

**Rapport Projet NF 18**

[Gestion du musée du Louvre]



Jinshan GUO

Qi XIA

Jules YVON

Bastien LE CALVE

UTC P20 Group 5

Table des matières

1. Rappel du Sujet1 / 15
2. MCD - Cardinalité2 / 15

* MCD
* Détail des cardinalités et des héritages

1. MLD normalisation4 / 15

* Clés primaires
* Contraintes
* Vue

1. Application Python7 / 15

* Introduction
* Connexion et Déconnexion avec la BD
* Login
* Interface du système
* Fonctions réalisées
* Afficher tous les enregistrements
* Insertion
* Mise à jour
* Suppression
* Recherche vague et précise
* Résultat de fonction statistique

1. Amélioration15 / 15
2. Conclusion15 / 15

I. Rappel du Sujet

L’objectif de notre projet est de concevoir une base de données pour le musée du Louvre à Paris. Cette base de données a pour objectif d’aider dans la gestion du musée, que ce soit pour les œuvres, des guides, ou encore les expositions du musée.

Nous avons à gérer de nombreux éléments pour notre base de données :

Pour les œuvres, nous devions faire intervenir celles du musée mais également celles de musées extérieurs que nous avons emprunté pour des expositions temporaires. Nous assurons également le suivi des œuvres que nous avons prêtées ainsi que celles que nous avons empruntées. La restauration des œuvres est également un facteur à prendre en compte.

Pour les expositions, nous en avions 2 types, les permanentes avec les œuvres du musée, les

temporaires avec les œuvres des autres musées.

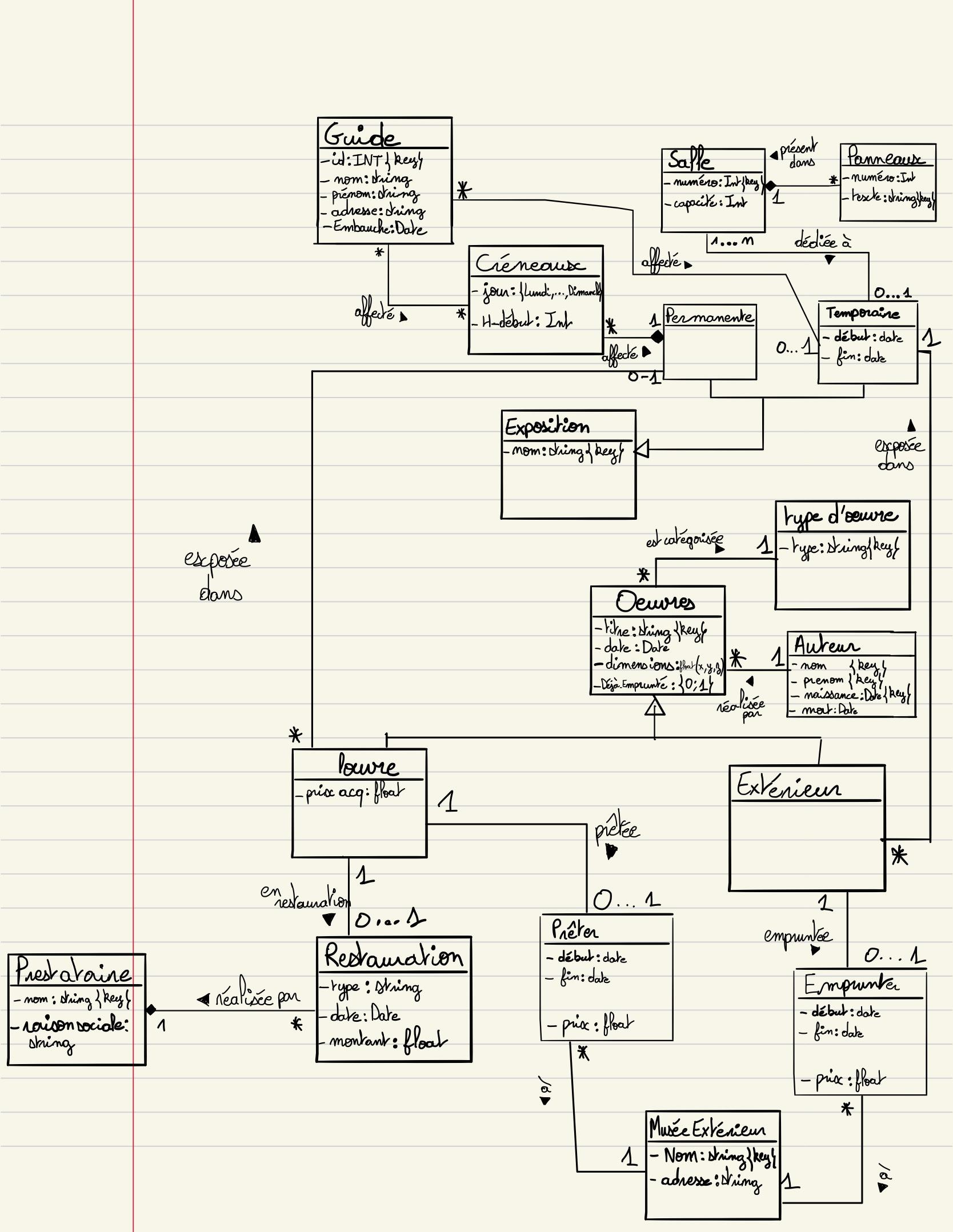
Pour les guides, nous devions nous assurer de l’attribution des créneaux pour les visites des expositions.

II. MCD - Cardinalité

* MCD

Suite à l’étude du sujet, nous avons commencé par réaliser notre MCD pour structurer notre base de données. La MCD nous a permis de rentrer en détail dans le sujet ainsi que de commencer à réfléchir aux contraintes proposées par l’énoncé.

Suite à plusieurs réunions, nous en sommes arrivé au résultat suivant :



Nous avons, grâce à cette MCD, défini l’ensemble de nos tables ainsi que leurs attributs. Nous avons ensuite réalisé nos associations entre les tables.

Nous avons un total de 17 tables reliées par association ou héritage. Nous avons également défini l’ensemble des attributs, avec leurs types. Nous avons également défini des clés pour certains.

Nous avons cependant rencontré quelques difficultés, notamment pour l’interprétation du sujet. Nous avons donc réalisé des choix pour définir nos cardinalités entre nos tables.

* Détail des cardinalités et des héritages:
* Héritage

Œuvres -> Louvre/Extérieur

Héritage par classe fille (classe mère abstraite)

Exposition -> Temporaire/Permanente

Héritage par classe fille (classe mère abstraite)

* Association
* Œuvres - auteur

Association 1:N, une œuvre a forcément un auteur associé.

* Œuvre - Type d’œuvre

Association 1:N, nous avons décidé de créer une table « type d’œuvre » pour faciliter la lecture de la BDD.

* Prestataire - Restauration

Association 1:N, un prestataire peut s’occuper de plusieurs restaurations.

* Restauration - louvre

Association 0...1:1, les œuvres peuvent être en restauration ou non, une restauration est forcément réalisée sur une œuvre du musée du Louvre.

* Louvre - Permanente

Association 0...1:N, nous sommes partis du principe qu’une œuvre n’est pas forcément associée à une exposition permanente.

* Louvre - Prêter

Association 1:0...1, Ici nous avons décidé de créer 2 tables (Prêter et Emprunter) pour gérer la contrainte du sujet (on ne peut emprunter/prêter qu'une seule fois chaque œuvre).

* Prêter - Musée Extérieur

Association 1:N, on prête l’œuvre à un musée extérieur.

* Emprunter - Extérieur

Association 1:0...1, idem à Louvre – Prêter.

* Musée Extérieur - Emprunter

Association 1:N, on emprunte l’œuvre à un musée extérieur.

* Extérieur-Temporaire

Association 1:N, Les œuvres empruntées sont forcément exposées en exposition temporaire (aucune dans les expositions permanente), et sont forcément liées à une exposition.

* Temporaire - Salle

Association 0...1:1...N, une exposition a au moins une salle dans laquelle elle est exposée, cependant une salle peut être sans exposition associée.

* Salle - Panneau

Association 1:N, Une salle peut posséder plusieurs panneaux. Un panneau est forcément associé à une unique salle.

* Temporaire - Guide

Association 0...1:N, un guide peut être associé ou non à une exposition temporaire, mais une seule à la fois.

* Permanente - Créneau

Association 1:N, Chaque créneau est associé à une unique exposition permanente (nous avons considéré qu’il pouvait exister plusieurs expositions permanentes en même temps).

* Créneau - Guide

Association N:N, Chaque guide a plusieurs créneau et ils peuvent être plusieurs guides sur un même créneau.

III. MLD normalisation

* Clés primaires

La clé primaire de ‘Auteur’ est le tuple (Prénom, Nom, Date de Naissance). La date de naissance est précisée au jour près, donc la probabilité d’avoir deux auteurs différents de même nom, prénom et même date au jour près est négligée. La date de mort peut être NULL pour les auteurs toujours vivants.

La clé primaire de Oeuvre\_Louvre Oeuvre\_Ext est le couple (Titre, Date). On aurait pu ajouter des clés étrangères vers l’auteur dans la clé primaire, mais cela n’aurait pas permis de recenser les œuvres d’auteurs inconnus.

La clé primaire des tables Emprunt et Prêt sont le couple (Oeuvre, Musée extérieur) donc deux références étrangères vers des tables : il s’agit de deux classes d’association. Chaque oeuvre ne peut être prêtée qu’une seule fois à un musée donnée, il n’est donc pas pertinent d’ajouter les dates des Emprunt/Prêt comme clé primaire.

Une restauration est identifiée par le couple (Date de la restauration, Oeuvre) et contient une référence étrangère vers un prestataire.

Tous les attributs de la table Créneau font partie de sa clé : un créneau donné précise l’heure de début, le jour de la semaine, et l’exposition qu’elle concerne.

Guide et Créneau sont reliés par une association N..N donc on a besoin d’une nouvelle table Créneau-Guide, composée de références étrangères vers Guide et Créneau.

Un panneau est identifié par son texte (qui est forcément unique) et non par son numéro.

* Contraintes

L’héritage des expositions et des œuvres est un héritage par classe fille, il faut donc vérifier qu’il n’y ait pas une exposition permanente et une exposition temporaire qui aient le même nom, ou une œuvre du Louvre et une œuvre extérieure qui aient le même nom et même date.

Il faut vérifier que toutes les dates de mort des auteurs, ou de fin d’exposition/de prêt, sont supérieures aux dates de naissance ou de début.

*Projection(Exposition\_Temporaire,nom)⊆Projection(Salle,exposition\_temporaire)*

Enfin, il faut vérifier qu’on ne prête pas d'œuvres qui soient en restauration, et qu’on ne mette pas en restauration des œuvres qui sont en prêt.

*NOT(Projection(Restauration, titre\_oeuvre, date\_oeuvre) AND ((Projection(Prêt, titre\_oeuvre, date\_oeuvre) OR Projection(Oeuvre\_Louvre, titre, date)))*

* Vue

Voici maintenant la liste des views que nous avons réalisées pour notre projet. Nous les avons séparées en 2 parties, les vues dites “principales” permettant d’accéder à certaines données particulières de la base de données.

Il y a également les vues dites ‘statistiques’ pour les calculs relatifs à certains attributs de notre base de données.

**Vues principales :**

* Vérifier la contrainte d’héritage de la classe ‘exposition’
* Vérifier la contraire d’héritage de la classe ‘œuvres’
* Vérifier la contrainte 'Une oeuvre en restauration ne peut être ni empruntée ni exposée'
* Vue Oeuvre Louvre disponible
* Vérifier la contrainte 'le prêt d'une œuvre entre le Louvre et un musée extérieur (dans un sens ou dans l'autre) ne peut avoir lieu qu'une seule fois'
* Vue Oeuvre déjà prêtée
* Vue Oeuvre déjà empruntée
* Vue Oeuvre ne pouvant être plus prêtée ou empruntée
* Vue Creneau et Creneau\_Guide: calculer l'horaire complet en fonction de l'horaire du début donné
* Vue Oeuvre Louvre à mettre à jour
* Vue Oeuvre Ext à mettre à jour

**Vues statistiques :**

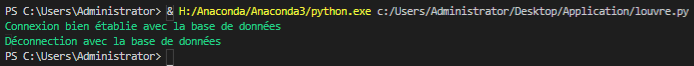
* Calculer le montant des restaurations
* Calculer le prix moyen des emprunts pour chaque musée
* Calculer le montant du prix des prêts pour chaque musée
* Calculer la durée des expositions temporaires
* Calculer le prix moyen d'acquisition des oeuvres d'une exposition
* Calculer le temps moyen des prêts avec les musées extérieur

IV. Application Python

* **Introduction**

Nous avons réalisé une application Python pour manipuler notre base de données (BD). Pour cette partie, on a créé un système de gestion interne, qui permet de se connecter, de consulter et de manipuler notre BD à l’aide de la librairie Tkinter du Python en proposant une interface d’utilisateur (GUI).

* **Connexion et Déconnexion avec la BD**



Au moment du lancement et de la fermeture du programme, un message de connexion et déconnexion va afficher dans la console pour s'assurer que l’on a bien établi puis fermé la connexion avec la BD.

* **Login**



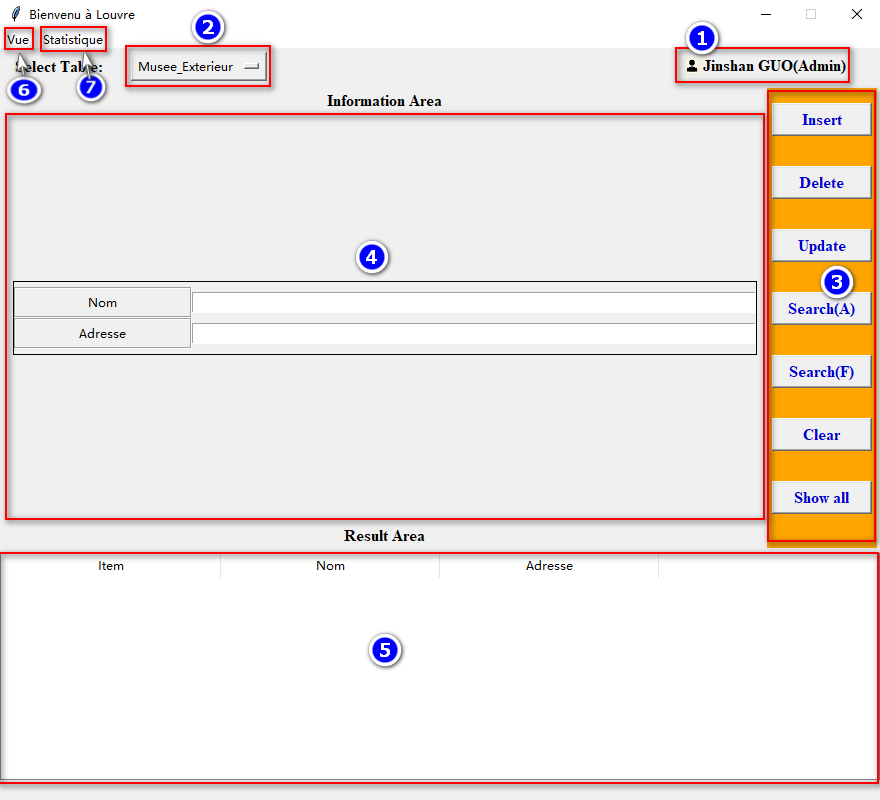
Cette interface permet à l’utilisateur d’entrer le login et mot de passe pour accéder au système de gestion Louvre, on a créé une table ‘Utilisateur’ dans notre BD pour stocker tous les utilisateurs en précisant leur rôle :

|  |  |  |
| --- | --- | --- |
| Login | Mot de passe | Rôle |
| Jinshan GUO | nf18 | Admin |
| Jules Yvon | nf18 | Membre |

* Si aucune information n'est entrée dans la zone de saisie, un message ‘*Erreur, Tous les champs requis*" va apparaître.
* Si le login et/ou mot de passe ne pouvant pas être trouvé(s) dans la BD, un message ‘*Erreur, Nom d'utilisateur ou Mot de passe incorrecte*" va apparaître.
* La table ‘Utilisateur’ ne peut être modifié que par l’administrateur, en éliminant les privilèges de tous les membres

*REVOKE ALL PRIVILEGES ON Utilisateur IN SCHEMA louvre FROM Membre;*

* Si le login et mot de passe sont corrects, la fenêtre login va être cachée et on entre dans l'interface du système.
* **Interface du système**

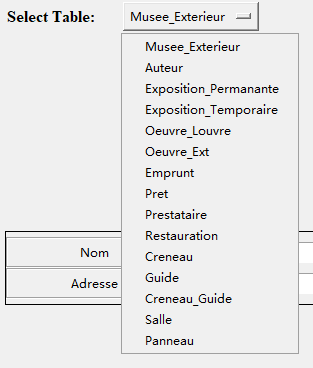


* Zone 1

Afficher l’utilisateur courant avec son rôle, toutes les modifications de l’administrateur (insertion, suppression et mise à jour de données) vont être mise à jour dans le système et dans la BD en même temps , et toutes les modifications d’autres membres seront simplement mis à jour dans le système mais non dans la BD, c’est l'intérêt de gestion du droit.

* Zone 2

Sélectionner la table créée dans la BD à manipuler



* Zone 3

Proposer le bouton permet aux utilisateurs d’insérer, de supprimer, de mettre à jour, de consulter et d’afficher tous les enregistrements pour la table sélectionnée.

* Zone 4

Permettre d'entrer les valeurs d’un tuple, en chaîne de caractère pour simplifier la programmation.

* Zone 5

Afficher les résultats en indiquant le nom de colonne, en numérotant chaque enregistrement.

Possible de glisser pour afficher tous les résultats au cas où il y en aurait beaucoup ou si la longueur des colonnes dépasse la taille de la fenêtre.

Mettre à jour en même temps lorsqu’on choisit une nouvelle table.

* Zone 6

Menu vue permet de vérifier toutes les contraintes, le résultat va être affiché dans la Zone 5.

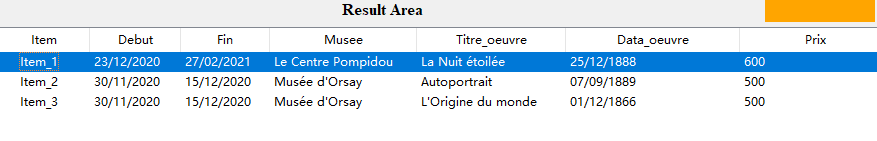
* Zone 7

Menu de fonction statistique permet de calculer de d’afficher le résultat dans la Zone 5.

* **Fonctions réalisées**

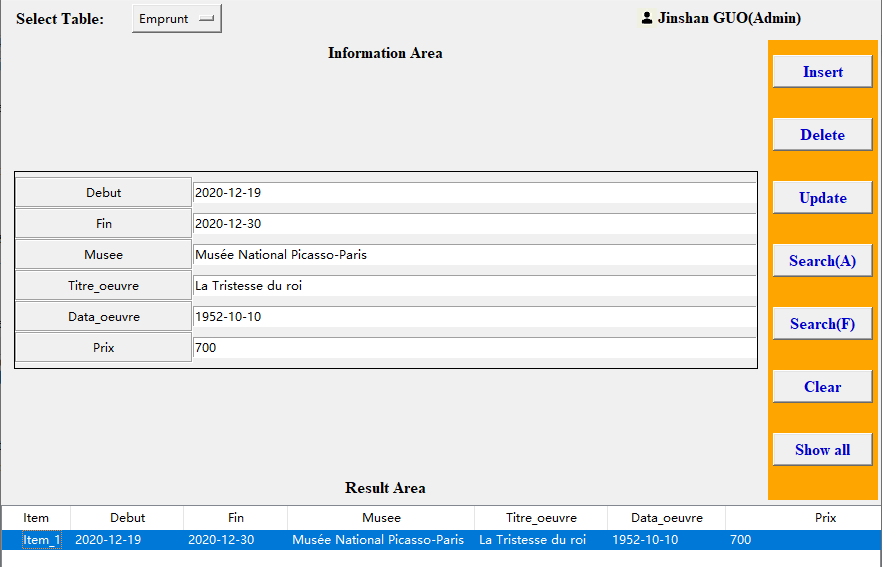
On prend l’exemple de table ‘Emprunt’ pour montrer que toutes les fonctions ont été bien réalisées:

* + Afficher tous les enregistrements



Au début, il y a trois enregistrements dans la BD pour la table ‘Emprunt’, on va maintenant insérer une nouvelle donnée.

* + Insertion



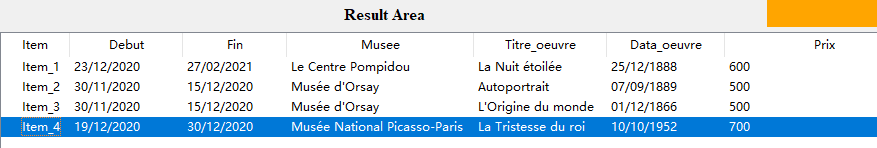
Si l'utilisateur courant est admin, la BD va être mise à jour en même temps(ici, un message de confirmation va être affiché dans la console)



Si l'utilisateur courant n’est pas admin, « insertion en échec » s’affiche comme message d’erreur:

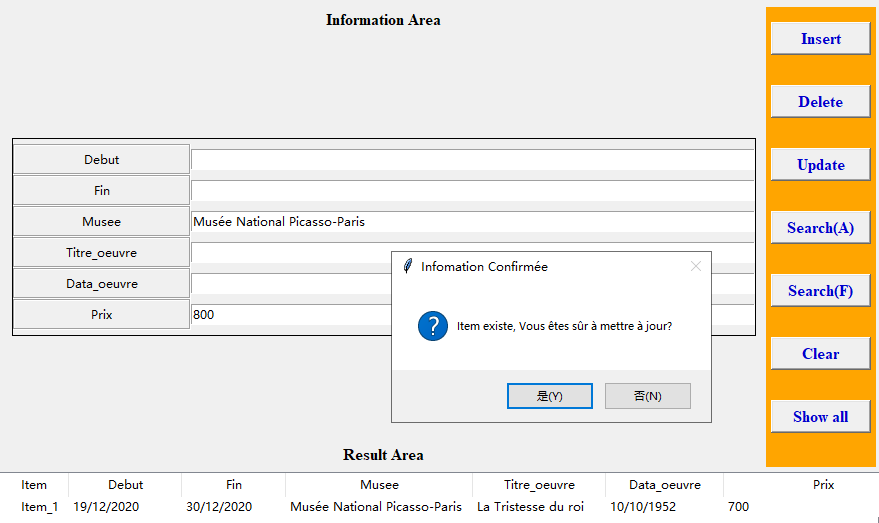


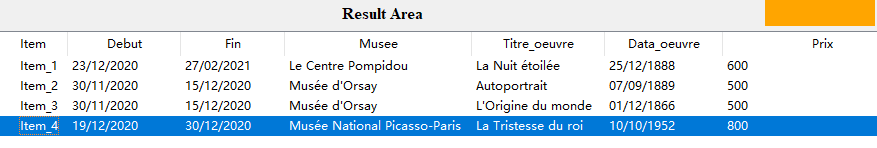
Maintenant, on affiche à nouveau tous les enregistrements, on peut voir que les données ont été bien insérées.



* + Mise à jour

Lorsque l’on veut modifier un enregistrement, on peut taper le mot clé dans la zone de saisie, une recherche précise va être effectuée. Un message de dialogue s’affiche si l’enregistrement existe dans la BD, et demande alors à l’utilisateur de confirmer la mise à jour. Par exemple, on va mettre le prix d’oeuvre qu’on vient d’insérer en 800:



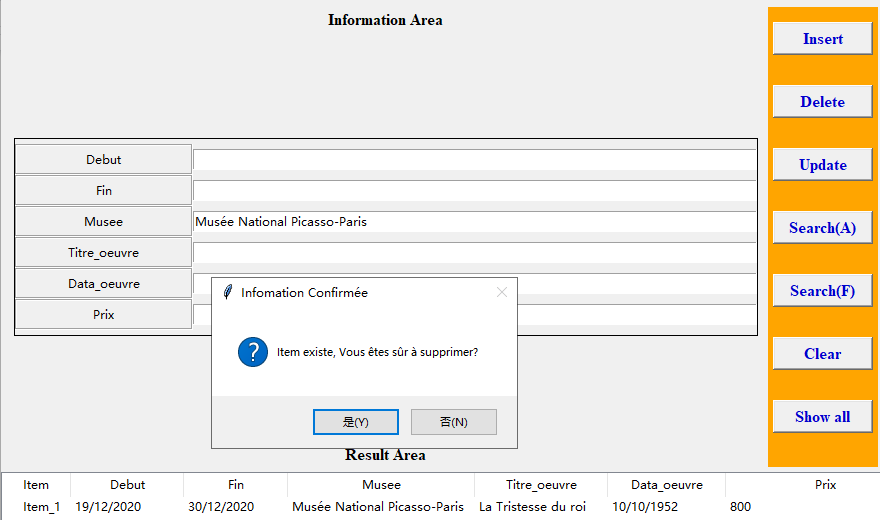


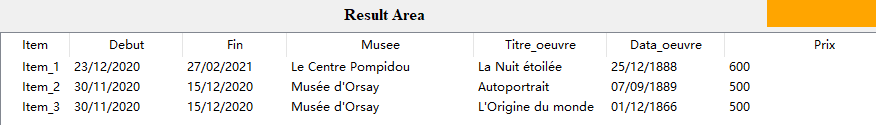
L’enregistrement a été mis à jour correctement, et un message de succès va être affiché aussi dans la console en tant qu’administrateur. Un message de refus va être affiché en tant que membre.



* + Suppression

Même démarche pour la suppression. Nous supprimons l’enregistrement que l’on a inséré:





L’enregistrement n’existe plus dans la BD et un message de succès de suppression a été affiché:

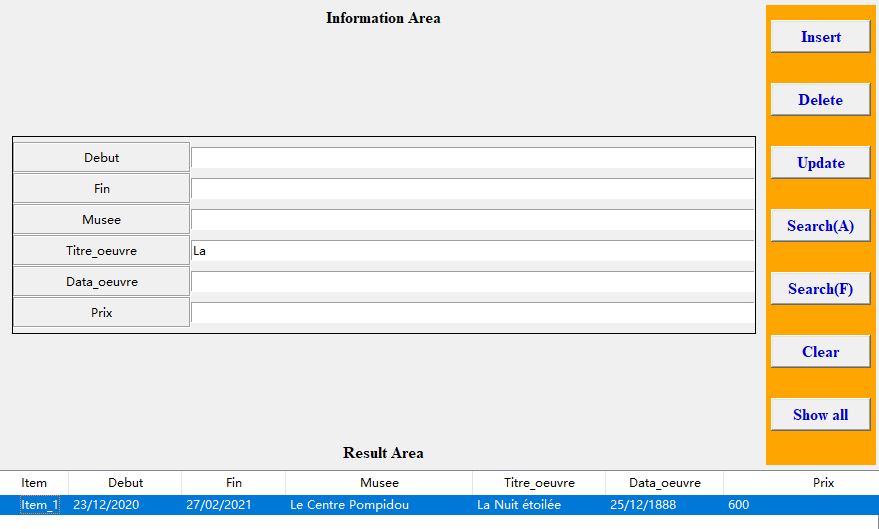


* + Recherche

Il y deux types de recherche: recherche précise et recherche vague, on va chercher maintenant l’oeuvre ‘La Nuit étoilée’ de deux manière:

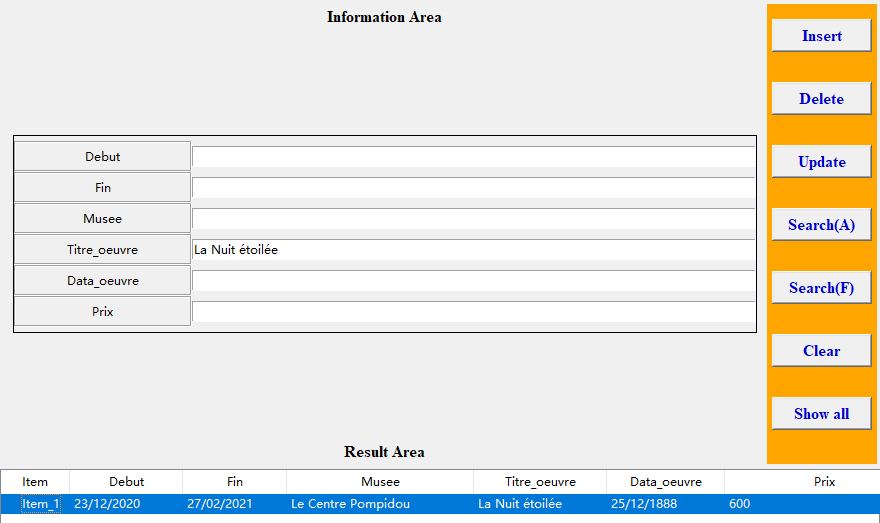
* Recherche vague

Il faut juste proposer une partie du nom de l’œuvre ‘La Nuit étoilée’.



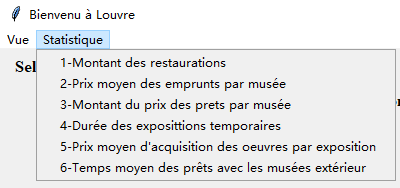
* Recherche précise

Il faut proposer le nom complet et exact

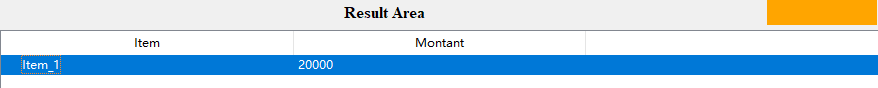


* + Résultat de fonction statistique

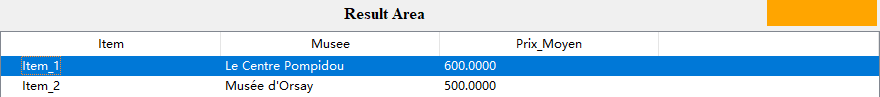
On propose un menu permettant de calculer les fonctions statistiques et d'afficher le résultat dans la zone de résultat(zone 5)

****

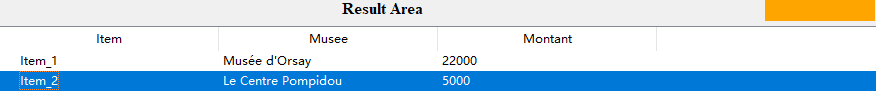
Calculer le montant des restaurations



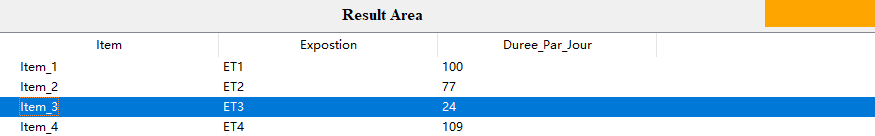
Calculer le prix moyen des emprunts pour chaque musée



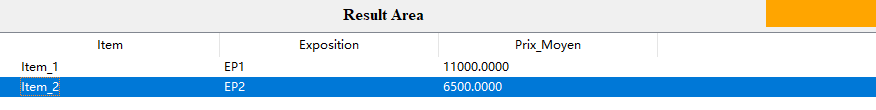
Calculer le montant du prix des prêts pour chaque musée



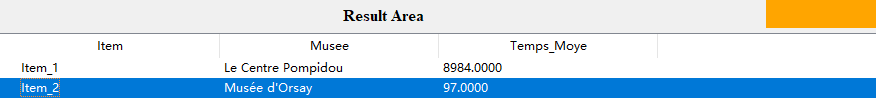
Calculer la durée des expositions temporaires



Calculer le prix moyen d'acquisition des oeuvres d'une exposition



Calculer le temps moyen des prêts avec les musées extérieur



1. Amélioration

Comme nous sommes débutants pour créer une GUI avec la librairie Tkinter en python, au lieu d'appliquer la programmation orientée objet (POO), on a choisi de concevoir cette application par la programmation orientée procédure. PPO peut résoudre le problème du domaine de variable, simplifier la programmation et rendre la logique plus claire.

Pour simplifier l’entrée des informations dans la zone de saisie, on a choisi de tout faire par chaîne de caractère. L’application n’est pas assez ergonomique et on peut l’améliorer en personnalisant chaque attribut pour chaque table. Par exemple, pour l’attribut de type booléen, on peut proposer un bouton d’option.

1. Conclusion

De la conception à l’application, on a bien intégré tout ce qu’on a appris dans le cours et td dans notre projet. On a bien profité le projet et aussi enrichi notre connaissance sur la base des données.