*MonAvenir*, vaste projet !

Objctif :Mini Parcoursup

Etape 1 : Requêtes SQL

En amont,

* Présentation de rapide de parcoursup.
* Identification des besoins : formulaires de saisie et de consultation des vœux, base de données pour traiter les candidatures, interactions client / serveur …

D’où l’intérêt des SGBD.

* Cours, utilisation de SQLITE Browser puis SQLITE3 de python
* Identification des entités et relations en jeu et choix du schéma de conception de la base de données.
* Travail sur requêtes SQL à partir de la base de données *MonAvenir*…

Ainsi les élèves commencent à s’approprier cette base de données, ce qui facilitera le travail par la suite…

Nous avons choisi de ne pas développer cette étape car, bien qu’au cœur du BLOC4, celle-ci est aisément transposable dans un cours.

Cependant, sans présenter le document élève final, nous avons pensé à quelques requêtes possibles qui aideront les élèves à s’approprier le langage SQL. Ceci peut aussi servir de base pour une **évaluation**.

Etape 2 : Utilisation Open Data, Fake pour constituer fichiers sources CSV

* Travail sur les requêtes
* Travail sur les contraintes
* Création de fichiers csv pour les lycées et les écoles supérieures avec Open Data.
* Création de fichiers csv avec des données fictives pour les élèves, admin et candidature

- révisions de première sur les données en tables, utilisation de fichiers CSV,

- prévoir des fonctions à compléter, différentiation…

Etape 3 : Création BD MonAvenir

* peuplement de la base à partir des fichiers csv ensuivant le schéma
* prévoir des fonctions à compléter, différentiation…

Etape 4 : Création du formulaire inscription et du fichier main

Detailler brievement

Etape 5 : Traitement

Ici, on ne détaillera pas, on donne code, mais on propose en prolongement…

Etape 6 : Création des formulaires de consultations

Detailler brievement

Etape 7 : CSS

En D.M. UX Design…

Etape 1 : Requêtes SQL

En utilisant la base suivante, écrire les requêtes SQL correspondant aux différents exercices.

admin(idAdmin, login, password, nom, prenom)

superieur(idSuperieur, login, password, nom, type, commune, lattitude, longitude, nbAdmis, nbAppel,   
 coefNote1, coefNote2)

lycee(idLycee, login, password, nom, commune)

eleve(idEleve, idLycee, login, password, nom, prenom, anneeNaissance, note1, note2)

candidature(idEleve, idSupérieur, statut)

j’ai enlevé les retours lignes pour que ça tienne sur une page…

Exercice1 : Donner tous les noms des lycées

SELECT nom FROM lycee ; #On en obtient 140…

Exercice2 : Donner les noms des écoles du supérieurs dans l’ordre alphabétique décroissant

SELECT nom FROM superieur ORDER BY nom DESC ; #On en obtient 181…

Exercice3 : Donner les 10 premiers noms, prénoms des élèves ayant obtenu au moins 19 en note1

SELECT nom,prenom FROM eleve WHERE note1 >= 19 LIMIT 10 ; #Loiseau Emmanuel…

Exercice4 : Donner les prénoms et moyennes des élèves dont le prénom contient la lettre A et dont la moyenne des notes 1 et 2, nommée moyenne, est strictement supérieure à 16

SELECT prenom, (note1 + note2)/2 AS moyenne FROM eleve

WHERE 16 <= moyenne AND prenom LIKE '%A%'; #On en obtient 415…

Exercice5 : Donner le nombre d’élèves du lycee ' Lycée polyvalent Aiguerande'

SELECT COUNT(\*) FROM eleve JOIN lycee USING(idLycee)

WHERE lycee.nom = 'Lycée polyvalent Aiguerande' ; #On en obtient 60…

Exercice6 : Donner le nombre d’élèves du lycée ' Lycée général et technologique Jean-Paul Sartre' qui ont postulé pour la faculté de droit #On en obtient 14…

SELECT COUNT (\*) FROM eleve JOIN lycee USING(IdLycee)

JOIN candidature USING(idEleve) JOIN superieur USING(Idsuperieur)

WHERE lycee.nom = 'Lycée général et technologique Jean-Paul Sartre' AND superieur.nom = 'Faculté de droit ';

Exercice7 : Donner les noms des filles prénommées 'Charlotte' qui sont dans le même lycée que 'Charlotte Weis' (sauf elle-même !) *auto jointure…*

SELECT E2.nom #On en obtient 2 : Bodin Moreno

FROM eleve E1 JOIN eleve E2 ON E1.idLycee = E2.idLycee

WHERE E1.nom = 'Weiss' AND E1.prenom = 'Charlotte' AND E2.nom <> 'Weiss' AND E2.prenom = 'Charlotte' ;

Exercice8 : Donner, pour chaque école du supérieur, son nom et le nombre de postulants *GROUP BY*

SELECT nom, COUNT(\*)

FROM superieur join candidature using(idSuperieur) GROUP BY idSuperieur ; #Institut des droits de l’ho, 462

Exercice9 : Donner, pour chaque école du supérieur, son nom et le nombre de postulants du lycée du Parc ayant plus de 16 de moyenne *GROUP BY + HAVING*

SELECT superieur.nom, COUNT(\*) #Faculté de droit,6…

FROM candidature JOIN eleve USING(idEleve) JOIN lycee USING(idLycee) JOIN superieur USING(idSuperieur)

WHERE lycee.nom = 'Lycée général du Parc' GROUP BY idSuperieur HAVING ((note1+note2)/2 >= 16) ;

Exercice10 : Trouver requête avec LEFT JOIN et COALESCE ??

Etape 2 : Utilisation Open Data, Fake pour constituer fichiers sources CSV

On pourrait travailler par groupe/étape, en fonction des niveaux de chacun, du fait qu’avec le confinement, les différences ont été accentuées, et aussi en tenant compte que tous les cours ne seront pas tous sur machine…

**Atelier 1 : papier crayon puis DB BROWSER**

Imaginez qu’il existe une base de données *MonAvenir* comme celle vue en classe…

Quelles requêtes voudriez-vous pouvoir effectuées en tant qu’administrateur ? en tant que lycéen ? en tant que chef d’établissement du supérieur ? en tant que proviseur ?

Ecrire ces requêtes en langage SQL en utilisant la BD *MonAvenir*.

Créer un fichier **requêtes.sql**

**Atelier 2 : papier crayon puis Éditeur de texteNote Pad++ ici j’hesite à faire faire direct avec DB BROWSER**

Imaginez que l’on veuille peupler la base de données *MonAvenir* comme celle vue en classe…

Réfléchir sur la construction du schéma entité/association et sur les contraintes

TRavil sur equ^tet pour créate la base de la BD.

Ecrire avec **NotePad++** unfichier nommé **creer\_base\_monavenir.sql** permettant la construction de cette BD

**Atelier 3 : travail sur OpenData et CSV**

BUT : peuplement des classes lycée et supérieur…

**1/** **Récupération des données lycées**

Sur internet, saisir l’adresse : [https://data.education.gouv.fr](https://data.education.gouv.fr/), puis cliquer sur l’onglet Données.

Effectuer des filtrages pour obtenir la liste des **154 lycées ouverts du Rhône** (ne pas tenir compte de celui qui va ouvrir ni de celui qui va fermer).

Exporter le jeu des 154 données au format CSV sous le nom **lycee-rhone-data.csv**

Ouvrir le fichier avec un tableur et supprimer les colonnes inutiles : garder uniquement **Appellation officielle** et **Commune**.

**2/** **Récupération des données supérieurs**

Aller sur le site internet [https://www.data.gouv.fr/en/datasets/etablissements-denseignement-superieur-2/#](https://www.data.gouv.fr/en/datasets/etablissements-denseignement-superieur-2/)

et exporter la liste des établissements d’enseignement supérieur au format csv et renommer sous

**superieur-rhone-data.csv**

Ouvrir le fichier avec un tableur et supprimer les colonnes inutiles : garder uniquement **type d’établissement**, **nom**, **commune, département, latitude** et **longitude**. Avec des tris/filtres, garder uniquement les 181 écoles du Rhône, puis supprimer la colonne département et inverser l’ordre des deux premières colonnes en nommant : **nom**, **type**, **commune, latitude** et **longitude.**

**3/ Modification des données lycées et supérieurs**

Ouvrir et compléter le fichier **generer\_base\_csv.py** dans lequel on utilise le module Faker (qu’il faudra importer si nécessaire) afin de créer des pseudos ou mots de passes fictifs.

Ce fichier contient 3 fonctions : la première generer\_login permet de générer un login en vérifie que celui-ci n’est pas déjà pris ; les 2 fonctions suivantes permettent de compléter les fichiers CSV pour obtenir les fichiers

**lycee-rhone.csv et superieur-rhone-data.csv**

Compléter la fonction **generer\_login**, puis en vous inspirant de la fonction **generer\_csv\_lycee**, écrire la fonction **generer\_csv\_superieur**.

*Rajouter questions sur Open Data ??…*

*Commentaires sur encodage… Ouvrir le fichier avec NotePad+++ pour vois rendre compte qu’il n’y a pas de problème d’encodage(…)*

**Atelier 4 : travail sur CSV**

BUT : peuplement des classes eleve, admin et candidature…

**1/** **Classe eleve**

Ecrire une fonction **anneeNaiss(a)** qui simule les années de naissances des élèves de terminales.

Elle prend en argument une année a et renvoie une année entre a et a +4 selon une certaine distribution correspondant aux pourcentage d’élèves ayant des années de retard ou d’avance…

Si a = 2000, la fonction doit renvoyer un nombre entier entre 2000 et 2004 compris, sachant qu’il y a environ 1% d’élèves qui ont 2 ans de retard (nés en 2000), 7% d’élèves qui ont 1 an de retard (nés en 2001), 8% d’élèves qui ont 1 an d’avance (nés en 2003) et 1% d’élèves qui ont 2 ans d’avance (nés en 2004).

Compléter la fonction **generer\_login** qui permet de générer un login en vérifiant que celui-ci n’est pas déjà pris.

Compléter la fonction **generer\_csv\_eleve** afin de créer des fichiers csv de 60 élèves pour chacun des lycées.

Compléter la fonction **generer\_csv\_candidature**

Moduler les quotas du nombre d’appel et du nombre d’admis en fonction du type de l’école.

Etape 3 : Création BD *MonAvenir*

**Atelier 1 : Peuplement à partir d’un fichier .sql (NotePad++)**

Compléter la fonction python **creer\_base\_monavenir** du fichier **peupler\_base\_bd.py**.

Elle prend en argument le fichier bd à créer et le fichier sql à partir duquel on va créer la bd.

**Atelier 2 : Peuplement à partir d’un fichier .csv ici on peut faire 3 groupes !!**

En vous inspirant de la fonction **peupler\_eleve**, compléter les fonctions python **peupler\_lycee** **peupler\_superieur, peupler\_admin,** du fichier **peupler\_base\_bd.py** en insérant les données lues, cette fois, dans les fichiers csv correspondants.

Attention les 3 fonctions sont plus simples et ne nécessitent pas l’utilisation d’un compteur…

**Atelier 3 : Peuplement dynamique à partir d’un script de python**

Compléter la fonction python **peupler\_candidature**

Puis regarder et commenter chaque ligne de la fonction **peupler\_base\_mon\_avenir**

En particulier, ils devront comprendre seul l'insertion d'un élève test...

Etape 4 : Création du formulaire inscription et du fichier main

**Atelier 1 : travail sur formulaire inscription HTML fait en classe entiere ensemble**

Regarder le site de David Roche <https://pixees.fr/informatiquelycee/n_site/nsi_prem_flask.html>

Et modifier le fichier TODO ?? afin de compléter le formulaire d’inscription élève.

On donne une image du formulaire d’inscription attendu ().

Correction complète faite en classe car sera réinvestie ultérieurement…

Etape 5 : Traitement

Ici, on ne détaillera pas, on donne code, mais on propose en prolongement…

Etape 6 : Création des formulaires de consultations

**Travail sur formulaire consultation HTML et fichier main TP noté par groupes**

En vous inspirant de ce qui a été fait en classe pour le formulaire d’inscription et pour le fichier main,

créer un formulaire de consultation pour un élève (resp. lycée, supérieur, admin) et compléter le fichier main corrrespondant.

On donne une image du formulaire attendu (4 images différentes pour chacun des quatre formulaires)

Chaque groupe est en charge de réaliser un formulaire de consultation html et les requêtes sql sous-jacentes à partir d’une photo du formulaire attendu, d’une photo de ce que sera la réponse à l’une des requêtes et de la base sql MonAvenir créée.

Le travail préliminaire a donné tous les outils nécessaires aux élèves pour réaliser cette tache finale et donc le travail sera rendu pour **évaluation**.

Etape 7 : CSS

En D.M. ON demande aux élèves de réfléchir à l’UX Design… et de prévoir un CSS pour le projet.

Ce travail peut encore être mené par petit groupe et fera l’objet d’un bonus.