

# BASES DE DONNÉES AVANCÉES

## *Optimisation des performances*

Travaux dirigés

30 novembre 2023

### **Exercice 1 : Structures simples**

Supposons une mémoire organisée en blocs de taille de 8k, soit 8192 octets. Un pointeur sera codé sur 8 octets. Soit une relation CLIENTS(numclient, nom, prenom, adresse) ; chaque tuple a une taille de  $(8+41+41+151)$  octets. La table contient 100.000.000 de tuples, sachant que les blocs ne sont remplis qu'à 80% :

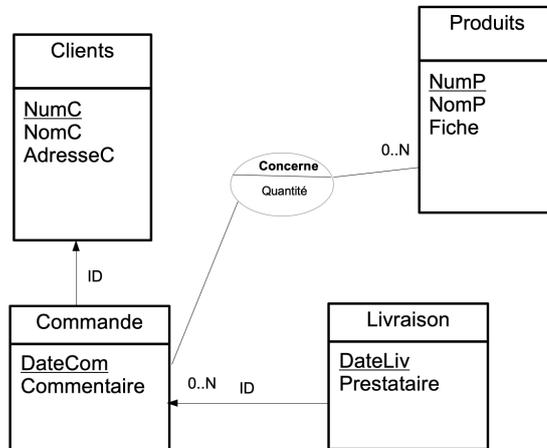
1. Calculez le nombre de blocs occupés par la relation
2. Calculez le nombre de blocs occupés par un index dense sur les numéros de clients
3. Combien de chargements de blocs faut-il effectuer au plus pour retrouver une valeur de numéro de client dans la relation sans index ? Avec l'index ?
4. Même question avec un index B-Arbre sur les numéros de clients.

### **Exercice 2 : B-Arbre**

1. Construire un B-Arbre pour les valeurs suivantes : 49, 17, 21, 16, 34, 42, 81, 39, 41, 27, 1, 5, 83, 6, 100, 51
  - Considérer  $N=3$  (un nœud contient 3 valeurs, 4 pointeurs).
  - Faire un nouveau dessin pour chaque nouveau nœud inséré.

### Exercice 3 : requêtes et chemins d'accès

On considère que le schéma conceptuel suivant a été traduit en relationnel selon l'algorithme vu en cours.



1. Quelle relation est susceptible d'être la plus volumineuse dans le temps ?
2. Quels sont les attributs présents dans cette relation ? Quelle est la clé ?
3. Essayez de prévoir, pour chaque requête suivante du cahier des charges, les index qui seront pertinents.
  1. Listez les numéro de produits en précisant, pour chacun, son nom.
  2. Même question mais dans l'ordre alphabétique des noms.
  3. Liste des produits qui portent un nom donné.
  4. Une requête très fréquente consistera à connaître les produits commandés par un client donné. Parfois, on restreindra cette requête à une date/période donnée. L'index sur la clé primaire de "Concerne", qui existe de toutes façons, est-il utilisé ?
4. Les requêtes suivantes posent des problèmes de performance. Que faire ?
  1. 

```
select distinct Nump, Nomp
from produits;
```
  2. 

```
select distinct nomc, nomp
from clients, produits, concerne, commandes, livraisons
where clients.numC = commandes.numc
and commandes.datecom = concerne.datecom
and commandes.numc=concerne.numc
and produits.nump=concerne.nump;
```
5. La relation "Concerne" devient trop lente. Que faire pour tenir compte du fait que la majorité des requêtes portent sur les commandes de l'année ?