

# Projet de Bases de Données

Grebic Christopher - Menin Valérian

8 janvier 2015

## Description du projet

Pour le semestre 5, nous avons dû développer une application pour informatiser la gestion d'un parc animalier.

## Choix d'analyse

Le modèle conceptuel de données choisi est assez simple. Les relations sont principalement de type 1 à N sauf pour les cas suivants :

- Historique d'enclos  
Celui-ci regroupe trois dates en tant que clé primaire. L'originalité est liée au fait que nous avons trois clés étrangères.
- Répartition géographique  
La répartition des espèces affecte une localisation géographique à une espèce.
- Composition de régime  
La composition d'un régime alimentaire est définie par un aliment ainsi qu'un régime pour lesquels une quantité est alors donnée.
- Gestion des substituts  
Les substituts sont simplement une clé secondaire qui fait référence à la clé primaire de la classe aliments. Cela pourrait même permettre de gérer des substituts de substituts et ainsi de suite de manière récursive.

D'une manière générale, les données sont gérées de façon à permettre la plus grande flexibilité de leur contenu. C'est-à-dire que les modifications y sont le plus facile possible.

A cet égard de simplicité, nous avons choisi certaines conventions concernant les types de données utilisés :

- Les identifiants et stocks  
Comme toujours les identifiants sont de type INT.  
Les stocks et quantités sont quant à eux de type DOUBLE car les plus petits animaux ne mangent bien évidemment pas plusieurs kilos de nourriture par jour.
- Les noms  
VARCHAR d'une taille de 63 en général sauf pour des cas particuliers tels que les noms scientifiques qui sont souvent plus longs possède une taille de 127.
- Les sexes et espèces menacées  
Pour les sexes, nous avons le choix entre le type TINYINT ou un

ENUM. Nous avons choisi l'utilisation d'un type ENUM avec comme possibilité 'M' ou 'F'. Cela simplifie la saisie et le traitement des données.

Pour En ce qui concerne le fait qu'une espèce soit menacée ou non, il était plus logique et pratique d'utiliser un TINYINT. C'est donc le choix que nous avons fait.

En cas de besoin d'une compatibilité de notre gestion du parc avec le système du zoo, les index suivants existent :

- Codes aliments
- Localisation
- Code enclos.

Nous avons définis plusieurs contraintes d'intégrités pour gérer au mieux nos données.

- Lorsqu'une famille est supprimée, toutes les relations entre les espèces et cette famille sont supprimées (SET NULL).
- On empêche la suppression d'une localisation si cette localisation est utilisée par une espèce (RESTRICT)
- Lors de la suppression d'une espèce, sa répartition et ses régimes sont supprimés (CASCADE) et ses références dans animaux sont mises à NULL (SET NULL).
- Lorsqu'on supprime un régime, sa composition est supprimée (CASCADE) mais on empêche la suppression d'un aliment utilisé dans un régime.
- Quand on supprime un aliment, les aliments qui utilisent ce même aliment en tant que substitut sont mis à NULL (SET NULL).
- Suite à la suppression d'un animal, on supprime son historique d'affiliation aux enclos.
- Après la suppression d'un enclos, celui-ci est supprimé de l'historique.
- Si on supprime un soigneur, toutes les relations entre les enclos et ce soigneur sont supprimées (SET NULL).

Le trigger permet l'utilisation du stock substitut en cas de rupture de celui de l'aliment primaire.

Nom d'attribut	Signification	Type (taille)	E / Co	Règle de calcul
idAnimal	-	INT	E	/
nomA	-	VARCHAR(63)	E	/
sexeA	-	ENUM('M','F')	E	/
dateNaissanceA	-	DATE	E	/
dateArriveeA	-	DATE	E	/
idEspece	-	INT	E	/
nomSciE	-	VARCHAR(127)	E	/
nomVulE	-	VARCHAR(63)	E	/
nbIndividuE	-	INT	E	/
menaceE	-	TINYINT(1)	E	/
idFamille	-	INT	E	/
nomF	-	VARCHAR(127)	E	/
idLocalisation	-	INT	E	/
codeL	-	CHAR(3)	E	/
nomL	-	VARCHAR(63)	E	/
idSoigneur	-	INT	E	/
nomS	-	VARCHAR(63)	E	/
prenomS	-	VARCHAR(63)	E	/
dateS	-	DATE	E	/
idEnclos	-	INT	E	/
codeE	-	CHAR(3)	E	/
nomE	-	VARCHAR(63)	E	/
surfaceE	-	INT	E	/
typeE	-	VARCHAR(63)	E	/
dateEntree	-	DATE	E	/
dateSortie	-	DATE	E	/
idRegime	-	INT	E	/
sexeR	-	ENUM('M','F')	E	/
quantite	-	DOUBLE	E	/
idAliment	-	INT	E	/
codeA	-	CHAR(3)	E	/
designationA	-	VARCHAR(63)	E	/
stockA	-	DOUBLE	E	/
substitut	-	INT	E	/

FIGURE 1 – Dictionnaire de données

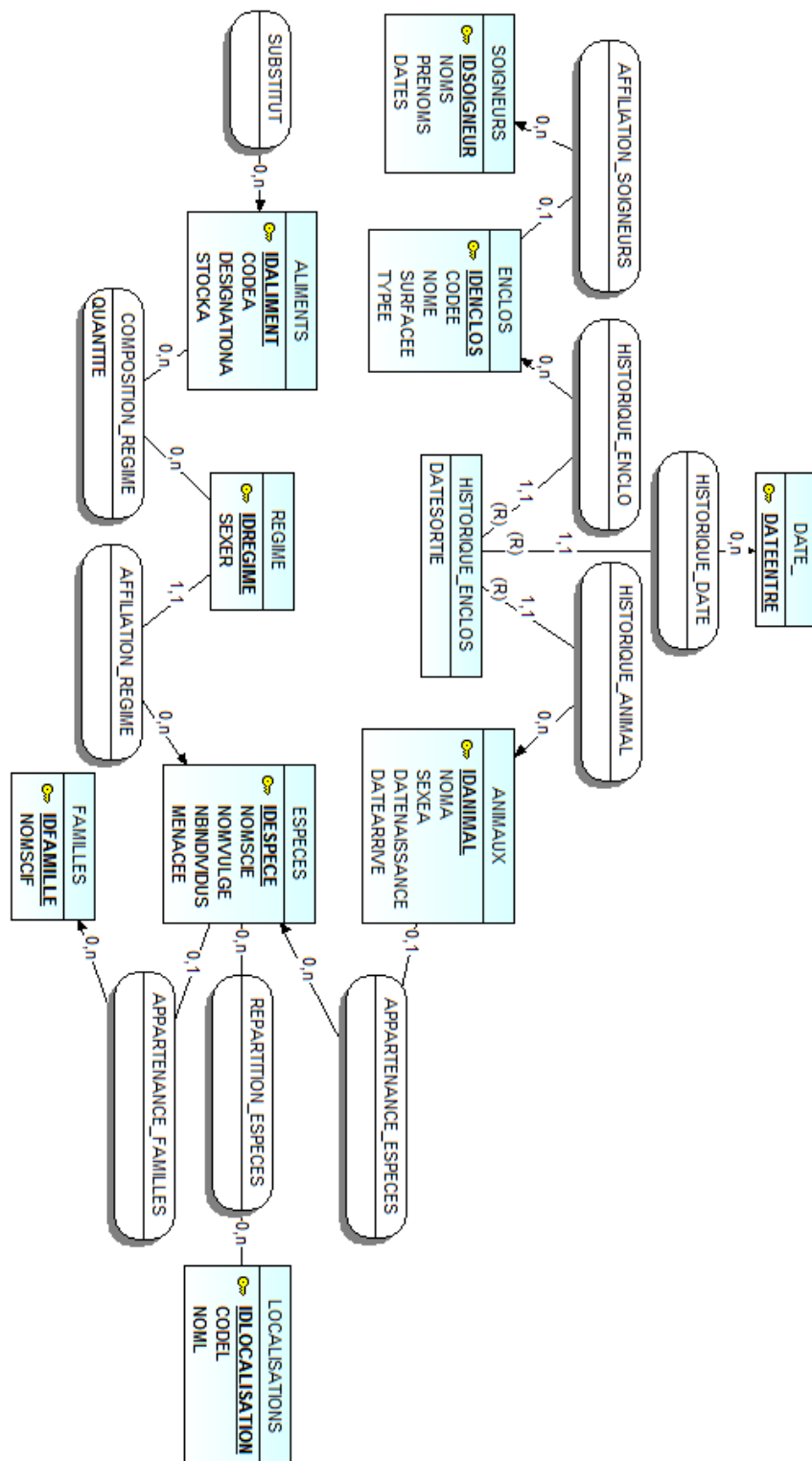


FIGURE 2 – Modèle Conceptuel de Données

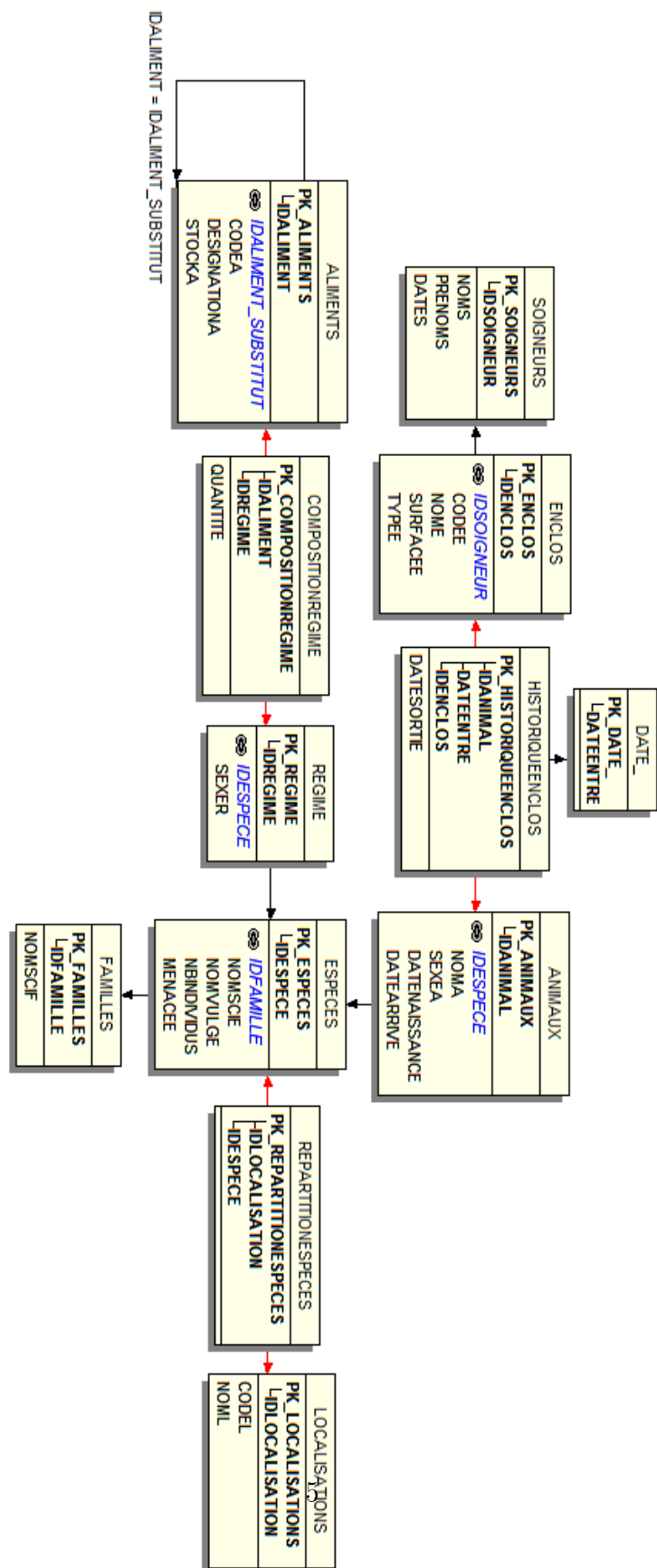


FIGURE 3 – Modèle Conceptuel de Données