*MonAvenir*, vaste projet !

Objctif :Mini Parcoursup

Etape 1 : Requêtes SQL

Faite en amont, pendant le cours sur le chapitre SQL.

* 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 et serveur / SGBD.
* 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 de notre 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.

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

* création de la base à partir du schéma,
* peuplement de la base par des données aléatoires :

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

- prévoir des fonctions à compléter, différentiation…

* travailler sur contraintes

Etape 3 : Création BD MonAvenir

* insertion dans la base des données lues dans les csv
* Un début de code de l'application avec Flask et sqlite3
* prévoir des fonctions à compléter, différentiation…

Etape 4 : Création Formulaires inscription, consultations

* prévoir des fonctions à compléter, différentiation…

Etape 5 : Traitement

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

Etape 6 : 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…

**Groupe1 : papier crayon**

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*.

**Groupe2 : papier crayon**

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

Quelles questions poser dans le formulaire élève lors de son inscription ?

Réfléchir sur les contraintes à imposer sur les réponses attendues…

**Groupe3 : travail sur formulaire inscription HTML**

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 peut donner une image du formulaire attendu…)

**Groupe4 : travail sur OpenData et CSV**

BUT : peuplement des classes lycées et supérieurs…

**1/** **Récupération de données**

Sur internet, saisir l’adresse : <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**.

Aller sur le site internet <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.**

**2/ Modification de données**

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 5 fonctions : la première generer\_login permet de générer un login en vérifient 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**

J’ai mis des TODO, dans les 2 dernières fct, une autre idée est de donner la première fct et à eux de faire la deuxième…

La quatrième fonction anneeNaiss(a) doit simuler les années de naissances des élèves de terminales de cette année. Elle renvoie un entier aléatoire 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).

Enfin la dernière fonction permet de créer des fichiers csv de 60 élèves pour chacun des lycées.

*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’encodahe(…)*

Etape 3 : Création BD *MonAvenir*

**Groupe1 : travail sur sqlite**

Insertion dans la base des données lues dans les csv

**Groupe2 : travail sur Flask**

Création de données fictives

BUT : peuplement de la calsse élève