C018SA-W2-S3

SEMAINE 2 : Indexation

1. Introduction
2. Hiérarchie de mémoire
3. **Fichiers indexés**
4. Arbre-B
5. Hachage
6. Hachage dynamique
7. Multi-hachage

Fichiers indexés

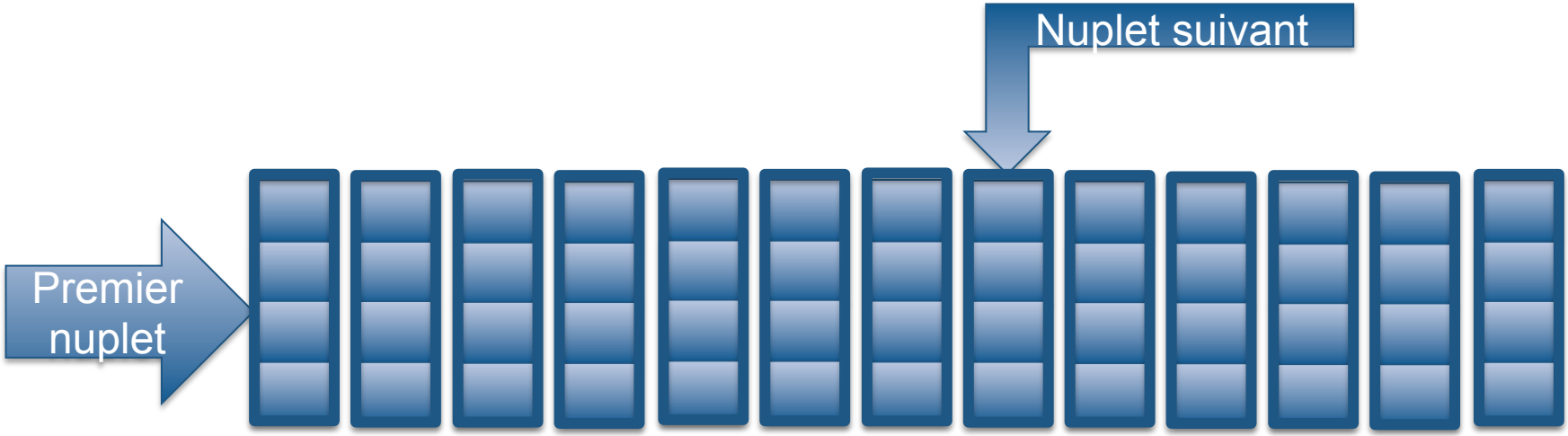
- Le fichier indexé consiste en
  - Une séquence de nuplets (on parle parfois d'enregistrement)
  - Un index sur un des attributs des nuplets pour accès direct : $\sigma_{A=a}(R)$
- Interface – accès séquentiel

```
LirePremier/Suivant(NomFichier,test,X)
EcrirePremier/Suivant(X,NomFichier,test)
SupprimerPremier/Suivant(NomFichier,test)
```

- Interface – accès direct

```
Lire(NomFichier,ValeurIndex,test,X)
```

Accès séquentiel



- Lire/écrire/supprimer
- Organisation ?
  - ✓ nuplet taille fixe et variable

Taille fixe

La taille du nuplet est fixe
- Nombre fixe d'attributs
- Taille fixe des attributs

Insertion en fin de fichier

Suppression
- Compression – coûteux
- Alternative ?
- Un bit pour indiquer que le nuplet est vide
- Insertion : trouver un nuplet vide

Les données sont rarement de taille fixe

- Que feriez-vous ?

Taille variable

La taille du nuplet est variable

- Suppression : un bit pour indiquer que le nuplet est vide
- Insertion : trouver un nuplet vide assez grand
- Mise-à-jour qui change la taille
  - ✓ Suppression/insertion
  - ✓ Réserver de la place
- Problème de fragmentation

Hybride

- Taille Fixe + débordement

Accès direct : trouver un nuplet

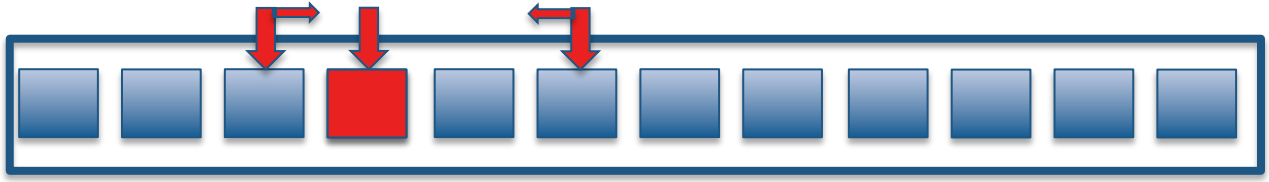
Le fichier est trié suivant un attribut

Solution 1 : parcourir tout le fichier pour trouver le nuplet

- Lecture de N/2 blocs en moyenne

Solution 2 : divide and conquer

- Lecture de log(N) blocs au maximum

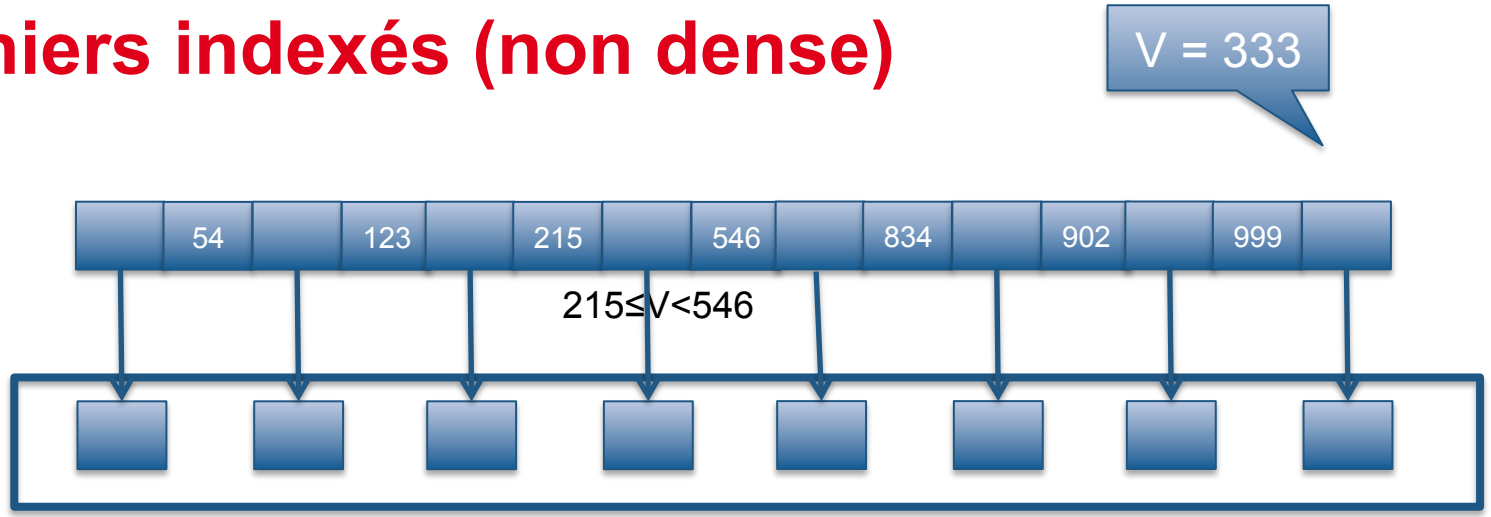


Accès direct : trouver un nuplet

Solution 3 : Utiliser une structure de données auxiliaire

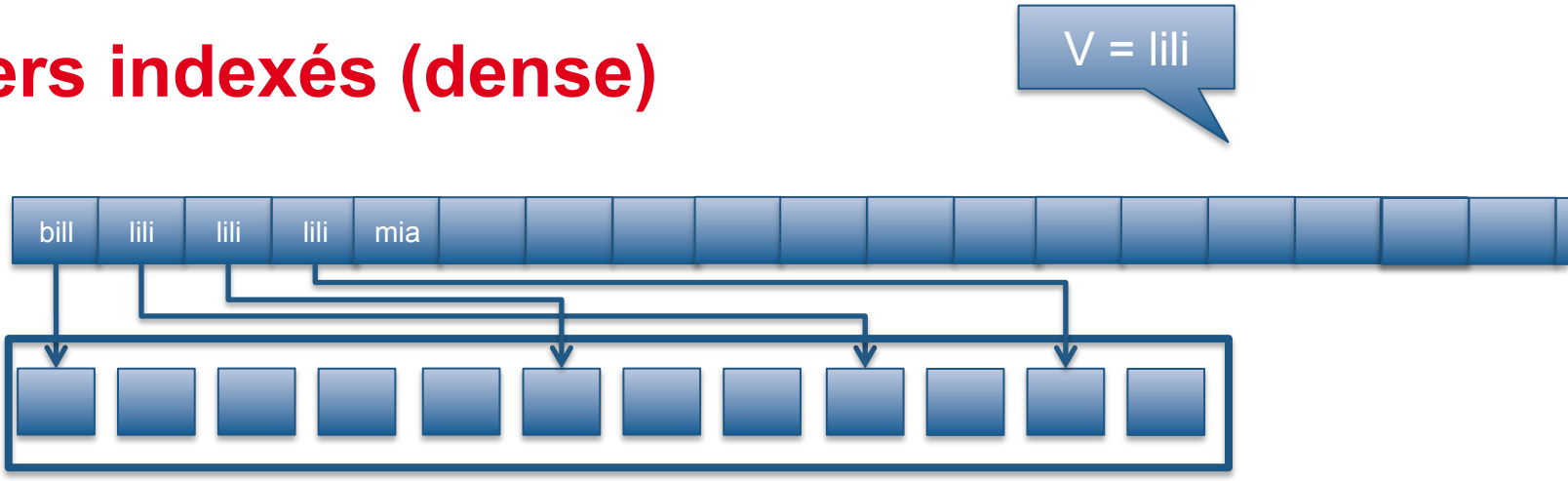
- ✓ On garde si possible cette structure en mémoire
- ✓ Permet d'accéder facilement au nuplet par son adresse
    1. Donne le bloc où se trouve le nuplet cherché
    2. Donne l'adresse dans le bloc
- ✓ Lecture uniquement du bloc utile (des blocs utiles)

Fichiers indexés (non dense)



- Le fichier est trié suivant un attribut
- On construit un index pour cet attribut
  - ✓ Seules les valeurs limites sont représentées
  - ✓ Non dense

Fichiers indexés (dense)



- Le fichier n'est pas trié dans l'ordre de cet attribut
  - ✓ On ne peut pas le trier suivant plusieurs valeurs
- Toutes les valeurs de l'attribut sont représentées
  - ✓ Dense : une adresse de nuplet par valeur
  - ✓ Pour une valeur d'index,

    autant d'adresses que de nuplets avec cette valeur

Comparaison (index en mémoire)

Non dense

Lecture

- On lit un seul bloc (sauf si les résultats ne tiennent pas sur un seul bloc)

Insertion dans un bloc

- Rien à faire s'il y a la place

Suppression

- Pas grand chose à faire en général

Dense

Lecture

- Lire un bloc pour chaque nuplet recherché (dans le pire des cas)

Insertion dans un bloc

- Insérer la clé dans l'index

Suppression

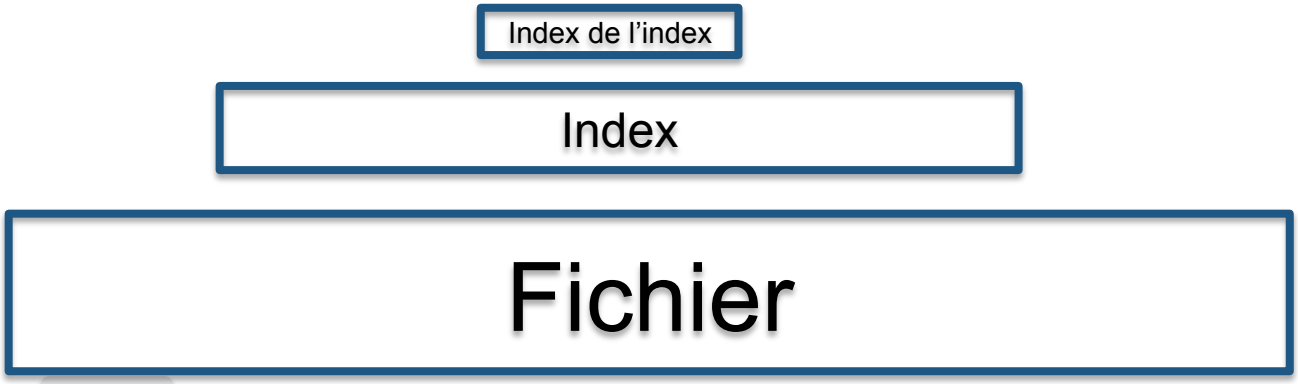
- Supprimer la clé de l'index

Fichiers indexés

Et si l'index devient trop grand ?

Que feriez-vous ?

On indexe l'index – On invente l'arbre B !



MERCI