

Architecture d'un SGBD
TD : Stockage et indexation

Exercice 1:

Soit la table Film = (Titre, Année) suivante :

Titre	Année
Vertigo	1958
Brazil	1984
Twin peaks	1990
Underground	1995
Easy Rider	1969
Psychose	1960
Greystoke	1984
Shining	1980
Annie Hall	1977
Jurassic park	1992
Metropolis	1926
Manhattan	1979
Reservoir Dogs	1992
Impitoyable	1992
Casablanca	1942
Smoke	1995

1. Expliquer l'organisation séquentielle et représenter un exemple sur la relation FILM donnée ci-dessus. Montrer le fichier après une insertion et quelques suppressions d'articles.
2. Stocker les données suivant une organisation d'arbre B d'ordre 2.
3. Stocker les données suivant une organisation d'arbre B+ d'ordre 2

Exercice 2 : Organisation directe

1. Construire une structure de hachage statique pour des enregistrements dont les clés sont : (2,3, 5, 11, 19, 23, 29, 31, 7, 9, 20, 22). On suppose qu'on peut stocker jusqu'à 4 enregistrements par bloc et qu'on a 5 blocs primaires. Pour la gestion de collisions, utilisez la technique :
 - a. *Unchained overflow*
 - b. *Chained overflow*
2. Construire une structure de hachage dynamique pour des enregistrements dont les clés sont : (2,3, 5, 11, 19, 23, 29, 31, 7, 9, 20, 22). On suppose qu'on peut stocker jusqu'à 4 enregistrements par bloc.
3. Décrire l'évolution de la structure précédente (2) après les opérations suivantes :
 - a) Suppression de l'enregistrement 11.
 - b) Suppression de l'enregistrement 31.

Exercice 3 :

1. Soit la table Musicien = (Nom, Année) suivante :

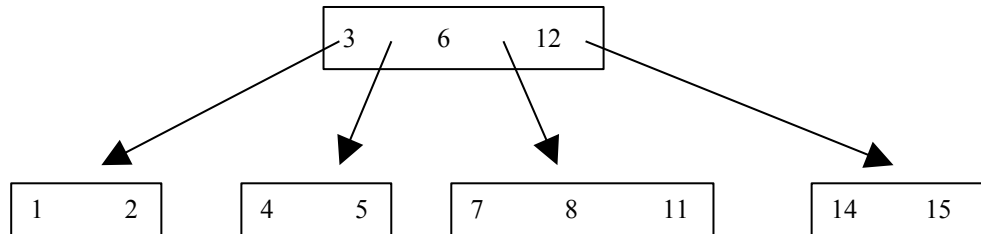
Nom	Année
Monteverdi	1589
Couperin	1668
Bach	1685
Rameau	1684
Debussy	1862
Ravel	1875
Mozart	1756
Faure	1856

- a) Construire un index sur la date de naissance des musiciens ayant la structure d'un B-arbre d'ordre 2

- b) Construire un index sur les noms des musiciens ayant la structure d'un B-arbre d'ordre 2

Exercice 4 :

Soit le fichier séquentiel suivant (on ne donne pour chaque article que la clé sur laquelle on construit l'arbre) : 1 15 3 12 6 4 11 7 2 5 14 8 9 17 10 13 16. L'index organisé selon la structure d'un arbre B d'ordre 2 après insertion des clés 1 15 3 12 6 4 11 7 2 5 14 8 est montré sur la figure suivante :



Donner l'arbre résultant de l'insertion de tous les articles du fichier séquentiel.

Combien de nœuds différents (racine et feuilles compris) doit-on parcourir dans l'index pour répondre à la requête qui cherche des articles dont la clé appartient à l'intervalle [5, 10] ?

Exercice 5 :

On propose de construire un arbre B+

- a. On suppose que l'arbre est initialement vide, que les clés sont insérées dans l'ordre croissant de leur valeur et que chaque nœud peut contenir au plus 2 clés et 3 pointeurs.

Construire l'arbre B+ contenant les clés suivantes : (2, 5, 7, 17, 19, 23, 29, 31).

- b. Décrire les différentes étapes composant les opérations suivantes :
- Recherche de l'enregistrement de clé 11
 - Recherche de l'enregistrement des clés comprises entre 7 et 17 inclus.
 - Insertion de la clé 9, de la clé 10, de la clé 8 puis de la clé 40.
 - Suppression de la clé 29, puis de la clé 5

Exercice 6 :

Index dense et non-dense

Soit un fichier tel que chaque page peut contenir 10 articles. On indexe ce fichier avec un niveau d'index (un seul), et on suppose qu'une page d'index contient 100 paires (clé, pointeur). Si n est le nombre d'articles, donner la fonction de n permettant d'obtenir le nombre minimum de pages pour les structures suivantes :

1. Fichier séquentiel non ordonné avec un index dense.
2. Fichier trié sur la clé avec un index non-dense.

B-Arbre

On reprend les hypothèses précédentes et on indexe le fichier avec un arbre-B+. Les feuilles de l'arbre contiennent donc des pointeurs vers le fichier, et les nœuds internes des pointeurs vers d'autres nœuds. On suppose qu'une page d'arbre B+ est pleine à 70 % (69 clés, 70 pointeurs).

1. Le fichier est trié sur la clé et indexé par un arbre B+ dense. Donner (1) le nombre de niveaux de l'arbre pour un fichier de 1 000 000 d'articles, (2) le nombre de pages utilisées (y compris l'index), et (3) le nombre d'accès disque pour rechercher un article par sa clé.
2. On effectue une recherche par intervalle ramenant 1000 articles. Décrire la recherche et donner le nombre d'accès disque dans le pire des cas.

Exercice 7 : Stockage e indexation

On a une relation R (A, B, C, D, E) organisée en arbre B+ sur l'attribut A, clé de la relation. Un index dense organisés en arbre B+ est associé à l'attribut B. Les pages de données de l'index sont composées de couples (B, A). Le nombre moyen de nuplets auxquels on accède à partir d'une valeur de l'attribut d'indexation B est de 10. La relation comporte 2.000.000 nuplets stockés dans des pages. On suppose la répartition des nuplets uniforme. On suppose que l'attribut A est stocké sur 6 octets, l'attribut B sur 8 octets, l'attribut C sur 10 octets, l'attribut D sur 20 octets, l'attribut E sur 6 octets et qu'une adresse de page est stockées sur 8 octets. On suppose que les pages de la base ont 4000 octets disponibles pour stocker les données et quelles sont en moyenne pleines aux 3/5.

1. Décrivez complètement toutes les structures de données et évaluez le nombre de pages de l'index primaire (non dense) sur A.
2. Évaluez le nombre de pages de l'index secondaire dense sur B.