
Binômontron

Kevin LE GRAND - Greoffroy DAUMER

SOMMAIRE :

- Présentation du projet page 2
- Organigramme page 3
- Structure de la base de données page 4
- Explication du programme page 5-8
- Démonstration page 10
- Visualisation dans la base de données page 11
- Conclusion page 12



Présentation du projet Binômontron :

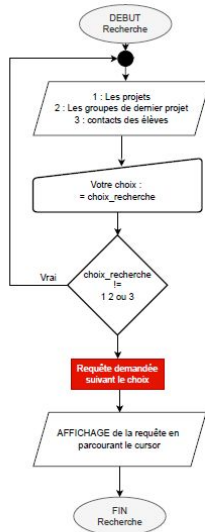
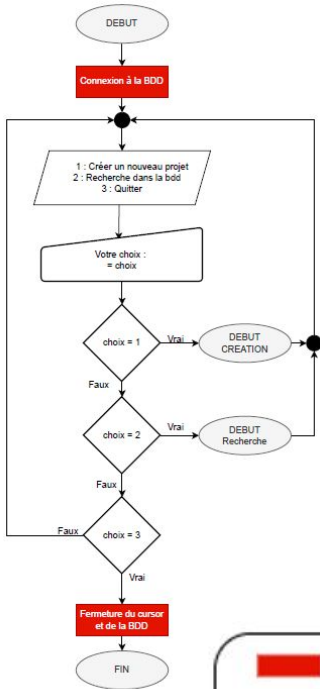
Le but : Créer une appli permettant de former aléatoirement des groupes d'élèves d'une promo

Conditions :

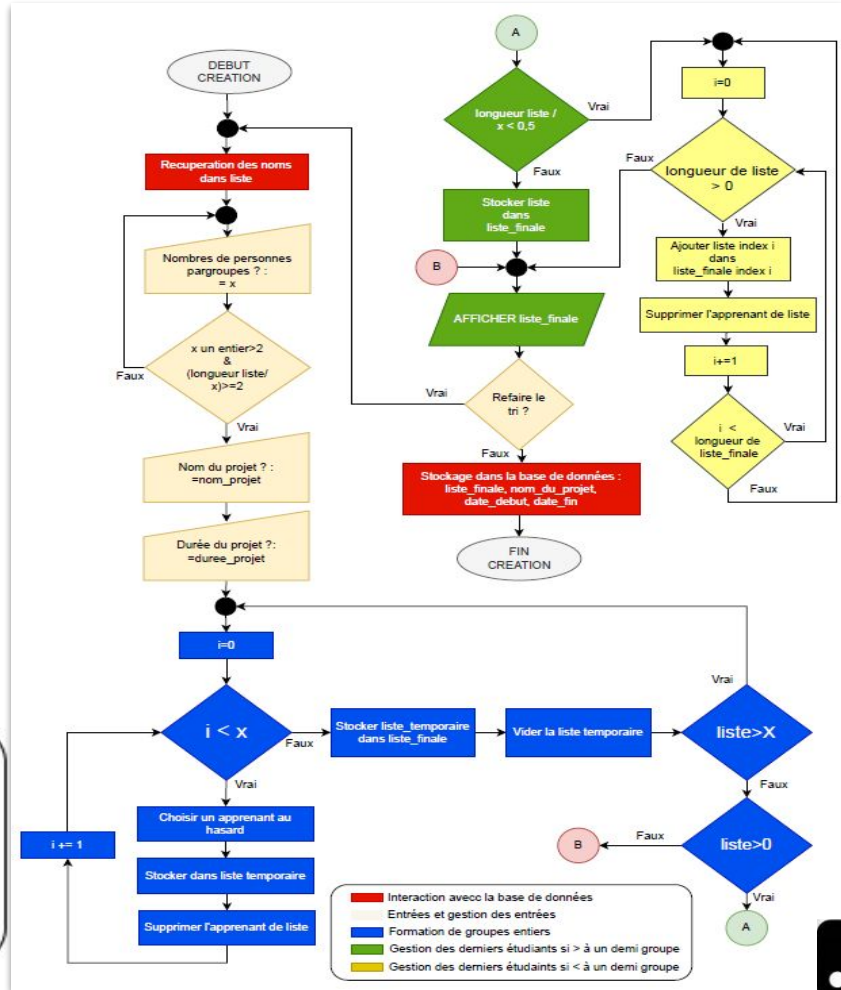
- Une base de données (en MySQL) qui contient une simple table qui liste les élèves de la promo.
- Le code(en python) crée des groupes en répartissant, au hasard, les apprenants par 2,3,4...
- Un l'élève n'appartient qu'à une seule et unique équipe.
- Tous les élèves doivent appartenir à un groupe.
- Les listes de groupes, ainsi que leur constitution, sont à afficher.
- Tous les échanges avec l'utilisateur (affichages et saisies) se font sur la console.
- Stocker en base les groupes constitués, avec leurs dates de création.



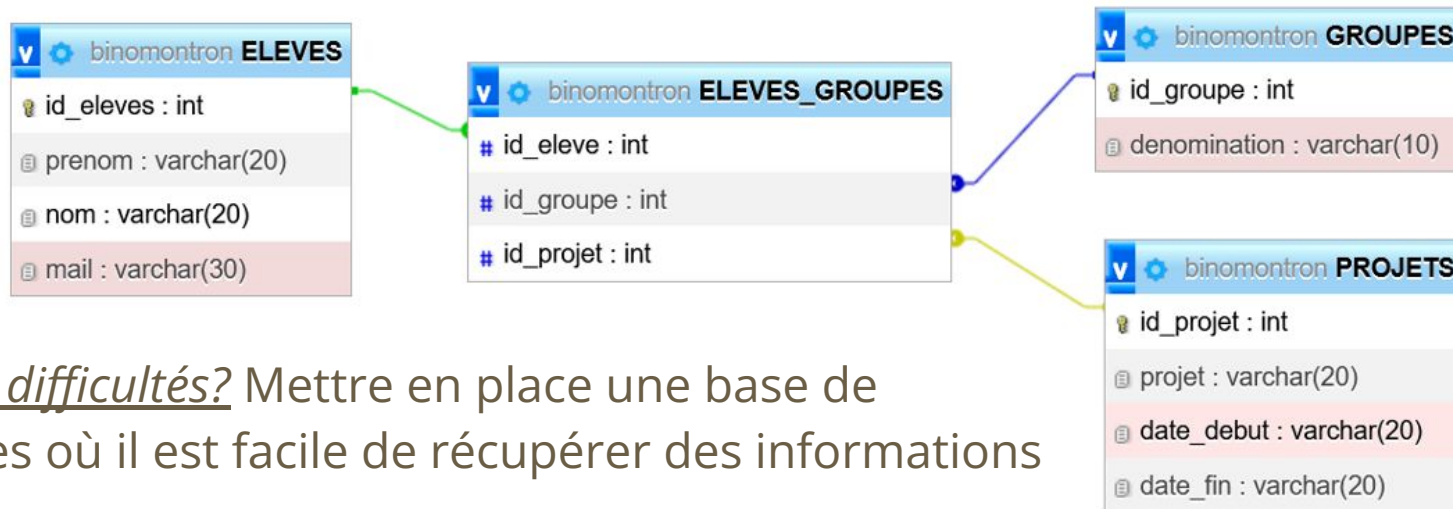
Organigramme :



- Interaction avec la base de données
- Entrées et gestion des entrées
- Formation de groupes entiers
- Gestion des derniers étudiants si $>$ à un demi groupe
- Gestion des derniers étudiants si $<$ à un demi groupe



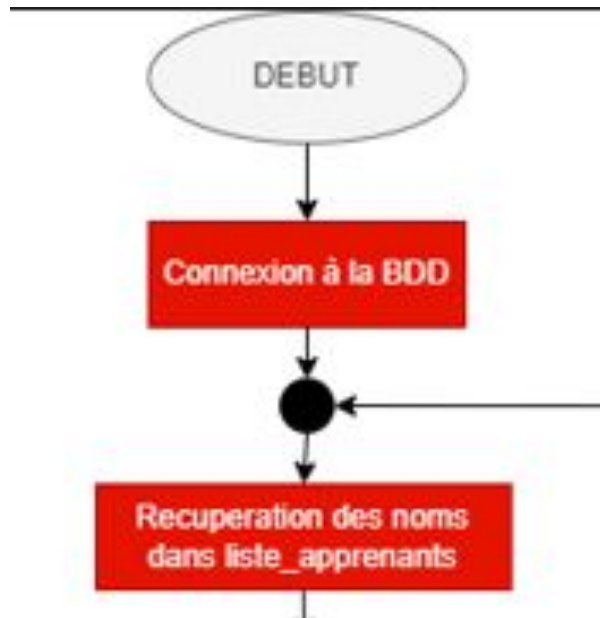
Structure de la base de données :



Quelles difficultés? Mettre en place une base de données où il est facile de récupérer des informations

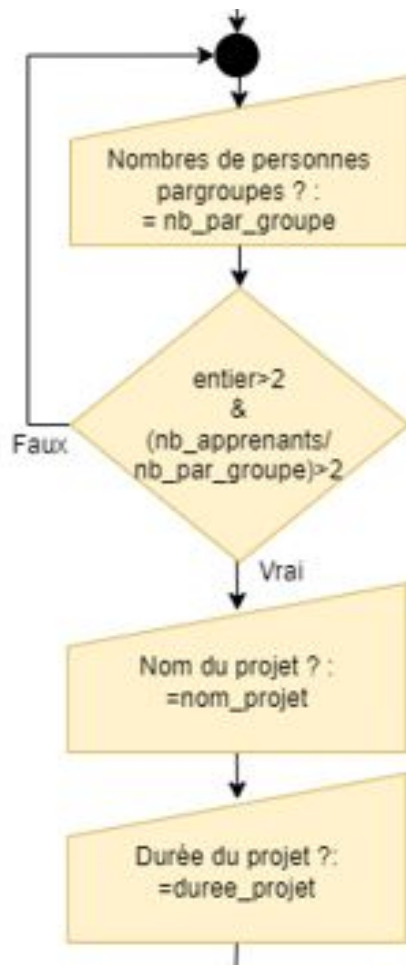
Quelles solutions? Faire une base de données comportant plusieurs table reliées entre elle par une colonne commune (clés étrangères et primaires)





```
# -----  
# Accès à la bdd  
#-----  
user = 'root'  
password = 'example'  
host = 'localhost'  
port = '3307'  
database = 'binomontron'  
  
bdd = mysqlpy.connect(user=user, password=password, host=host, port=port, database=dat  
cursor = bdd.cursor()  
  
#-----  
# Boucle pour un tirage  
#-----  
  
tirage=bool(1) # Variable pour refaire un tirage ou non  
while tirage==1:  
    os.system('clear')  
    query = "SELECT * FROM ELEVES;" # Requête pour récupérer la table ELEVES  
    cursor.execute(query)          # Exécution de la requête  
    liste= [i[1] for i in cursor]  # Stockage des étudiants dans liste  
    x=0 # Variable qui correspond au nombre d'etudiants dans un groupe
```





```

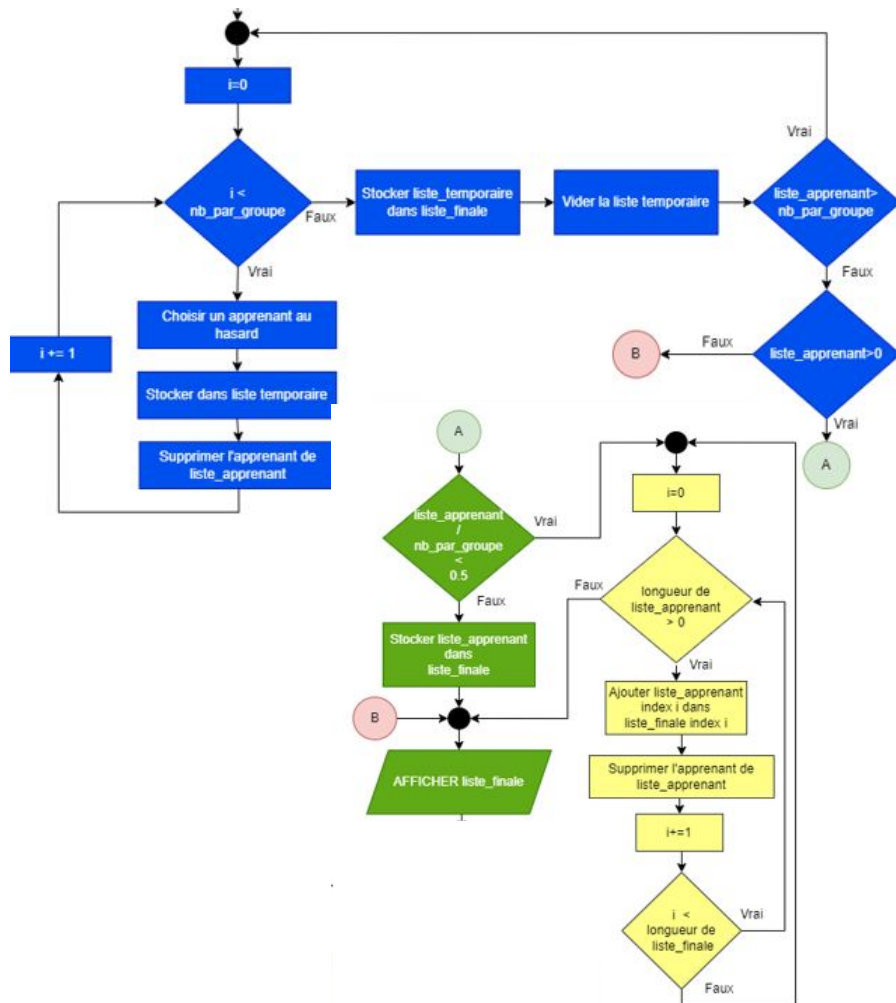
#-----
#   Info projet
#-----
projet = input('Nom du projet :')
# A ajouter au programme : verifier que le nom n'est pas déjà présent dans la bdd
# et qu'il ne dépasse pas 20 caractère

test, error = bool(0), bool(0)
while test == 0:
    duree = input('\nDurée du projet (en jours) : ')
    try:
        duree= int(duree)
    except ValueError:
        print("Il faut un nombre entier :")
        error=1
        continue
    if error==1:
        test=1

date_debut = datetime.date.today()
date_fin = date_debut + datetime.timedelta(duree)

#-----
# RECUPERATION du nombre de personnes par groupe voulu
# par l'utilisateur et gestion des erreurs:
#-----
test = 0
while test == 0:
    x = input('\nNombre de personnes voulu par groupe : ')
    try:
        x= int(x)
    except ValueError:

```

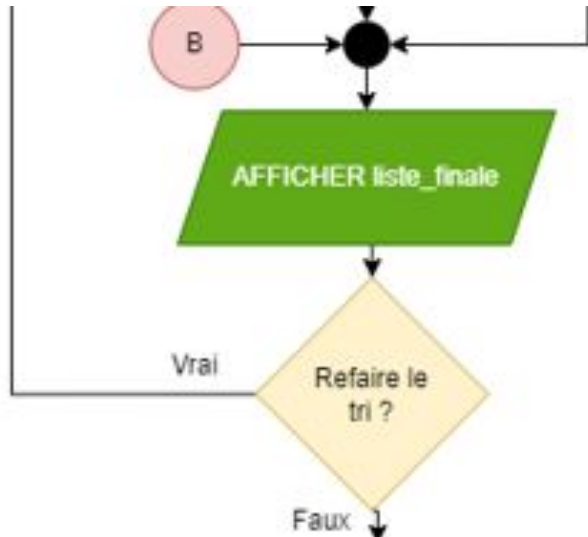



```

#-----
#                               Creation des groupes
#-----
# permettant de connaitre la proportion des etudiants restant pour faire un groupe
ou_mettre_les_derniers= float(float(len(liste)/x)-int(len(liste)/x))

# Groupe complet
liste_finale=[] # Variable pour stocker la liste des groupes(sera vidée à chaque boucle)
while len(liste)>=x:
    liste_temp=[]
    for i in range(x):
        a=str(random.choice(liste))
        liste_temp.append(a)
        liste.remove(a)
    liste_finale.append(liste_temp)

# Etudiant restant:
if len(liste)>0: # Il reste encore des étudiants
    if ou_mettre_les_derniers>0.5: # S'il y a plus d'un demi groupe on ajoute
        liste_finale.append(liste) # les élèves restant dans un groupe à part
    else: # Sinon on inclus les élèves restant dans les groupes déjà créés
        i=0
        for eleves_restant in range(len(liste)):
            liste_finale[i].append(liste[i])
            liste.remove(liste[i])
            i+=1
        if i==len(liste_finale)-1:
            i=0
  
```

```

#-----
# Affichage des elements à envoyer dans la base de donnee
#-----
os.system('clear')
print("Nom du projet :",projet)
print("Debut du projet :{} \nFin du projet :{}\n".format(date_debut.strftime('%A-%d-%B-%Y'),date_fin.strftime('%A-%d-%B-%Y')))
for i in range(len(liste_finale)):
    print("Groupe", i+1, ":", end="")
    for j in range(len(liste_finale[i])):
        print(liste_finale[i][j] , end=" ")
    print("")

#-----
#Demander à l'utilisateur s'il veut refaire un tirage ou stocker la liste de son choix
#-----
test,refaire = 0,0
while test == 0:
    try:
        refaire = int(input("\n1 : Refaire un tirage \n2 : Stocker les informations dans la base de donnee\n"))
    except ValueError:
        print("Vous n'avez pas saisi 1 ou 2 :)")
    if 1>refaire>2:
        print("Vous n'avez pas saisi 1, 2 :)")
    if refaire==1 or refaire==2:
        test =1

```

```

#-----
#                               STOCKAGE
#-----

if refaire==2:
    query = f'INSERT INTO PROJETS(projet, date_debut, date_fin) VALUES ("{projet}", "{date_debut}", "{date_fin}");'
    cursor.execute(query)
    bdd.commit()

    for i in range(len(liste_finale)):
        for j in range(len(liste_finale[i])):
            reqeleve= 'SELECT id_eleves FROM ELEVES WHERE prenom="'+ liste_finale[i][j] +"'";'
            cursor.execute(reqeleve)
            for c in cursor:
                id_eleve=c[0]

            reqprojet= 'SELECT id_projet FROM PROJETS WHERE projet="'+ projet +"'";'
            cursor.execute(reqprojet)
            for result in cursor:
                id_projet=result[0]

            query = f"INSERT INTO ELEVES_GROUPE(id_eleve, id_groupe, id_projet) VALUES ('{id_eleve}', '{i+1}', '{id_projet}')"
            cursor.execute(query)
            bdd.commit()

print("Votre liste de groupes est stockée dans la base de données, au boulot !!!")
tirage=0

#-----
# Fermeture de l'accès à la bdd
#-----

cursor.close()
bdd.close()

```

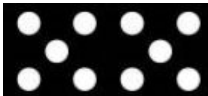
Faux

Stockage dans la base de données :
liste_finale, nom_du_projet,
date_debut, date_fin

FIN



Démonstration...



Visualisation dans la base de données :

```
1: Les projet existant
2: Les groupes sur le dernier projet
3: Contact des eleves
```

Votre choix :2

```
Groupe 1 Dimitri JULES
Groupe 1 Amadou DIABY
Groupe 1 Charley LEBARBIER
Groupe 1 Harold PETIARD
Groupe 2 Johann JOURNAUX
Groupe 2 Viacheslav OSWALD
Groupe 2 Geoffroy DAUMER
Groupe 3 Soraya DUSSART
Groupe 3 Kevin LE GRAND
Groupe 3 Basile GUERIN
Groupe 4 Antoine ANCELIN
Groupe 4 Guinel ZOUITEN
Groupe 4 David SCANU
Press enter to continue...
```

d_eleve_groupe	id_eleve	id_groupe	id_projet
30	9	1	63
31	7	1	63
32	6	1	63
33	2	1	63
34	5	2	63
35	11	2	63
36	8	2	63
37	12	3	63
38	1	3	63
39	10	3	63
40	4	4	63
41	13	4	63
42	3	4	63

id_projet	projet	date_debut	date_fin
61	HHHH	2022-10-27	2022-10-30
62	essai n_???	2022-10-28	2022-11-11
63	dernier_essai	2022-10-28	2022-11-07

id_eleves	prenom	nom	mail
1	Kevin	LE GRAND	kevin.le-grand@isen-ouest.yncr
2	Harold	PETIARD	harold.petiard@isen-ouest.yncr
3	David	SCANU	david.scanu@isen-ouest.yncrea.
4	Antoine	ANCELIN	antoine.ancelin@isen-ouest.ync
5	Johann	JOURNAUX	johann.journaux@isen-ouest.ync
6	Charley	LEBARBIER	charley.lebarbier@isen-ouest.y
7	Amadou	DIABY	amadou.diaby@isen-ouest.yncrea
8	Geoffroy	DAUMER	geoffroy.daumer@isen-ouest.ync
9	Dimitri	JULES	dimitri.jules@isen-ouest.yncr
10	Basile	GUERIN	basile.guerin@isen-ouest.yncr
11	Viacheslav	OSWALD	viacheslav.oswald@isen-ouest.y
12	Soraya	DUSSART	ramata-soraya.dussart@isen-ou
13	Guinel	ZOUITEN	guinel.zouiten@isen-ou



Conclusion :

L'appli fonctionne pour toutes les conditions demandées cependant elle reste largement améliorable.

Les améliorations à faire :

- Finir de gérer toutes les entrées faites par l'utilisateur pour éviter tout bug du programme ou de conflit avec la BDD.
- Ajouter la possibilité à l'utilisateur de faire une recherche dans la BDD par projet, dates, prénoms, noms...
- Ajouter une table dans la BDD permettant de connaître le niveau d'un élève.
- Ajouter des conditions au tri des élèves : par niveau, qu'ils n'aient pas été déjà ensemble sur le dernier projet...
-

Tous les fichiers de ce projet sont récupérable sur : <https://github.com/rastakoer?tab=projects>

