

# Bases de données et SQL

## Un mini-Parcoursup

### 1 Structure de la base de données

On examine ici une base de données se rapprochant de celle de Parcoursup (en version très simplifiée) : elle contient des informations sur des étudiants, sur des formations, et les vœux hiérarchisés<sup>1</sup> que font les candidats pour rejoindre les formations.

On considérera trois tables :

1. Une table `etudiants` contient des informations sur les étudiants :
  - Nom (attribut `nom_etudiant`) ;
  - Numéro de dossier (attribut `numero_dossier`) ;
  - Lycée d'origine (attribut `lycee_origine`) ;
  - Moyenne générale (attribut `moyenne`).
  - Spécialités suivies.  
On simplifiera en créant 3 colonnes `spe_math`, `spe_ses`, `spe_lettres` dont le contenu vaudra 1 si l'étudiant suit la spécialité et 0 sinon.
2. Une table `formations` contient des informations sur les formations :
  - Un identifiant (attribut `id_formation`, qui prendra les valeurs 1,2,...) ;
  - Le nom de la formation (attribut `nom_formation`) ;
  - L'établissement où est dispensée la formation (attribut `etablissement`) ;
  - La dominante de la formation : scientifique, économique ou littéraire. (attribut `dominante`, prendra les valeurs `sci`, `eco` ou `lettres`).
3. Une table `vœux` recense les trois vœux (au maximum) que peut faire un étudiant. Elle contient :
  - Une colonne avec les numéros de dossier des étudiants (attribut `dossier`) ;
  - trois colonnes `vœu1`, `vœu2`, `vœu3` correspondant aux formations demandées, qui seront repérées par leur identifiant de la table des Formations.

**Exercice 1.** Quelles colonnes peut-on déclarer comme clés primaires ? Doit-on mettre en place des clés étrangères ?

**Exercice 2.** Donner les syntaxes créant les tables précédentes, en tenant compte des clés décidées ci-dessus.

### 2 Ajout de données

**Exercice 3.** Peupler ces tables :

- Ajouter Lucie, numéro de dossier 1111, de Berthelot, qui a 16.1 de moyenne générale et qui suit Spé Math mais pas SES ni Lettres.
- Ajouter Victor, numéro de dossier 2000, de Teilhard, qui a 11.4 de moyenne générale et qui suit Spé Math et Spé SES mais pas Spé Lettres.

---

<sup>1</sup>Ce serait donc plutôt APB, l'ancêtre de Parcoursup.

### 3 Recherche dans les tables

Vous trouverez sur Cahier de Prépa le fichier parcoursup.db qui contient ces tables remplies de diverses informations. Nous allons maintenant ajouter et extraire des informations de ce fichier.

Dans l'exercice suivant, on aura à faire au maximum une jointure INNER JOIN.

#### Exercice 4.

1. Dans la table des vœux, rentrer les vœux de Lucie (dossier 1111), qui demande les formations 6,1,4 (dans l'ordre des vœux) ; et ceux de Victor (dossier 2000) qui demande les formations 5 et 8 (dans l'ordre des vœux).  
*On utilisera des commandes SQL, et pas l'interface graphique !!!*
2. Afficher les noms des étudiants suivant les spécialités Maths et SES.
3. Afficher la moyenne des moyennes des candidats venant de Berthelot.
4. Y a-t-il des dossiers ayant fait 2 fois le même vœu ?
5. Afficher les noms et moyennes des candidats suivant la Spé SES, triés par moyenne croissante.
6. Lister les lycées de provenance des étudiants de cette table (sans répétition de nom de lycée).
7. Donner les numéros de dossier des étudiants ayant mis la formation 1 en premier vœu.
8. Donner les numéros de dossier des étudiants ayant candidaté à la formation 1 (en position quelconque).
9. Afficher les noms des étudiants ayant candidaté à la formation 1 (en position quelconque).
10. Combien de formations à dominante économique sont répertoriées dans la base ? (rappel : c'est l'ordi qui doit répondre !)
11. Combien d'étudiants ont demandé une MPSI en premier vœu ?
12. Combien ont demandé une prépa ECG en premier vœu ?
13. Donner les noms et numéros de dossier des étudiants ayant demandé la formation 5 en premier vœu, et n'ayant pas fait Spé SES.
14. Lister les noms et vœux des élèves de Louis Le Grand ayant demandé la formation 2 en position quelconque.
15. ...

Dans l'exercice suivant, on opère une double jointure en partant de la table vœux : la première jointure permet de remonter aux étudiants à l'aide de leur numéro de dossier, et la seconde permet de récupérer les informations sur la formation mentionnée dans la colonne Vœu1 de la table des vœux – si on s'intéresse au premier vœu !

#### Exercice 5 (Double jointure).

1. Lister les noms des étudiants qui candidatent en premier vœu à une formation à dominante économique sans avoir suivi la Spé SES.
2. Donner les noms et moyennes des candidats à une MPSI en premier vœu, classés par moyenne décroissante. Recommencer en retirant ceux qui n'ont pas suivi la Spé Math.
3. Lister les noms, numéros de dossier, et lycées d'origine des étudiants qui candidatent en **second** vœu à une formation dispensée par leur lycée d'origine.
4. Donner la valeur moyenne des moyennes des élèves qui candidatent en premier vœu à une formation dispensée par le Lycée Louis Le Grand.
5. ...

## 4 Les commandes UNION, INTERSECT, EXCEPT

On présente ici ces trois commandes :

- La commande UNION renvoie la réunion de deux résultats de recherche (la syntaxe est donc `SELECT (...) UNION SELECT(...)`.  
Si les deux résultats ont des lignes en commun, celles-ci ne sont renvoyées qu'une fois.
- La commande INTERSECT renvoie les lignes communes à deux résultats de recherche (la syntaxe est donc `SELECT (...) INTERSECT SELECT(...)`).
- La commande EXCEPT renvoie les résultats d'une première requête, en y enlevant les résultats qui apparaîtraient dans une autre requête.

NB : ceci ressemble fort à ce qui pourrait être obtenu en enchaînant des conditions (derrière un WHERE) sur une requête, avec des AND et des OR. Ces syntaxes ont en fait un intérêt quand on manipule deux requêtes sur deux tables différentes (ou alors sur des jointures différentes des mêmes tables).

### Exercice 6.

1. Lister les numéros de dossiers des étudiants ayant demandé une MPSI en premier vœu ou en second vœu.  
(on pourra faire l'UNION de deux requêtes s'inspirant de celle de la question 11 de l'exercice 4).  
*Remarque la non-répétition du candidat 9984 qui demande deux MPSI*
2. Lister les noms et numéros de dossier des élèves donc les deux premiers vœux sont pour des formations à dominante économique.
3. Le lycée Louis Le Grand décide<sup>2</sup> de ne pas examiner les dossiers qui ont classé une de ses formations en second vœu, tout en classant une formation d'Henri IV en premier vœu.
  - (a) Afficher les numéros de dossier des étudiants ayant fait un second vœu pour une formation dispensée par le Lycée Louis Le Grand.
  - (b) Afficher les numéros de dossier des étudiants ayant fait un premier vœu pour une formation dispensée par le Lycée Henri IV.
  - (c) À l'aide des deux questions précédentes, générer la liste des numéros de dossier que Louis le Grand refusera d'examiner.

## 5 Modification de tables

**Exercice 7.** Donner les syntaxes SQL permettant les opérations suivantes :

1. Le Lycée Louis Le Grand est rebaptisé Lycée Zinédine Zidane. Modifier la table des formations en conséquence.
2. Suite à la dernière réforme, les prépas ECG ferment leur portes.
  - Exécuter des commandes permettant de retirer toutes les formations ECG de la table des formations. Que se passe-t-il ? Pourquoi ?
  - Retirer tous les vœux ECG des élèves (vœux 5,6,7).  
On devra pour cela passer par 3 requêtes UPDATE / SET : une pour enlever les Voew1, une pour les Voew2, et une pour les Voew3.  
Puis réessayer les commandes précédentes.

---

<sup>2</sup>Ceci est impossible dans la vraie procédure : l'établissement ne connaît pas les autres vœux du candidat.

## 6 Bonus : jointure avec alias

On cherche ici à construire une requête qui détecte si les établissements des formations demandées dans les deux premiers vœux sont identiques.

Quelle jointure doit-on utiliser pour récupérer l'établissement associé à une formation spécifiée en vœu 1 ? et pour un vœu 2 ?

On voit que si on veut faire la double jointure, on se retrouve à joindre deux fois la table `formations` : toutes les colonnes correspondantes seront dédoublées, sans moyen de les distinguer. Un moyen de contourner cela est de joindre la table *sous un alias*.

La syntaxe

```
1 table1 INNER JOIN table2 AS t ON ...
```

effectue une jointure des tables `table1` et `table2`, mais en appelant `t` la table `table2`. Les colonnes issues de `table2` s'appelleront donc par la syntaxe `t.attribut1`, `t.attribut2`, etc.

On peut ainsi joindre plusieurs fois la même table : il suffit de donner à chaque fois un alias différent.

### Exercice 8.

1. Proposer une double jointure avec alias sous la forme

```
1 vœux
2 INNER JOIN formations AS f1 ON ...
3 INNER JOIN formations AS f2 ON ...
```

qui permettra de placer des requêtes pouvant prendre en compte les caractéristiques du vœu1 et du vœu2.

2. En interrogeant cette jointure, donner les numéros de dossier des étudiants ayant fait leurs premier et second vœux dans le même établissement.
3. En effectuant encore une jointure (!) avec la table `etudiants`, donner les noms des élèves associés à ces dossiers.