

Bases de Données Avancées : Optimisation, Les fichiers

Thomas Gerald

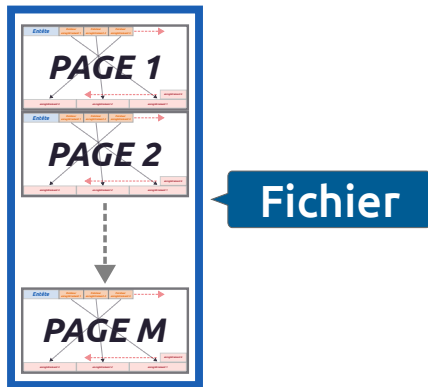
22 novembre 2024

Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, CNRS

`thomas.gerald@lisn.upsaclay.fr`

Organisation des relations

- Une relation est stockée dans un fichier
- Les fichiers contiennent des pages
- Il existe différentes organisations dans un fichier



Organisation en fichier(s) : accéder à une page/un enregistrement

Le record ID

Identifiant pour les enregistrements permettant :

- Retrouver la page de l'enregistrement
- Retrouver l'enregistrement dans la page

Dans postgresSQL ?

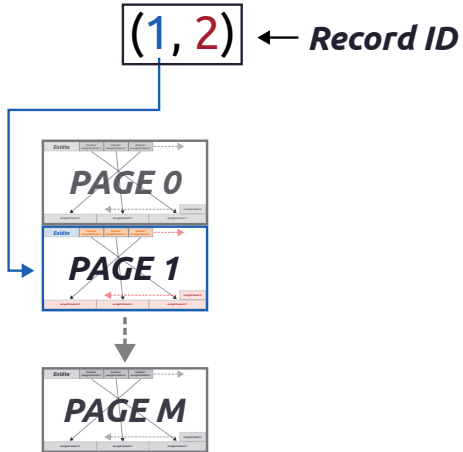
→ Appelé **ctid**, il s'agit du couple :

- Le numéro de la page/block
- Le numéro de l'enregistrement dans la page

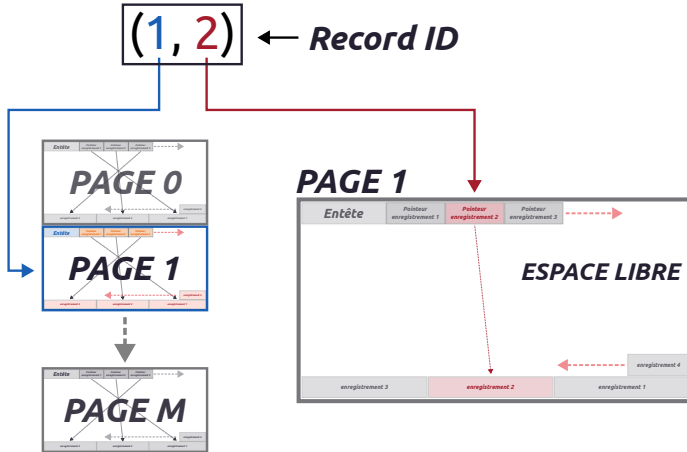
ctid	title
(0,1)	Toy Story
(0,2)	Jumanji
⋮	⋮
(0,22)	Hyvä poika
(1,1)	Copycat
⋮	⋮
(1,20)	Richard III
(2,1)	Dead Presidents
⋮	⋮
(2349,2)	Queerama

Table 1 : **ctid** et titre de la table movie
(`SELECT ctid,title FROM movies`)

Organisation en fichier(s) : accéder à une page/un enregistrement



Organisation en fichier(s) : accéder à une page/un enregistrement



Accéder à un enregistrement ?

- Depuis le numéro de page retrouver l'adresse de la page sur le disque
- Depuis le numéro de l'enregistrement, retrouver l'adresse de l'enregistrement

Organisation en fichier(s) : Retrouver l'adresse de l'enregistrement

Accéder au **troisième enregistrement** de la cinquième page :

- l'adresse de la page est 0xAF00 (données de l'énoncé)
- La taille d'un pointeur le disque est de 4 octets (taille fixe)
- La taille de l'entête est de 20 octets (taille fixe)

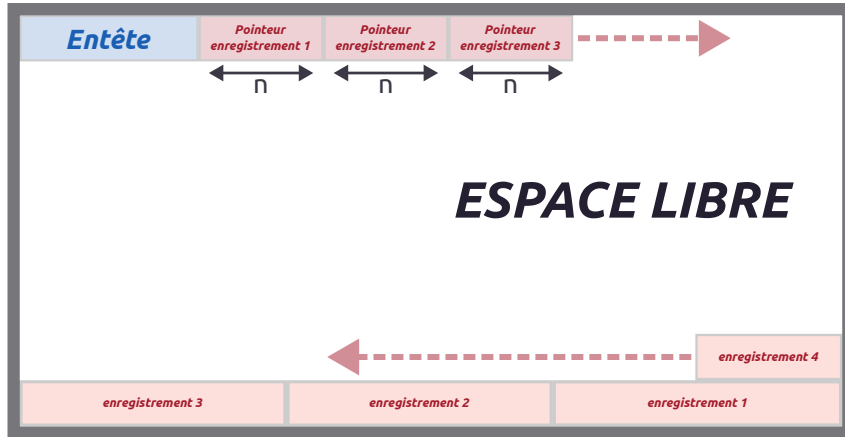
L'adresse du pointeur est donc (20 s'écrit 14 en hexadécimal) :

•

$$e_3 = 0xAF00 + 4 \times 2 + 14 = 0xAF1C$$

e_3 pointe sur le début de l'enregistrement!!!

Organisation en fichier(s) : accéder à une page/un enregistrement



Organisation en fichier(s) : accéder à une page/un enregistrement

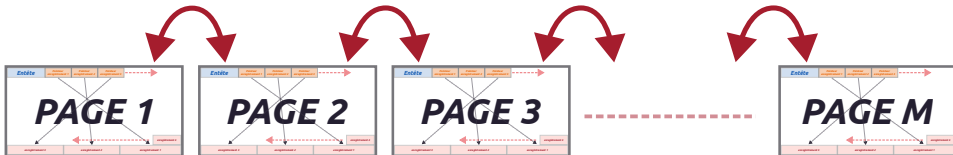
Comment accéder à la page n°k (adresse)

Organisation en liste chaînées

- Chaque page contient l'adresse (sur le disque) de la page suivante
- Pour accéder à une page il faut lire la page précédente (ou du moins une partie)...

Organisation en répertoires

- Des pages spéciales existent contenant les adresses des pages disque
- Moins de lecture de pages pour accéder à n'importe quelle page



Le système de fichier (système)

En pratique on va s'abstraire de cette contrainte
(dans ce cours)

→ On considère les pages stockées dans un espace
contiguë

Organisation en fichier(s) : accéder à une page/un enregistrement

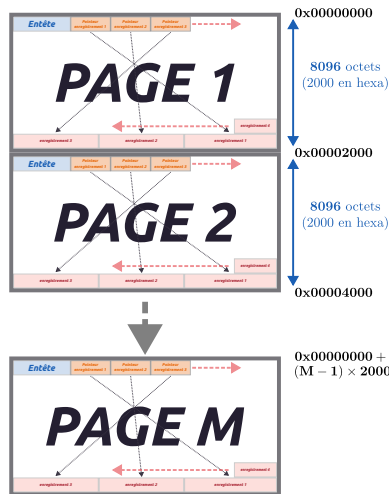
Le système de fichier (système)

En pratique on va s'abstraire de cette contrainte
(dans ce cours)

→ On considère les pages stockées dans un espace
contiguë

Dans Postgres ?

→ Utilisation du système de fichier du système
d'exploitation (OS)



Des opérations

- **Parcours (SCAN)** : parcourir tous les enregistrements d'un fichier
- **Recherche sur égalité** : Retrouver les enregistrement avec un attributs ayant une valeur données
- **Recherche sur intervalle** : Retrouver les enregistrements avec un attributs dans un intervalle données
- **Insertion** : Ajouter un nouvel enregistrement
- **Suppression** : Supprimer un enregistrement (éventuellement par attribut ou rid)

Organisation en fichier(s) : Le tas

Le tas (Heap File)

- Structure simple
- Pas d'ordre dans les pages (enregistrements)

Ajouter un éléments ?

- Ajouter à la fin du fichier
 - dernière pages si assez d'espace
 - création d'une nouvelle page sinon

Un problème ?

	EID	NOM	PRENOM	AGE
Page 1	1	Zeblouse	Agathe	24
	2	Huai	Odile	26
	7	Peuplu	Jean	23
Page 2	4	Gator	Ali	23
	3	Roïd	Paula	22
	8	Banylon	Philémon	21
↓				
Page M	5	Dévent	Rose	23
	21	Dubois	Yvan	19
	23	Cevite	Sakura	22

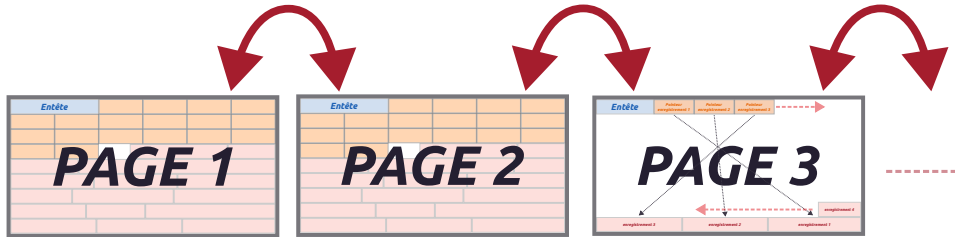
Organisation en fichier(s) : Le tas

⚠ **Des enregistrements peuvent être supprimés** → dispersion des enregistrements (clairemée)

→ Des pages sous-utilisées (mais prenant autant de place que les pages pleines)

Solutions?

→ Parcourir les pages jusqu'à trouver une page avec un espace libre suffisant



⚠ **A chaque ajout d'enregistrement parcourir l'intégralité du fichier**

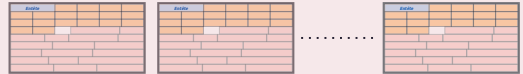
Organisation en fichier(s) : Le tas

Maintenir une liste des pages

Maintenir une liste avec les pages contenant un espace libre!!!

- À la création d'une page
- À la suppression d'un enregistrement

PAGES PLEINES



PAGES AVEC ESPACE LIBRE



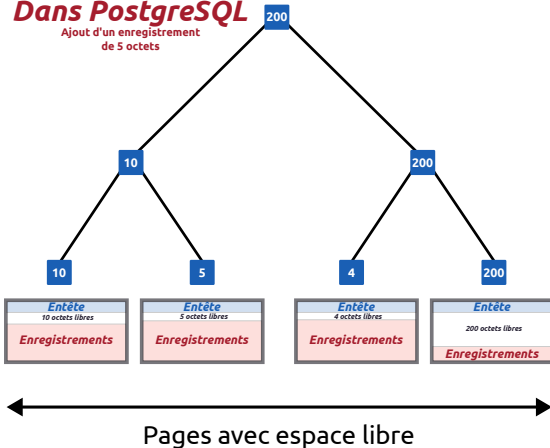
Dans postgres ?

Les pages libres sont stockées dans à l'aide d'une structure arborescente

- Chaque nœud de l'arbre contient l'espace maximum des pages du sous arbre
- Chaque feuille pointe sur une page avec de l'espace libre

Dans PostgreSQL

Ajout d'un enregistrement de 5 octets



Organisation en fichier(s) : Le tas

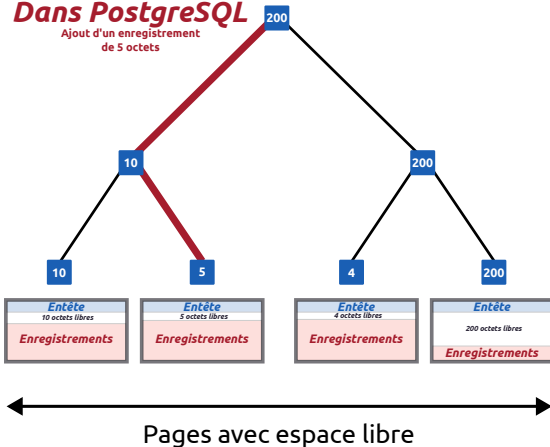
Dans postgres ?

Les pages libres sont stockées dans à l'aide d'une structure arborescente

- Chaque nœud de l'arbre contient l'espace maximum des pages du sous arbre
- Chaque feuille pointe sur une page avec de l'espace libre

Dans PostgreSQL

Ajout d'un enregistrement de 5 octets



Organisation en fichier(s) : Le tas

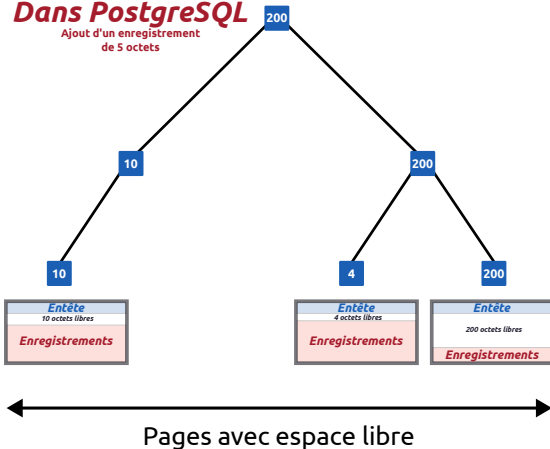
Dans postgres ?

Les pages libres sont stockées dans à l'aide d'une structure arborescente

- Chaque nœud de l'arbre contient l'espace maximum des pages du sous arbre
- Chaque feuille pointe sur une page avec de l'espace libre

Dans PostgreSQL

Ajout d'un enregistrement de 5 octets



Organisation en fichier(s) : Opérations sur un tas

$P \rightarrow$ Nombre de pages

$T_P \rightarrow$ Temps d'ouverture d'une page

Le parcours

Ouverture de toutes les pages

$\rightarrow P$ pages sont ouvertes

$$\Rightarrow P \times T_P \times R \times T_R$$

	EID	NOM	PRENOM	AGE
Page 1	1	Zeblouse	Agathe	24
	2	Huai	Odile	26
	7	Peuplu	Jean	23
Page 2	4	Gator	Ali	23
	3	Roïd	Paula	22
	8	Banylon	Philémon	21
↓				
Page M	5	Dévent	Rose	23
	21	Dubois	Yvan	19
	23	Cevite	Sakura	22

Organisation en fichier(s) : Opérations sur un tas

M pages dans le fichier organiser en tas (Heap-File)

Recherche sur égalité (unicité de l'attribut)

Pages non triées → ouverture de toutes les pages avant la bonne valeur

- Au minimum une page
- Au maximum P pages
- En moyenne $\frac{M}{2}$ pages

$$\Rightarrow \frac{P}{2} \times T_P$$

	EID	NOM	PRENOM	AGE
Page 1	1	Zebrouse	Agathe	24
	2	Huai	Odile	26
	7	Peuplu	Jean	23
Page 2	4	Gator	Ali	23
	3	Roïd	Paula	22
	8	Banylon	Philémon	21
↓				
Page M	5	Dévent	Rose	23
	21	Dubois	Yvan	19
	23	Cevite	Sakura	22

Organisation en fichier(s) : Opérations sur un tas

M pages dans le fichier organiser en tas (Heap-File)

Recherche sur égalité

Pages non triées $\rightarrow$ ouverture de toutes les pages

$\rightarrow P$ pages sont ouvertes

$$\Rightarrow P \times T_p$$

	EID	NOM	PRENOM	AGE
Page 1	1	Zeblouse	Agathe	24
	2	Huai	Odile	26
	7	Peuplu	Jean	23
Page 2	4	Gator	Ali	23
	3	Roïd	Paula	22
	8	Banylon	Philémon	21
↓				
Page M	5	Dévent	Rose	23
	21	Dubois	Yvan	19
	23	Cevite	Sakura	22

Organisation en fichier(s) : Opérations sur un tas

Recherche d'un intervalle

Pages non triées → ouverture de toutes les pages
→ P pages sont ouvertes

$$\Rightarrow P \times T_p$$

	EID	NOM	PRENOM	AGE
Page 1	1	Zeblouse	Agathe	24
	2	Huai	Odile	26
	7	Peuplu	Jean	23
Page 2	4	Gator	Ali	23
	3	Roïd	Paula	22
	8	Banylon	Philémon	21
↓				
Page M	5	Dévent	Rose	23
	21	Dubois	Yvan	19
	23	Cevite	Sakura	22

Organisation en fichier(s) : Opérations sur un tas

Insertion

Pages non triées → Placement libre

Si structure de l'espace libre sur $< N$ pages ($N \ll P$)

- Trouver une page libre $\rightarrow < N$ lectures
- Lecture de la page avec suffisamment d'espace libre (1 lecture)
- Écriture des modifications (1 écriture)

$$\Rightarrow N \times T_P + T_P + T_P = (N + 2)T_P$$

	EID	NOM	PRENOM	AGE
Page 1	1	Zeblouse	Agathe	24
	2	Huai	Odile	26
	7	Peuplu	Jean	23
Page 2	4	Gator	Ali	23
	3	Roïd	Paula	22
	8	Banylon	Philémon	21
↓				
Page M	5	Dévent	Rose	23
	21	Dubois	Yvan	19
	23	Cevite	Sakura	22

La suppression

- Trouver la page contenant l'enregistrement → recherche
- Écriture des modifications (1 écriture)

$$\Rightarrow P \times T_p \times R$$

	EID	NOM	PRENOM	AGE
Page 1	1	Zeblouse	Agathe	24
	2	Huai	Odile	26
	7	Peuplu	Jean	23
Page 2	4	Gator	Ali	23
	3	Roïd	Paula	22
	8	Banylon	Philémon	21
↓				
Page M	5	Dévent	Rose	23
	21	Dubois	Yvan	19
	23	Cevite	Sakura	22

Organisation en fichier(s) : Opérations sur un tas (recapitulatif)

	Scan	Rech (unicité)	Rech (intervalle)	Insert	Suppr
I/O (pages)	P	$\frac{P}{2}$	P	$N + 2$	$P + 1$
Complexité	P	P	P	1 (ne dépend pas de P)	P

→  Peu efficace, excepté pour l'insertion

Des structures de fichier plus efficaces ?

De meilleures Solutions

- Si les enregistrements et les pages étaient triés sur un/des attributs
- Si les enregistrements et les pages étaient regroupés sur un/des attributs

Organisation en fichier(s) : Un fichier trié

Le fichier trié

- Les enregistrements sont triés selon un attribut
- Les pages sont triés selon le même attribut

Données :

$P \rightarrow$ Nombre de pages

$T_p \rightarrow$ Temps d'ouverture d'une page

EID	NOM	PRENOM	AGE
21	Dubois	Yvan	19
8	Banylon	Philémon	21
3	Roïd	Paula	22

23	Cevite	Sakura	22
7	Peuplu	Jean	23
5	Dévent	Rose	23

4	Gator	Ali	23
1	Zeblouse	Agathe	24
2	Huai	Odile	26

Organisation en fichier(s) : Opérations sur fichier trié

Parcours

Ouverture de toutes les pages

$$\Rightarrow P \times T_p$$

EID	NOM	PRENOM	AGE
21	Dubois	Yvan	19
8	Banylon	Philémon	21
3	Roïd	Paula	22
23	Cevite	Sakura	22
7	Peuplu	Jean	23
5	Dévent	Rose	23
4	Gator	Ali	23
1	Zeblouse	Agathe	24
2	Huai	Odile	26

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zeblouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 1 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zeblouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 2 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

 if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zeblouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 3 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

 if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

 return $p_{\frac{e-s}{2}}$

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zeblouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 4 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

return $p_{\frac{e-s}{2}}$

else if $v > \max p_{\frac{e-s}{2}}$ then

▷ Valeur max page $P_{\frac{e-s}{2}}$

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zablouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 5 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

return $p_{\frac{e-s}{2}}$

else if $v > \max p_{\frac{e-s}{2}}$ then

▷ Valeur max page $P_{\frac{e-s}{2}}$

$s \leftarrow \frac{e-s}{2}$

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zeblouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 6 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

return $p_{\frac{e-s}{2}}$

else if $v > \max p_{\frac{e-s}{2}}$ then

▷ Valeur max page $P_{\frac{e-s}{2}}$

$s \leftarrow \frac{e-s}{2}$

else if $v < \min p_{\frac{e-s}{2}}$ then

▷ Valeur min dans $P_{\frac{e-s}{2}}$

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zebrouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 7 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

return $p_{\frac{e-s}{2}}$

else if $v > \max p_{\frac{e-s}{2}}$ then

▷ Valeur max page $P_{\frac{e-s}{2}}$

$s \leftarrow \frac{e-s}{2}$

else if $v < \min p_{\frac{e-s}{2}}$ then

▷ Valeur min dans $P_{\frac{e-s}{2}}$

$e \leftarrow \frac{e-s}{2}$

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zeblouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 8 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

return $p_{\frac{e-s}{2}}$

else if $v > \max p_{\frac{e-s}{2}}$ then

▷ Valeur max page $P_{\frac{e-s}{2}}$

$s \leftarrow \frac{e-s}{2}$

else if $v < \min p_{\frac{e-s}{2}}$ then

▷ Valeur min dans $P_{\frac{e-s}{2}}$

$e \leftarrow \frac{e-s}{2}$

else

Organisation en fichier(s) : Opérations sur fichier trié

EID	NOM	PRENOM	AGE
1	Zebrouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Algorithm 9 Recherche par dichotomie

Require : P, a, v

▷ nb page, attribut, valeur

$s \leftarrow 0$

$e \leftarrow P$

while $e - s > 0$ do

if $\exists e_a \in p_{\frac{e-s}{2}}$ tq $e_a = v$ then

return $p_{\frac{e-s}{2}}$

else if $v > \max p_{\frac{e-s}{2}}$ then

▷ Valeur max page $P_{\frac{e-s}{2}}$

$s \leftarrow \frac{e-s}{2}$

else if $v < \min p_{\frac{e-s}{2}}$ then

▷ Valeur min dans $P_{\frac{e-s}{2}}$

$e \leftarrow \frac{e-s}{2}$

else

return NULL

end if

end while

Organisation en fichier(s) : Opérations sur fichier trié

Recherche (unicité)

Recherche par dichotomie → division par deux de l'espace de recherche à chaque étape

- Au maximum $\leq \log_2(P)$ étapes
- Une page lue par étape
- Lecture de $\leq \log_2(P)$ pages max

$$\Rightarrow \log_2(P) \times T_P$$

EID	NOM	PRENOM	AGE
1	Zeblouse	Agathe	24
2	Huai	Odile	26
3	Roïd	Paula	22
4	Gator	Ali	23
5	Dévent	Rose	23
7	Peuplu	Jean	23
8	Banylon	Philémon	21
21	Dubois	Yvan	19
23	Cevite	Sakura	22

Organisation en fichier(s) : Opérations sur fichier trié

Recherche (intervalle)

Recherche par dichotomie puis parcours des pages ayant l'attribut la valeur dans le bon intervalle (N)

$$\Rightarrow < (\log_2(P) + N) \times T_p$$

Exemple

Trouver les enregistrements avec l'attribut $age < 23$

- Recherche d'un élément $< \sup(\log_2(3)) = 2$ pages (arrondi supérieur)
- Parcourir les pages (ici 2 pages où $age < 23$)

EID	NOM	PRENOM	AGE
21	Dubois	Yvan	19
8	Banylon	Philémon	21
3	Roïd	Paula	22
23	Cevite	Sakura	22
7	Peuplu	Jean	23
5	Dévent	Rose	23
4	Gator	Ali	23
1	Zebrouse	Agathe	24
2	Huai	Odile	26

Insertion

1. Recherche (un élément) $\rightarrow \approx \log_2(P)$
2. Insertion de l'enregistrement $\rightarrow 1$
3. Décalage dans toutes les pages suivantes...

⚠ $\rightarrow 2 \times \frac{P}{2}$ (lecture/écriture)

$$\Rightarrow < (\log_2(P) + 2 \times \frac{P}{2} + 1) \times T_P$$

EID	NOM	PRENOM	AGE
21	Dubois	Yvan	19
8	Banylon	Philémon	21
3	Roïd	Paula	22
23	Cevite	Sakura	22
7	Peuplu	Jean	23
5	Dévent	Rose	23
4	Gator	Ali	23
1	Zebrouse	Agathe	24
2	Huai	Odile	26

Suppression

1. Recherche (un élément) $\rightarrow \approx \log_2(P)$
2. Suppression de l'enregistrement $\rightarrow 1$
3. Décalage dans toutes les pages suivantes...

⚠ $\rightarrow 2 \times \frac{P}{2}$ (lecture/écriture)

$$\Rightarrow < (\log_2(P) + 2 \times \frac{P}{2} + 1) \times T_P$$

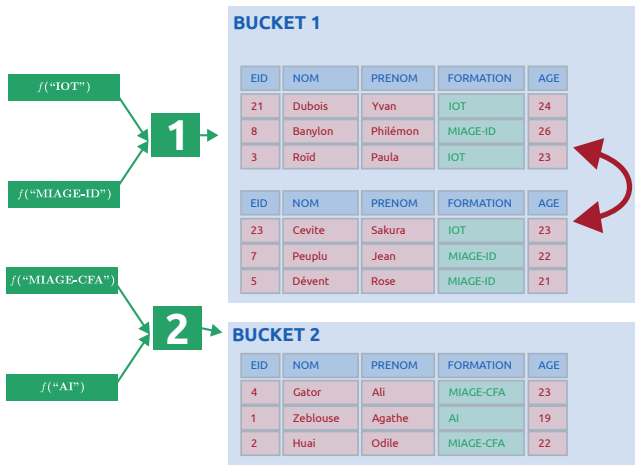
EID	NOM	PRENOM	AGE
21	Dubois	Yvan	19
8	Banylon	Philémon	21
3	Roïd	Paula	22
23	Cevite	Sakura	22
7	Peuplu	Jean	23
5	Dévent	Rose	23
4	Gator	Ali	23
1	Zebrouse	Agathe	24
2	Huai	Odile	26

Organisation en fichier(s) : Opérations sur fichier trié (récapitulatif)

	Scan	Rech (unicité)	Rech (intervalle)	Insert	Suppr
I/O (pages)	P	$\log_2(P)$	$\log_2(P) + N$	$\log_2(P) + P$	$\log_2(P) + P$
Complexité	P	$\log(P)$	$\log(P)$	P	P

→  Efficace pour la recherche

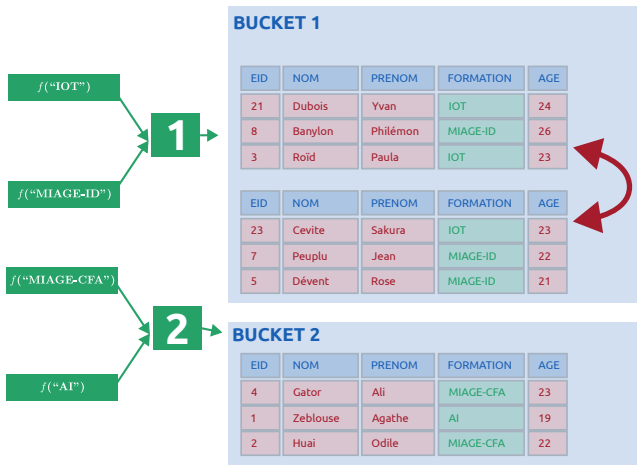
Organisation en fichier(s) : Fichier Haché



Fichier Haché

- Les enregistrements sont regroupés par valeurs
- On supposera une page par valeur dans cet exemple
- Il existe une fonction prenant la valeur de l'attribut retournant la/les pages correspondantes

Organisation en fichier(s) : Fichier Haché

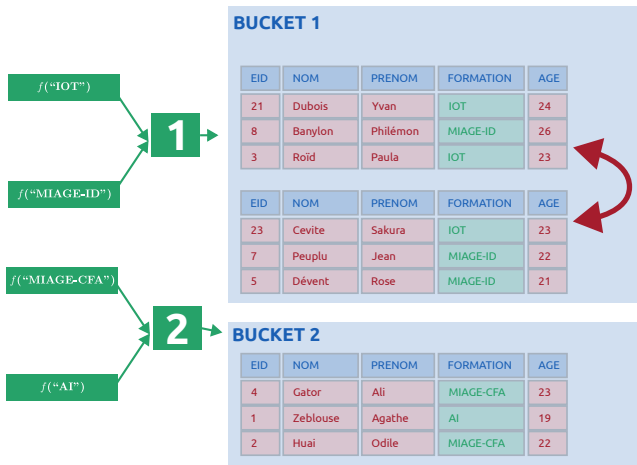


Parcours

Toutes les pages :

$$\Rightarrow P \times T_p$$

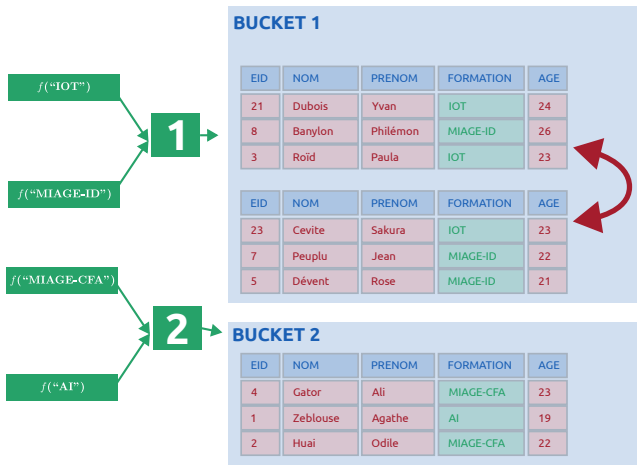
Organisation en fichier(s) : Fichier Haché



Recherche (égalité)
La fonction de hachage nous renvoi sur la bonne page donc une ouverture

T_p

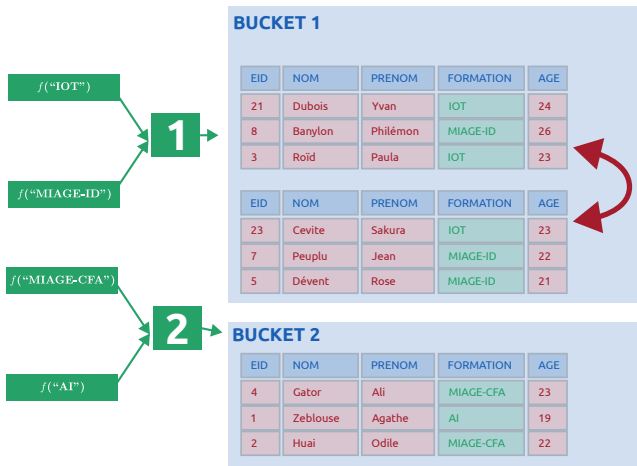
Organisation en fichier(s) : Fichier Haché



Recherche (intervalle)
Il faut parcourir toutes les valeurs :

$$\Rightarrow P \times T_p$$

Organisation en fichier(s) : Fichier Haché

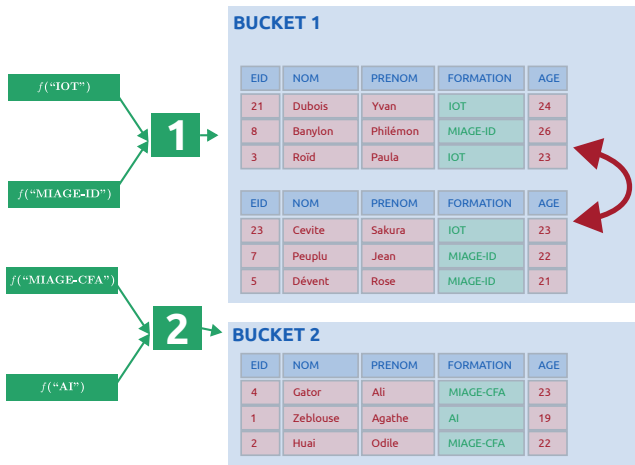


Insertion

- Rechercher la page (1 ouverture)
- Ajouter l'enregistrement (1 écriture)

$$\Rightarrow 2 \times T_p$$

Organisation en fichier(s) : Fichier Haché



Suppression

- Rechercher la page (1 ouverture)
- supprimer l'enregistrement (1 écriture)

$$\Rightarrow 2 \times T_p$$

Organisation en fichier(s) : Fichier Haché(recapitulatif)

	Scan	Rech (unicité)	Rech (intervalle)	Insert	Suppr
I/O (pages)	P	1	P	2	2
Complexité	P	1	P	1	1

→ Efficace pour la recherche sur égalité et la mise à jour

→ ⚠ Équilibrer les groupes (dépend de la fonction de hash)

Comparaison des organisations des fichiers :

	Scan	Rech (égalité)	Rech (intervalle)	Insert	Suppr
Tas (Heap File)	P	P	P	1	1
Trié (Sorted File)	P	$\log(P)$	$\log(P)$	P	P
Haché (Hashed File)	P	1	P	1	1

- Pour la recherche, préférence pour les fichiers triés
- Pour la recherche sur égalité et l'insertion/suppression, les fichiers hachés

Ce que nous avons vu :

- L'organisation en fichier des relations
- Les différents types de fichier

⚠ Les fichiers triés et hachés sont organisés selon un unique attribut!!!

Comment accélérer les opérations sur des attributs différents?

Utilisation d'index!!!