

Rapport projet BDD

Sujet: SPA

Le but de ce projet est de créer une base de données contenant des informations répondant au cahier des charges donnée et de mettre en place un site qui fera usage de cette base de données. Notre Sujet, la SPA, consiste en la création d'un site qui permettrait aux employés de la SPA, un ensemble de refuges pour animaux de pouvoir accéder aux informations des refuges, des animaux, des employés, et de pouvoir insérer de nouvelles données, nouveau pensionnaire, nouvel employé, nouveau transfert d'un animal, nouveau soin, adoption... De plus, il nous est demandé de faire une partie pour les particuliers pour leur permettre d'accéder aux informations sur les refuges et animaux, ils ne pourront pas cependant ajouter de nouvelle information contrairement aux employés ou accéder aux informations sur les employés.

Sommaire

Manuel d'utilisateur :	2
I - Etape 1 : Modélisation	3
II - Etape 2 : Création de la base de données	3
III - Description technique	5
A - Session	5
B - Navbar	5
C - Requêtes SQL	5
IV - Piste d'amélioration	6
V - Organisation et répartition du travail	7

Code couleur:

En violet: les noms de fichier

En rouge: les noms de table de la base de données

En vert: les noms de fonctions

Manuel d'utilisateur :

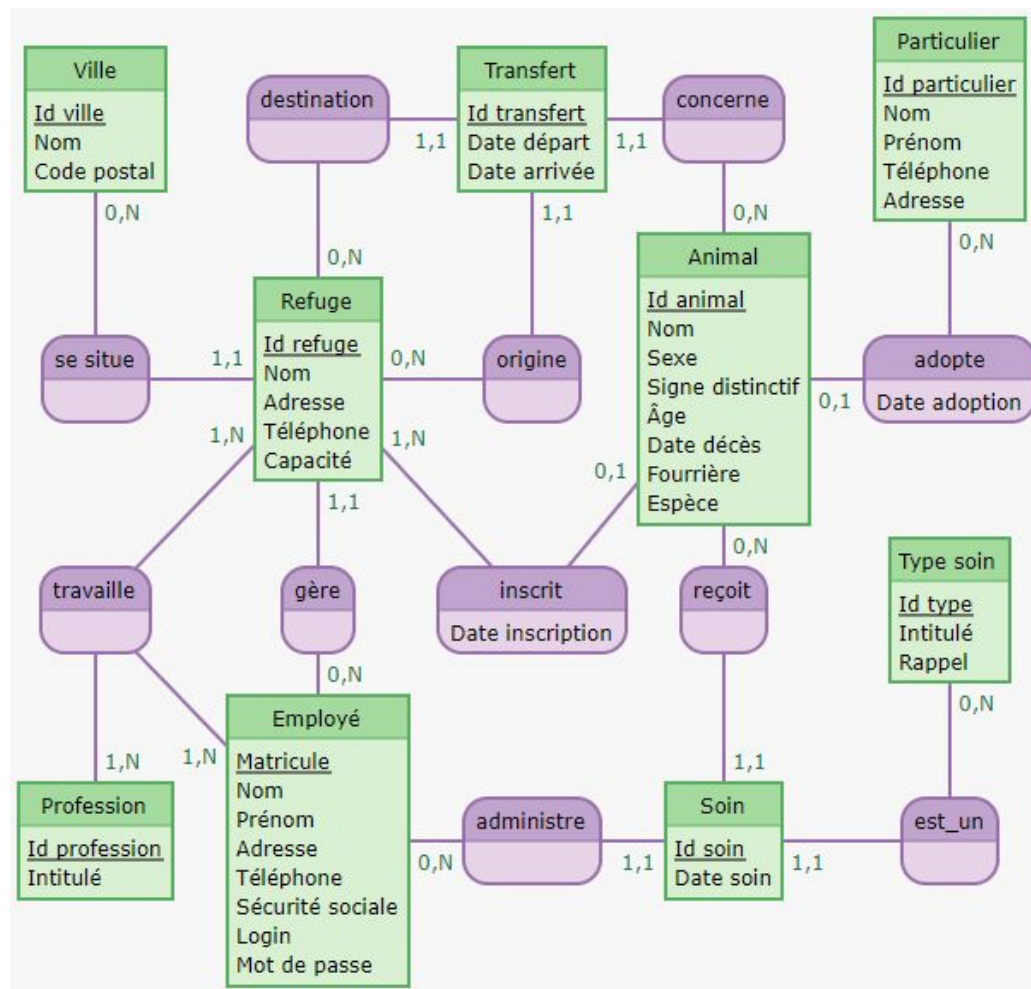
<https://etudiant.u-pem.fr/~kquach01/SPA/>

Pour commencer, veuillez extraire les fichiers contenue dans le fichier zip puis exécutez le fichier *SPA.sql* dans votre base de donnée ou bien dans phpPgadmin pour importer les tables nécessaires au fonctionnement du site. Ensuite, ouvrez le fichier *connection.inc.php* à l'aide d'un éditeur et remplacez les données des variable \$user et \$pass par les login et mot de passes de votre base de données. Cela fait, déposer l'ensemble des fichiers dans un espace dédié au développement de page Web et y accéder. Vous vous trouvez maintenant sur la page d'accueil de notre site. Vous n'avez plus qu'à vous connecter en utilisant l'un des login et mot de passe ci joint . Une fois connecté, vous pouvez naviguer sur le site grâce aux boutons. Utilisez la barre de navigation pour revenir en arrière sur les pages précédentes. Sur la page listes des animaux, les pages dédiées à chaque animal et la liste des employés, vous pouvez insérer de nouvelle information en cliquant sur le bouton dédié sur chacune de ces pages, ce qui vous donnera accès à un formulaire à remplir. Enfin, à droite de la barre de navigation se trouve un bouton de déconnexion qui, comme son nom l'indique, permet de se déconnecter et de revenir à l'écran d'accueil. Par ailleurs, vous pouvez vous connecter avec le login et mot de passe d'employé que vous ajouter via le formulaire du site.

login	mdp
jchirac	chiracj
sbourpalet	bourpalets
ckent	kentc
bdupuit	dupuitb
bjean	jeanb
ppierre	pierrep
cader	aderc
prose	rosep
jdujardin	dujardinj
rjacques	jacquesr
ymauri	mauriy
cchateau	chateauc
mrobert	robertm
gtimbre	timbreg
mpanne	pannem
vmona	monav
pgrégorie	grégoriep
amarine	marinea
jparain	parainj
jjonathan	jonathanj

I - Etape 1 : Modélisation

L'étape 1 consistait à modéliser une base de données sous forme de schéma entité-relation répondant aux attentes du cahier des charges. Ci joint le schéma que nous avons suivi pour notre base de données (correction donnée par notre chargé de TD) :



II - Etape 2 : Création de la base de données

L'étape 2 était dédiée à la création de la base de données directement à partir de la modélisation de celle-ci via le modèle relationnel. Cette étape, si la précédente était bien faite, était assez simple et rapide à faire. Il s'agissait simplement de traduire le modèle relationnel en créant toutes les entités et associations qu'on avait défini dans celui-ci, remplir les tables en pensant aux différents cas possibles et surtout mettre en place les contraintes correctes sur les attributs de la table pour que le tout soit cohérent avec notre sujet.

Voici donc le modèle relationnel utilisé :

- Ville(Id_ville, Nom, Code_postal)
- Transfert(Id_transfert, Date_départ, Date_arrivée, #Id_refuge_destination, #Id_animal, #Id_refuge_origine)
- Particulier(Id_particulier, Nom, Prénom, Téléphone, Adresse)
- Refuge(Id_refuge, Nom, Adresse, Téléphone, Capacité, #Id_ville, #Matricule)
- Animal(Id_animal, Nom, Sexe, Signe_Distinctif, Âge, Date_décès, Fourrière, Espèce, #Id_particulier, Date_adoption, #Id_refuge, Date_inscription)
- Type_soin(Id_type, Intitulé, Rappel)
- Profession(Id_profession, Intitulé)
- Employé(Matricule, Nom, Prénom, Adresse, Téléphone, Sécurité_sociale, Login, Mot_de_passe)
- Soin(Id_soin, Date_soin, #Id_type, #Id_animal, #Matricule)
- travaille(#Id_profession, #Id_refuge, #Matricule)

Dans Refuge:

- Id_ville fait référence à Id_ville dans Ville.
- Matricule fait référence à Matricule dans Employé

Dans Transfert:

- Id_animal fait référence à Id_animal dans Animal
- Id_refuge_destination fait référence à Id_refuge dans Refuge
- Id_refuge_origine fait référence à Id_refuge dans Refuge

Dans Animal:

- Id_particulier fait référence à Id_particulier dans Particulier
- Id_refuge fait référence à Id_refuge dans Refuge

Dans Soin:

- Id_type fait référence à Id_type dans Type_soin
- Id_animal fait référence à Id_animal dans Animal
- Matricule fait référence à Matricule dans Employé

Dans travaille:

- Id_profession fait référence à Id_profession dans Profession
- Id_refuge fait référence à Id_refuge dans Refuge
- Matricule fait référence à Matricule dans Employé

Quelques explications sur les contraintes :

- Les mots de passe ne sont pas uniques, deux utilisateurs peuvent avoir le même mot de passe.
- Les codes postaux ne sont pas uniques non plus.
- Un animal peut avoir un trait distinctif ou non.
- Un animal peut ne pas avoir de date d'adoption et donc de particulier, l'animal n'est pas encore adopté.

- Chaque refuge ne peut avoir qu'un seul gérant.
- Date de rappel en jours, c'est une période de temps (ex: 90 jours). Mais la base de données ne gère pas la prochaine date d'elle-même.
- On ne peut pas gérer la capacité maximale d'un refuge. Il faudrait pouvoir bloquer le nombre d'insertions de lignes par rapport à la capacité.

III - Description technique

A - Session

Notre site utilise des sessions pour permettre aux employés de se connecter. En entrant le login et mot de passe sur la page de connection [index.php](#) , le formulaire renvoie sur un autre fichier [auth_employe.php](#) dans lequel on vérifie que les informations saisie, login et mot de passe sont contenue dans la base de données, dans la table [employe](#), et qu'il correspondent tout les deux à un employé. Si c'est le cas, une session est créée et le fichier renvoie l'utilisateur vers [acceuil_employe.php](#) , page qui servira de page d'accueil pour les employés connectés.

B - Navbar

En ce qui est de la barre de navigation, elle est codée dans le fichier [navbar.php](#) et on l'inclut dans chaque fichier qui compose le site. Cette "navbar" affiche un historique des pages visité depuis la page d'accueil et , en cliquant sur le nom d'une des pages précédente, permet d'y retourner immédiatement dessus. Quant à son fonctionnement, on crée une variable de session à chaque page qu'on visite contenant les informations nécessaires (numéro du refuge, nom de l'animal,...), le fichier [navbar.php](#) vérifie si ces variables existent via la fonction [isset](#) et, si c'est le cas, crée un bouton renvoyant sur ces pages sur la navbar. De plus, à chaque fois qu'on accède à une page, on vérifie si des variables pour des pages suivantes existent et si c'est le cas, on les détruit avec [unset](#) . On précisera par ailleurs que cette navbar a été faite à l'aide *bootstrap*.

C - Requêtes SQL

Les requêtes SQL de notre site internet sont principalement des requêtes de sélection de tables, récupérant des lignes de certaines tables de la base de données (quand la donnée est présente dans la table), pour ensuite être utilisées dans des requêtes d'insertions pour étoffer la base de données.

On a par exemple besoin à 4 reprises d'utiliser des INSERT INTO, pour permettre d'ajouter des nouveaux employés et pensionnaires dans les différents refuges, ou encore ajouter les différents soins qu'un animal a pu recevoir au cours de son séjour

dans le refuge, et aussi tenir un historique des transferts de l'animal dans chaque SPA pour avoir un historique de déplacements.

Exemple de requête que notre site web utilise :

```
<?php
$req_id = "SELECT id_animal FROM animal WHERE nom = '$_SESSION[\"animal\"]'";
$res_id = $cnx->query($req_id);
$id = $res_id->fetch();

$req_insert_soin = "INSERT INTO soin VALUES('$_POST[\"id_soin\"]',
'$_POST[\"date_soin\"]', '$_POST[\"type_soin\"]', '$_POST[\"id_animal\"]', '$_POST[\"soigneur\"]')";
$cnx->exec($req_insert_soin);

$_SESSION["insert_soin"] = "ok";
header('location: animal.php');
?>
```

Dans ce cas là, on veut pouvoir ajouter un soin effectué à un animal précis. Il faut donc d'abord récupérer l'id de l'animal en question, ça passe dans un premier temps par une requête SELECT. Une fois l'id récupéré et stocké dans une variable on va pouvoir s'en servir dans la requête INSERT INTO, les autres variables de la requête étant les informations que l'utilisateur remplit via le formulaire du site web.

```
<?php
$req2 = "SELECT id_animal,nom,sexe,signe_distinct,age,date_deces,fourriere,espece,id_refuge,
date_inscription,id_particulier,date_adoption FROM animal WHERE id_refuge = '$_SESSION[\"refuge\"]' ORDER BY id_animal";
$res2 = $cnx->query($req2);
?>
```

Autre cas, ici le but est l'affichage de la table **animal** on récupère donc chaque ligne de la table avec la requête SELECT qu'on stocke et qu'on affiche une par une avec un while juste après. La même chose est utilisée pour l'affichage des animaux ou des employés pour un refuge en particulier, avec une condition supplémentaire dans la requête pour correspondre au refuge pour lequel on veut les différentes listes.

IV - Piste d'amélioration

Le site est, en l'état, incomplet, en effet, il manque la partie dédiée aux particuliers. Cela est notamment dû à un manque de temps dû au fait que le serveur de l'université était HS pendant un certain temps durant les vacances.

Mis à part la partie pour les particuliers, d'autres idées que nous avons voulu implantée, mais ne l'avons pas fait dû aux mêmes raisons cité plus haut, inclus la possibilité au employés de saisir une adoption qui mettra à jour les informations de

l'animal, notamment l'identifiant de son particulier et sa date d'adoption dans la table **animal**, informations qui restent vides tant que l'animal n'a pas été adopté. Dans le même genre, on pourrait inclure un tableau à jour des animaux adoptables et ceux déjà adoptés pour plus de clarté. De la même manière, nous avons aussi voulu donner aux employés la possibilité de saisir un nouveau transfert qui mettra à jour le refuge dans lequel l'animal est abrité actuellement. Une différence par rapport à l'ajout d'un ancien déplacement que nous avons implanté permettant de saisir un transfert qui ne modifie pas le refuge de l'animal.

On remarquera aussi que l'insertion d'un animal ne marche pas si tous les champs du formulaire ne sont pas saisis, cela inclut donc la date de décès, l'identifiant d'un particulier et la date d'adoption ainsi qu'un signe distinctif.

De plus, il aurait été plus intéressant de donner le type *serial* à nos attribut id ou matricule pour les tables de nos base de données. Cela aurait permis d'insérer des animaux sans avoir à aller vérifier quel(les) id / matricules sont déjà utilisé(es). Nous n'avons malheureusement pas réussi à changer leur type en serial, les ayant déjà définie en tant que int durant l'étape 2.

V - Organisation et répartition du travail

Nous avons prévu au départ de travailler ensemble par vocal sur Discord. Mais en raison de la panne du serveur de l'université, cela n'a pas été possible pour le début du projet. Nous avons ensuite essayé de mettre en place un serveur et base de données local pour essayer mais n'avons pas réussi avant pas mal de temps. Alexis étant indisponible durant quelques jours, et Kevin ayant réussi à mettre en place un serveur et base de données entre temps (en utilisant wamp), Kevin a commencé à travailler sur l'architecture du site, les structures des pages et des redirections, les tables et les sessions. Une fois qu'Alexis a été de nouveau disponible, nous avons alors continuer ensemble sur Discord sur les formulaire d'insertion, les insertions en elles même dans les tables de la base de données, l'affichage des tableaux dynamiques, l'ajout de certaine information sur les pages telle que les transferts et soins pour les animaux et le débogage.