

Projet
Système de gestion d'un centre équestre

Partie 1 : Le cahier des charges

Un centre équestre associatif situé en région nouvelle aquitaine vous sollicite afin que vous mettiez en place un système de gestion informatisé permettant de gérer les activités du centre équestre.

L'objectif est de proposer un logiciel Python (en mode texte) permettant de satisfaire au cahier des charges ci-dessous. Celui-ci se connectera à une base de données de type sqlite3. Pour simplifier au maximum, nous imaginerons que le logiciel sera accessible physiquement uniquement sur un ordinateur situé au centre équestre.

Le logiciel sera accessible à 3 publics différents :

- Les gestionnaires du centre équestre (Régis X, Cécilia Y et Leslie Z)
- Les membres de l'associations qui utiliseront les chevaux du centre
- Les membres de l'association qui sont propriétaires d'un ou plusieurs chevaux et qui laisse en pension leur chevaux dans ce centre équestre.

Le but principal du centre équestre, vous l'aurez compris, est d'aider les gestionnaires à mettre en place un système de facturation afin de les aider dans leur quotidien.

Le centre équestre s'entend sur plusieurs hectares :

- Le bâtiment principal est constitué de vestiaires, d'une pièce avec 3 bureaux (un pour chaque gestionnaire), d'une salle de lavage avec eau chaude pour les soins des chevaux, d'une aire de pansage abritée, d'une sellerie pour y entreposer les selles des chevaux du club et des propriétaires et d'une partie box où vivent les chevaux. Il existe 30 box disponibles qui permettent à certains chevaux de vivre avec tout le confort imaginable. Le bâtiment est chauffé l'hiver, les box sont nettoyés quotidiennement, le foin est à volonté avec des compléments alimentaires (en option) donnés tous les soirs. Les chevaux en box ne vivent pas toute la journée dans le bâtiment, ils sont sortis et mis au pré de 8h à 17h tous les jours de l'année.
- Il y a en tout 55 hectares de pré afin d'accueillir les chevaux du centre et des chevaux de propriétaires. Il y a 5 champs de 11 hectares chacun.
- Pour le bien-être animal, il est obligatoire d'octroyer 1,5 hectares par cheval. Au bout de 5 jours il doit être changé de pré.
- Il existe un manège couvert où les cours sont donnés. Attention, sa surface est petite, l'association ne peut donc accueillir au maximum que 10 chevaux par cours.
- Il existe un autre manège non couvert réservé aux propriétaires pour s'entraîner avec son cheval. Il ne peut y avoir qu'un seul cheval par créneau d'une heure. Un propriétaire doit réserver le créneau horaire dans l'application. Il ne peut pas réserver plus d'une heure par jour. Créneaux disponibles : 17h-18h et 18h-19h.

L'association est propriétaire de 18 chevaux. Pour simplifier la gestion de l'application, nous excluons la possession de poneys. Sur ce cheptel, 10 sont des hongres et 8 sont des juments. 2 hongres sont « à la retraite », c'est-à-dire qu'ils ne peuvent plus être mobilisés pour des cours et sont rentrés au box chaque soir pour service rendu.

Afin d'augmenter les recettes du centre équestre, des propriétaires peuvent laisser en pension leur chevaux tout au long de l'année ... dans la limite de capacité d'accueil. Le propriétaire à le choix entre :

- Une pension pré (le cheval vit toute l'année dehors)
- Une pension pré/box (le cheval est rentré tous les soirs et sorti le matin)
- Une option à 5€ par jour est possible pour la pension pré/box : le palefrenier soigneur donne des compléments alimentaires quotidiennement (chevaux de compétition, etc ..)

Afin d'éviter des naissances incontrôlées, dans le contrat passé entre le propriétaire du cheval et le centre équestre, les étalons ne sont jamais mis dans le même pré que des juments.

Pour obtenir des prix particulièrement attractifs, les chevaux de propriétaire peuvent être eux aussi utilisés par le centre équestre. Cependant le propriétaire peut voir la fréquence hebdomadaire d'utilisation de son cheval et peut s'il le souhaite suspendre temporairement le prêt de son cheval afin de préserver la santé mentale de l'équidé.

Les cours

Il y a 3 cours d'une heure par jour du lundi au samedi.

- Le premier créneau horaire est de 17h30 à 18h30 (réservé aux cours particuliers) animé par Cécilia.
- Le deuxième créneau horaire est de 18h30 à 19h30 animé par Régis.
- Le troisième créneau horaire est de 19h30 à 20h30 animé par Leslie¹.

L'application ne gèrera pas les jours fériés, ni les années bissextiles, ni les vacances scolaires.

Les prix pratiqués

- 20€ pour une heure de cours collectif en prenant le cheval du club (assignation aléatoire)
- 12€ pour heure de cours collectif (pour les propriétaires qui viennent avec leur chevaux)
- 25€ l'heure de cours particulier (pour les propriétaires)
- 30€ l'heure de cours particulier (en utilisant le cheval du club)
- 200€ par mois pour une pension pré (foin à volonté)
- 350€ par mois pour la pension pré/box (foin à volonté + alimentation)

¹ Uniquement des cours pour Adultes, car Leslie n'aime pas se répéter durant une séance...

Les objectifs de l'application

Un membre de l'association (propriétaire ou non) peut réserver un cours dans les créneaux disponibles. Il a un récapitulatif à chaque instant de sa facture qu'il devra payer à la fin du mois.

Lors de la réservation d'un cours, un propriétaire ne peut être assigné qu'au cheval dont il est propriétaire. S'il possède plusieurs chevaux, il choisit quel cheval il va monter. Un membre non-propriétaire peut choisir son cheval (dans les chevaux disponibles).

Un propriétaire de cheval peut voir dans quel pré a été mis son cheval dans le mois, peut savoir quelle est la fréquence d'utilisation de son cheval par le club et bloquer l'utilisation du cheval par le centre équestre.

Bien entendu, si un cheval est utilisé par son propriétaire pour suivre un cours ou pour son usage personnel, il ne peut être associé à une autre personne pour cette heure de cours.

Le gestionnaire peut voir les chevaux de l'association les plus utilisés et les laisser au repos si besoin.

Les nouveaux membres de l'association sont intégrés dans le système par le gestionnaire, ainsi que les nouveaux propriétaires.

Un propriétaire peut retirer son cheval du centre équestre à tout moment.

Les gestionnaires doivent pouvoir voir l'ensemble du cheptel, avec leur assignation du pré et leur fréquence d'utilisation, s'ils ceux-ci doivent être rentré aux box le soir, etc ...

Compléments

Voici la liste des 18 noms de chevaux :

Hongres :

1. Écuyer d'Or
2. Tempête du Vent
3. Loup des Champs
4. Ciel d'Argent (en retraite)
5. Prince du Soleil
6. Mystère Noir
7. Roi de l'Étoile
8. Élan de Feu
9. Baron des Bois (en retraite)
10. Titan des Prairies

Juments :

1. Perle du Lac
2. Reine de l'Aube
3. Étoile Filante
4. Fleur de Lune
5. Belle de Nuit
6. Lumière d'Or
7. Vénus des Cieux
8. Douce Mélodie

Voici les cinq noms de prés où vivent les chevaux :

1. Prairie des Écuyers
2. Champ des Sabots d'Or
3. Prairie des Cieux
4. Verger des Étoiles
5. Plaine des Cavaliers

Partie 2 – La gestion de projet

Les objectifs du projet sont décrits par un cahier des charges détaillé qui vous a été présenté précédemment. Ce cahier des charges pourra être réajusté ou affiné, par l'équipe de projet en fonction de votre analyse.

Dans la gestion de projet, il est important de planifier des moments d'échanges au sein de l'équipe, appelés revues de projet.

1. Objectifs intermédiaires n°1 du 9 janvier 2025

Analyser le besoin :

1. Analysez **finement** le cahier des charges. Déterminer les besoins. En déduire les fonctionnalités demandées. Produire une liste des fonctionnalités structurée.
2. Schématiser une arborescence de "navigation" entre les pages (*voir section les rendus*)
3. Lister chaque "écrans" prévus présentant la fonctionnalité en français (*voir les rendus*).
4. Produire une première version du Schéma entité-association

2. Première revue de projet (Fin de la première semaine)

Les objectifs de cette revue de projet sont :

1. Rechercher les premières pistes de solutions (découpage du problème).
2. S'organiser :
 - Partage des tâches ;
 - Amorce de planification.

Cette première revue de projet se fait à la fin de la première semaine avec l'équipe au complet.

Objectif : S'assurer que le groupe a effectivement compris le problème à résoudre et qu'il a mis en place une organisation adéquate.

À l'issue de cette revue de projet, l'équipe doit rédiger un relevé de conclusion. L'existence et la pertinence de ce document peut entrer dans l'évaluation du projet.

Coup de pouce :

- Commencer au début du projet par réaliser les tâches simples.
- Ne devenez pas « expert » d'un module. Chacun doit pouvoir travailler sur le module de son camarade.

3. Revues de projet suivantes

Durant le projet, chaque élève effectue sa tâche en autonomie avec l'aide ponctuelle du professeur. Le code qu'il crée doit pouvoir s'intégrer dans le projet global de l'équipe. Des moments de rencontres entre les membres de l'équipe, appelés revues de projet, doivent donc avoir été planifiés avec ou sans le professeur.

Chaque début et fin des séances doivent être consacrés à un moment, même court, durant lequel l'équipe identifie le travail fait et celui restant à réaliser. L'équipe peut utiliser un cahier de suivi résumant les bilans de ces moments.

À la fin du projet, l'équipe doit s'assurer de l'interopérabilité des différentes briques (par exemple : les noms de variables ou de fonctions sont-ils cohérents ? partagés par tous ?).

Objectifs intermédiaires n°2 du 13 janvier 2025

Document simplifié d'architecture générale du projet :

- Lister les modules nécessaires
- Lister les noms des fonctions (et des paramètres)

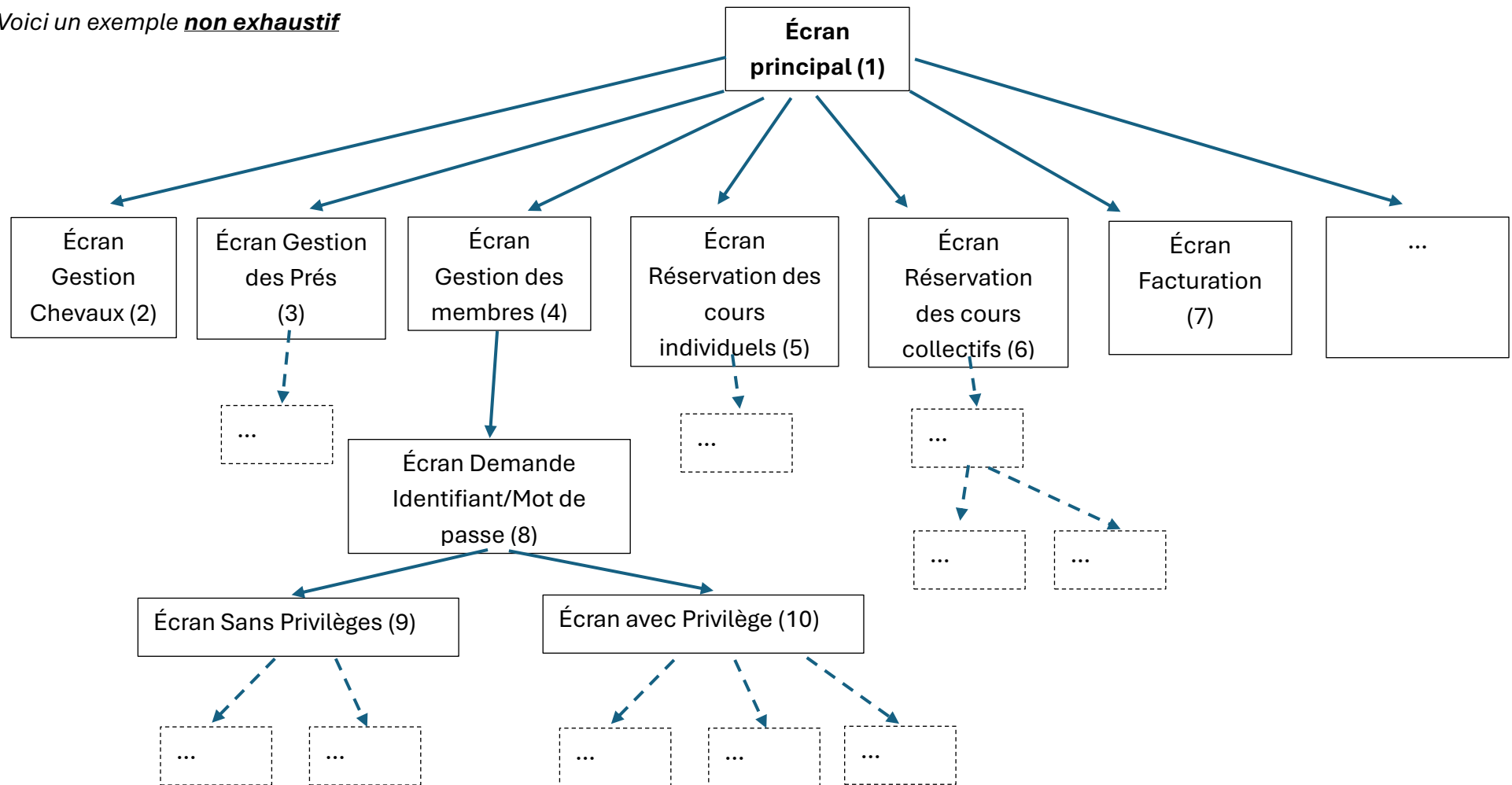
Objectifs intermédiaires n°3 du 16 janvier 2025

- Envoyer l'ensemble des tests unitaires : Le répertoire test zippé.

Partie 3 – Les rendus

1. Exemple d'arborescence de "navigation" entre les pages

Voici un exemple ***non exhaustif***



2. Exemple de liste de chaque "écrans" prévus présentant la fonctionnalité (en français)

Voici un exemple ***non exhaustif***

Écran principal (1)

Cet écran affichera les éléments suivants :

Gestion des chevaux

Gestion des prés

...

Charger le schéma de la base

Charger les données en base

Arrêt du programme

Écran gestion des chevaux (2)

Permet de lister tous les chevaux (cheval_id, nom, genre_id, statut_id) sans critères spécifiques.

Ajouter un cheval (son nom, son genre, son statut).

Modifier les données d'un cheval par son identifiant (*clé primaire*).

Supprimer la gestion d'un cheval par son identifiant (*clé primaire*).

3. Exemple de document simplifié d'architecture générale du projet

Nom du module : Gestion cheval		Niveau de difficulté Simple/Moyen/Complexe	Tâche affectée à
Section Gestion de la base de données :			
lister_tous_chevaux()			
lister_genre()			
lister_statut()			
insérer_cheval(...)			
...			
Section Gestion des écrans (Affichage) :			
ecran_gestion_chevaux()			
ecran_liste_tous_chevaux()			
ecranajouter_un_cheval()			
...			

4. Le rendu final

Structure du rapport

1. Page de titre

- Titre du projet
- Noms des élèves du binôme

2. Sommaire

- Liste des parties avec numéros de page

3. Introduction

- Contexte du projet
- Objectifs

4. Décomposition du projet

- Présentation du projet en sous-problèmes
- Justification de cette décomposition

5. Organisation du travail

- Répartition des tâches au sein du binôme
- Qui fait quoi ? (Détails des responsabilités de chaque membre)

En dehors de l'affectation de chaque fonction à un membre du binôme, expliquer la décomposition des tâches annexes au sein de votre binôme et la méthode de travail.

- Planning et organisation temporelle
- Moments d'échanges et de synchronisation
- Intégrer le relevé de conclusion (première revue de projet)
- Intégrer le cahier de suivi (revues de projet suivantes)

6. Méthodologie de développement

- Expliquez votre processus de mise au point des programmes
- Méthodes de validation des programmes
- Méthodes de tests (tests unitaires, tests d'intégration, etc.)
 - Description des tests effectués
 - Avez-vous fait des tests d'intégrations, comment les avez-vous intégrés ?

7. Développement

- Intégrer le schéma entité / association définitif
- Intégrer l'arborescence de "navigation" entre les pages définitive
- Pour des parties spécifiques (que vous aurez choisies) expliquer les solutions apportées
 - Description détaillée des solutions apportées à chaque sous-problème
 - Structures de données choisies
- Difficultés rencontrées et solutions apportées

8. Analyse et réflexion

- Analyse critique de la solution développée
- Comparaison avec les objectifs initiaux
- Réflexion sur le processus de développement et l'organisation du travail

9. Conclusion

- Apprentissages réalisés (techniques et organisationnels)
- Perspectives d'amélioration ou d'extension du projet

5. Le barème (indicatif)

Fonctionnalités (7 points)

- Palier 1 (0~2/7) : Le programme correspond à peu près aux spécifications demandées
- Palier 2 (2~4/7) : Le programme correspond aux spécifications demandées
- Palier 3 (4~6/7) : Le programme ne présente aucun bug et correspond aux spécifications demandées
- Palier 4 (6~7/7) : Des fonctionnalités supplémentaires ont été ajoutées

Algorithmes et code (6 points)

- Utilisation pertinente de structures conditionnelles, de boucles, de variables, etc.
- Utilisation correcte des différents types de valeurs et des fonctions de conversion quand nécessaire
- Décomposition fonctionnelle pertinente (chaque fonction est suffisamment petite, on évite la répétition de code)

Style du code (2 points)

- Utilisation de noms pertinents pour les variables et les fonctions
- Documentation pertinente des fonctions grâce à des docstrings
- Utilisation pertinente des commentaires

Rédaction du rapport (5 points)

Au travers du compte-rendu, les critères d'évaluation seront :

- L'organisation ;
- L'analyse des problèmes à résoudre ;
- La clarté de l'expression et des illustrations ;
- L'explicitation des méthodes et des choix effectués ;
- L'identification et le réinvestissement de notions vues en cours ;
- L'analyse critique des solutions.

Partie 4 – Technique

1. Liste des fonctionnalités à prévoir

Fonctionnalités
Ajouter un nouveau cheval - Ne pas oublier d'ajouter le genre "Etalon"
Modifier/Supprimer les données d'un cheval
Ajouter / Modifier / Supprimer un pré.
Placer un cheval dans un autre pré.
Pour un pré donné connaître le taux de remplissage.
Respecter le placement en fonction du taux de remplissage et de son genre (étalon vs Jument).
Pour un pré donné, lister les chevaux placés dans ce pré avec le nom du cheval, son genre et son statut. Avoir une information pour savoir si le club doit changer le cheval de pré.
Pour un propriétaire, réserver le créneau horaire dans le manège réservé aux propriétaires (réservation max 1h/jour).
Pour un mois donné, lister l'utilisation précise du manège réservé au propriétaire.
Ajouter un nouvel adhérent non-propriétaire (Adulte / Enfant)
Ajouter un nouvel adhérent propriétaire.
Associer un cheval à son propriétaire - choisir le type de pension associée (pré / pré-box / option complémentaire).
Activer ou non l'option d'autorisation de prêt par le club d'un cheval de propriétaire.
Un propriétaire peut avoir plusieurs chevaux.
Avoir un calcul de capacité d'accueil pour savoir si le centre est capable d'accepter un nouveau propriétaire avec un nouveau cheval (contrôle du taux de remplissage des prés).
Si ajout d'un nouveau champ alors le taux de remplissage redescend ...
Le propriétaire peut voir la fréquence hebdomadaire d'utilisation de son cheval (prise en compte de l'utilisation du propriétaire et de l'utilisation par le club).
Si le propriétaire accepte de prêter son cheval, possibilité de suspendre temporairement le prêt de son cheval.
Un propriétaire de cheval peut voir dans quel pré a été mis son cheval dans le mois.
Propriétaire - Prise d'un cours individuel - choix de la date + créneau.
Propriétaire - Prise d'un cours collectif - choix de la date + créneau.
Adhérent non-propriétaire - Prise d'un cours individuel - choix de la date + créneau.
Adhérent non-propriétaire - Prise d'un cours collectif - choix de la date + créneau.
Lors de la réservation d'un cours, un propriétaire ne peut être assigné qu'au cheval dont il est propriétaire. S'il possède plusieurs chevaux, il choisit quel cheval il va monter.
Lors de la réservation d'un cours, un membre non-propriétaire peut choisir son cheval (dans les chevaux disponibles).
Si un cheval est utilisé par son propriétaire pour suivre un cours ou pour son usage personnel, il ne peut être associé à une autre personne pour cette heure de cours.
Facturation mensuelle pour un propriétaire - Lister le coût de la pension pour chaque cheval en fonction du type de pension (Réduction de 20% si prêt au club)
Facturation mensuelle pour un propriétaire - Compléter le coût précédent avec le coût des cours réservés (individuel/collectif) avec date + créneau utilisé + nom du moniteur

Facturation mensuelle pour un adhérent - Lister tous les cours réservés (individuel/collectif) avec date + créneau utilisé + nom du moniteur
Facturation mensuelle avec tous les propriétaires et tous adhérents pour un mois donné.
Les gestionnaires doivent pouvoir voir l'ensemble du cheptel, avec leur assignation du pré et leur fréquence d'utilisation, s'ils ceux-ci doivent être rentré aux box le soir.
Les gestionnaires doivent pouvoir voir les chevaux de l'association les plus utilisés et les laisser au repos si besoin.

2. Logiciel préconisé

Schéma entité-association : <https://www.looping-mcd.fr/>

3. API à utiliser pour les interactions Python / Bases de données

Voir : [DB-API 2.0 interface for SQLite databases](#)

4. Objectifs et bonnes pratiques techniques

- Le logiciel est découpé en différents modules en fonction des grands axes fonctionnels.
- Les fonctions sont documentées par des docstrings.
- Appliquer le type hinting sur toutes les fonctions (déclarer les types utilisés).
- Les noms des variables sont correctement choisis.
- Des commentaires sont placés uniquement aux endroits judicieux (explication d'un algorithme complexe).
- Les problématiques d'affichages et de traitements sont différenciés. Il existe des fonctions qui gèrent uniquement l'affichage et des fonctions qui ne gèrent que les traitements.
- Respecter **les règles de nommage** des fonctions :
 - `def ecran_xxx() -> int` Pour la gestion des **interactions** avec les utilisateurs
 - `def lister_xxx() -> list` Pour **lister** des éléments de la bdd
 - `def inserer_xxx(...) -> None` Pour **insérer** des éléments de la bdd
 - `def modifier_xxx(...) -> None` Pour **modifier** des éléments de la bdd
 - `def supprimer_xxx(...) -> None` Pour **supprimer** des éléments de la bdd
- Pour les tests, utiliser pytest. S'inspirer de `test_gestion_chevaux.py`
- Mettre en place un fichier de tests unitaire par module testé.
- Les tests unitaires ne testent pas les fonctions d'affichage. Les tests unitaires tests uniquement les fonctions de traitement (calculs, requête SQL).
 - Exécuter la fonction
 - Contrôler le nombre de résultats renvoyés
 - Vérifier le 1^{er} et dernier élément renvoyé
 - Ne pas oublier de supprimer l'impact du test unitaire ...
- **Un test unitaire doit pouvoir se répéter à l'infini** sans intervention humaine.
- **Un test unitaire teste une fonction et une seule** (*on isole le cas à tester*) :

Exemple du cas `supprimerXXX()` :

- Insérer par requête SQL direct une donnée temporaire (*ne pas utiliser une fonction de votre programme qui supprime afin d'éviter un éventuel effet de bord de cette fonction*).
 - Appeler la fonction `supprimerXXX()`.
 - Tester par requête SQL direct que la donnée n'existe plus.
-
- Une table ne stocke jamais le résultat de calculs. Les résultats sont calculés à la demande.
 - La programmation objet peut être utile pour aider à stocker des données à calculer.