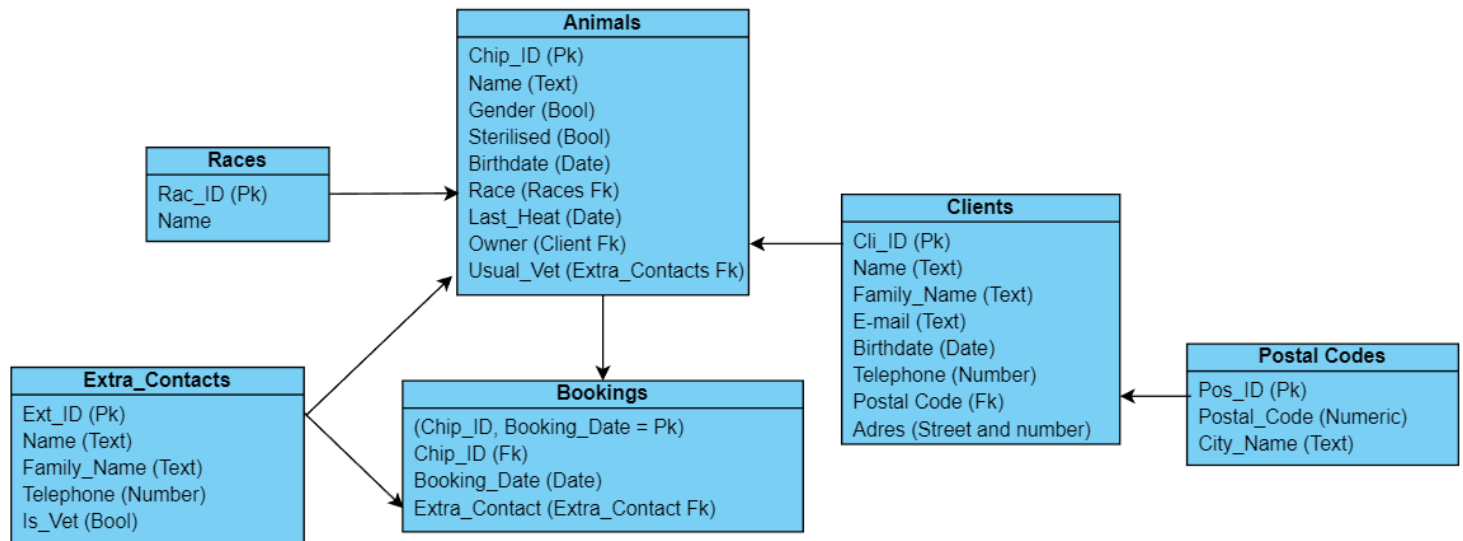


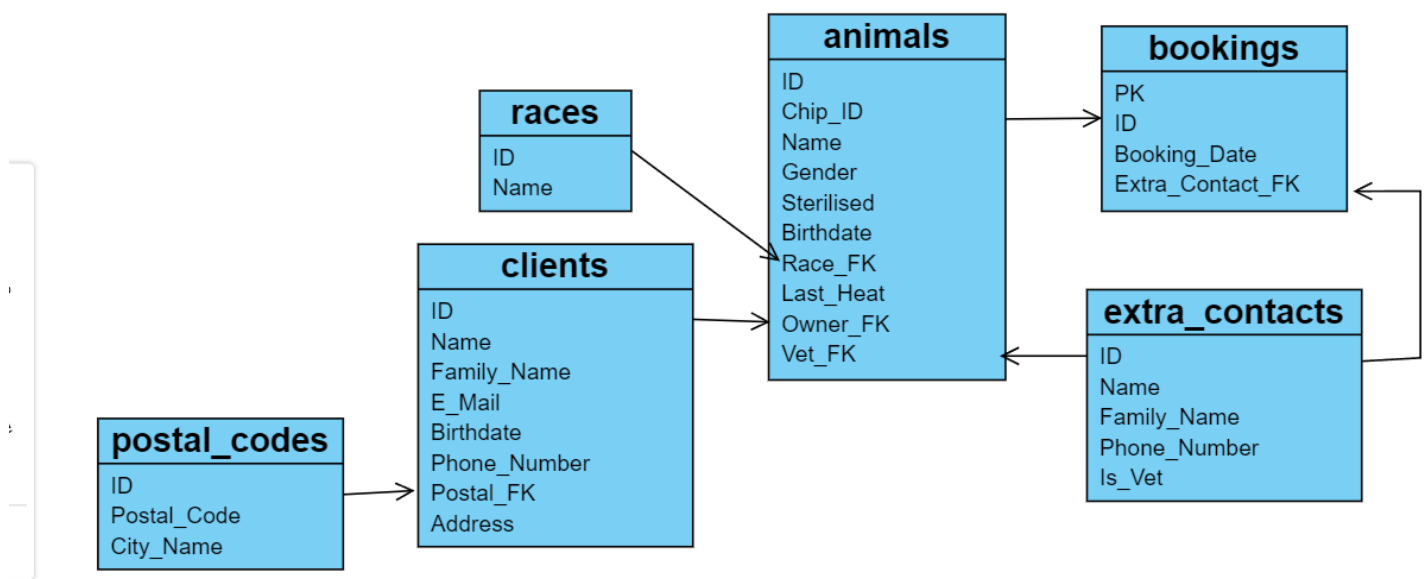
Examen 5IDV2 –Juin 2022  
Vroonen Kyran

## Schéma de la base de données :

Au début j'avais fait ce schéma pour la DB, mais pendant le développement je me suis rendu compte qu'il y avait des soucis.



J'avais rencontré des problèmes parce que je n'avais pas mis de ID dans bookings et animals, j'avais donc changé ça et ça a bien fonctionné. Mais c'est pour cela que les noms dans ces deux tables se ressemblent. Le ID dans bookings reprend le chip ID de l'animal (avec plus de temps je changerais ça pour faire référence à l'ID de animals mais j'ai laissé tel quel pour ne pas risquer de perdre trop de temps).



## Fonctionnalités :

On commence sur la page qui montre la liste des animaux et de là il y a un bouton pour afficher la liste de chaque table (ces boutons se trouvent dans public>index donc ils sont visibles sur chaque page). Dans chaque page de liste il y a une requête xhr pour récupérer les données automatiquement. De là on peut :

- Passer à la page pour ajouter un nouvel élément dans la base donnée,
- Supprimer un des éléments
- Ou modifier un élément déjà existant (à l'exception des races, et code postaux).

Sur les pages de modifications il y a également un bouton retour pour retourner à la page liste d'avant.

## Problèmes :

J'avais essayé d'utiliser le jQuery pour gérer les boutons des forms mais à ce moment-là je recevais une erreur disant "Uncaught SyntaxError: Unexpected token <" à la ligne 1:1 donc le "<" de mon !doctype html, j'avais essayé de régler ça mais finalement pour contourner cela j'ai utilisé dans les forms le post avec l'action vers par exemple /clients/store au lieu d'utiliser le JQuery.

Dans la liste des animaux il y a le ID de owner\_fk qui ne s'affiche pas. Quand je fais un dump il me montre bien les bonnes infos mais quand j'essaie de l'afficher il ne le fait pas. Je n'ai pas trouvé une solution.

Quand on fait un refresh de la page après avoir ajouté un élément dans une table il répète les actions. J'estime que c'est la même chose après une modification mais là on ne le voit pas visuellement. Je pense que le problème vient du fait que je ne vide pas mon url après une commande et aussi que si j'aurais réussi à utiliser le JQuery au lieu des forms pour gérer les modifications et ajouts je n'aurais pas ce problème.

Pour les ajouts j'avais un problème avec les Animals et les Bookings parce que la structure que j'utilisais chèque si le id existait et faisait une update au lieu d'un store s'il existait et de la manière que j'ai fait le id existait chaque fois. La première chose au quel j'ai pensé c'était d'ajouter à ces tables une colonne ID, c'était du chipot mais ça a fonctionné.

Les fonctions CRUD fonctionnent il y a quelque jour... mais maintenant il est bourré d'erreurs sur certains pages. Il est trop tard pour y travailler.

## Routes :

Options(animals, bookings, clients, extraContacts, postalCodes, races)

Foreach(Options as object)

POST /object/destroy

Paramètre	Type	Status	Description
id	int	Obligatoire	Id correspondant à l'objet à supprimer
Réponse	Type	Status	Description
Succès -> update le html avec le résultat	html	200 - OK (done)	Réponse reçue si l'objet est supprimé correctement
Erreur -> l'affiche sur l'écran	warn	404 - Not found (fail)	Réponse reçue si l'id ne correspond à aucun enregistrement / l'id n'est pas précisé

## POST /object/store

Paramètre	Type	Status	Description
Tous les inputs du form concerné	array	Obligatoire	Infos qu'on veut ajouter dans la table liée à l'objet
Réponse	Type	Status	Description
Succès -> update le html avec le résultat	html	200 - OK (done)	Réponse reçue si les infos sont ajoutées correctement a la table liée à l'objet
Erreur -> l'affiche sur l'écran	warning	404 - Not found (fail)	Réponse reçue si une erreur s'est produite

## POST /object/edit

Paramètre	Type	Status	Description
Tous les inputs du form concerné	array	Obligatoire	Infos qu'on veut changer pour l'objet concerné
Réponse	Type	Status	Description
Succès -> update le html avec le résultat	html	200 - OK (done)	Réponse reçue si les infos sont changées correctement à la table liée à l'objet
Erreur -> l'affiche sur l'écran	warning	404 - Not found (fail)	Réponse reçue si une erreur s'est produite

## POST /object/show

Paramètre	Type	Status	Description
id	int	Obligatoire	Id correspondant à l'objet à afficher
Réponse	Type	Status	Description
Succès -> update le html avec le résultat	html	200 - OK (done)	Réponse reçue si l'objet est affiché correctement
Erreur -> l'affiche sur l'écran	warn	404 - Not found (fail)	Réponse reçue si l'id ne correspond à aucun enregistrement / l'id n'est pas précisé