*MonAvenir*, vaste projet !

Objectif : créer un mini Parcoursup

application web destinée à recueillir et gérer les vœux d'affectation de lycéens du Rhône   
pour un établissement de l'enseignement supérieur dans le Rhône.

par Véronique Reynaud, Brigitte Mougeot, Frédéric Junier

Etape 0 : En amont

* Présentation de rapide de parcoursup.
* Présentation du projet de BD *MonAvenir* qui tiendra lieu de fil rouge pour cette partie du programme.
* Cahier des charges de l’application - 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 sur les BD, utilisation de SQLITE Browser puis SQLITE3 de python

*Dans certaines étapes, nous proposons de travailler par ateliers pour tenir compte des différents niveaux.*

*Les élèves pourront faire tout ou partie des ateliers et auront pour objectif commun d’achever ce projet.*

Etape 1 : Requêtes SQL

* Discussion avec le groupe classe : notion de dépendances fonctionnelles, identification des entités et relations en jeu dans la BD *MonAvenir* et choix du schéma de conception.
* Ecrire des requêtes SQL d’interrogation de la DB *MonAvenir*.
* Ecrire les requêtes SQL permettant la construction de la BD *MonAvenir*.
* Enoncer les fonctionnalités attendues de l’application *MonAvenir*.

Les élèves commencent à s’approprier la BD *MonAvenir*, ce qui facilitera le travail par la suite.

Etape 2 : Données Open Data et Fake

Construire les fichiers source au format CSV

* Création de fichiers csv pour les lycées et les écoles supérieures avec des données open data.
* Création de fichiers csv pour les élèves et les candidatures avec des données fictives

Cette étape permet de faire des révisions de première sur le traitement des données en tables, l’utilisation de fichiers CSV

Nous fournissons du code à comprendre puis des fonctions à compléter ou à écrire en utilisant l’existant.

Les tâches sont de difficultés variées ce qui permet de gérer l’hétérogénéité entre les élèves.

Etape 3 : Création et peuplement de la BD MonAvenir

Etape 4 : Développement WEB côté serveur en Python, Flask

* Découverte de Flask
* Création du formulaire de connexion HTML (connexion à la BD) et du fichier main (traitement)
* Une fois la connexion réalisée, on souhaite qu’un utilisateur de profil élève puisse interroger la base : par exemple rechercher les établissements du supérieur par type et/ou par commune.

Etape 5 : Traitement des candidatures Non développé. Prolongement. Code est fourni.

Etape 6 : Création des formulaires de consultations

* Création de formulaires de consultations en fonction du profil utilisateur

sous forme de tP noté par groupe.

Etape 7 : CSS

* Devoir Maison : Bonus UX design

*MonAvenir* Etape 1 : Requêtes SQL 30 à 60 minutes

**Atelier 1 :** En utilisant la base suivante, écrire les requêtes SQL correspondant aux différents exercices.

*Difficulté croissante (les dernières ne sont pas au programme officiellement). Peut servir d’évaluation.*

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)

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 : Donner, pour chaque école du supérieur, son identifiant et le nombre de postulants du lycée d’identifiant 4 en écrivant 0 si aucun élève n’a postulé *LEFT JOIN + COALESCE*

SELECT cadidature.idSuperieur, COALESCE(COUNT(\*),0) # …

FROM candidature LEFT JOIN eleve USING(idEleve) WHERE eleve.ideLycee = '4' GROUP BY idSuperieur ;

**Atelier 2 : Requêtes SQL de création de la BD MonAvenir**

…. Commentaires ???

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

**Atelier 3 : Quelles fonctionnalités attendre de notre application ?**

**papier crayon puis DB BROWSER**

L’idée ici n’est pas seulement de savoir comment formuler une requête mais de savoir de quelles requêtes on va avoir besoin pour le projet...

Consigne aux élèves :

- Travailler en groupes à partir de la base de données *MonAvenir*

- Chaque groupe d’élèves prend un rôle parmi : administrateur, lycéen, chef d’établissement du supérieur, proviseur d’un établissement du secondaire.

- Selon le rôle endossé, quelles requêtes voudriez-vous pouvoir effectuer sur la base de données une fois que les vœux des élèves ont été formulés ?

- Ecrire ces requêtes en langage SQL.

- Les tester et corriger si besoin en utilisant la BD *MonAvenir* et le logiciel DB BROWSER.

- Enregistrer vos requêtes dans un fichier **requêtes\_administrateur.sql**, resp **requêtes\_lycéen.sql**,

resp **requêtes\_superieur.sql**, resp resp **requêtes\_ proviseur sql** selon les cas.

*MonAvenir* Etape 2 : Open Data et Fake pour obtenir des fichiers CSV

30 à 60 minutes

**Atelier 1 : Travail sur des données open data et sur le format CSV**

**Génération des données sur les lycées et les établissements du supérieur**

Lors de cet atelier, nous serons amenés à échanger avec les élèves sur :

- les données open-data

- les enjeux éthiques du stockage de données nominatives sur les élèves (d’où des données fictives)

- les normes d’encodage du texte (plus de problème d’encodage lorsqu’on ouvre avec NotePad…)

**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). 155 ce soir !!! lycées ????

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

Ouvrir le fichier avec un tableur et garder uniquement les colonnes **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, travail sur python**

materiel

lycee-rhone-data.csv

superieur-rhone-data.csv

generer\_base\_csv.py

Extraire de l’archive **materiel.zip** le fichier **generer\_base\_csv.py .**

Dans ce fichier, 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 vérifiant 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** du fichier **generer\_base\_csv.py**,

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

**Atelier 2 : Génération des données sur les élèves et leurs candidatures**

**1/ Générer les élèves**

- 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 pourcentages 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).

En utilisant le fichier **generer\_base\_csv.py :**

- Compléter la fonction **generer\_login** qui génère 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.

2/ **Générer les candidatures**

- Compléter la fonction **generer\_csv\_candidature** ou plutôt generer\_csv\_superieur et generer\_admin ??? non

Moduler les quotas de nombre d’appel/nombre d’admis et prévoir un lycée où une école n’est pas demandée.

*MonAvenir* Etape 3 : Création du formulaire inscription et du fichier main

30 minutes

Dans cette partie, il est intéressant de remarquer les différentes façons pour créer une base de données :

* drectement à partir de DB Browser ou Sqlite
* à partir d’un fichier texte
* à partir d’un fichier CSV
* à la volée dans python…

materiel

peupler\_base\_bd.py

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

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

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...

*MonAvenir* Etape 4 : Création du formulaire inscription et du fichier main

Cf autre doc à insérer apres… 120 minutes

*MonAvenir* Etape 5 : Traitement des candidatures, classements…

Non développé. En prolongement. Le code est fourni.

*MonAvenir* Etape 6: Création des formulaires de consultations 30 à 60 minutes

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

**Travail sur formulaire consultation HTML et fichier main**

Consigne aux élèves :

- Chaque groupe se voit attribuer un rôle parmi : élève, lycée, établissement du supérieur, administrateur.

- Suivant son rôle et en s’inspirant de ce qui a été fait en classe pour le formulaire d’inscription et pour le fichier main, chaque groupe devra :

\* créer un formulaire de consultation HTML

\* compléter le fichier main correspondant avec les requêtes SQL sous-jacentes

\* créer le fichier HTML renvoyé par le serveur en réponse à la requête

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

IMAGES

*MonAvenir* 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.