

SAÉ "Conception et implémentation d'une base de données"

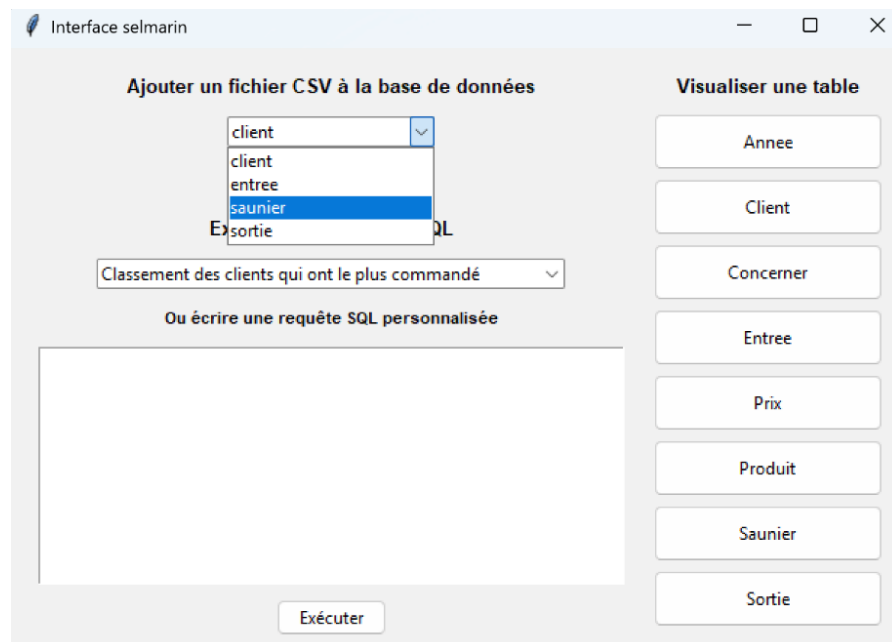
Dans le cadre de la Saé, nous avons travaillé sur le projet intitulé "**Les sauniers de l'île de Ré**". L'objectif de ce projet était d'informatiser la gestion des approvisionnements et des ventes de la coopérative de sel marin, en concevant une base de données complète, en la remplissant de données réelles, et en créant une application graphique pour interagir avec celle-ci.

Après avoir conçu le modèle conceptuel de données (MCD) puis le schéma relationnel, nous avons créé et alimenté la base de données MySQL en utilisant des scripts SQL. En complément, nous avons développé une interface graphique en Python.

Au début du projet, nous avons travaillé en commun pour concevoir la base de données. Ensuite, la répartition des tâches s'est organisée naturellement : Nathaël s'est chargé de commencer l'interface Python, tandis que Paul s'est concentré sur l'écriture et le test des différentes requêtes SQL. Enfin, lors de la phase finale, nous avons travaillé ensemble pour corriger les bugs, ajuster les dernières fonctionnalités de l'interface, et vérifier que tout fonctionnait correctement avant la livraison du projet.

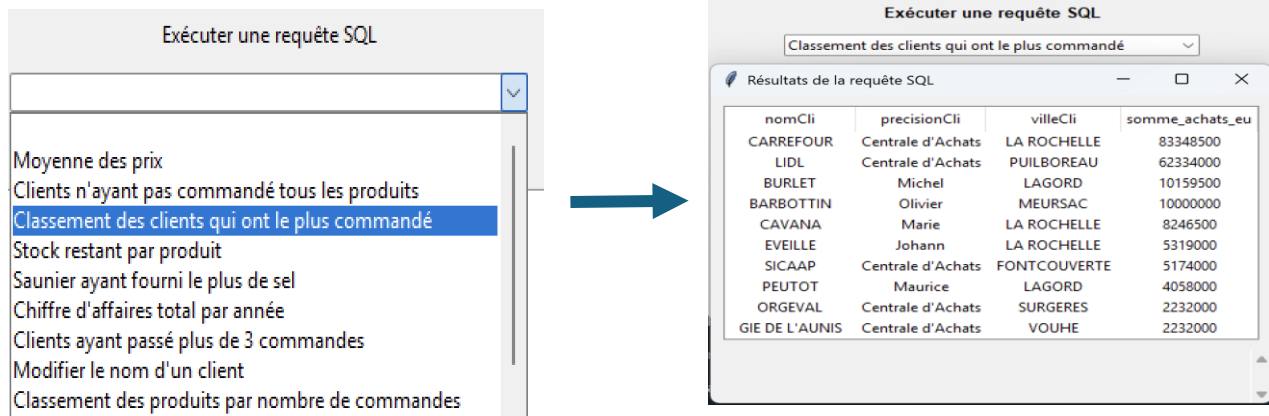
Ce document présente un bilan de la réalisation de cette application, accompagné d'une démonstration de ses fonctionnalités, d'un retour d'expérience sur ce qui fonctionne ou non, ainsi que des pistes d'amélioration envisagées.

DEMONSTRATION DE NOTRE TKINTER :



Voici notre interface tkinter qui est simple et fonctionnelle. Pour le style de l'interface, nous avons utilisé ttk (Tk themed widgets), une extension moderne de tkinter, qui permet d'obtenir des composants plus esthétiques, cohérents avec le système d'exploitation, et mieux adaptés à une interface professionnelle.

L'ajout de fichiers CSV fonctionne correctement, car la liste déroulante prend en compte les fichiers .csv présents dans le dossier. Si les fichiers ne sont pas les bons, des erreurs sont affichées car les tables ne peuvent pas être alimentées correctement.



The image shows two windows from a software application. The left window, titled "Exécuter une requête SQL", contains a dropdown menu with several predefined SQL queries. The query "Classement des clients qui ont le plus commandé" is selected and highlighted in blue. An arrow points from this window to the right window. The right window, also titled "Exécuter une requête SQL", shows the results of the selected query in a table format. The table has four columns: "nomCli", "precisionCli", "villeCli", and "somme_achats_eu". The results are sorted by the total purchase amount in descending order.

nomCli	precisionCli	villeCli	somme_achats_eu
CARREFOUR	Centrale d'Achats	LA ROCHELLE	83348500
LIDL	Centrale d'Achats	PUILBOREAU	62334000
BURLET	Michel	LAGORD	10159500
BARBOTTIN	Olivier	MEURSAC	10000000
CAVANA	Marie	LA ROCHELLE	8246500
EVEILLE	Johann	LA ROCHELLE	5319000
SICAAP	Centrale d'Achats	FONTCOUVERTE	5174000
PEUTOT	Maurice	LAGORD	4058000
ORGEVAL	Centrale d'Achats	SURGERES	2232000
GIE DE L'AUNIS	Centrale d'Achats	VOUHE	2232000

La sélection d'une requête SQL prédéfinie fonctionne comme prévu : une fois la requête choisie dans la liste déroulante, son exécution affiche les résultats dans une nouvelle fenêtre. Les colonnes s'adaptent dynamiquement en fonction des données retournées par la requête, ce qui permet une visualisation claire et structurée des informations extraites de la base de données. Toutes les requêtes fonctionnent correctement.



The image shows a window titled "Ou écrire une requête SQL personnalisée". It contains a text area with the placeholder text "select * from...|". Below the text area is a button labeled "Exécuter".

Écriture manuelle d'une requête SQL : La zone de texte dédiée permet à l'utilisateur de saisir librement une requête SQL personnalisée. Une fois la requête rédigée et validée, elle est exécutée sur la base de données, et les résultats sont automatiquement affichés dans

une nouvelle fenêtre. Nous avons testé la zone avec des requêtes fausses et cela affiche les erreurs correspondantes sans faire planter le programme. Les requêtes qui commencent par autre chose que 'select' sont programmées pour ne pas renvoyer de résultats hormis un message d'exécution. Nous avons fait en sorte dans notre programme que DROP database sur selmarin_final soit impossible. Pour les autres 'drop database', la base correspondante est bien supprimée.

A droite on a une visualisation possible d'une table complète via des boutons : chaque table peut être visualisée dynamiquement, ce qui permet de vérifier l'état de la base de données à tout moment.

Nous avons essayé d'intégrer la création de la base de données à partir de l'interface python mais nous n'avons pas réussi. La création de nouvelles bases de données est possible mais seulement si selmarin_final existe déjà. Notre programme est basé sur l'existence de selmarin_final, par conséquent, on peut créer et supprimer d'autres bases mais l'interaction avec ces dernières est impossible.

Nous aurions aimé pouvoir gérer plusieurs bases de données à partir de l'ihm sans être dépendant de selmarin_final.

Ce que nous avons le plus apprécié lors de ce projet est la création de la base en entier à partir de rien, ce qui nous était inconnu, et le fait de tout développer par nous-même. En revanche, les problèmes d'installation de 'mysql.connector' sont les seuls défauts du projet qui nous ont déplu.

Nathaël BENOIT Paul BABIN