---

- Fonctions d'agrégation (renvoient une seule valeur pour un groupe) :

<b>AVG</b>	Moyenne des valeurs
<b>MIN</b>	Valeur minimum
<b>MAX</b>	Valeur maximum
<b>SUM</b>	Somme les valeurs
<b>COUNT</b>	Compte le nombre d'éléments

- Fonctions mathématiques :

<b>ABS</b>	Valeur absolue
<b>ROUND</b>	Valeur arrondie
<b>SQUARE</b>	Valeur au carré
<b>SQRT</b>	Racine carrée de la valeur
<b>POWER</b>	Puissance de la valeur

## Groupement des résultats d'une requête

---

### Structure type :

```
SELECT {liste des attributs ou * pour tous}
FROM {une ou plusieurs tables}
[WHERE {liste de conditions ou critères}]
[GROUP BY {attribut(s)}];
```

### Exemples :

- Sélectionner les durées totales d'interventions par objet :

```
SELECT objet, SUM(duree)
FROM intervention
GROUP BY objet;
```

## Renommer les attributs dans les résultats

---

### Structure type :

```
SELECT {liste des attributs [AS nom choisi]}  
FROM {une ou plusieurs tables}  
[WHERE {liste de conditions ou critères}];
```

### Exemples :

- Sélectionner les durées totales d'interventions par objet :

```
SELECT objet, SUM(duree) AS duree_totale  
FROM intervention  
GROUP BY objet;
```

## Jointure de plusieurs tables (JOIN)

---

### Structure type :

```
SELECT {liste des attributs}
FROM {une table}
JOIN {une table} ON {lien entre les deux tables}
[WHERE {liste de conditions ou critères}];
```

### Exemples :

- Sélectionner les matricules d'avions, leur type et leur constructeur :

```
SELECT matricule, nom, constructeur
FROM avion
JOIN type_avion ON avion.type=type_avion.id;
```

## Jointure de plusieurs tables (sans JOIN)

---

### Structure type :

```
SELECT {liste des attributs}
FROM {plusieurs tables}
WHERE {lien(s) entre les tables}
[AND {liste de conditions ou critères}];
```

### Exemples :

- Sélectionner les matricules d'avions, leur type et leur constructeur :

```
SELECT matricule, nom, constructeur
FROM avion, type_avion
WHERE avion.type=type_avion.id;
```

## Fonctions sur les chaînes de caractères

- Fonctions de manipulation sur les chaînes de caractères :

Fonction	Exemple	Résultat
Concaténation	<code>'Post'    'greSQL'</code>	<code>'PostgreSQL'</code>
Concaténation	<code>'Numéro'    ' '    42</code>	<code>'Numéro 42'</code>
Minuscules	<code>LOWER('PostgreSQL')</code>	<code>'postgresql'</code>
Majuscules	<code>UPPER('PostgreSQL')</code>	<code>'POSTGRESQL'</code>
Initiales en majuscules	<code>INITCAP('alain DELOIN')</code>	<code>'Alain Deloin'</code>

### Exemple :

- Reconstituer les adresses complètes de tous les mécaniciens :

```
SELECT nom, num || ' ' || rue || ', ' || cp AS adresse
FROM mecanicien;
```

## Exercice Observation Faune

---

Dans le cadre de leur travail de l'observation de la faune dans le parc national de Keskecébeau, les agents notent toutes les observations obtenues lors de transects à travers le parc.

La direction du parc possède un fichier avec les données brutes, identifiant chaque observation. Mais ce fichier ne s'avère pas très pratique pour la gestion au quotidien, et la direction du parc demande donc de restructurer tout cela en une base de données propre composée d'autant de tables séparées que nécessaire.

## Exercice Observation Faune

date	heure	long	lat	nom_fr	nom_lat	menace	nbr_ind	typ_renc	nom_ag	prenom_ag	num_tel	email_ag	service	bur	rue	N°	cp	fax	resp_serv
2011-10-05	12:44:00	3,45735	8,765423	Elephant	Oreillus gigantes	oui	5	rencontre	Attara	Bienvenu	44/8537765	bienvenu.attara@yahoo.fr	Foresterie	345B	Rue de la Forêt	23	5678	44/785234	Dr. Baobab
2011-10-05	13:38:00	3,45812	8,765111	Cheval sauvage	Caballus wildus	non	2	observation de loin	Attara	Bienvenu	44/8537765	bienvenu.attara@yahoo.fr	Foresterie	345B	Rue de la Forêt	23	5678	44/785234	Dr. Baobab
2011-10-05	14:23:00	3,47128	8,76423	Gorille	Gorillus grandus	oui	8	observation rapprochée	Attara	Bienvenu	44/8537765	bienvenu.attara@yahoo.fr	Foresterie	345B	Rue de la Forêt	23	5678	44/785234	Dr. Baobab
2011-07-04	08:00:00	3,72313	8,553098	Elephant, Lion, Serpent brillant	Oreillus gigantes, Leo leo, Vipera luminosa	oui	2, 4, 1	obs. éloignée	Meyers	Alonso	44/8041288	ameyers@gmail.com	Faune générale	43	Rue des fauves	2	5677	44/4208945	Dr. Rat
2011-08-14	19:00:00	3,6511	8,601929	Elephant	Oreillus gigantes	oui	0	feces	Conaca	Karl	5438765	kc@fauves.fr	Grande faune	41	Rue des Elephants	33	5677	44/4286312	Prof. Lion
2011-08-14	19:20:00	3,65118	8,601937	Lion	Leo leo	non	1	renc.	Conaca	Karl	5438765	kc@fauves.fr	Grande faune	41	Rue des Elephants	33	5677	44/4286312	Prof. Lion
2011-08-14	19:34:00	3,65218	8,602061	Gorille	Gorillus grandus	oui	0	tr. de pass	Conaca	Karl	5438765	kc@fauves.fr	Grande faune	41	Rue des Elephants	33	5677	44/4286312	Prof. Lion
2011-08-14	20:18:00	3,661	8,602712	Braconnier	Homo fatalus	non	2	obs. Rapp.	Conaca	Karl	5438765	kc@fauves.fr	Grande faune	41	Rue des Elephants	33	5677	44/4286312	Prof. Lion
2011-10-05	14:59:00	3,47129	8,76841	Humain	Homo sapiens	non	5	rencontre	Attara	Bienvenu	44/8537765	bienvenu.attara@yahoo.fr	Foresterie	345B	Rue de la Forêt	23	5678	44/785234	Dr. Baobab

## Structure type de quelques requêtes

```
SELECT attribut1, attribut2, attribut3
FROM table1, table2
WHERE condition1
  AND condition2
  OR condition3
GROUP BY attribut3
ORDER BY attribut2;
```

Colonne à afficher

Tables où récupérer les informations

Conditions délimitant les lignes à afficher

Comment grouper les lignes

Comment ordonner les lignes

```
INSERT INTO table (attribut1, attribut2, attribut3)
VALUES (valeur1, valeur2, valeur3);
```

Nom de la table

Ordre des colonnes

Valeurs à insérer

Mise à jour et modification des données  
Gestion des accès à la base de données  
Cas d'utilisation et types de systèmes

---

## **4. Maintenance et accès à la base de données**

---

## Maintenance et accès à la base de données

---

De nombreuses autres commandes permettent d'assurer la maintenance d'une base de données en corrigeant les erreurs, améliorant les performances ou en donnant un accès (limité) à d'autres utilisateurs :

- UPDATE
- DELETE
- DROP
- INDEX
- VIEW
- USER (ROLE)

## Mise à jour avec UPDATE

---

### Structure type :

**UPDATE** {une table}

**SET** {opération à effectuer}

[**WHERE** {liste de conditions ou critères}];

### Exemples :

- Remplacer la superficie de la parcelle 3 par 550 :

```
UPDATE parcelles SET superficie=550 WHERE id=3;
```

## Suppression avec DELETE

---

### Structure type :

```
DELETE FROM {une table}
[WHERE {liste de conditions ou critères}];
```

### Exemples :

- Supprimer la parcelle 3 :

```
DELETE FROM parcelles WHERE id=3;
```

- Vider la table parcelles :

```
DELETE FROM parcelles;
```

## Suppression avec DROP

---

### Structure type :

```
DROP {un type d'élément} {un nom d'élément}
```

Drop permet de supprimer complètement un élément de la base de données tel qu'une table, une colonne, un utilisateur, une vue, la base de données, ...

### Exemples :

- Supprimer une table :

```
DROP TABLE proprietaires;
```

- Supprimer un utilisateur :

```
DROP USER moritz;
```

- Supprimer une vue :

```
DROP VIEW parcelles_par_proprio;
```

- Supprimer une base de données :

```
DROP DATABASE cadastre;
```

## Accélération des recherches avec un index

---

Lorsque de nombreuses requêtes font usage d'une même colonne pour des conditions ou critères, il est parfois efficace de définir un index.

Cela permet de retrouver plus rapidement les lignes concernées par une sélection. Par contre, les modifications de la table (mise à jour, suppression, ...) deviennent plus lentes car l'index doit être maintenu à jour aussi.

### Structure type :

```
CREATE INDEX {nom de l'index}  
ON {nom de la table} ( {nom de la colonne} )
```

### Exemple :

- Créer un index sur la colonne ville de la table propriétaires :

```
CREATE INDEX idx_proprio_ville ON proprietaires  
(ville);
```

## Créer des vues avec CREATE VIEW

---

Les vues sont des requêtes qui sont traitées comme des tables. Elles permettent de sélectionner certaines informations accessibles difficilement au sein du modèle et d'en partager l'accès de manière plus aisée.

Cette vue étant le résultat d'une requête, elle reste à jour même lorsque les données de base des tables d'origine sont modifiées.

### Structure type :

```
CREATE VIEW {nom de la vue} AS {requête à utiliser}
```

### Exemple :

- Créer une vue des parcelles associées à leur propriétaire :

```
CREATE VIEW parcelles_par_proprio AS  
SELECT numero_national, nom, adresse, num_parcelle  
FROM proprietaires  
    JOIN proprios_parcelles ON (numero=numero_proprio);
```

## Créer des utilisateurs et leur accorder des privilèges

---

Pour limiter l'accès à des données confidentielles, il est possible de créer des comptes utilisateurs qui n'auront accès qu'à certaines tables, requêtes, vues,

...

- **CREATE ROLE / CREATE USER** :

Permet de définir des groupes d'utilisateurs avec des privilèges différents

- **GRANT / REVOKE** :

Permet d'accorder ou retirer des droits d'accès :

- Aux requêtes de sélection, modification, insertion, suppression, ...
- Sur des bases de données, tables, colonnes, ...



---

## 5. Programmation et utilisation d'interfaces

---

## Programmation et utilisation d'interfaces

---

Il existe de nombreuses façons de donner accès à certaines fonctions d'une base de données :

- En utilisant la ligne de commande (pour certains systèmes)
- En utilisant des programmes dédiés, conçus à l'aide d'un langage de programmation (beaucoup plus spécifiques à une utilisation particulière)
- En utilisant des interfaces génériques pouvant accéder à toutes sortes de systèmes (logiciels tels que pgAdminIII, OpenOffice Base, Microsoft Access, ...)

## Exemple 1 : Utilisation de pgAdmin III

---

pgAdmin III permet la connexion à des bases de données sur d'autres serveurs connectés en réseau.

### Procédure :

- Créer un compte utilisateur sur le serveur PostgreSQL :

```
CREATE USER stagiaire WITH PASSWORD 'ulb';
```

- Donner l'accès à certaines requêtes sur certaines tables :

```
GRANT SELECT ON TABLE agent TO stagiaire;
```

- Donner l'accès à certaines requêtes sur toutes les tables :

```
GRANT SELECT ON ALL TABLES IN SCHEMA public  
TO stagiaire;
```

- Autoriser l'accès à distance (varie en fonction du système)
- Transmettre les informations de connexion (adresse IP, login, password)

## Autoriser l'accès à distance à PostgreSQL

---

Modifier les fichiers `pg_hba.conf` et `postgresql.conf` dans :

- `/etc/postgresql/x.x/main/` (sur Linux)
- `C:/Program Files/PostgreSQL/x.x/data/` (sur Windows)

Dans `pg_hba.conf`, ajouter une ligne :

```
host all all 127.0.0.1/32 md5
```

où `127.0.0.1/32` sera remplacé par :

- l'adresse IP de la machine et `/32` est le masque CIDR indiquant le nombre de connexions simultanées autorisées (`/32` pour une seule, `/28` pour 16, `/0` pour plus de 4 milliards)
- ou l'expression `samehost` pour prendre en compte toutes les adresses actuelles du serveur
- ou l'expression `samenet` pour prendre en compte toutes les adresses du même sous-réseau que le serveur

Dans `postgresql.conf`, assurez-vous que la ligne `listen_addresses = '*'` n'est pas précédée du symbole `#`

## Exemple 2 : Programmation d'une page web

---

Une page web dynamique peut être hébergée sur une machine et donner un accès personnalisé à une base de données présente dessus

### Procédure :

- Installer un serveur web Apache et le support pour le langage PHP (packages : apache2, php5, php5-pgsql )
- Placer un fichier php dans /var/www/ pour tester le serveur
- Créer un répertoire de travail
- Rendre le répertoire accessible grâce à un alias
- Relancer le serveur
  
- Dans le répertoire des pages php peuvent à présent être créées et seront accessibles depuis d'autres machines sur le réseau

## Exercice

---

- Créer une nouvelle connexion dans pgAdmin III vers un serveur de votre choix
  - Récupérer l'adresse IP de l'hôte ainsi que les login et mot de passe
- Tenter quelques requêtes SELECT, CREATE, UPDATE, ... pour tester les privilèges disponibles

## Exercices supplémentaires Observation Faune

---

- Karl Conaca s'est fait voler son chapeau à l'instant par un babouin (primatus cleptomanus, non menacé). Faites les insertions nécessaires.
- Donnez les noms et prénoms des agents ayant réalisé au moins 4 observations
  - Ecrivez 3 requêtes différentes permettant d'obtenir ce résultat
- Donnez le nom de l'agent, la date et l'heure de la dernière observation enregistrée (en date et en heure!)

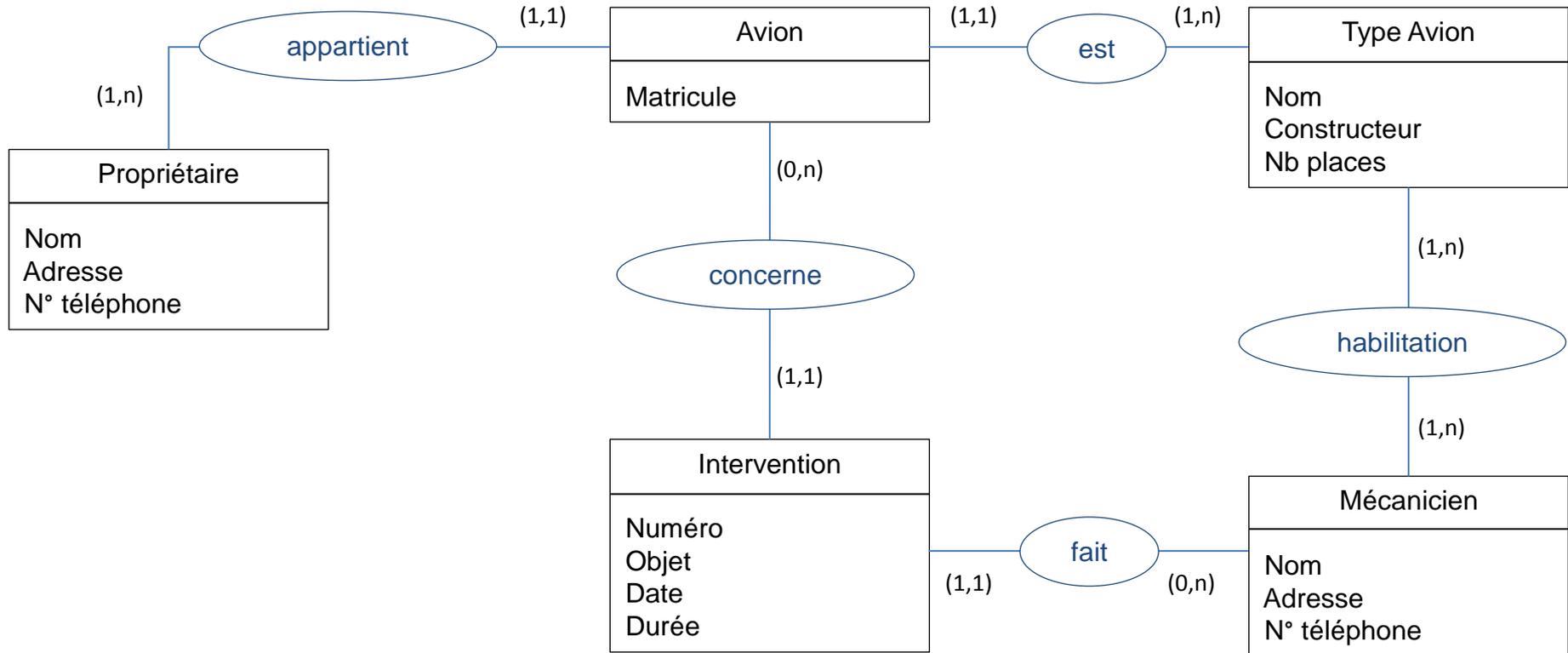
Exercice Gestion Aéroport (modélisation)  
Exercice Remembrement (modélisation)  
Exercice Observation Faune (modélisation)

---

## **Solutions exercices**

---

# Solution Gestion Aéroport



# Solution Gestion Aéroport

Propriétaire			
<u>id</u>	nom	adresse	no_telephone
1	Jean KAGINA	99, rue des dunes	12345678
2	Lionel NOBLET	1, Place de la Concorde	87654321
3	Guillaume VAN BERGEN	45, rue des plaines	88800000

Habilitation	
<u>mecanicien</u>	<u>type</u>
1	4
1	1
2	2
2	3
3	3
4	5
4	4
4	1

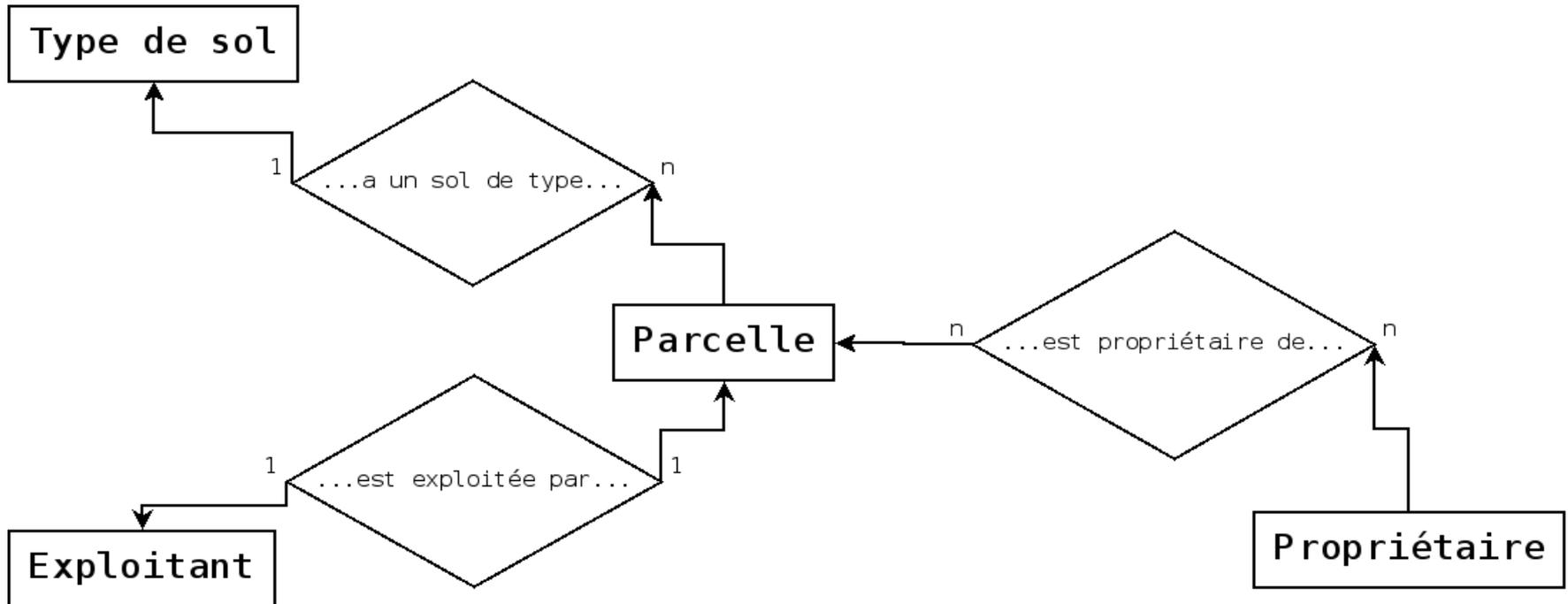
Avion		
<u>matricule</u>	<u>type</u>	<u>proprietaire</u>
T112	1	1
M125	2	2
S175	3	2
T459	4	1
SN115	5	1
T489	1	3
SN116	5	3

Mécanicien			
<u>id</u>	nom	adresse	no_telephone
1	Martin ALONSI	18, Chemin des Rois	11223344
2	Tiery MITCHELL	152, Avenue Jacobi	26262626
3	Jacques DUPONT	4, Boulevard Lucas	98989898
4	Yvain ALONSO	5, Chaussée du pauvre	12121212

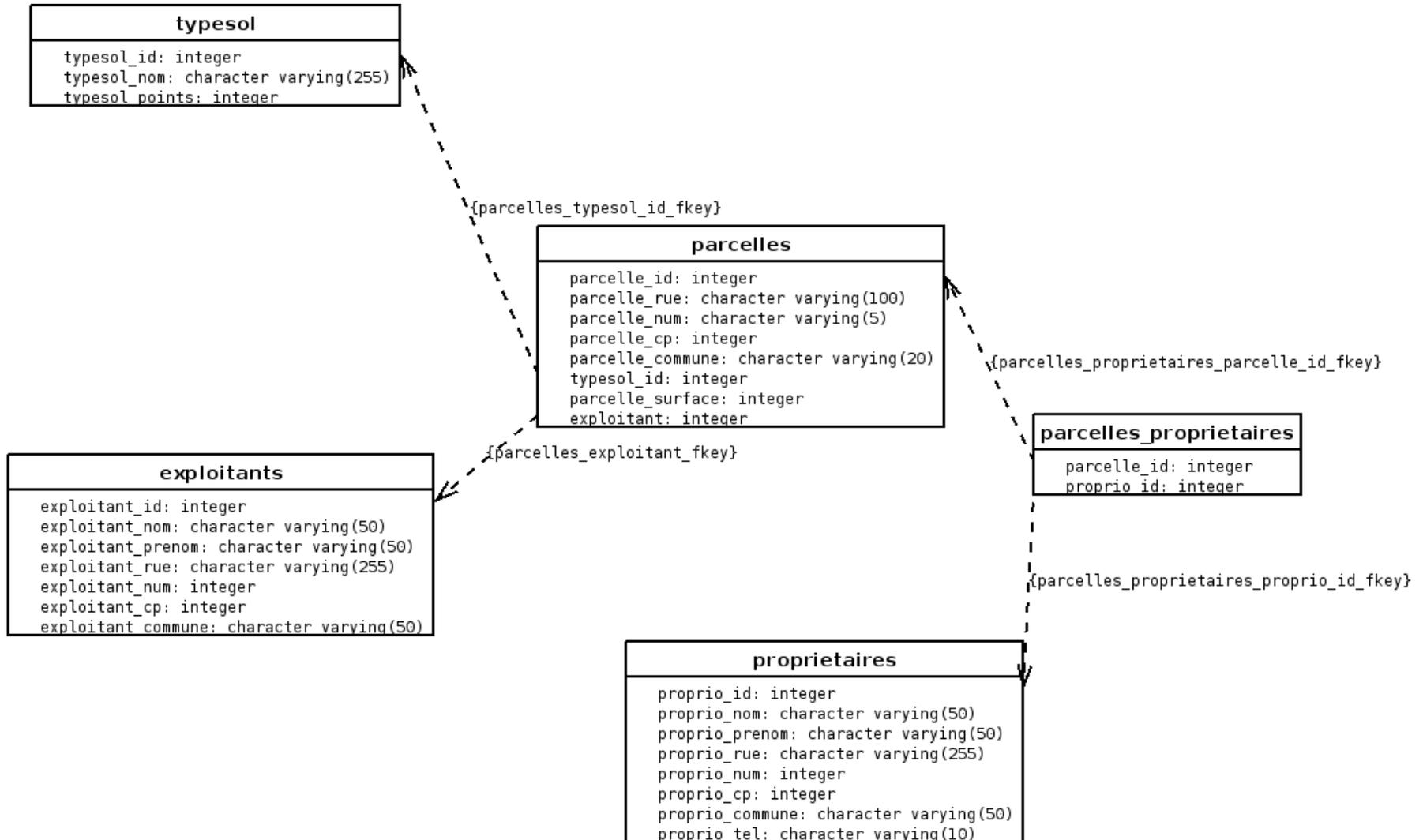
TypeAvion			
<u>id</u>	nom	constructeur	nb_places
1	Airbus A320	Airbus	150
2	AASI Jetcruzer	AASI	5
3	Gulfstream G500	General Dynamics	19
4	Airbus A310	Airbus	150
5	Boeing 747	Boeing	404

Intervention					
<u>numero</u>	objet	date	duree	<u>avion</u>	<u>mecanicien</u>
126	Turbine	21/08/2013	45	T112	1
127	Air conditionné	21/08/2013	30	M125	2
128	Moteur	22/08/2013	180	S175	3
129	Moteur	22/08/2013	210	T459	1
130	Tableau de bord	23/08/2013	90	SN115	4
131	Eclairage	23/08/2013	5	T489	1
132	Ajustement siège	24/08/2013	15	M125	2
133	Tableau de bord	24/08/2013	30	SN116	4

# Solution Remembrement



# Solution Remembrement



## Solution Remembrement

Types_sol		
id	nom	pts
1	Limons à tâches	750
2	Sable argileux	400
3	Argile lourde	800

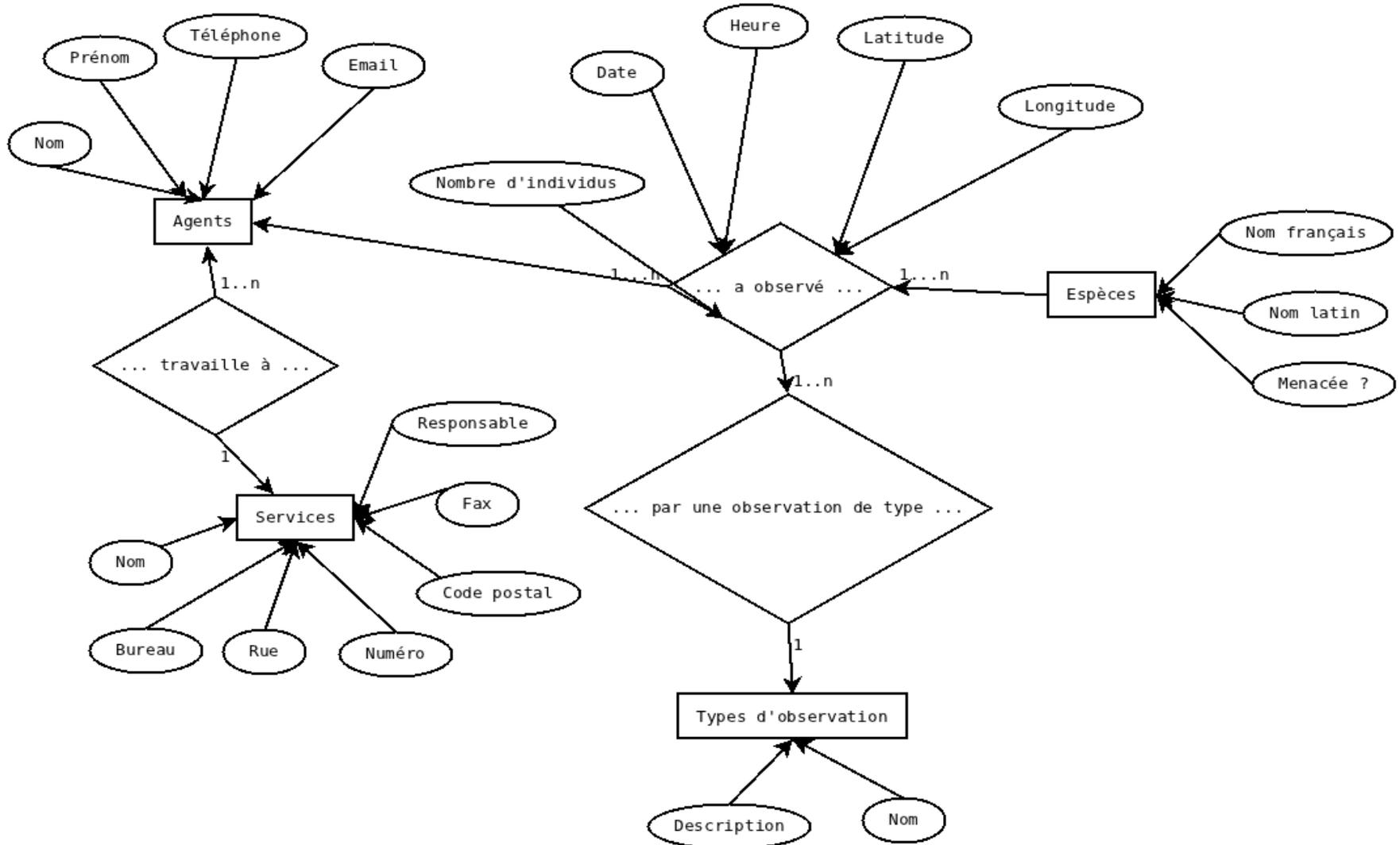
Exploitants						
id	nom	prenom	rue	num	cp	commune
1	Dupont	Ferdinant	Rue de l'Ecole	4	5430	Beaulieu
2	George	Pierre	Rue du Château	1	5430	Beaulieu
3	Laferme	Hilaire	Rue de la Poste	15	5430	Beaulieu

Parcelles_proprio	
parc_id	propr_id
1	1
2	2
3	2
4	1
5	3
6	3
7	1
8	2
9	4
10	4
10	5

Proprietaires							
id	nom	prenom	rue	num	cp	commune	tel
1	Dupont	Bernard	Rue de la Longue Vie	15	1050	Bruxelles	02/5432187
2	Decocq	Micheline	Rue de l'Eglise	3	7777	Beauvillage	067/12345
3	Dufer	Piet	Quai de la brique	65	6654	Montusine	045/335241
4	Hilo	Martin	Kerkstraat	7	6649	Kempenhout	023/776643
5	Keyser	Marc	Rue du village	8	5922	Chezmoi	043/739122

Parcelles							
id	rue	num	cp	commune	typsol_id	surf	exploit_id
1	Au Champ	8	5430	Beaulieu	1	120	1
2	Chant d'Oiseau	15b	5430	Beaulieu	2	110	2
3	Chemin de la Haie	9	5430	Beaulieu	2	450	3
4	Au Champ	5	5430	Beaulieu	1	340	3
5	Rue du Chemin de Fer	9	5431	Mochelieu	3	67	3
6	Chemin vert	11	5430	Beaulieu	3	220	2
7	Au Champ	7	5430	Beaulieu	1	123	1
8	C. de la Haie	24	5430	Beaulieu	3	90	2
9	Rue courte	15b	5430	Beaulieu	3	66	2
10	Rue Longue	23	5431	Mochelieu	3	307	2

# Solution Observation Faune



# Solution Observation Faune

Especies			
id	nom_fr	nom_lat	menace
1	Elephant	Oreillus gigantes	oui
2	Cheval sauvage	Caballus wildus	non
3	Gorille	Gorillus grandus	oui
4	Lion	Leo leo	non
5	Serpent brillant	Vipera luminosa	oui
6	Braconnier	Homo fatalus	non
7	Humain	Homo sapiens	non

Types_observation		
id	nom	description
1	rencontre	Moins de 20m
2	observation rapprochée	Moins de 50m
3	observation éloignée	Plus de 50m
4	traces de passage	Traces de passage
5	feces	Matière fécale

Services							
id	nom_serv	bur	rue	num	cp	fax	resp
1	foresterie	345B	rue de la Forêt	23	5678	44/785234	Dr. Baobab
2	faune generale	43	rue des Fauves	2	5677	44/4208945	Dr. Rat
3	grande faune	41	rue des Elephants	33	5677	44/4286312	Pr. Lion

Agents					
id	nom	prenom	tel	email	id_serv
1	Attara	Bienvenu	44/8537765	bienvenu.attara@yahoo.fr	1
2	Meyers	Alonso	44/8041288	ameyers@gmail.com	2
3	Conaca	Karl	5438765	kc@fauves.fr	3

Observations								
id	date	heure	lat	long	nbr_ind	id_ag	id_esp	id_typ
1	2011-10-05	12:44:00+01	8.765423	3.457345	5	1	1	1
2	2011-10-05	13:38:00+01	8.765111	3.458123	2	1	2	3
3	2011-10-05	14:23:00+01	8.76423	3.471278	8	1	3	2
4	2011-07-04	08:00:00+01	8.553098	3.723129	2	2	1	3
5	2011-07-04	08:00:00+01	8.553098	3.723129	4	2	4	3
6	2011-07-04	08:00:00+01	8.553098	3.723129	1	2	5	3
7	2011-08-14	19:00:00+01	8.601929	3.651098	0	3	1	5
8	2011-08-14	19:20:00+01	8.601937	3.651178	1	3	4	1
9	2011-08-14	19:34:00+01	8.602061	3.622176	0	3	3	4
10	2011-08-14	20:18:00+01	8.602712	3.651002	2	3	6	2
11	2011-10-05	14:59:00+01	8.76841	3.471293	5	1	7	1