



TD 01 - Conception physique

- Expliquez le processus d'une conception physique.

Exercice 00 : « Révision »

Soit une base de données portant sur les cours d'amphi dont le schéma est le suivant :

ETUDIANT (Num_etudiant, NomE, E-mail, Age, Moyenne)
 PROF (Num_prof, NomP, Salaire)
 MODULE (Num_module, NomM, DuréeAnnuelle)
 SUIT (Num_etudiant, Num_module, Section, Note)
 ENSEIGNE (Num_prof, Num_module, Section)

On note que, comme les étudiants sont nombreux, ils sont répartis, pour chaque module, en plusieurs sections, chaque section correspondant à un amphi et donc à un prof. La répartition en sections peut varier d'un module à un autre.

- ✓ Quelles sont les contraintes d'intégrité référentielle de cette base ?
- ✓ Exprimer les requêtes suivantes en algèbre relationnelle, et en SQL :

1. Noms des professeurs de 'bases de données'.
2. Nom des étudiants qui ne suivent pas le module de 'réseaux'.
3. Numéro(s) du ou des étudiant(s) le(s) plus âgé(s).
4. Numéros des étudiants ayant réussi (Note ≥ 10) tous les modules qu'ils suivent.

- ✓ Exprimer les requêtes suivantes en SQL :

1. Noms des professeurs de 'bases de données'.
2. Moyenne d'âge par module sauf pour 'réseaux'.
3. Noms des professeurs qui enseignent à moins de 10 étudiants en tout.

Exercice 01

Soit la table Départements contenant les 20 tuples suivants :

N°	Département	Région
3	Allier	Auvergne
36	Indre	Centre
18	Cher	Centre
75	Paris	Ile-de-France
39	Jura	Franche Comté
9	Ariège	Midi Pyrénées
81	Tarn	Midi Pyrénées
11	Aude	Languedoc Roussillon
12	Aveyron	Midi Pyrénées
25	Doubs	Franche Comté
73	Savoie	Rhone Alpes
55	Meuse	Lorraine
15	Cantal	Auvergne
51	Marne	Champagne - Ardenne
42	Loire	Rhone Alpes
40	Landes	Aquitaine
14	Calvados	Basse Normandie
30	Gard	Languedoc Roussillon
84	Vaucluse	Provence Alpes Cote d'Azur
7	Ardèche	Rhone Alpes

Nous supposons que les n° de département sont codés sur 2 octets, les départements sur 30 octets et la région de même. Pour les pointeurs, nous considérerons une taille de 18 octets. Nous supposons que notre SGBD constitue des pages (pages) de 512 octets.

- Quel est la taille en nombre de pages de la table Départements ? Comment peut-on identifier le ROWID du Doubs ?
- Retirez le département 55. Comment pourrait-on garder l'information du tuple vide ?
- Créez un index dense, sur la colonne Région. (On représentera ici les ROWID par des flèches vers les données et non leur valeur réelle).
- Combien faut-il d'accès disques pour récupérer les départements du Languedoc-Roussillon ? (Nous supposons que l'index est en mémoire)
- Créez un index non dense sur ces pages (pages ordonnés sur la colonne Région).
- Combien faut-il d'accès disques pour récupérer les départements du Languedoc-Roussillon ? (Nous supposons que l'index est en mémoire)

Exercice 02

Une structure de hachage extensible contient des entrées x^* dont les valeurs de hachage $h(x)$ sont les suivantes : 4,12,32,16,1,5,21,10,15,7,19

- Avec au plus 4 entrées par paquets, combien faut-il de paquets pour avoir la structure la plus compacte possible ?
- Dessiner la structure de hachage pour ces valeurs (répertoire et paquets, en indiquant profondeur globale et profondeurs locales)
- Insérer 20*

Exercice 03

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 :

- Fichier séquentiel non ordonné avec un index dense.
- 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 noeuds internes des pointeurs vers d'autres noeuds. On suppose qu'une page d'arbre B+ est pleine à 70 %.

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 04

Soit le schéma relationnel suivant, utilisé pour la base d'une agence de location : **Lot** (Id Lot, Id Propriétaire, Adresse, Loyer, Etat)

Personne (Id Personne, Nom, Prenom, Adresse, Coord bancaires)

Location (Id Lot, Id Locataire, Date arrivee, Date depart, Note)

Taille des différents types :

Les identifiants (Id ...) et les nombres (Loyer, Note) sont codés sur 4 octets,

Les dates sur 4 octets, - Nom, Prenom, Etat sur 50 octets

Adresse et Coord bancaires sur 100 octets,

Statistiques :

2 100 tuples dans la table Lot (sur 1 000 propriétaires),

600 000 tuples Personne (Contenant locataire et propriétaires),

1 840 000 tuples Location (Avec une note entre 0 et 5),

Configuration Base de données :

Taille d'un ROWID : 18 octets

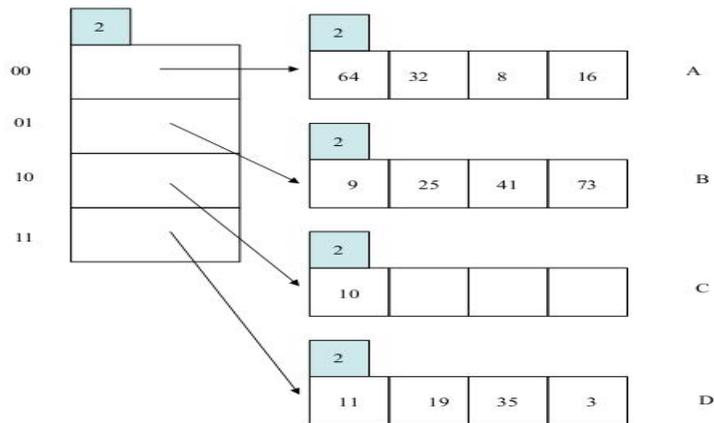
10 pages en mémoire centrale,

Une page fait 8 ko, dont 10% sont utilisées pour les mises à jour et les pointeurs. Nous avons également 10 pages disponibles en mémoire pour l'exécution de chaque requête.

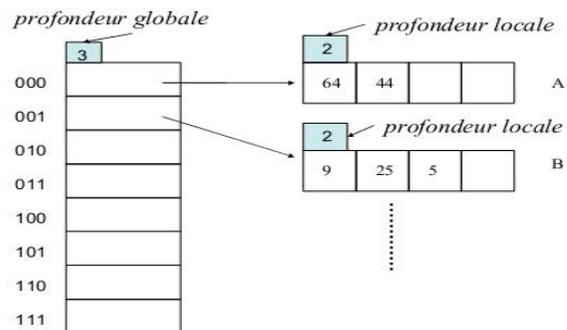
- 1- Calculer en nombre de pages, la taille de chaque table (donner le détail du calcul)
- 2- On construit un index bitmap sur la note de la location. Quel est la taille totale de ce bitmap en nombre de pages ?
- 3- Créer l'index B+Tree sur Lot(Id Lot) d'ordre 3 avec les valeurs suivantes, vous ferez apparaître les différentes étapes d'éclatement : 100, 25, 72, 48, 10, 33, 58, 110, 40, 52, 115, 80, 5, 28, 49, 75, 60
- 4- On souhaite construire un index B+Tree couvrant sur Location (Id Lot, Note). Quel est la taille d'une case d'une feuille de l'index (avec le ROWID) ? Calculer l'ordre idéal de cet index (Sachant que les 8 ko de la page sont utilisés).

Exercice 05

On considère un index utilisant une technique de hachage extensible. La profondeur globale est indiquée au dessus du répertoire. La profondeur locale est indiquée au dessus de chaque paquet. Un paquet peut contenir de 1 à 4 entrées. Les entrées correspondent à des valeurs de clé hachées. On utilise un algorithme de suppression complète, c'est-à-dire que la suppression d'un paquet entraîne la diminution de la taille du répertoire, si cela est possible. On suppose qu'il y a fusion seulement si et seulement si un paquet est vide. Toute insertion dans un paquet plein provoque un éclatement et donc la création d'un nouveau paquet. Soit l'index suivant dans l'état initial



- 1- Représenter l'état E2 de l'index après insertion successive des entrées 4, 5, 7.
- 2- Complétez l'état initial de cet index de telle façon que celui-ci contiennent les entrées (en plus de celle déjà apparentes sur la figure ci-dessous) : 10, 3, 11, 19, 27, 63.



- 3- Donnez l'état de l'index après suppression de l'entrée 63.