

# Algèbre

## Partie 1 : Gestion de matrices creuses en SQL

**Base de données fournies.** Pour cet exercice, on vous fournit un fichier `mat.db` qui stocke une base de donnée SQLite3. Pour pouvoir exécuter des commandes sur ce fichier on peut lancer la commande `sqlite3 poly.db` depuis le dossier où se trouve ce fichier. Cette commande lance un interpréteur SQLite3. On peut par exemple taper la commande suivante pour avoir le nombre d'enregistrements dans la table `mat` :

```
SELECT COUNT(*) FROM mat ;
```

**Affichage dans SQLite3.** Pour avoir un affichage plus lisible avec SQLite3, on pourra utiliser les deux commandes suivantes (à copier directement dans le shell SQLite3) :

```
.header on  
.mode column
```

**Représentation creuse des matrices.** Cette base de données stocke des matrices de réels en représentation creuse. Dans la représentation creuse, une matrice  $M$  est représentée par des triplets  $(x, y, z)$  avec  $x, y$  des coordonnées de la matrice  $M$  et  $z$  non nul indique que  $M_{x,y} = z$ . Ainsi dans la représentation creuse d'une matrice, pour chaque paire de coordonnées  $(x, y)$  de la matrice il existe au plus un élément  $(x, y, z)$  et cet élément n'existe que si  $M_{x,y} \neq 0$ .

**Question 1.** Peut-on calculer la représentation creuse de la somme ou du produit de deux matrices en représentation creuse dont on ne connaît pas les dimensions ?

**Base de donnée fournie.** Notre base de donnée contient une table `mat` qui stocke plusieurs matrices  $M^0, \dots, M^k$ . Cette table a quatre attributs `identifiant`, `x`, `y` et `z`. Chaque enregistrement (`identifiant=i`, `x=x`, `y=y`, `z=z`) décrit une partie de la représentation creuse de la matrice  $M^i$  (et donc  $M_{x,y}^i = z$ ).

**Question 2.** Combien de matrices différentes sont stockées dans la base de données ? On supposera qu'il n'existe pas de matrices complètement nulles qui ne serait décrite par aucun enregistrement.

**Représentations valides et canoniques.** Une représentation creuse est *valide* s'il n'existe pas deux triplets  $(x, y, z)$  et  $(x', y', z')$  avec  $x = x'$  et  $y = y'$  et elle est *canonique* s'il n'existe pas de triplet  $(x, y, z)$  avec  $z = 0$ .

**Question 3.** On souhaite modifier le schéma de la base de données pour interdire les représentations qui ne sont pas valides. Comment peut-on faire ?

**Question 4.** Écrire une requête SQL qui détermine les matrices qui ne sont pas canoniques. On supposera qu'il n'existe pas de matrices complètement nulles qui ne serait décrite par aucun enregistrement.

**Question 5.** Écrire une requête SQL qui détermine les matrices qui sont invalides. On supposera qu'il n'existe pas de matrices complètement nulles qui ne serait décrite par aucun enregistrement.

**Question 6.** Proposer une requête SQL qui calcule le produit des matrices d'identifiants 0 et 1. Votre résultat doit être une représentation valide et canonique.

## Partie 2 – Résolution de systèmes linéaires et application à l'algèbre

Pour traiter cette partie, vous pouvez choisir librement entre les langages C ou Ocaml.

### 2.1 Implémentation des rationnels

On représentera les rationnels en utilisant un couple d'entiers.  $(n, d)$  représentera le nombre  $\frac{n}{d}$ . Bien entendu,  $d$  ne pourra jamais être nul. Par ailleurs, plusieurs couples peuvent représenter le même nombre.

1. Ecrire une fonction `pgcd` prenant en paramètre deux entiers non nuls en renvoyant leur pgcd.
2. Ecrire une fonction `irreductible` qui prend en paramètre un couple d'entiers  $(n, d)$  et renvoie le couple  $(0, 1)$  si  $n$  est nul, et  $(u, v)$  avec  $u$  et  $v$  premiers entre eux et  $v$  positif. Pour ceux qui ont choisi Ocaml, on lèvera une exception si le couple rentré en paramètre ne représente pas un rationnel. Pour ceux qui ont choisi le langage C, on arrêtera le programme avec un `exit` et un message d'erreur.
3. Ecrire des fonctions `somme`, `produit` et `quotient` qui effectue respectivement la somme, le produit, le quotient de deux rationnels. On renverra une version simplifiée du nombre obtenu.
4. Enfin, écrire une fonction `produit_par_entier` qui prend en paramètre un rationnel et un entier et renvoie le produit des deux, sous forme rationnelle.

### 2.2 Inversion de matrices

On va implémenter la méthode de Gauss-Jordan pour inverser une matrice. On rappelle que cette méthode s'effectue en deux temps :

- On effectue des opérations élémentaires sur la matrice pour rendre cette dernière triangulaire supérieure. Cette phase est appelée la phase de descente.
- En normalisant les coefficient diagonaux de la matrice, et en effectuant à nouveau des opérations élémentaires sur les lignes, on transforme la matrice en la matrice identité. Cette phase est appelée la phase de remontée.

En parallèle de ces opérations effectuées sur la matrice initiale, on effectue les mêmes opérations, dans le même ordre sur une matrice qui initialement est la matrice identité. A la fin de l'algorithme, cette dernière matrice est alors l'inverse de la matrice de départ. Si le processus bloque à un moment, cela signifie que la matrice initiale n'est pas inversible.

1. Ecrire trois fonctions permettant d'implémenter les trois opérations élémentaires sur les matrices, à savoir :
  - $L_i \leftrightarrow L_j$  : permute les lignes  $i$  et  $j$  de la matrice.
  - $L_i \leftrightarrow L_i + aL_j$  : ajoute  $a$  fois la ligne  $j$  à la ligne  $i$  de la matrice.
  - $L_i \leftrightarrow aL_i$  : multiplie la ligne  $i$  de la matrice par un nombre  $a$  non nul.
2. Ecrire une fonction `descente` qui prend en paramètre deux matrices de rationnels, et qui effectue la phase de descente, en fonction de la première matrice.
3. Ecrire une fonction `inverse` qui prend une matrice en paramètre et renvoie sa matrice inverse. On gèrera le cas où la matrice n'est pas inversible.

Quelques exemples pour tester vos algorithmes.

$$\text{— } A = \begin{pmatrix} 0 & 2 & 3 \\ -1 & 3 & 2 \\ 4 & 7 & -2 \end{pmatrix} \text{ est inversible et } A^{-1} = \begin{pmatrix} 4/9 & -5/9 & 1/9 \\ -2/15 & 4/15 & 1/15 \\ 19/45 & -8/45 & -2/45 \end{pmatrix}$$

—  $A = \begin{pmatrix} 7 & -2 & 5 \\ 2 & 8 & 1 \\ 9 & 6 & 6 \end{pmatrix}$  n'est pas inversible.

### 2.3 Résolution de systèmes linéaires

On cherche à résoudre un système de la forme  $AX = B$ , avec  $A \in \mathcal{M}_{n,p}(\mathbb{Q})$  et  $B \in \mathcal{M}_{p,1}(\mathbb{Q})$ . Quitte à appliquer la fonction `descente` de la partie précédente, on peut supposer que le système est échelonné.

On obtiendra par exemple :

$$\begin{cases} 3x_1 + 4x_2 - 5x_3 + 8x_4 + 21x_5 = 2 \\ \phantom{3x_1} + 1x_3 + 4x_4 - 2x_5 = 9 \\ \phantom{3x_1} \phantom{1x_3} + 5x_4 = 12 \\ \phantom{3x_1} \phantom{1x_3} \phantom{5x_4} 0 = 0 \end{cases}$$

Après échelonnement, on aura :  $A = \begin{pmatrix} 3 & 4 & -5 & 8 & 21 \\ 0 & 0 & 1 & 4 & -2 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$  et  $B = \begin{pmatrix} 2 \\ 9 \\ 12 \\ 0 \end{pmatrix}$ .

Notons dès à présent que si, après échelonnement, le dernier coefficient de  $B$  était non nul, ce système n'aurait aucune solution. Par ailleurs, ici, les variables  $x_2$  et  $x_5$  sont des variables libres. En les passant de l'autre côté des égalités, le système devient :

$$\begin{cases} 3x_1 - 5x_3 + 8x_4 = 2 - 4x_2 - 21x_5 \\ \phantom{3x_1} + 1x_3 + 4x_4 = 9 + 2x_5 \\ \phantom{3x_1} \phantom{1x_3} + 5x_4 = 12 \\ \phantom{3x_1} \phantom{1x_3} \phantom{5x_4} 0 = 0 \end{cases}$$

Pour résoudre le système, on peut donc trouver une solution particulière en fixant  $x_2 = 0$ ,  $x_5 = 0$  et résoudre :

$$\begin{cases} 3x_1 - 5x_3 + 8x_4 = 2 \\ \phantom{3x_1} + 1x_3 + 4x_4 = 9 \\ \phantom{3x_1} \phantom{1x_3} + 5x_4 = 12 \end{cases}$$

Pour trouver une base du sous-espace vectoriel des solutions du système homogène, on peut fixer  $x_2 = 1$  et  $x_5 = 0$ , puis  $x_2 = 0$  et  $x_5 = 1$ . On va donc résoudre :

$$\begin{cases} 3x_1 - 5x_3 + 8x_4 = 4 \\ \phantom{3x_1} + 1x_3 + 4x_4 = 0 \\ \phantom{3x_1} \phantom{1x_3} + 5x_4 = 0 \end{cases} \text{ puis } \begin{cases} 3x_1 - 5x_3 + 8x_4 = -21 \\ \phantom{3x_1} + 1x_3 + 4x_4 = 2 \\ \phantom{3x_1} \phantom{1x_3} + 5x_4 = 0 \end{cases}$$

1. On va représenter les solutions du système par un type constitué de deux champs : une valeur particulière, et une liste de vecteurs, qui formera une base du sous-espace vectoriel des solutions du système homogène. Quelle implémentation proposez-vous ? Que faire en cas d'absence de solution ?
2. Ecrire une fonction qui prend en paramètre les matrices  $A$  et  $B$  et renvoie l'ensemble des solutions du système associé. Pour tester, l'ensemble des solutions de l'exemple ci-dessus est :

$$\left\{ \left( -\frac{101}{15} - \frac{4}{3}x - \frac{11}{3}y, x, -\frac{3}{5} + 2y, \frac{12}{15}, y \right), (x, y) \in \mathbb{Q}^2 \right\}.$$

## 2.4 Application à l'algèbre

On utilisera le type précédemment défini pour les espaces affines, pour représenter les espaces vectoriels en mettant systématiquement la valeur du point au vecteur nul.

1. Ecrire une fonction prenant en paramètre une famille de vecteurs, et renvoyant un booléen, indiquant si la famille est libre.
2. Ecrire une fonction prenant en paramètre une famille de vecteurs, et qui extraira de cette famille une base de l'espace vectoriel engendré par les combinaisons linéaires de ces derniers.
3. Ecrire une fonction prenant en paramètre deux espaces vectoriels et qui renverra l'espace vectoriel obtenu par intersection des deux.

### Quelques exemples pour tester vos algorithmes.

— Dans  $\mathbb{Q}^5$ , on se donne :

$$e_1 = (1, 3, 2, 4, -2), \quad e_2 = (0, 3, 2, -3, 1), \quad e_3 = (-4, 1, 6, 2, 1), \quad e_4 = (-3, 7, 10, 3, 0).$$

Alors la famille  $(e_1, e_2, e_3)$  est libre, mais pas  $(e_1, e_2, e_3, e_4)$ .

— Dans  $\mathbb{Q}^5$ , on se donne :

$$f_1 = (-12, 3, 18, 6, 3), \quad f_2 = (-3, 7, 10, 3, 0), \quad f_3 = (1, 3, 2, 4, -2), \quad f_4 = (0, 6, 4, -6, 2), \\ f_5 = (-2, 10, 12, 7, -2).$$

On note  $F = Vect(f_1, f_2, f_3, f_4, f_5)$ . Alors  $\dim(F) = 3$