

1. Questions de cours (3 Points)

Répondez brièvement les questions ci-dessous, justifiez chaque réponse.

- 1.1. Pourquoi l'organisation primaire d'un fichier est déterminante pour l'insertion et la suppression d'enregistrements ?
- 1.2. Quelle est la différence entre un index primaire et un index secondaire.
- 1.3. Que sont les propriétés ACID dans un SGBD ?

2. Stockage (2 points)

Soit la base de données, de l'hôpital *H*, ci-dessus, contenant 3.000 Médecins de 10 spécialités différentes.

Médecin (*id-médecin*, *nom_médecin*, *adresse*, *téléphone*, *spécialité*)

La relation *Médecin* est stockée en utilisant en arbre B+ sur l'attribut *id-médecin*, clé de la relation.

On suppose que l'attribut *id-médecin* est stocké sur 7 octets, l'attribut *nom_médecin* sur 50 octets, l'attribut *adresse* sur 50 octets, *téléphone* sur 20 octets et *spécialité* sur 23 octets.

On suppose que :

- Une adresse de page est stockée sur 8 octets et les pages de la base ont 3008 octets.
- Les attributs de la relation suivent une distribution uniforme.

Répondez les questions ci-dessous, justifiez vos réponses :

- 2.1. Combien de pages il y a dans les nœuds feuilles de l'arbre B+ ?
- 2.2. Qu'elle est la structure d'un nœud de l'index ?
- 2.3. Quelle est la hauteur de l'arbre B+ ?
- 2.4. Combien de pages sont-elles nécessaires pour stocker la table ?

3. Indexation (4 points)

Soit la table *E* = (*ID*, *Nom*, *Age*, *salaire*, *adresse*, *téléphone*, *département*), ci-dessous. L'attribut *ID* est la clé de la relation.

ID	Nom	Age	Salaire	Adresse	Téléphone	département
11(0001011)	J	43 (0101011)	4500
25(0011001)	F	57 (0111001)	4500
32(0100000)	D	54 (0110110)	5004
45(0101101)	R	44 (0101100)	3000
100(1100100)	L	55 (0110111)	6003
61(0111101)	H	58 (0111010)	5000
73 (1001001)	C	40 (0101000)	4006
78 (1001110)	S	48 (0110000)	3000
94 (1011110)	L	42 (0101010)	6003
104 (1101000)	H	61(0111101)	5000
117 (1110101)	FF	55 (0110111)	5300
54(0110110)	H	44 (0101100)	5000
87 (1010111)	FF	57 (0111001)	5300

Cette table est stockée séquentiellement non-ordonnée, sur 7 pages (2 enregistrements par pages).

Répondez les questions ci-dessous, justifiez vos réponses :

- 3.1. Combien d'entrées/sorties sont-ils nécessaires pour chercher les employés de 44 ans ?
- 3.2. Supposons qu'on dispose d'un index secondaire d'hachage extensible type (clé de recherche, clé primaire) sur l'attribut âge. Construisez cet index en considérant qu'une page peut stocker jusqu'à 4 enregistrements de l'index et qu'initialement on ne considère que le dernier bit de la clé de recherche.
- 3.3. Décrivez en français la procédure générale à suivre, en utilisant l'index, pour chercher
 - 3.3.1. Tous les employés (Nom de l'employé) qu'ont moins de 54 ans. Combien de pages faut-il lire pour résoudre cette requête ?
 - 3.3.2. Tous les employés (ID de l'employé) de 44 ans. Combien de pages faut-il lire pour résoudre cette requête ?

4. Optimisation des requêtes (3,5 points)

Soit la base de données suivante :

Médecin (Mnum, spécialité, Nom, Adresse, Téléphone)

Acte (Anum, Mnum, SS, Date, Description, Total, NMutuelle, PourcentageM)

Les attributs soulignés sont les clés des relations.

L'attribut Mnum de la relation Acte est une clé étrangère qui fait référence à l'attribut Mnum de la relation Médecin

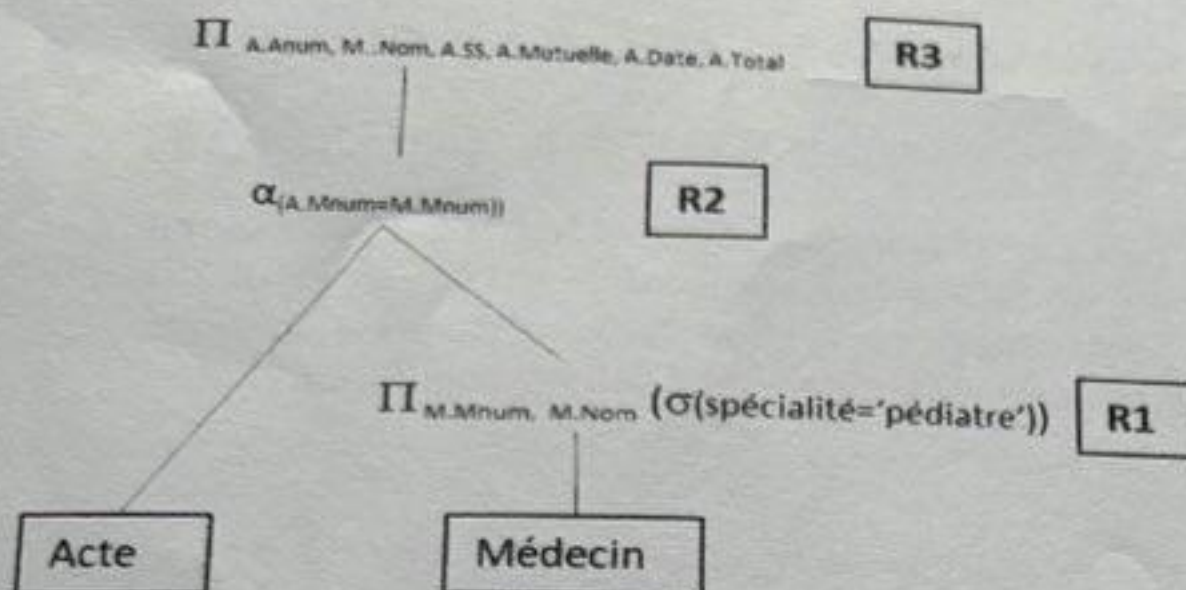
Données spécifiques à la base :

- La relation Médecin contient 500 tuples, dont 80 cardiologues et 150 pédiatres. Elle est stockée suivant un arbre B+ sur l'attribut Mnum, clé de la relation. Cet arbre est stocké sur 3 pages avec une hauteur de 3 niveaux (index + données). L'attribut Mnum est stocké sur 10 octets, spécialité sur 15 octets, Nom sur 15 octets, Adresse sur 20 octets, et Téléphone sur 10 octets.
- La relation Acte est stockée avec une organisation directe sur un bloc primaire de 3000 pages sans pages d'overflow en utilisant une fonction d'hachage sur l'attribut Anum, clé de la relation. Elle contient 20 000 tuples. L'attribut Anum est stocké sur 10 octets, Mnum est stocké sur 10 octets, SS sur 10 octets, Date sur 8 octets, Description sur 20 octets, Total sur 12 octets, NMutuelle sur 10 octets, et PourcentageM sur 4 octets.
- On suppose que :
 - Les pages de la base occupent 600 octets et une adresse mémoire est stockée sur 8 octets.
 - La mémoire peut contenir 40 pages pour les opérations.
 - Un médecin n'a qu'une seule spécialité

Pour l'arbre algébrique ci-dessous correspondant à la requête : "Pour tous les interventions suivies par un médecin pédiatre, donnez le nom du médecin, le numéro de la Sécurité Social (SS) et le nom de la mutuelle du patient ainsi que la date de l'intervention, le montant total de l'intervention et le pourcentage payé par la mutuelle" :

SELECT M.Nom, A.SS, A.Mutuelle, A.Date, A.Total, A.PourcentageM

FROM Acte A, Medecin M
WHERE A.Mnum = M.Mnum AND M.Spécialité = 'pédiatre'



Évaluez et comparez le coût d'exécution de la requête, avec les plans d'exécutions suivants :

Plan d'exécution 1 :

- R1) La requête de R1 est une sélection sans index → parcours total de la table **Médecin**
- R2) L'algorithme de jointure **BI** en utilisant au maximum l'espace de mémoire entre les tables **Acte** et la table résultante **R1** sur l'attribut **Mnum**
- R3) Projection → parcours total de la table résultante de **R2**

Plan d'exécution 2 :

- R1) La requête de R1 est une sélection sans index → parcours total de la table **Médecin**
- R2) L'algorithme de jointure **HashJoin** entre les tables **Acte** et la table résultante **R1** sur l'attribut **Mnum**
- R3) Projection → parcours total de la table résultante de **R2**

5. Contrôle de la concurrence (2,5 points)

Les trois programmes suivants peuvent s'exécuter dans un système de gestion bancaire. **Débit** diminue le solde d'un compte **c** avec un montant donné **m**. Pour simplifier, tout débit est permis (on accepte les découverts). **Crédit** augmente le solde d'un compte **c** d'un montant **m**. **Transfert** transfère un montant **m** à partir d'un compte source **s** vers un compte destination **d**. L'exécution de chaque programme démarre par un **Begin** et se termine par un **Commit** (non montré ci-dessous).

Débit (c:Compte; m:Montant)	Crédit (c:Compte; m:Montant)	Transfert(s,d:Compte; m:Montant)
begin	begin	begin
t := Read(c);	t = Read(c);	Débit(s,m) .
Write(c,t-m);	Write(c, t+m);	Crédit(d,m);
end	end	end

Le système exécute en même temps les trois opérations suivantes :

- un transfert d'un montant 100 du compte A vers le compte B

- un crédit de 200 pour le compte A
- un débit de 50 pour le compte B

5.1. Ecrire les transactions T1, T2, T3 qui correspondent à ces opérations. Montrer que l'histoire

H: $r1[A]$ $r3[B]$ $w1[A]$ $r2[A]$ $w3[B]$ $r1[B]$ $c3$ $w2[A]$ $c2$ $w1[B]$ $c1$ est une exécution concurrente de T1, T2 et T3.

5.2. Mettre en évidence les conflits dans H et construire le graphe de sérialisation de cette histoire. H est-elle sérialisable ?

5.3. Quelle est l'exécution H' obtenue à partir de H par verrouillage à 2 phases ? On suppose que les verrous d'une transaction sont relâchés après le COMMIT de celle-ci. Une opération bloquée en attente d'un verrou bloque le reste de sa transaction. Au moment du relâchement des verrous, les opérations en attente sont exécutées en priorité.

6. MongoDB (5 points)

Soit la base de données document MongoDB 'livres', dont un petit échantillon est montré ci-dessous :

```
[ { "type": "Book",
  "title": "Modern Database Systems: The Object Model, Interoperability, and Beyond.",
  "year": 1995,
  "publisher": "ACM Press and Addison-Wesley",
  "authors": ["Won Kim"],
  "source": "DBLP" },
  { "type": "journal",
  "title": "An All-Round Sweep Algorithm for 2-Dimensional Nearest-Neighbor Problems",
  "year": 1992,
  "publisher": "ACM TODS",
  "authors": ["Klaus H. Hinrichs",
    "Juuml Nievergelt",
    "Peter Schorn"],
  "source": "DBLP" },
  { "type": "Book",
  "title": "Programming HIVE",
  "year": 201,
  "publisher": "O'RELLY",
  "authors": ["Edward Capriolo", "Dean Wampler", "Jason Rutherglen"] } ]
```

Répondez les requêtes suivantes :

- Liste de tous les livres (type "Book")
- Liste des publications depuis 2011
- Liste des livres depuis 2014
- Compter le nombre des publications de "Edward Capriolo"
- Compter le nombre de publications depuis 2011 et par type
- Compter le nombre de publications par auteur et trier le résultat par ordre croissant