

# LIFBDW2 – Bases de données avancées

## TD9 – Principales structures d'index

Licence informatique – Automne 2020–2021

*Les questions marquées du symbole (†) sont à préparer pour la séance*

### Exercice 1 : Structures simples

Supposons une mémoire organisée en blocs de taille de 4k, soit 4096 octets. Un pointeur sera codé sur 8 octets. Soit une relation *FILMS*(*numfilm*, *titre*, *date*, *numstudio*) triée sur sa clé primaire; chaque tuple a une taille de  $(8 + 16 + 16 + 8)$  octets. Calculez le nombre de blocs occupés par les objets suivants lorsque la table contient 1.000.000 de tuples, sachant que les blocs ne sont remplis qu'à 80% :

1. La relation
2. Un index primaire dense sur les numéros de films
3. Un index primaire creux sur les numéros de films
4. Combien de chargements de blocs faut-il effectuer au plus pour retrouver une valeur dans la relation sans index? Avec l'index primaire creux?
5. Même question avec un deuxième niveau sur cet index.
6. On suppose qu'il y a 80% de valeurs distinctes dans les titres. Calculez le nombre de blocs d'un index sur le titre. N'oubliez pas le bucket utilisé pour chaque valeur; rappelons qu'un bucket est une liste de pointeurs elle-même organisée en blocs. On supposera que les blocs stockant les buckets sont pleins.

### 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.