

## Devoir n°1

### Problème : base de données en simulation de la cinétique d'un gaz parfait

La théorie cinétique des gaz vise à expliquer le comportement macroscopique d'un gaz à partir des mouvements des particules qui le composent. Depuis la naissance de l'informatique, de nombreuses simulations numériques ont permis de retrouver les lois de comportement de différents modèles de gaz comme celui du gaz parfait. Ce sujet s'intéresse à un gaz parfait monoatomique. Nous considérerons que le gaz étudié est constitué de particules sphériques, toutes identiques, de masse  $m$  et de rayon  $R$ , confinées dans un récipient rigide. Les simulations seront réalisées dans un espace à une, deux ou trois dimensions ; le récipient contenant le gaz sera, suivant le cas, un segment de longueur  $L$ , un carré de côté  $L$  ou un cube d'arête  $L$  avec des coordonnées modélisées de 0 à  $L$ .

Dans le modèle du gaz parfait, les particules ne subissent aucune force (leur poids est négligé) ni aucune autre action à distance. Elle n'interagissent que par l'intermédiaire de chocs, avec une autre particule ou avec la paroi du récipient. Ces chocs sont toujours élastiques, c'est-à-dire que l'énergie cinétique totale est conservée.

À partir d'une situation initiale, où  $N$  particules de rayon  $R$  sont placées aléatoirement à l'intérieur d'un récipient de taille  $L$  dans un espace à  $D$  dimensions, les positions et vitesses des particules vont évoluer au gré du déplacement des particules, des différents chocs entre elles et des rebonds sur les parois. On appelle événement chaque choc ou rebond.

On suppose que l'on dispose d'une base de données bdd dans laquelle ont été enregistré lors de différentes simulations (effectuées en simulation python), dans lesquelles les particules ne sont pas nécessairement identiques, les données des événements (rebonds ici). On donne la structure de cette base de données dans la figure 1

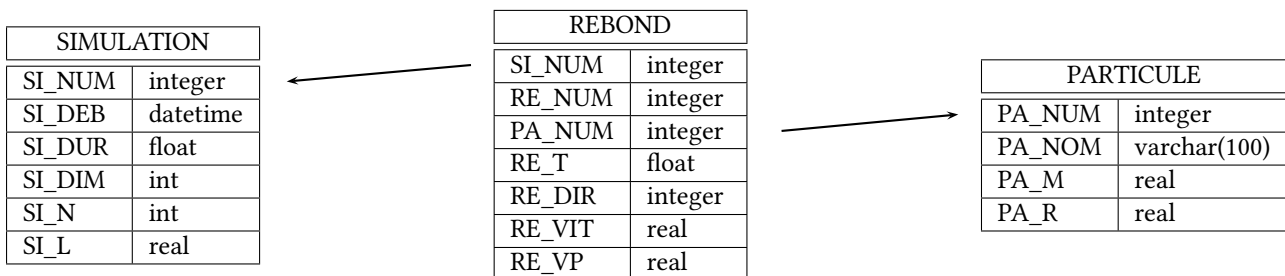


Figure 1: Structure physique de la base de données des résultats des simulations

Cette base comporte les trois tables suivantes :

- la table SIMULATION, donne les caractéristiques de chaque simulation effectuée. Elle contient les colonnes
  - SI\_NUM numéro d'ordre de la simulation
  - SI\_DEB date et heure du lancement du programme de simulation
  - SI\_DUR durée (en secondes) de la simulation (il ne s'agit pas du temps d'exécution du programme, mais du temps simulé)
  - SI\_DIM nombre de dimensions de l'espace de simulation
  - SI\_N nombre de particules pour cette simulation
  - SI\_L (en mètres) taille du récipient utilisé pour la simulation
- la table PARTICULE, des types de particules considérées. Elle contient les colonnes
  - PA\_NUM numéro (entier) identifiant le type de particule
  - PA\_NOM nom de ce type de particule
  - PA\_M masse de la particule (en grammes)
  - PA\_R rayon (en mètres) de la particule

– la table REBOND, liste les chocs des particules avec les parois du récipient.

Elle contient les colonnes

- SI\_NUM numéro d'ordre de la simulation ayant généré ce rebond
- RE\_NUM numéro d'ordre du rebond au sein de cette simulation
- PA\_NUM numéro du type de particule concernée par ce rebond
- RE\_T temps de simulation (en secondes) auquel ce rebond est arrivé
- RE\_DIR paroi concernée : entier non nul de l'intervalle  $[-SI\_DIM, SI\_DIM]$  donnant la direction de la normale à la paroi. Ainsi -2 désigne la paroi située en  $y = 0$  alors que 1 désigne la paroi située en  $x = L$
- RE\_VIT norme de la vitesse de la particule qui rebondit (en  $m \cdot s^{-1}$ )
- RE\_VP valeur absolue de la composante de la vitesse normale à la paroi (en  $m \cdot s^{-1}$ )

1. Donner pour chaque table SIMULATION, PARTICULE et REBOND une clé primaire.
2. Combien d'attribut possède la table REBOND ?
3. Expliquer le type de l'attribut PA\_NOM.
4. Quel est le domaine de l'attribut SI\_NUM ?
5. Comment désigne t-on l'attribut SI\_NUM de la table REBOND ?
6. Dans cette base de données, combien y-a-t-il d'entités et d'associations ?
7. De quel type est l'association entre les tables SIMULATION et REBOND ? Entre les tables REBOND et PARTICULE ?
8. Écrire une requête qui détermine la masse minimale des particules des simulations.
9. Écrire une requête qui détermine le nombre de simulation de durée supérieure à une minute.
10. Écrire une requête qui détermine le ou les noms des particules qui possèdent la plus grande masse, classé par ordre alphabétique croissant.
11. Écrire une requête SQL qui donne le nombre de simulations effectuées pour chaque nombre de dimensions de l'espace de simulation.
12. Écrire une requête SQL qui donne, pour chaque simulation, le nombre de rebonds enregistrés et la vitesse moyenne des particules qui frappent une paroi.
13. Écrire une requête SQL qui, pour une simulation  $n$  donnée, calcule, pour chaque paroi, la variation de quantité de mouvement due aux chocs des particules sur cette paroi tout au long de la simulation. On se rappellera que lors du rebond d'une particule sur une paroi la composante de sa vitesse normale à la paroi est inversée, ce qui correspond à une variation de quantité de mouvement de  $2m|v_{\perp}|$  où  $m$  désigne la masse de la particule et  $v_{\perp}$  la composante de sa vitesse normale à la paroi.
14. Écrire une requête SQL qui renvoie les simulations et la taille du récipient dont le nombre de rebonds de particules de nom nomp dépasse 1000.



« Vous ne pouvez pas comprendre la récursivité sans d'abord avoir compris la récursivité. »

« Si vous ne savez toujours pas ce qu'est la récursivité, relisez cette phrase. »

- On donne les clés primaires :
  - table SIMULATION : SI\_NUM
  - table PARTICULE : PA\_NUM
  - table REBOND : le couple (SI\_NUM, RE\_NUM).
- La table REBOND possède 7 attributs.
- Expliquer le type de l'attribut PA\_NOM. Il s'agit d'un varchar(100), c'est à dire une chaîne de caractère de longueur 100 au maximum.
- Quel est le domaine de l'attribut SI\_NUM ? Il s'agit du domaine des entiers.
- Comment désigne t-on l'attribut SI\_NUM de la table REBOND ? C'est une clé étrangère. Plusieurs enregistrement de la table REBOND peuvent avoir le même attribut SI\_NUM, qui désigne le numéro de simulation de la table SIMULATION, dans laquelle il joue le rôle de clé primaire.
- Dans cette base de données, nous avons 3 entités : SIMULATION, REBOND, PARTICULE.  
On a 2 associations : SIMULATION - REBOND et REBOND - PARTICULE.
- L'association entre les tables SIMULATION et REBOND est de type \* - 1 : une simulation contient un ou plusieurs rebonds, et un rebond fait partie d'une simulation. L'association entre les tables REBOND et PARTICULE est aussi de type 1 - \* : un rebond concerne un type de particule et un type de particule concerne un ou plusieurs rebonds.
- Écrire une requête qui détermine la masse minimale des particules des simulations. Si on suppose que toute les particules participent à au moins une simulation (sinon, elles ne seraient pas dans la base de données) :

```
SELECT MIN(PA_M) FROM PARTICULE
```

Si un type de particule ne participe à aucune simulation, il peut s'avérer impossible de le déterminer et de les éliminer. En effet, on peut imaginer des particules de simulations qui n'ont pas eu de rebond, et donc il peut-être alors impossible de déterminer les types de particules participant à au moins une simulation.

- Écrire une requête qui détermine le nombre de simulation de durée supérieure à une minute.

```
SELECT COUNT(*) FROM SIMULATION
WHERE SI_DUR >= 60
```

La condition WHERE effectue une pré-sélection (en amont), le comptage est effectué après cette pré-sélection.

- Écrire une requête qui détermine le ou les noms des particules qui possèdent la plus grande masse, classé par ordre alphabétique croissant.

```
SELECT PA_NOM FROM PARTICULE
WHERE PA_M = (SELECT MAX(PA_M) FROM PARTICULE)
ORDER BY PA_NOM
```

- Écrire une requête SQL qui donne le nombre de simulations effectuées pour chaque nombre de dimensions de l'espace de simulation. On accompagne le compteur de la dimension (on pourrait logiquement mettre la dimension en première colonne) :

```
SELECT COUNT(*), SI_DIM FROM SIMULATION
GROUP BY SI_DIM
```

- Écrire une requête SQL qui donne, pour chaque simulation, le nombre de rebonds enregistrés et la vitesse moyenne des particules qui frappent une paroi.

```
SELECT SI_NUM, COUNT(RE_NUM), AVG(RE_VIT) FROM REBOND
GROUP BY SI_NUM
```

13. Écrire une requête SQL qui, pour une simulation  $n$  donnée, calcule, pour chaque paroi, la variation de quantité de mouvement due aux chocs des particules sur cette paroi tout au long de la simulation. On se rappellera que lors du rebond d'une particule sur une paroi la composante de sa vitesse normale à la paroi est inversée, ce qui correspond à une variation de quantité de mouvement de  $2m|v_{\perp}|$  où  $m$  désigne la masse de la particule et  $v_{\perp}$  la composante de sa vitesse normale à la paroi.

```
SELECT RE_DIR, SUM(2 * PA_M * RE_VP) FROM REBOND
JOIN PARTICULE ON REBOND.PA_NUM = PARTICULE.PA_NUM
WHERE SI_NUM = n
GROUP BY RE_DIR
```

14. Écrire une requête SQL qui renvoie les simulations et la taille du récipient dont le nombre de rebonds de particules de nom `nomp` dépasse 1000.

```
SELECT SI_NUM, SI_L FROM REBOND AS R
JOIN PARTICULE AS P ON R.PA_NUM = P.PA_NUM
JOIN SIMULATION AS S ON S.SI_NUM = R.SI_NUM
WHERE PA_NOM = nomp
GROUP BY SI_NUM
HAVING COUNT(*) >= 1000
```

La condition HAVING s'effectue en aval, c'est une post-sélection.